LinuxQMISDK

SLQS04.00.10

Generated by Doxygen 1.8.6

Fri Sep 8 2017 03:30:10

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 pFwAutoSDM . . . 85
8.27.2.6 pFwPromptdownload . . . 85
8.27.2.7 pFwPromptUpdate . . . 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 pFwAutoSDM . . . 85
8.28.1.6 pFwPromptdownload . . . 85
8.28.1.7 pFwPromptUpdate . . . 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 cdmaMsgDecodingParamsExt Struct Reference . . . 167
8.110.1 Detailed Description . . . 168
8.110.2 Field Documentation . . . 169
8.110.2.1 absoluteValidity . . . 169
8.110.2.2 mcTimeStamp . . . 169
8.110.2.3 messageLength . . . 170
8.110.2.4 pAlertPriority . . . 170
8.110.2.5 pCallbkAddr . . . 170
8.110.2.6 pCallbkAddrLength . . . 170
8.110.2.7 pDisplayMode . . . 170
8.110.2.8 pIsUDHPresent . . . 170
8.110.2.9 pLanguage . . . 170
8.110.2.10 pMessage . . . 170
8.110.2.11 pMessageID . . . 170
8.110.2.12 pPartNum . . . 170
8.110.2.13 pPriority . . . 170
8.110.2.14 pPrivacy . . . 170
8.110.2.15 pReadAcknowledgementReq . . . 170
8.110.2.16 pReferenceNum . . . 170
8.110.2.17 pRelativeValidity . . . 170
8.110.2.18 pSenderAddr . . . 170
8.110.2.19 pSenderAddrLength . . . 170
8.110.2.20 pTextMsg . . . 170
8.110.2.21 pTextMsgLength . . . 170
8.110.2.22 pTotalNum . . . 170
8.110.2.23 pUserAcknowledgementReq . . . 170
8.111 cdmaMsgEncodingParams Struct Reference . . . 170
8.111.1 Detailed Description . . . 171
8.111.2 Field Documentation . . . 172
8.111.2.1 messageId . . . 172
8.111.2.2 pCallbackAddr . . . 172
8.111.2.3 pDestAddr . . . 172
8.111.2.4 pEncodingAlphabet . . . 172
8.111.2.5 pMessage . . . 172
8.111.2.6 pMessageSize . . . 172
8.111.2.7 pPriority . . . 172
8.111.2.8 pRelValidity . . . 172
8.111.2.9 pTextMsg . . . 172
8.111.2.10 textMsgLength . . . 172
8.112 CDMARSSIThresh Struct Reference . . . 172

8.112.1 Detailed Description . . . 172
8.112.2 Field Documentation . . . 172
8.112.2.1 CDMARSSIThreshListLen . . . 172
8.112.2.2 pCDMARSSIThreshList . . . 172
8.113 CDMASSInfo Struct Reference . . . 172
8.113.1 Detailed Description . . . 173
8.113.2 Field Documentation . . . 173
8.113.2.1 ecio . . . 173
8.113.2.2 rssi . . . 173
8.114 cdmaSSInfo Struct Reference . . . 173
8.114.1 Detailed Description . . . 173
8.114.2 Field Documentation . . . 173
8.114.2.1 ecio . . . 173
8.114.2.2 rssi . . . 174
8.115 CDMASysInfo Struct Reference . . . 174
8.115.1 Detailed Description . . . 174
8.115.2 Field Documentation . . . 176
8.115.2.1 baseId . . . 176
8.115.2.2 baseLat . . . 177
8.115.2.3 baseLong . . . 177
8.115.2.4 bsInfoValid . . . 177
8.115.2.5 bsPRev . . . 177
8.115.2.6 bsPRevValid . . . 177
8.115.2.7 ccsSupported . . . 177
8.115.2.8 ccsSupportedValid . . . 177
8.115.2.9 cdmaSysIdValid . . . 177
8.115.2.10 isSysPrlMatch . . . 177
8.115.2.11 isSysPrlMatchValid . . . 177
8.115.2.12 MCC . . . 177
8.115.2.13 MNC . . . 177
8.115.2.14 networkID . . . 177
8.115.2.15 networkIdValid . . . 177
8.115.2.16 packetZone . . . 177
8.115.2.17 packetZoneValid . . . 177
8.115.2.18 pRevInUse . . . 177
8.115.2.19 pRevInUseValid . . . 177
8.115.2.20 sysInfoCDMA . . . 177
8.115.2.21 systemID . . . 177
8.116 CDMASysInfoExt Struct Reference . . . 177
8.116.1 Detailed Description . . . 177

8.116.2 Field Documentation . . . 178
8.116.2.1 imsi_11_12 . . . 178
8.116.2.2 MCC . . . 178
8.117 CellDb Struct Reference . . . 178
8.117.1 Detailed Description . . . 178
8.117.2 Field Documentation . . . 178
8.117.2.1 mask . . . 178
8.118 cellParams Struct Reference . . . 178
8.118.1 Detailed Description . . . 179
8.118.2 Field Documentation . . . 179
8.118.2.1 pci . . . 179
8.118.2.2 rsrp . . . 179
8.118.2.3 rsrq . . . 179
8.118.2.4 rssi . . . 179
8.118.2.5 srxlev . . . 179
8.119 changeUIMPIN Struct Reference . . . 179
8.119.1 Detailed Description . . . 180
8.119.2 Field Documentation . . . 180
8.119.2.1 oldPINLen . . . 180
8.119.2.2 oldPINVal . . . 180
8.119.2.3 pinID . . . 180
8.119.2.4 pinLen . . . 180
8.119.2.5 pinVal . . . 180
8.120 channelRate Struct Reference . . . 180
8.120.1 Detailed Description . . . 180
8.120.2 Field Documentation . . . 181
8.120.2.1 CurrChanRxRate . . . 181
8.120.2.2 CurrChanTxRate . . . 181
8.121 ChannelRate Struct Reference . . . 181
8.121.1 Detailed Description . . . 181
8.121.2 Field Documentation . . . 181
8.121.2.1 CurrChanRxRate . . . 181
8.121.2.2 CurrChanTxRate . . . 181
8.121.2.3 MaxChanRxRate . . . 181
8.121.2.4 MaxChanTxRate . . . 181
8.122 CLIPResp Struct Reference . . . 182
8.122.1 Detailed Description . . . 182
8.122.2 Field Documentation . . . 182
8.122.2.1 ActiveStatus . . . 182
8.122.2.2 ProvisionStatus . . . 182

8.123 CLIRResp Struct Reference . . . 182
8.123.1 Detailed Description . . . 182
8.123.2 Field Documentation . . . 183
8.123.2.1 ActiveStatus . . . 183
8.123.2.2 ProvisionStatus . . . 183
8.124 ClkInfo Struct Reference . . . 183
8.124.1 Detailed Description . . . 183
8.124.2 Field Documentation . . . 184
8.124.2.1 mask . . . 184
8.125 CNAPResp Struct Reference . . . 184
8.125.1 Detailed Description . . . 185
8.125.2 Field Documentation . . . 185
8.125.2.1 ActiveStatus . . . 185
8.125.2.2 ProvisionStatus . . . 185
8.126 COLPResp Struct Reference . . . 185
8.126.1 Detailed Description . . . 185
8.126.2 Field Documentation . . . 186
8.126.2.1 ActiveStatus . . . 186
8.126.2.2 ProvisionStatus . . . 186
8.127 COLRResp Struct Reference . . . 186
8.127.1 Detailed Description . . . 186
8.127.2 Field Documentation . . . 186
8.127.2.1 ActiveStatus . . . 186
8.127.2.2 ProvisionStatus . . . 186
8.128 CommInfo Struct Reference . . . 186
8.128.1 Detailed Description . . . 187
8.128.2 Field Documentation . . . 188
8.128.2.1 imsRegState . . . 188
8.128.2.2 modemMode . . . 188
8.128.2.3 psState . . . 188
8.128.2.4 systemMode . . . 188
8.128.2.5 temperature . . . 188
8.129 ConnectionStatus Struct Reference . . . 188
8.129.1 Detailed Description . . . 188
8.129.2 Field Documentation . . . 189
8.129.2.1 MDMCallDuration . . . 189
8.129.2.2 MDMConnStatus . . . 189
8.130 connectionStatus Struct Reference . . . 189
8.130.1 Detailed Description . . . 189
8.130.2 Field Documentation . . . 189

8.130.2.1 MDMCallDuration . . . 190
8.130.2.2 MDMConnStatus . . . 190
8.131 connectNumInfo Struct Reference . . . 190
8.131.1 Detailed Description . . . 190
8.131.2 Field Documentation . . . 191
8.131.2.1 callerID . . . 191
8.131.2.2 callerIDLen . . . 191
8.131.2.3 numPlan . . . 191
8.131.2.4 numPresInd . . . 191
8.131.2.5 numType . . . 191
8.131.2.6 screeningInd . . . 191
8.132 CrashInfo Struct Reference . . . 191
8.132.1 Detailed Description . . . 191
8.132.2 Field Documentation . . . 192
8.132.2.1 crashData . . . 192
8.132.2.2 crashId . . . 192
8.132.2.3 crashStrLen . . . 192
8.132.2.4 gcDumpStrLen . . . 192
8.132.2.5 numCrashes . . . 192
8.132.2.6 pCrashString . . . 192
8.132.2.7 pGCDumpString . . . 192
8.133 crashInfoParams Struct Reference . . . 192
8.133.1 Detailed Description . . . 192
8.133.2 Field Documentation . . . 193
8.133.2.1 crashInfo . . . 193
8.133.2.2 crashStatus . . . 193
8.134 CrashInfoParams Struct Reference . . . 193
8.134.1 Detailed Description . . . 193
8.134.2 Field Documentation . . . 193
8.134.2.1 pCrashInfo . . . 193
8.134.2.2 pDevCrashStatus . . . 193
8.135 crashInformation Struct Reference . . . 193
8.135.1 Detailed Description . . . 194
8.135.2 Field Documentation . . . 194
8.135.2.1 crashData . . . 194
8.135.2.2 crashId . . . 194
8.135.2.3 crashString . . . 194
8.135.2.4 crashStrlen . . . 194
8.135.2.5 gcdumpString . . . 194
8.135.2.6 gcdumpStrlen . . . 194

8.135.2.7 numCrashes . . . 194
8.136 CreateProfileIn Struct Reference . . . 194
8.136.1 Detailed Description . . . 195
8.136.2 Field Documentation . . . 195
8.136.2.1 curProfile . . . 195
8.136.2.2 pProfileID . . . 195
8.136.2.3 pProfileType . . . 195
8.137 CreateProfileOut Struct Reference . . . 195
8.137.1 Detailed Description . . . 195
8.137.2 Field Documentation . . . 196
8.137.2.1 pExtErrorCode . . . 196
8.137.2.2 pProfileIndex . . . 196
8.137.2.3 pProfileType . . . 196
8.138 CsgId Struct Reference . . . 196
8.138.1 Detailed Description . . . 196
8.138.2 Field Documentation . . . 196
8.138.2.1 csgId . . . 196
8.138.2.2 TlvPresent . . . 196
8.139 CSGID Struct Reference . . . 196
8.139.1 Detailed Description . . . 197
8.139.2 Field Documentation . . . 197
8.139.2.1 id . . . 197
8.139.2.2 mcc . . . 197
8.139.2.3 mnc . . . 197
8.139.2.4 mncPcsDigits . . . 197
8.139.2.5 rat . . . 197
8.140 CUGInfo Struct Reference . . . 197
8.140.1 Detailed Description . . . 198
8.140.2 Field Documentation . . . 198
8.140.2.1 CUGIndex . . . 198
8.140.2.2 SuppOA . . . 198
8.140.2.3 SuppPrefCUG . . . 198
8.141 curAMRConfig Struct Reference . . . 198
8.141.1 Detailed Description . . . 198
8.141.2 Field Documentation . . . 199
8.141.2.1 gsmAmrStat . . . 199
8.141.2.2 wcdmaAmrStat . . . 199
8.142 CurrDataSysStat Struct Reference . . . 199
8.142.1 Detailed Description . . . 199
8.142.2 Field Documentation . . . 200

8.142.2.1 pCurrNetworkInfo . . . 200
8.142.2.2 pNetworkInfoLen . . . 200
8.142.2.3 pPrefNetwork . . . 200
8.143currentCatEvent Union Reference . . . 200
8.143.1 Detailed Description . . . 200
8.143.2 Field Documentation . . . 200
8.143.2.1 CatAlphaIdtfr . . . 200
8.143.2.2 CatEndPS . . . 200
8.143.2.3 CatEventLst . . . 200
8.143.2.4 CatEvIDData . . . 200
8.143.2.5 CatRefresh . . . 200
8.144CurrentImgList Struct Reference . . . 200
8.144.1 Detailed Description . . . 201
8.144.2 Field Documentation . . . 201
8.144.2.1 carrier . . . 201
8.144.2.2 fwvers . . . 201
8.144.2.3 numEntries . . . 201
8.144.2.4 pCurrImgInfo . . . 201
8.144.2.5 pkgver . . . 201
8.144.2.6 priver . . . 201
8.145currentPLMN Struct Reference . . . 201
8.145.1 Detailed Description . . . 202
8.145.2 Field Documentation . . . 202
8.145.2.1 MCC . . . 202
8.145.2.2 MNC . . . 202
8.145.2.3 netDescr . . . 202
8.145.2.4 netDescrLength . . . 202
8.146CurrImageInfo Struct Reference . . . 202
8.146.1 Detailed Description . . . 203
8.146.2 Field Documentation . . . 203
8.146.2.1 buildID . . . 203
8.146.2.2 buildIDLen . . . 203
8.146.2.3 imageType . . . 203
8.146.2.4 uniqueID . . . 203
8.147CurrNetworkInfo Struct Reference . . . 203
8.147.1 Detailed Description . . . 203
8.147.2 Field Documentation . . . 204
8.147.2.1 NetworkType . . . 205
8.147.2.2 RATMask . . . 205
8.147.2.3 SOMask . . . 205

8.148currNetworkInfo Struct Reference . . . 205
8.148.1 Detailed Description . . . 205
8.148.2 Field Documentation . . . 206
8.148.2.1 NetworkType . . . 206
8.148.2.2 RATMask . . . 206
8.148.2.3 SOMask . . . 206
8.149custFeaturesInfo Struct Reference . . . 206
8.149.1 Detailed Description . . . 206
8.149.2 Field Documentation . . . 208
8.149.2.1 GpsEnable . . . 208
8.149.2.2 pDHCPRelayEnabled . . . 208
8.149.2.3 pDisableIMSI . . . 208
8.149.2.4 pGPSLPM . . . 208
8.149.2.5 pGPSSel . . . 208
8.149.2.6 pIPFamSupport . . . 208
8.149.2.7 pIsVoiceEnabled . . . 208
8.149.2.8 pRMAutoConnect . . . 208
8.149.2.9 pSMSSupport . . . 209
8.150custFeaturesSetting Struct Reference . . . 209
8.150.1 Detailed Description . . . 209
8.150.2 Field Documentation . . . 210
8.150.2.1 pDHCPRelayEnabled . . . 210
8.150.2.2 pGPSEnable . . . 210
8.150.2.3 pGPSLPM . . . 210
8.150.2.4 pGPSSel . . . 210
8.150.2.5 pIsVoiceEnabled . . . 210
8.151custSettingInfo Struct Reference . . . 210
8.151.1 Detailed Description . . . 210
8.151.2 Field Documentation . . . 211
8.151.2.1 cust_attr . . . 211
8.151.2.2 cust_id . . . 211
8.151.2.3 cust_value . . . 211
8.151.2.4 id_length . . . 211
8.151.2.5 value_length . . . 211
8.152custSettingList Struct Reference . . . 211
8.152.1 Detailed Description . . . 211
8.152.2 Field Documentation . . . 211
8.152.2.1 custSetting . . . 211
8.152.2.2 list_type . . . 212
8.152.2.3 num_instances . . . 212

8.153dataBearers Struct Reference . . . 212
8.153.1 Detailed Description . . . 212
8.153.2 Field Documentation . . . 212
8.153.2.1 dataBearerMask . . . 212
8.153.2.2 pCurDataBearerTechnology . . . 212
8.153.2.3 pLastCallDataBearerTechnology . . . 212
8.154DataBearerTech Struct Reference . . . 212
8.154.1 Detailed Description . . . 212
8.154.2 Field Documentation . . . 214
8.154.2.1 ratValue . . . 214
8.154.2.2 soMask . . . 214
8.154.2.3 techType . . . 214
8.155DataBearerTechExt Struct Reference . . . 214
8.155.1 Detailed Description . . . 215
8.155.2 Field Documentation . . . 215
8.155.2.1 pBearerTech . . . 215
8.155.2.2 pLastBearerTech . . . 215
8.156dataBearerTechnology Struct Reference . . . 215
8.156.1 Detailed Description . . . 215
8.156.2 Field Documentation . . . 216
8.156.2.1 currentNetwork . . . 216
8.156.2.2 ratMask . . . 216
8.156.2.3 soMask . . . 216
8.157dataRate Struct Reference . . . 216
8.157.1 Detailed Description . . . 216
8.157.2 Field Documentation . . . 217
8.157.2.1 dataRateMax . . . 217
8.157.2.2 guaranteedRate . . . 217
8.158dataSrvCapabilities Struct Reference . . . 217
8.158.1 Detailed Description . . . 217
8.158.2 Field Documentation . . . 217
8.158.2.1 dataCapabilities . . . 217
8.158.2.2 dataCapabilitiesLen . . . 218
8.159DataStatusDetail Struct Reference . . . 218
8.159.1 Detailed Description . . . 218
8.159.2 Field Documentation . . . 219
8.159.2.1 IPAddress . . . 219
8.159.2.2 LastErrCode . . . 219
8.160DataULongLongTlv Struct Reference . . . 220
8.160.1 Field Documentation . . . 220

8.160.1.1 TlvPresent . . . 220
8.160.1.2 ullData . . . 220
8.161 DataULongTlv Struct Reference . . . 220
8.161.1 Field Documentation . . . 220
8.161.1.1 TlvPresent . . . 220
8.161.1.2 ulData . . . 220
8.162 DcsUsbPortNames Struct Reference . . . 220
8.162.1 Field Documentation . . . 220
8.162.1.1 AtCmdPort . . . 220
8.162.1.2 DmPort . . . 220
8.162.1.3 NmeaPort . . . 220
8.163 delAssistDataStatus Struct Reference . . . 220
8.163.1 Detailed Description . . . 221
8.163.2 Field Documentation . . . 221
8.163.2.1 status . . . 221
8.164 depersonalizationInformation Struct Reference . . . 221
8.164.1 Detailed Description . . . 221
8.164.2 Field Documentation . . . 222
8.164.2.1 ckLen . . . 222
8.164.2.2 ckVal . . . 222
8.164.2.3 feature . . . 222
8.164.2.4 operation . . . 222
8.165 detailSvcInfo Struct Reference . . . 222
8.165.1 Detailed Description . . . 222
8.165.2 Field Documentation . . . 223
8.165.2.1 hdrHybrid . . . 223
8.165.2.2 hdrSrvStatus . . . 223
8.165.2.3 isSysForbidden . . . 223
8.165.2.4 srvCapability . . . 224
8.165.2.5 srvStatus . . . 224
8.166 DeviceConfigDetail Struct Reference . . . 224
8.166.1 Detailed Description . . . 224
8.166.2 Field Documentation . . . 224
8.166.2.1 Chipset . . . 224
8.166.2.2 HWVersion . . . 225
8.166.2.3 QLIC . . . 225
8.166.2.4 Technology . . . 225
8.167 DHCPOption Struct Reference . . . 225
8.167.1 Detailed Description . . . 225
8.167.2 Field Documentation . . . 225

8.167.2.1 optCode . . . 225
8.167.2.2 optValLen . . . 225
8.167.2.3 pOptVal . . . 225
8.168 DHCPOptionList Struct Reference . . . 225
8.168.1 Detailed Description . . . 225
8.168.2 Field Documentation . . . 226
8.168.2.1 numOpt . . . 226
8.168.2.2 pOptions . . . 226
8.169 diagInfo Struct Reference . . . 226
8.169.1 Detailed Description . . . 226
8.169.2 Field Documentation . . . 226
8.169.2.1 diagInfoLen . . . 226
8.169.2.2 diagnosticInfo . . . 226
8.170 dirNum Struct Reference . . . 226
8.170.1 Detailed Description . . . 227
8.170.2 Field Documentation . . . 227
8.170.2.1 dirNum . . . 227
8.170.2.2 dirNumLen . . . 227
8.171 dms_ActivationStatusTlv Struct Reference . . . 227
8.171.1 Detailed Description . . . 227
8.171.2 Field Documentation . . . 228
8.171.2.1 activationStatus . . . 228
8.171.2.2 TlvPresent . . . 228
8.172 dms_OperatingModeTlv Struct Reference . . . 228
8.172.1 Detailed Description . . . 228
8.172.2 Field Documentation . . . 228
8.172.2.1 operatingMode . . . 228
8.172.2.2 TlvPresent . . . 228
8.173 dmsCurrentPRLInfo Struct Reference . . . 228
8.173.1 Detailed Description . . . 228
8.173.2 Field Documentation . . . 229
8.173.2.1 pPRLPreference . . . 229
8.173.2.2 pPRLVersion . . . 229
8.174 DMScustSettingInfo Struct Reference . . . 229
8.174.1 Detailed Description . . . 229
8.174.2 Field Documentation . . . 230
8.174.2.1 cust_attr . . . 230
8.174.2.2 cust_id . . . 230
8.174.2.3 cust_value . . . 230
8.174.2.4 id_length . . . 230

8.174.2.5 value_length . . . 230
8.175 DMScustSettingList Struct Reference . . . 230
8.175.1 Detailed Description . . . 230
8.175.2 Field Documentation . . . 230
8.175.2.1 custSetting . . . 230
8.175.2.2 list_type . . . 230
8.175.2.3 num_instances . . . 230
8.176 DMSgetCustomFeatureV2 Struct Reference . . . 230
8.176.1 Detailed Description . . . 231
8.176.2 Field Documentation . . . 231
8.176.2.1 pCustSettingInfo . . . 231
8.176.2.2 pCustSettingList . . . 231
8.176.2.3 pGetCustomInput . . . 231
8.177 DMSgetCustomInput Struct Reference . . . 231
8.177.1 Detailed Description . . . 231
8.177.2 Field Documentation . . . 231
8.177.2.1 cust_id . . . 231
8.177.2.2 list_type . . . 231
8.178 dmsIndicationRegisterReq Struct Reference . . . 232
8.178.1 Detailed Description . . . 232
8.178.2 Field Documentation . . . 232
8.178.2.1 pSwiGetResetInd . . . 232
8.179 dmsSwiGetResetInfo Struct Reference . . . 232
8.179.1 Detailed Description . . . 232
8.179.2 Field Documentation . . . 233
8.179.2.1 source . . . 233
8.179.2.2 type . . . 233
8.180 Domain Struct Reference . . . 233
8.180.1 Detailed Description . . . 233
8.180.2 Field Documentation . . . 233
8.180.2.1 domainLen . . . 233
8.180.2.2 domainName . . . 233
8.181 DomainNameList Struct Reference . . . 233
8.181.1 Detailed Description . . . 234
8.181.2 Field Documentation . . . 234
8.181.2.1 domain . . . 234
8.181.2.2 numInstances . . . 234
8.182 DRCParams Struct Reference . . . 234
8.182.1 Detailed Description . . . 234
8.182.2 Field Documentation . . . 234

8.182.2.1 DRCCover . . . 234
8.182.2.2 DRCValue . . . 234
8.183 DTMFInfo Struct Reference . . . 234
8.183.1 Detailed Description . . . 235
8.183.2 Field Documentation . . . 235
8.183.2.1 callID . . . 235
8.183.2.2 digitBuff . . . 235
8.183.2.3 digitCnt . . . 235
8.183.2.4 DTMFEvent . . . 235
8.184 DTMFLengths Struct Reference . . . 235
8.184.1 Detailed Description . . . 235
8.184.2 Field Documentation . . . 236
8.184.2.1 DTMFInterdigitInterval . . . 236
8.184.2.2 DTMFPulseWidth . . . 236
8.185 DUNCallInfoInd Struct Reference . . . 236
8.185.1 Field Documentation . . . 236
8.185.1.1 CallEndReason . . . 236
8.185.1.2 ChannelRate . . . 236
8.185.1.3 DataBearerTech . . . 236
8.185.1.4 DormancyStatus . . . 236
8.185.1.5 MdmConnStatus . . . 236
8.185.1.6 RXOKBytesCount . . . 236
8.185.1.7 TXOKBytesCount . . . 236
8.186 dunchannelRate Struct Reference . . . 237
8.186.1 Detailed Description . . . 237
8.186.2 Field Documentation . . . 237
8.186.2.1 CurrChanRxRate . . . 237
8.186.2.2 CurrChanTxRate . . . 237
8.186.2.3 MaxChanRxRate . . . 237
8.186.2.4 MaxChanTxRate . . . 237
8.187 ecioListElement Struct Reference . . . 237
8.187.1 Detailed Description . . . 237
8.187.2 Field Documentation . . . 238
8.187.2.1 ecio . . . 238
8.187.2.2 radioIf . . . 238
8.188 ECIOThresh Struct Reference . . . 238
8.188.1 Detailed Description . . . 238
8.188.2 Field Documentation . . . 238
8.188.2.1 ECIOThresListLen . . . 238
8.188.2.2 pECIOThresList . . . 239

8.189 ECTNum Struct Reference . . . 239
8.189.1 Detailed Description . . . 239
8.189.2 Field Documentation . . . 239
8.189.2.1 ECTCallState . . . 239
8.189.2.2 number . . . 239
8.189.2.3 presentationInd . . . 239
8.190 encryptedPIN1 Struct Reference . . . 239
8.190.1 Detailed Description . . . 239
8.190.2 Field Documentation . . . 240
8.190.2.1 pin1Len . . . 240
8.190.2.2 pin1Val . . . 240
8.191 eriDataparams Struct Reference . . . 240
8.191.1 Detailed Description . . . 240
8.191.2 Field Documentation . . . 240
8.191.2.1 eriData . . . 240
8.191.2.2 eriDataLen . . . 240
8.192 ERIFileparams Struct Reference . . . 240
8.192.1 Detailed Description . . . 241
8.192.2 Field Documentation . . . 241
8.192.2.1 pFile . . . 241
8.192.2.2 pFileSize . . . 241
8.193 errorRateListElement Struct Reference . . . 241
8.193.1 Detailed Description . . . 241
8.193.2 Field Documentation . . . 242
8.193.2.1 errorRate . . . 242
8.193.2.2 radioIf . . . 242
8.194 eTWSPLMNInfoTlv Struct Reference . . . 242
8.194.1 Detailed Description . . . 242
8.194.2 Field Documentation . . . 242
8.194.2.1 ETWSPLMNInfo . . . 242
8.194.2.2 TlvPresent . . . 242
8.195 extDispRecInfo Struct Reference . . . 242
8.195.1 Detailed Description . . . 243
8.195.2 Field Documentation . . . 243
8.195.2.1 dispType . . . 243
8.195.2.2 extDispInfo . . . 243
8.195.2.3 extDispInfoLen . . . 243
8.196 FactorySequenceNumber Struct Reference . . . 243
8.196.1 Detailed Description . . . 243
8.196.2 Field Documentation . . . 243

8.196.2.1 FSNumber . . . 243
8.197 fileAttributes Struct Reference . . . 243
8.197.1 Detailed Description . . . 244
8.197.2 Field Documentation . . . 246
8.197.2.1 fileID . . . 246
8.197.2.2 fileSize . . . 246
8.197.2.3 fileType . . . 246
8.197.2.4 rawLen . . . 246
8.197.2.5 rawValue . . . 246
8.197.2.6 recordCount . . . 246
8.197.2.7 recordSize . . . 246
8.197.2.8 secActivate . . . 246
8.197.2.9 secActivateMask . . . 246
8.197.2.10 secDeactivate . . . 246
8.197.2.11 secDeactivateMask . . . 247
8.197.2.12 secIncrease . . . 247
8.197.2.13 secIncreaseMask . . . 247
8.197.2.14 secRead . . . 247
8.197.2.15 secReadMask . . . 247
8.197.2.16 secWrite . . . 247
8.197.2.17 secWriteMask . . . 247
8.198 fileInfo Struct Reference . . . 247
8.198.1 Detailed Description . . . 247
8.198.2 Field Documentation . . . 247
8.198.2.1 fileID . . . 247
8.198.2.2 path . . . 247
8.198.2.3 pathLen . . . 247
8.199 FirmwareUpdatStat Struct Reference . . . 247
8.199.1 Detailed Description . . . 248
8.199.2 Field Documentation . . . 249
8.199.2.1 pImgType . . . 249
8.199.2.2 pLogString . . . 249
8.199.2.3 pLogStringLen . . . 249
8.199.2.4 pRefData . . . 249
8.199.2.5 pRefString . . . 249
8.199.2.6 pRefStringLen . . . 249
8.199.2.7 ResCode . . . 249
8.200 FMSImageElement Struct Reference . . . 249
8.200.1 Detailed Description . . . 249
8.200.2 Field Documentation . . . 249

8.200.2.1 buildId . . . 250
8.200.2.2 buildIdLength . . . 250
8.200.2.3 imageId . . . 250
8.200.2.4 imageType . . . 250
8.201 FMSImageIdElement Struct Reference . . . 250
8.201.1 Detailed Description . . . 250
8.201.2 Field Documentation . . . 250
8.201.2.1 buildID . . . 250
8.201.2.2 buildIDLength . . . 250
8.201.2.3 failureCount . . . 250
8.201.2.4 imageID . . . 250
8.201.2.5 storageIndex . . . 250
8.202 FMSImageIDEntries Struct Reference . . . 251
8.202.1 Detailed Description . . . 251
8.202.2 Field Documentation . . . 251
8.202.2.1 executingImage . . . 251
8.202.2.2 imageIDElement . . . 251
8.202.2.3 imageIDSize . . . 251
8.202.2.4 imageType . . . 251
8.202.2.5 maxImages . . . 251
8.203 FMSImageList Struct Reference . . . 251
8.203.1 Detailed Description . . . 252
8.203.2 Field Documentation . . . 252
8.203.2.1 imageIDEntries . . . 252
8.203.2.2 listSize . . . 252
8.204 FMSPrefImageList Struct Reference . . . 252
8.204.1 Detailed Description . . . 252
8.204.2 Field Documentation . . . 252
8.204.2.1 listEntries . . . 252
8.204.2.2 listSize . . . 252
8.205 fwinfo_s Struct Reference . . . 252
8.205.1 Detailed Description . . . 253
8.205.2 Field Documentation . . . 253
8.205.2.1 Carrier . . . 253
8.205.2.2 FirmwareID . . . 253
8.205.2.3 GPSCapability . . . 253
8.205.2.4 Region . . . 253
8.205.2.5 Technology . . . 253
8.206 GERANInfo Struct Reference . . . 253
8.206.1 Detailed Description . . . 253

8.206.2 Field Documentation . . . 254
8.206.2.1 arfcn . . . 254
8.206.2.2 bsic . . . 255
8.206.2.3 cellID . . . 255
8.206.2.4 insNmrCellInfo . . . 255
8.206.2.5 lac . . . 255
8.206.2.6 nmrInst . . . 255
8.206.2.7 plmn . . . 255
8.206.2.8 rxLev . . . 255
8.206.2.9 timingAdvance . . . 255
8.207 geranInstInfo Struct Reference . . . 255
8.207.1 Detailed Description . . . 255
8.207.2 Field Documentation . . . 255
8.207.2.1 geranArfcn . . . 255
8.207.2.2 geranBsicBcc . . . 255
8.207.2.3 geranBsicNcc . . . 255
8.207.2.4 geranRssi . . . 256
8.208 getAllCallInformation Struct Reference . . . 256
8.208.1 Detailed Description . . . 256
8.208.2 Field Documentation . . . 256
8.208.2.1 ALS . . . 256
8.208.2.2 Callinfo . . . 256
8.208.2.3 isEmpty . . . 256
8.209 getAllCallRmtPtyName Struct Reference . . . 256
8.209.1 Detailed Description . . . 256
8.209.2 Field Documentation . . . 257
8.209.2.1 callID . . . 257
8.209.2.2 RemotePartyName . . . 257
8.210 getAllCallRmtPtyNum Struct Reference . . . 257
8.210.1 Detailed Description . . . 257
8.210.2 Field Documentation . . . 257
8.210.2.1 callID . . . 257
8.210.2.2 RemotePartyNum . . . 257
8.211 GetAudioPathConfigReq Struct Reference . . . 257
8.211.1 Detailed Description . . . 257
8.211.2 Field Documentation . . . 258
8.211.2.1 Item . . . 258
8.211.2.2 Profile . . . 258
8.212 GetAudioPathConfigResp Struct Reference . . . 258
8.212.1 Detailed Description . . . 258

8.212.2 Field Documentation . . . 259
8.212.2.1 pCodecSTGain . . . 260
8.212.2.2 pDTMFTXGain . . . 260
8.212.2.3 pECMode . . . 260
8.212.2.4 pMICGainSelect . . . 260
8.212.2.5 pNSEnable . . . 260
8.212.2.6 pRXAGCList . . . 260
8.212.2.7 pRXAVCAGCSwitch . . . 260
8.212.2.8 pRXAVCList . . . 260
8.212.2.9 pRXPCMIIRFltr . . . 260
8.212.2.10 pTXAGCList . . . 260
8.212.2.11 pTXAVCSwitch . . . 260
8.212.2.12 pTXGain . . . 260
8.212.2.13 pTXPCMIIRFltr . . . 260
8.213 GetAudioProfileReq Struct Reference . . . 260
8.213.1 Detailed Description . . . 260
8.213.2 Field Documentation . . . 260
8.213.2.1 Generator . . . 260
8.214 GetAudioProfileResp Struct Reference . . . 260
8.214.1 Detailed Description . . . 261
8.214.2 Field Documentation . . . 261
8.214.2.1 EarMute . . . 261
8.214.2.2 MicMute . . . 261
8.214.2.3 Profile . . . 261
8.214.2.4 Volume . . . 261
8.215 GetAudioVolTLBConfigReq Struct Reference . . . 262
8.215.1 Detailed Description . . . 262
8.215.2 Field Documentation . . . 262
8.215.2.1 Generator . . . 262
8.215.2.2 Item . . . 262
8.215.2.3 Profile . . . 262
8.215.2.4 Volume . . . 262
8.216 GetAudioVolTLBConfigResp Struct Reference . . . 262
8.216.1 Detailed Description . . . 263
8.216.2 Field Documentation . . . 263
8.216.2.1 ResCode . . . 263
8.217 getCallFWExtInfo Struct Reference . . . 263
8.217.1 Detailed Description . . . 263
8.217.2 Field Documentation . . . 263
8.217.2.1 CallFWExtInfo . . . 263

8.217.2.2 numInstances . . . 263
8.218 getCallFWInfo Struct Reference . . . 263
8.218.1 Detailed Description . . . 263
8.218.2 Field Documentation . . . 264
8.218.2.1 CallFWInfo . . . 264
8.218.2.2 numInstances . . . 264
8.219 getCustomFeatureV2 Struct Reference . . . 264
8.219.1 Detailed Description . . . 264
8.219.2 Field Documentation . . . 264
8.219.2.1 pCustSettingInfo . . . 264
8.219.2.2 pCustSettingList . . . 264
8.219.2.3 pGetCustomInput . . . 264
8.220 getCustomInput Struct Reference . . . 265
8.220.1 Detailed Description . . . 265
8.220.2 Field Documentation . . . 265
8.220.2.1 cust_id . . . 265
8.220.2.2 list_type . . . 265
8.221 getDUNCallInfoReq Struct Reference . . . 265
8.221.1 Detailed Description . . . 265
8.221.2 Field Documentation . . . 266
8.221.2.1 Mask . . . 266
8.221.2.2 pReportChannelRate . . . 266
8.221.2.3 pReportConnStatus . . . 266
8.221.2.4 pReportDataBearerTech . . . 266
8.221.2.5 pReportDormStatus . . . 266
8.221.2.6 pTransferStatInd . . . 266
8.222 getDUNCallInfoResp Struct Reference . . . 267
8.222.1 Detailed Description . . . 267
8.222.2 Field Documentation . . . 269
8.222.2.1 pCallEndReason . . . 269
8.222.2.2 pChannelRate . . . 269
8.222.2.3 pConnectionStatus . . . 269
8.222.2.4 pDataBearerTech . . . 269
8.222.2.5 pDormancyStatus . . . 269
8.222.2.6 pLastCallDataBearerTech . . . 269
8.222.2.7 pLastCallRXOKBytesCnt . . . 269
8.222.2.8 pLastCallTXOKBytesCnt . . . 269
8.222.2.9 pMdmCallDurationActive . . . 269
8.222.2.10 pRXOKBytesCount . . . 269
8.222.2.11 pTXOKBytesCount . . . 269

8.223 getDyingGaspCfg Struct Reference . . . 269
8.223.1 Detailed Description . . . 270
8.223.2 Field Documentation . . . 270
8.223.2.1 pDestSMSContent . . . 270
8.223.2.2 pDestSMSNum . . . 270
8.224 getDyingGaspStatistics Struct Reference . . . 270
8.224.1 Detailed Description . . . 270
8.224.2 Field Documentation . . . 270
8.224.2.1 pSMSAttemptedFlag . . . 271
8.224.2.2 pTimeStamp . . . 271
8.225 GetErrRateResp Struct Reference . . . 271
8.225.1 Detailed Description . . . 271
8.225.2 Field Documentation . . . 271
8.225.2.1 pCDMAFrameErrRate . . . 271
8.225.2.2 pGSMBER . . . 271
8.225.2.3 pHDRPackErrRate . . . 271
8.225.2.4 pWCDMABER . . . 271
8.226 GetHRPDStatsResp Struct Reference . . . 272
8.226.1 Detailed Description . . . 272
8.226.2 Field Documentation . . . 272
8.226.2.1 pDRCParams . . . 272
8.226.2.2 pPilotSetData . . . 272
8.226.2.3 pUATI . . . 272
8.227 GetIMSSMSConfigParams Struct Reference . . . 272
8.227.1 Detailed Description . . . 272
8.227.2 Field Documentation . . . 273
8.227.2.1 pPhoneCtxtURI . . . 273
8.227.2.2 pPhoneCtxtURILen . . . 273
8.227.2.3 pSettingResp . . . 273
8.227.2.4 pSMSFormat . . . 273
8.227.2.5 pSMSOverIPNwInd . . . 273
8.228 GetIMSUserConfigParams Struct Reference . . . 273
8.228.1 Detailed Description . . . 273
8.228.2 Field Documentation . . . 274
8.228.2.1 pIMSDomain . . . 274
8.228.2.2 pIMSDomainLen . . . 274
8.228.2.3 pSettingResp . . . 274
8.229 GetIMSVoIPConfigResp Struct Reference . . . 274
8.229.1 Detailed Description . . . 274
8.229.2 Field Documentation . . . 276

8.229.2.1 pAmrMode . . . 276
8.229.2.2 pAmrOctetAligned . . . 276
8.229.2.3 pAmrWbEnable . . . 276
8.229.2.4 pAmrWBMode . . . 276
8.229.2.5 pAmrWBOctetAligned . . . 276
8.229.2.6 pMinSessionExpiryTimer . . . 276
8.229.2.7 pRingBackTimer . . . 276
8.229.2.8 pRingingTimer . . . 276
8.229.2.9 pRTPRTCPInactTimer . . . 276
8.229.2.10 pScrAmrEnable . . . 276
8.229.2.11 pScrAmrWbEnable . . . 276
8.229.2.12 pSessionExpiryTimer . . . 276
8.229.2.13 pSettingResp . . . 276
8.230 GetInstIDResp Struct Reference . . . 276
8.230.1 Field Documentation . . . 276
8.230.1.1 pInstanceID . . . 276
8.230.1.2 pIPFamily . . . 277
8.231 GetM2MAudioProfileReq Struct Reference . . . 277
8.231.1 Detailed Description . . . 277
8.231.2 Field Documentation . . . 277
8.231.2.1 pGenerator . . . 277
8.232 GetM2MAudioProfileResp Struct Reference . . . 277
8.232.1 Detailed Description . . . 277
8.232.2 Field Documentation . . . 278
8.232.2.1 CwtMute . . . 278
8.232.2.2 EarMute . . . 278
8.232.2.3 Generator . . . 278
8.232.2.4 MicMute . . . 278
8.232.2.5 Profile . . . 278
8.232.2.6 Volume . . . 278
8.233 GetM2MAudioVolumeReq Struct Reference . . . 278
8.233.1 Detailed Description . . . 278
8.233.2 Field Documentation . . . 279
8.233.2.1 Generator . . . 279
8.233.2.2 Profile . . . 279
8.234 GetM2MAudioVolumeResp Struct Reference . . . 279
8.234.1 Detailed Description . . . 279
8.234.2 Field Documentation . . . 279
8.234.2.1 Level . . . 279
8.235 GetM2MAVMuteReq Struct Reference . . . 279

8.235.1 Detailed Description . . . 279
8.235.2 Field Documentation . . . 279
8.235.2.1 Profile . . . 280
8.236 GetM2MAVMuteResp Struct Reference . . . 280
8.236.1 Detailed Description . . . 280
8.236.2 Field Documentation . . . 280
8.236.2.1 CwtMute . . . 280
8.236.2.2 EarMute . . . 280
8.236.2.3 MicMute . . . 280
8.237 GetM2MSpkrGainReq Struct Reference . . . 280
8.237.1 Detailed Description . . . 280
8.237.2 Field Documentation . . . 281
8.237.2.1 Profile . . . 281
8.238 GetM2MSpkrGainResp Struct Reference . . . 281
8.238.1 Detailed Description . . . 281
8.238.2 Field Documentation . . . 281
8.238.2.1 Value . . . 281
8.239 getMsgWaitingInfo Struct Reference . . . 281
8.239.1 Detailed Description . . . 281
8.239.2 Field Documentation . . . 282
8.239.2.1 msgWaitInfo . . . 282
8.239.2.2 numInstances . . . 282
8.240 GetNetworkTimeResp Struct Reference . . . 282
8.240.1 Detailed Description . . . 282
8.240.2 Field Documentation . . . 282
8.240.2.1 p3GPP2TimeInfo . . . 282
8.240.2.2 p3GPPTimeInfo . . . 282
8.241 GetRegMgrConfigParams Struct Reference . . . 282
8.241.1 Detailed Description . . . 282
8.241.2 Field Documentation . . . 283
8.241.2.1 pIMSTestMode . . . 283
8.241.2.2 pPCSCFPort . . . 283
8.241.2.3 pPriCSCFPortName . . . 283
8.241.2.4 pPriCSCFPortNameLen . . . 283
8.241.2.5 pSettingResp . . . 283
8.242 GetSessionIDResp Struct Reference . . . 283
8.242.1 Field Documentation . . . 283
8.242.1.1 pSessionIDv4 . . . 283
8.242.1.2 pSessionIDv6 . . . 283
8.243 GetSIPConfigResp Struct Reference . . . 283

8.243.1 Detailed Description . . . 284
8.243.2 Field Documentation . . . 284
8.243.2.1 pSettingResp . . . 284
8.243.2.2 pSigCompEnabled . . . 284
8.243.2.3 pSIPLocalPort . . . 284
8.243.2.4 pSubscribeTimer . . . 284
8.243.2.5 pTimerSIPReg . . . 284
8.243.2.6 pTimerT1 . . . 284
8.243.2.7 pTimerT2 . . . 284
8.243.2.8 pTimerTf . . . 284
8.244 GnssData Struct Reference . . . 285
8.244.1 Detailed Description . . . 285
8.244.2 Field Documentation . . . 286
8.244.2.1 mask . . . 286
8.245 gnssSvInfoNotification Struct Reference . . . 286
8.245.1 Detailed Description . . . 287
8.245.2 Field Documentation . . . 287
8.245.2.1 bAltitudeAssumed . . . 287
8.245.2.2 pSatelliteInfo . . . 287
8.246 GPRSQoS Struct Reference . . . 287
8.246.1 Detailed Description . . . 287
8.246.2 Field Documentation . . . 288
8.246.2.1 delayClass . . . 288
8.246.2.2 meanThroughputClass . . . 288
8.246.2.3 peakThroughputClass . . . 288
8.246.2.4 precedenceClass . . . 288
8.246.2.5 reliabilityClass . . . 288
8.247 GPRSRequestedQoS Struct Reference . . . 288
8.247.1 Detailed Description . . . 288
8.247.2 Field Documentation . . . 289
8.247.2.1 delayClass . . . 289
8.247.2.2 meanThroughputClass . . . 289
8.247.2.3 peakThroughputClass . . . 289
8.247.2.4 precedenceClass . . . 289
8.247.2.5 reliabilityClass . . . 289
8.248 GPSStateInfo Struct Reference . . . 289
8.248.1 Detailed Description . . . 289
8.248.2 Field Documentation . . . 292
8.248.2.1 Altitude . . . 292
8.248.2.2 EngineState . . . 292

8.248.2.3 glo_almanac_sv_msk . . . 292
8.248.2.4 glo_ephemeris_sv_msk . . . 292
8.248.2.5 glo_health_sv_msk . . . 292
8.248.2.6 glo_visible_sv_msk . . . 292
8.248.2.7 gps_almanac_sv_msk . . . 292
8.248.2.8 gps_ephemeris_sv_msk . . . 292
8.248.2.9 gps_health_sv_msk . . . 292
8.248.2.10 gps_visible_sv_msk . . . 292
8.248.2.11 HorizontalUncertainty . . . 292
8.248.2.12 iono_valid . . . 292
8.248.2.13 Latitude . . . 292
8.248.2.14 Longitude . . . 292
8.248.2.15 sbas_almanac_sv_msk . . . 292
8.248.2.16 sbas_ephemeris_sv_msk . . . 292
8.248.2.17 sbas_health_sv_msk . . . 293
8.248.2.18 sbas_visible_sv_msk . . . 293
8.248.2.19 Time_uncert_ms . . . 293
8.248.2.20 TimeStmp_gps_week . . . 293
8.248.2.21 TimeStmp_tow_ms . . . 293
8.248.2.22 ValidMask . . . 293
8.248.2.23 VerticalUncertainty . . . 293
8.248.2.24 xtra_start_gps_minutes . . . 293
8.248.2.25 xtra_start_gps_week . . . 293
8.248.2.26 xtra_valid_duration_hours . . . 293
8.249 gpsTime_s Struct Reference . . . 293
8.249.1 Detailed Description . . . 293
8.249.2 Field Documentation . . . 293
8.249.2.1 gpsTimeOfWeekMs . . . 293
8.249.2.2 gpsWeek . . . 293
8.250 gsmCellInfo Struct Reference . . . 293
8.250.1 Detailed Description . . . 294
8.250.2 Field Documentation . . . 294
8.250.2.1 arfcn . . . 294
8.250.2.2 band1900 . . . 294
8.250.2.3 bsicId . . . 294
8.250.2.4 cellIdValid . . . 294
8.250.2.5 rssi . . . 294
8.250.2.6 srxlev . . . 294
8.251 GSMRSSIThresh Struct Reference . . . 295
8.251.1 Detailed Description . . . 295

8.251.2 Field Documentation . . . 295
8.251.2.1 GSMRSSIThreshListLen . . . 295
8.251.2.2 pGSMRSSIThreshList . . . 295
8.252 GSMSrvStatusInfo Struct Reference . . . 295
8.252.1 Detailed Description . . . 295
8.252.2 Field Documentation . . . 296
8.252.2.1 isPrefDataPath . . . 296
8.252.2.2 srvStatus . . . 296
8.252.2.3 trueSrvStatus . . . 296
8.253 GSMSysInfo Struct Reference . . . 296
8.253.1 Detailed Description . . . 296
8.253.2 Field Documentation . . . 298
8.253.2.1 cellId . . . 298
8.253.2.2 cellIdValid . . . 298
8.253.2.3 dtmSupp . . . 298
8.253.2.4 dtmSuppValid . . . 298
8.253.2.5 egprsSupp . . . 298
8.253.2.6 egprsSuppValid . . . 298
8.253.2.7 lac . . . 299
8.253.2.8 lacValid . . . 299
8.253.2.9 MCC . . . 299
8.253.2.10 MNC . . . 299
8.253.2.11 networkIdValid . . . 299
8.253.2.12 regRejectInfoValid . . . 299
8.253.2.13 rejCause . . . 299
8.253.2.14 rejectSrvDomain . . . 299
8.253.2.15 sysInfoGSM . . . 299
8.254 gyroAcceptReady_s Struct Reference . . . 299
8.254.1 Detailed Description . . . 299
8.254.2 Field Documentation . . . 299
8.254.2.1 batchPerSec . . . 299
8.254.2.2 injectEnable . . . 300
8.254.2.3 samplesPerBatch . . . 300
8.255 gyroTempAcceptReady_s Struct Reference . . . 300
8.255.1 Detailed Description . . . 300
8.255.2 Field Documentation . . . 300
8.255.2.1 batchPerSec . . . 300
8.255.2.2 injectEnable . . . 300
8.255.2.3 samplesPerBatch . . . 300
8.256 HDRECIOThresh Struct Reference . . . 300

8.256.1 Detailed Description . . . 301
8.256.2 Field Documentation . . . 301
8.256.2.1 HDRECIOThreshListLen . . . 301
8.256.2.2 pHDRECIOThreshList . . . 301
8.257 HDRIOThresh Struct Reference . . . 301
8.257.1 Detailed Description . . . 301
8.257.2 Field Documentation . . . 301
8.257.2.1 HDRIOThreshListLen . . . 301
8.257.2.2 pHDRIOThreshList . . . 301
8.258 HDRPersonalityInd Struct Reference . . . 301
8.258.1 Field Documentation . . . 302
8.258.1.1 pCurrentPersonality . . . 302
8.258.1.2 pPersonalityListLength . . . 302
8.258.1.3 pProtocolSubtypeElement . . . 302
8.259 HDRPersonalityResp Struct Reference . . . 302
8.259.1 Detailed Description . . . 302
8.259.2 Field Documentation . . . 302
8.259.2.1 pCurrentPersonality . . . 302
8.259.2.2 pPersonalityListLength . . . 302
8.259.2.3 pProtocolSubtypeElement . . . 302
8.260 HDRProtSubtypResp Struct Reference . . . 302
8.260.1 Detailed Description . . . 303
8.260.2 Field Documentation . . . 303
8.260.2.1 pAppSubType . . . 303
8.260.2.2 pCurrentPrsnlty . . . 303
8.260.2.3 pPersonalityListLength . . . 303
8.260.2.4 pProtoSubTypElmnt . . . 303
8.261 HDRRSSIThresh Struct Reference . . . 303
8.261.1 Detailed Description . . . 303
8.261.2 Field Documentation . . . 304
8.261.2.1 HDRRSSIThreshListLen . . . 304
8.261.2.2 pHDRRSSIThreshList . . . 304
8.262 HDRSINRThresh Struct Reference . . . 304
8.262.1 Detailed Description . . . 304
8.262.2 Field Documentation . . . 304
8.262.2.1 HDRSINRThresListLen . . . 304
8.262.2.2 pHDRSINRThresList . . . 304
8.263 HDRSINRThreshold Struct Reference . . . 304
8.263.1 Detailed Description . . . 305
8.263.2 Field Documentation . . . 305

8.263.2.1 HDRSINRThreshListLen . . . 305
8.263.2.2 pHDRSINRThreshList . . . 305
8.264 HDRSSInfo Struct Reference . . . 305
8.264.1 Detailed Description . . . 305
8.264.2 Field Documentation . . . 306
8.264.2.1 ecio . . . 306
8.264.2.2 io . . . 306
8.264.2.3 rssi . . . 306
8.264.2.4 sinr . . . 306
8.265 hdrSSInfo Struct Reference . . . 306
8.265.1 Detailed Description . . . 306
8.265.2 Field Documentation . . . 307
8.265.2.1 ecio . . . 307
8.265.2.2 io . . . 307
8.265.2.3 rssi . . . 307
8.265.2.4 sinr . . . 307
8.266 HDRSysInfo Struct Reference . . . 307
8.266.1 Detailed Description . . . 308
8.266.2 Field Documentation . . . 309
8.266.2.1 hdrActiveProt . . . 309
8.266.2.2 hdrActiveProtValid . . . 309
8.266.2.3 hdrPersonality . . . 309
8.266.2.4 hdrPersonalityValid . . . 309
8.266.2.5 is856SysId . . . 309
8.266.2.6 is856SysIdValid . . . 309
8.266.2.7 isSysPrlMatch . . . 309
8.266.2.8 isSysPrlMatchValid . . . 309
8.266.2.9 sysInfoHDR . . . 309
8.267 homeSIDNID Struct Reference . . . 309
8.267.1 Detailed Description . . . 309
8.267.2 Field Documentation . . . 310
8.267.2.1 numInstances . . . 310
8.267.2.2 SidNid . . . 310
8.268 hotSwapStatus Struct Reference . . . 310
8.268.1 Detailed Description . . . 310
8.268.2 Field Documentation . . . 310
8.268.2.1 hotSwap . . . 310
8.268.2.2 hotSwapLength . . . 310
8.269 image_info_t Struct Reference . . . 310
8.269.1 Detailed Description . . . 310

8.269.2 Field Documentation . . . 311
8.269.2.1 buildID . . . 311
8.269.2.2 buildIDLen . . . 311
8.269.2.3 imageType . . . 311
8.269.2.4 uniqueID . . . 311
8.270 ImageElement Struct Reference . . . 311
8.270.1 Detailed Description . . . 311
8.270.2 Field Documentation . . . 312
8.270.2.1 buildId . . . 312
8.270.2.2 buildIdLength . . . 312
8.270.2.3 imageId . . . 312
8.270.2.4 imageType . . . 312
8.271 ImageIdElement Struct Reference . . . 312
8.271.1 Detailed Description . . . 312
8.271.2 Field Documentation . . . 312
8.271.2.1 buildID . . . 313
8.271.2.2 buildIDLength . . . 313
8.271.2.3 failureCount . . . 313
8.271.2.4 imageID . . . 313
8.271.2.5 storageIndex . . . 313
8.272 ImageIDEntries Struct Reference . . . 313
8.272.1 Detailed Description . . . 313
8.272.2 Field Documentation . . . 313
8.272.2.1 executingImage . . . 313
8.272.2.2 imageIDElement . . . 313
8.272.2.3 imageIDSize . . . 313
8.272.2.4 imageType . . . 313
8.272.2.5 maxImages . . . 314
8.273 ImageList Struct Reference . . . 314
8.273.1 Detailed Description . . . 314
8.273.2 Field Documentation . . . 314
8.273.2.1 imageIDEntries . . . 314
8.273.2.2 listSize . . . 314
8.274 IMSAIndRegisterInfo Struct Reference . . . 314
8.274.1 Detailed Description . . . 314
8.274.2 Field Documentation . . . 315
8.274.2.1 pPdpStatusConfig . . . 315
8.274.2.2 pRatHandoverStatusConfig . . . 315
8.274.2.3 pRegStatusConfig . . . 315
8.274.2.4 pServiceStatusConfig . . . 315

8.275imsaPdpStatusInfo Struct Reference . . . 315
8.275.1 Detailed Description . . . 315
8.275.2 Field Documentation . . . 316
8.275.2.1 connetionState . . . 316
8.275.2.2 pFailErrorCode . . . 316
8.276imsaRatStatusInfo Struct Reference . . . 316
8.276.1 Detailed Description . . . 316
8.276.2 Field Documentation . . . 316
8.276.2.1 pErrorCodeStr . . . 316
8.276.2.2 pRATStatus . . . 316
8.276.2.3 pSrcRAT . . . 316
8.276.2.4 pTgtRAT . . . 317
8.277IMSARegistrationStatus Struct Reference . . . 317
8.277.1 Detailed Description . . . 317
8.277.2 Field Documentation . . . 317
8.277.2.1 pImsRegErrCode . . . 317
8.277.2.2 pImsRegStatus . . . 317
8.277.2.3 pNewImsRegStatus . . . 317
8.278imsaRegStatusInfo Struct Reference . . . 317
8.278.1 Detailed Description . . . 318
8.278.2 Field Documentation . . . 318
8.278.2.1 pbIMSRegistered . . . 318
8.278.2.2 pImsRegStatus . . . 318
8.278.2.3 pRegStatusErrorCode . . . 318
8.279IMSAServiceStatus Struct Reference . . . 318
8.279.1 Detailed Description . . . 318
8.279.2 Field Documentation . . . 320
8.279.2.1 pSmsServiceRat . . . 320
8.279.2.2 pSmsServiceStatus . . . 320
8.279.2.3 pUtServiceRat . . . 320
8.279.2.4 pUtServiceStatus . . . 320
8.279.2.5 pVoipServiceRat . . . 320
8.279.2.6 pVoipServiceStatus . . . 320
8.279.2.7 pVsServiceRat . . . 320
8.279.2.8 pVsServiceStatus . . . 320
8.279.2.9 pVtServiceRat . . . 320
8.279.2.10pVtServiceStatus . . . 320
8.280IMSASupportedFieldsResp Struct Reference . . . 320
8.280.1 Detailed Description . . . 320
8.280.2 Field Documentation . . . 321

8.280.2.1 pIndFieldsList . . . 321
8.280.2.2 pReqFieldsList . . . 321
8.280.2.3 pRespFieldsList . . . 321
8.281 IMSASupportedMsgInfo Struct Reference . . . 321
8.281.1 Detailed Description . . . 321
8.281.2 Field Documentation . . . 321
8.281.2.1 pSupportedMsgList . . . 321
8.282 imsaSvcStatusInfo Struct Reference . . . 321
8.282.1 Detailed Description . . . 321
8.282.2 Field Documentation . . . 322
8.282.2.1 pSMSSvcRAT . . . 322
8.282.2.2 pSMSSvcStatus . . . 322
8.282.2.3 pUTSvcRAT . . . 322
8.282.2.4 pUTSvcStatus . . . 322
8.282.2.5 pVOIPSvcRAT . . . 322
8.282.2.6 pVOIPSvcStatus . . . 322
8.282.2.7 pVTSvcRAT . . . 322
8.282.2.8 pVTSvcStatus . . . 322
8.283 imsCfgIndRegisterInfo Struct Reference . . . 322
8.283.1 Detailed Description . . . 322
8.283.2 Field Documentation . . . 323
8.283.2.1 pRegMgrConfigEvents . . . 323
8.283.2.2 pSIPConfigEvents . . . 323
8.283.2.3 pSMSConfigEvents . . . 323
8.283.2.4 pUserConfigEvents . . . 323
8.283.2.5 pVoIPConfigEvents . . . 323
8.284 imsRegMgrConfigInfo Struct Reference . . . 323
8.284.1 Detailed Description . . . 324
8.284.2 Field Documentation . . . 324
8.284.2.1 pCSCFPortName . . . 324
8.284.2.2 pIMSTestMode . . . 324
8.284.2.3 pPriCSCFPort . . . 324
8.285 imsSIPConfigInfo Struct Reference . . . 324
8.285.1 Detailed Description . . . 324
8.285.2 Field Documentation . . . 325
8.285.2.1 pSigCompEnabled . . . 325
8.285.2.2 pSIPLocalPort . . . 325
8.285.2.3 pSubscribeTimer . . . 325
8.285.2.4 pTimerSIPReg . . . 325
8.285.2.5 pTimerT1 . . . 325

8.285.2.6 pTimerT2 . . . 325
8.285.2.7 pTimerTf . . . 325
8.286 imsSMSConfigInfo Struct Reference . . . 325
8.286.1 Detailed Description . . . 325
8.286.2 Field Documentation . . . 326
8.286.2.1 pPhoneCtxtURI . . . 326
8.286.2.2 pSMSFormat . . . 326
8.286.2.3 pSMSOverIPNwInd . . . 326
8.287 imsUserConfigInfo Struct Reference . . . 326
8.287.1 Detailed Description . . . 326
8.287.2 Field Documentation . . . 326
8.287.2.1 pIMSDomain . . . 326
8.288 imsVoIPConfigInfo Struct Reference . . . 326
8.288.1 Detailed Description . . . 327
8.288.2 Field Documentation . . . 328
8.288.2.1 pAmrMode . . . 328
8.288.2.2 pAmrOctetAligned . . . 328
8.288.2.3 pAmrWbEnable . . . 328
8.288.2.4 pAmrWBMode . . . 328
8.288.2.5 pAmrWBOctetAligned . . . 328
8.288.2.6 pMinSessionExpiryTimer . . . 328
8.288.2.7 pRingBackTimer . . . 329
8.288.2.8 pRingingTimer . . . 329
8.288.2.9 pRTPRTCPInactTimer . . . 329
8.288.2.10 pScrAmrEnable . . . 329
8.288.2.11 pScrAmrWbEnable . . . 329
8.288.2.12 pSessionExpiryTimer . . . 329
8.289 IndFieldsList Struct Reference . . . 329
8.289.1 Detailed Description . . . 329
8.289.2 Field Documentation . . . 329
8.289.2.1 indicationFields . . . 329
8.289.2.2 indicationFieldsLen . . . 329
8.290 infoInterFreq Struct Reference . . . 329
8.290.1 Detailed Description . . . 329
8.290.2 Field Documentation . . . 330
8.290.2.1 cell_resel_priority . . . 330
8.290.2.2 cellInterFreqParams . . . 330
8.290.2.3 cells_len . . . 330
8.290.2.4 earfcn . . . 330
8.290.2.5 threshXHigh . . . 330

8.290.2.6 threshXLow . . . 330
8.291 IOThresh Struct Reference . . . 330
8.291.1 Detailed Description . . . 331
8.291.2 Field Documentation . . . 331
8.291.2.1 IOThresListLen . . . 331
8.291.2.2 pIOThresList . . . 331
8.292 IPv4Addr Struct Reference . . . 331
8.292.1 Detailed Description . . . 331
8.292.2 Field Documentation . . . 331
8.292.2.1 addr . . . 331
8.292.2.2 subnetMask . . . 331
8.293 IPv4Info Struct Reference . . . 331
8.293.1 Detailed Description . . . 332
8.293.2 Field Documentation . . . 332
8.293.2.1 address . . . 332
8.293.2.2 port . . . 332
8.293.2.3 TlvPresent . . . 332
8.294 IPv6Addr Struct Reference . . . 332
8.294.1 Detailed Description . . . 332
8.294.2 Field Documentation . . . 332
8.294.2.1 addr . . . 332
8.294.2.2 prefixLen . . . 332
8.295 IPV6AddressInfo Struct Reference . . . 333
8.295.1 Detailed Description . . . 333
8.295.2 Field Documentation . . . 333
8.295.2.1 IPAddressV6 . . . 333
8.295.2.2 IPV6PrefixLen . . . 333
8.296 ipv6AddressInfo Struct Reference . . . 333
8.296.1 Detailed Description . . . 333
8.296.2 Field Documentation . . . 333
8.296.2.1 IPAddressV6 . . . 333
8.296.2.2 IPV6PrefixLen . . . 333
8.297 IPV6GWAddressInfo Struct Reference . . . 334
8.297.1 Detailed Description . . . 334
8.297.2 Field Documentation . . . 334
8.297.2.1 gwAddressV6 . . . 334
8.297.2.2 gwV6PrefixLen . . . 334
8.298 IPv6Info Struct Reference . . . 334
8.298.1 Detailed Description . . . 334
8.298.2 Field Documentation . . . 335

8.298.2.1 address . . . 335
8.298.2.2 port . . . 335
8.298.2.3 TlvPresent . . . 335
8.299 IPv6TrafCls Struct Reference . . . 335
8.299.1 Detailed Description . . . 335
8.299.2 Field Documentation . . . 335
8.299.2.1 mask . . . 335
8.299.2.2 val . . . 335
8.300 LibPackGPRSRequestedQoS Struct Reference . . . 335
8.300.1 Detailed Description . . . 335
8.300.2 Field Documentation . . . 336
8.300.2.1 delayClass . . . 336
8.300.2.2 meanThroughputClass . . . 336
8.300.2.3 peakThroughputClass . . . 336
8.300.2.4 precedenceClass . . . 336
8.300.2.5 reliabilityClass . . . 336
8.301 LibpackProfile3GPP Struct Reference . . . 336
8.301.1 Detailed Description . . . 337
8.301.2 Field Documentation . . . 341
8.301.2.1 pAddrAllocPref . . . 341
8.301.2.2 pAPNClass . . . 341
8.301.2.3 pAPNDisabledFlag . . . 341
8.301.2.4 pAPNName . . . 342
8.301.2.5 pAPNnameSize . . . 342
8.301.2.6 pAuthenticationPref . . . 342
8.301.2.7 pGPRSMinimumQoS . . . 342
8.301.2.8 pGPRSRequestedQos . . . 342
8.301.2.9 pImCnFlag . . . 342
8.301.2.10 pIPv4AddrPref . . . 342
8.301.2.11 pIPv6AddPref . . . 342
8.301.2.12 pPassword . . . 342
8.301.2.13 pPasswordSize . . . 342
8.301.2.14 pPcscfAddrUsingDhcp . . . 342
8.301.2.15 pPcscfAddrUsingPCO . . . 342
8.301.2.16 pPDNInactivTimeout . . . 342
8.301.2.17 pPdpAccessConFlag . . . 342
8.301.2.18 pPdpContext . . . 342
8.301.2.19 pPdpDataCompType . . . 342
8.301.2.20 pPdpHdrCompType . . . 342
8.301.2.21 pPDPtype . . . 342

8.301.2.22 pPriDNSIPv4AddPref . . . 342
8.301.2.23 pPriDNSIPv6addpref . . . 342
8.301.2.24 pPrimaryID . . . 342
8.301.2.25 pProfilename . . . 342
8.301.2.26 pProfilenameSize . . . 342
8.301.2.27 pQosClassID . . . 342
8.301.2.28 pSecDNSIPv4AddPref . . . 342
8.301.2.29 pSecDNSIPv6addpref . . . 342
8.301.2.30 pSecondaryFlag . . . 342
8.301.2.31 pTFTID1Params . . . 342
8.301.2.32 pTFTID2Params . . . 343
8.301.2.33 pUMTSMinQoS . . . 343
8.301.2.34 pUMTSMinQosSigInd . . . 343
8.301.2.35 pUMTSReqQoS . . . 343
8.301.2.36 pUMTSReqQoSSigInd . . . 343
8.301.2.37 pUsername . . . 343
8.301.2.38 pUsernameSize . . . 343
8.302 LibpackProfile3GPP2 Struct Reference . . . 343
8.302.1 Detailed Description . . . 344
8.302.2 Field Documentation . . . 347
8.302.2.1 pAllowLinger . . . 347
8.302.2.2 pAPNClass3GPP2 . . . 347
8.302.2.3 pAPNEnabled3GPP2 . . . 347
8.302.2.4 pApnString . . . 347
8.302.2.5 pApnStringSize . . . 347
8.302.2.6 pAppPriority . . . 347
8.302.2.7 pAppType . . . 347
8.302.2.8 pAuthPassword . . . 347
8.302.2.9 pAuthPasswordSize . . . 347
8.302.2.10 pAuthProtocol . . . 347
8.302.2.11 pAuthRetryCount . . . 347
8.302.2.12 pAuthTimeout . . . 347
8.302.2.13 pDataMode . . . 347
8.302.2.14 pDataRate . . . 347
8.302.2.15 pIpcpAckTimeout . . . 348
8.302.2.16 pIpcpCreqRetryCount . . . 348
8.302.2.17 pIsPcscfAddressNedded . . . 348
8.302.2.18 pLcpAckTimeout . . . 348
8.302.2.19 pLcpCreqRetryCount . . . 348
8.302.2.20 pNegoDnsSrvrPref . . . 348

8.302.2.21 pPDNInactivTimeout3GPP2 . . . 348
8.302.2.22 pPdnType . . . 348
8.302.2.23 pPppSessCloseTimer1x . . . 348
8.302.2.24 pPppSessCloseTimerDO . . . 348
8.302.2.25 pPrimaryV4DnsAddress . . . 348
8.302.2.26 pPriV6DnsAddress . . . 348
8.302.2.27 pRATType . . . 348
8.302.2.28 pSecondaryV4DnsAddress . . . 348
8.302.2.29 pSecV6DnsAddress . . . 348
8.302.2.30 pUserId . . . 348
8.302.2.31 pUserIdSize . . . 348
8.303 LibPackprofile_3GPP Struct Reference . . . 348
8.303.1 Detailed Description . . . 349
8.303.2 Field Documentation . . . 353
8.303.2.1 pAddrAllocPref . . . 353
8.303.2.2 pAPNClass . . . 353
8.303.2.3 pAPNDisabledFlag . . . 353
8.303.2.4 pAPNName . . . 353
8.303.2.5 pAPNnameSize . . . 353
8.303.2.6 pAuthenticationPref . . . 353
8.303.2.7 pGPRSMinimumQoS . . . 353
8.303.2.8 pGPRSRequestedQos . . . 353
8.303.2.9 pImCnFlag . . . 353
8.303.2.10 pIPv4AddrPref . . . 353
8.303.2.11 pIPv6AddPref . . . 353
8.303.2.12 pPassword . . . 353
8.303.2.13 pPasswordSize . . . 353
8.303.2.14 pPcscfAddrUsingDhcp . . . 353
8.303.2.15 pPcscfAddrUsingPCO . . . 353
8.303.2.16 pPDNInactivTimeout . . . 353
8.303.2.17 pPdpAccessConFlag . . . 353
8.303.2.18 pPdpContext . . . 353
8.303.2.19 pPdpDataCompType . . . 353
8.303.2.20 pPdpHdrCompType . . . 353
8.303.2.21 pPDPtype . . . 353
8.303.2.22 pPriDNSIPv4AddPref . . . 353
8.303.2.23 pPriDNSIPv6addpref . . . 353
8.303.2.24 pPrimaryID . . . 354
8.303.2.25 pProfilename . . . 354
8.303.2.26 pProfilenameSize . . . 354

8.303.2.27 pQosClassID . . . 354
8.303.2.28 pSecDNSIPv4AddPref . . . 354
8.303.2.29 pSecDNSIPv6addpref . . . 354
8.303.2.30 pSecondaryFlag . . . 354
8.303.2.31 pTFTID1Params . . . 354
8.303.2.32 pTFTID2Params . . . 354
8.303.2.33 pUMTSMinQoS . . . 354
8.303.2.34 pUMTSMinQosSigInd . . . 354
8.303.2.35 pUMTSReqQoS . . . 354
8.303.2.36 pUMTSReqQoSSigInd . . . 354
8.303.2.37 pUsername . . . 354
8.303.2.38 pUsernameSize . . . 354
8.304 LibPackprofile_3GPP2 Struct Reference . . . 354
8.304.1 Detailed Description . . . 355
8.304.2 Field Documentation . . . 358
8.304.2.1 pAllowLinger . . . 358
8.304.2.2 pAPNClass3GPP2 . . . 358
8.304.2.3 pAPNEnabled3GPP2 . . . 358
8.304.2.4 pApnString . . . 358
8.304.2.5 pApnStringSize . . . 358
8.304.2.6 pAppPriority . . . 358
8.304.2.7 pAppType . . . 358
8.304.2.8 pAuthPassword . . . 358
8.304.2.9 pAuthPassword_tSize . . . 358
8.304.2.10 pAuthProtocol . . . 359
8.304.2.11 pAuthRetryCount . . . 359
8.304.2.12 pAuthTimeout . . . 359
8.304.2.13 pDataMode . . . 359
8.304.2.14 pDataRate . . . 359
8.304.2.15 pIpcpAckTimeout . . . 359
8.304.2.16 pIpcpCreqRetryCount . . . 359
8.304.2.17 pIsPcscfAddressNedded . . . 359
8.304.2.18 pLcpAckTimeout . . . 359
8.304.2.19 pLcpCreqRetryCount . . . 359
8.304.2.20 pNegoDnsSrvrPref . . . 359
8.304.2.21 pPDNInactivTimeout3GPP2 . . . 359
8.304.2.22 pPdnType . . . 359
8.304.2.23 pPppSessCloseTimer1x . . . 359
8.304.2.24 pPppSessCloseTimerDO . . . 359
8.304.2.25 pPrimaryV4DnsAddress . . . 359

8.304.2.26 pPriV6DnsAddress . . . 359
8.304.2.27 pRATType . . . 359
8.304.2.28 pSecondaryV4DnsAddress . . . 359
8.304.2.29 pSecV6DnsAddress . . . 359
8.304.2.30 pUserId . . . 359
8.304.2.31 pUserIdSize . . . 359
8.305 LibPackQosClassID Struct Reference . . . 359
8.305.1 Detailed Description . . . 360
8.305.2 Field Documentation . . . 360
8.305.2.1 gDlBitRate . . . 360
8.305.2.2 gUlBitRate . . . 360
8.305.2.3 maxDlBitRate . . . 360
8.305.2.4 maxUlBitRate . . . 360
8.305.2.5 QCI . . . 360
8.306 LibPackTFTIDParams Struct Reference . . . 360
8.306.1 Detailed Description . . . 361
8.306.2 Field Documentation . . . 361
8.306.2.1 destPortRangeEnd . . . 362
8.306.2.2 destPortRangeStart . . . 362
8.306.2.3 eValid . . . 362
8.306.2.4 filterId . . . 362
8.306.2.5 flowLabel . . . 362
8.306.2.6 IPSECSPI . . . 362
8.306.2.7 ipVersion . . . 362
8.306.2.8 nextHeader . . . 362
8.306.2.9 pSourceIP . . . 362
8.306.2.10 sourceIPMask . . . 362
8.306.2.11 srcPortRangeEnd . . . 362
8.306.2.12 srcPortRangeStart . . . 362
8.306.2.13 tosMask . . . 362
8.307 LibPackUMTSQoS Struct Reference . . . 362
8.307.1 Detailed Description . . . 362
8.307.2 Field Documentation . . . 364
8.307.2.1 deliveryErrSDU . . . 364
8.307.2.2 grntDownlinkBitrate . . . 364
8.307.2.3 grntUplinkBitrate . . . 364
8.307.2.4 maxDownlinkBitrate . . . 364
8.307.2.5 maxSDUSize . . . 364
8.307.2.6 maxUplinkBitrate . . . 364
8.307.2.7 qosDeliveryOrder . . . 365

8.307.2.8 resBerRatio . . . 365
8.307.2.9 sduErrorRatio . . . 365
8.307.2.10 trafficClass . . . 365
8.307.2.11 trafficPriority . . . 365
8.307.2.12 transferDelay . . . 365
8.308 LibPackUMTSReqQoSSigInd Struct Reference . . . 365
8.308.1 Detailed Description . . . 365
8.308.2 Field Documentation . . . 365
8.308.2.1 SigInd . . . 365
8.308.2.2 UMTSReqQoS . . . 365
8.309 lineCtrlInfo Struct Reference . . . 365
8.309.1 Detailed Description . . . 366
8.309.2 Field Documentation . . . 366
8.309.2.1 polarityIncluded . . . 366
8.309.2.2 pwrDenialTime . . . 366
8.309.2.3 revPolarity . . . 366
8.309.2.4 toggleMode . . . 366
8.310 loc_BdsSV Struct Reference . . . 366
8.310.1 Detailed Description . . . 366
8.310.2 Field Documentation . . . 367
8.310.2.1 id . . . 367
8.310.2.2 mask . . . 367
8.311 loc_BdsSVInfo Struct Reference . . . 367
8.311.1 Detailed Description . . . 367
8.311.2 Field Documentation . . . 367
8.311.2.1 len . . . 367
8.311.2.2 pSV . . . 367
8.312 loc_CellDb Struct Reference . . . 367
8.312.1 Detailed Description . . . 367
8.312.2 Field Documentation . . . 368
8.312.2.1 mask . . . 368
8.313 loc_ClkInfo Struct Reference . . . 368
8.313.1 Detailed Description . . . 368
8.313.2 Field Documentation . . . 369
8.313.2.1 mask . . . 369
8.314 loc_GnssData Struct Reference . . . 369
8.314.1 Detailed Description . . . 370
8.314.2 Field Documentation . . . 370
8.314.2.1 mask . . . 371
8.315 loc_gpsTime Struct Reference . . . 371

8.315.1 Detailed Description . . . 371
8.315.2 Field Documentation . . . 371
8.315.2.1 gpsTimeOfWeekMs . . . 371
8.315.2.2 gpsWeek . . . 371
8.316loc_IPv4Info Struct Reference . . . 371
8.316.1 Detailed Description . . . 371
8.316.2 Field Documentation . . . 372
8.316.2.1 address . . . 372
8.316.2.2 port . . . 372
8.316.2.3 TlvPresent . . . 372
8.317loc_IPv6Info Struct Reference . . . 372
8.317.1 Detailed Description . . . 372
8.317.2 Field Documentation . . . 372
8.317.2.1 address . . . 372
8.317.2.2 port . . . 372
8.317.2.3 TlvPresent . . . 372
8.318loc_LocApplicationInfo Struct Reference . . . 372
8.318.1 Detailed Description . . . 373
8.318.2 Field Documentation . . . 373
8.318.2.1 appNameLength . . . 373
8.318.2.2 appProviderLength . . . 373
8.318.2.3 appVersionLength . . . 373
8.318.2.4 appVersionValid . . . 373
8.318.2.5 pAppName . . . 373
8.318.2.6 pAppProvider . . . 373
8.318.2.7 pAppVersion . . . 373
8.319loc_precisionDilution Struct Reference . . . 373
8.319.1 Detailed Description . . . 374
8.319.2 Field Documentation . . . 374
8.319.2.1 HDOP . . . 374
8.319.2.2 PDOP . . . 374
8.319.2.3 VDOP . . . 374
8.320loc_satelliteInfo Struct Reference . . . 374
8.320.1 Detailed Description . . . 374
8.320.2 Field Documentation . . . 376
8.320.2.1 azimuth . . . 376
8.320.2.2 elevation . . . 376
8.320.2.3 gnssSvId . . . 376
8.320.2.4 healthStatus . . . 376
8.320.2.5 snr . . . 376

8.320.2.6 svInfoMask . . . 376
8.320.2.7 svListLen . . . 376
8.320.2.8 svStatus . . . 376
8.320.2.9 system . . . 376
8.320.2.10validMask . . . 376
8.321loc_sensorDataUsage Struct Reference . . . 376
8.321.1 Detailed Description . . . 376
8.321.2 Field Documentation . . . 377
8.321.2.1 aidingIndicatorMask . . . 377
8.321.2.2 usageMask . . . 377
8.322loc_SV Struct Reference . . . 377
8.322.1 Detailed Description . . . 377
8.322.2 Field Documentation . . . 378
8.322.2.1 id . . . 378
8.322.2.2 mask . . . 378
8.322.2.3 system . . . 378
8.323loc_SVInfo Struct Reference . . . 378
8.323.1 Detailed Description . . . 378
8.323.2 Field Documentation . . . 378
8.323.2.1 len . . . 378
8.323.2.2 pSV . . . 378
8.324loc_svUsedforFix Struct Reference . . . 378
8.324.1 Detailed Description . . . 379
8.324.2 Field Documentation . . . 379
8.324.2.1 gnssSvUsedList . . . 379
8.324.2.2 gnssSvUsedList_len . . . 379
8.325loc_urlAddr Struct Reference . . . 379
8.325.1 Detailed Description . . . 379
8.325.2 Field Documentation . . . 379
8.325.2.1 address . . . 379
8.325.2.2 TlvPresent . . . 379
8.326LocApplicationInfo Struct Reference . . . 380
8.326.1 Detailed Description . . . 380
8.326.2 Field Documentation . . . 380
8.326.2.1 appNameLength . . . 380
8.326.2.2 appProviderLength . . . 380
8.326.2.3 appVersionLength . . . 380
8.326.2.4 appVersionValid . . . 381
8.326.2.5 pAppName . . . 381
8.326.2.6 pAppProvider . . . 381

8.326.2.7 pAppVersion . . . 381
8.327 LocDelAssDataReq Struct Reference . . . 381
8.327.1 Detailed Description . . . 381
8.327.2 Field Documentation . . . 381
8.327.2.1 pBdsSVInfo . . . 381
8.327.2.2 pCellDb . . . 381
8.327.2.3 pClkInfo . . . 381
8.327.2.4 pGnssData . . . 382
8.327.2.5 pSVInfo . . . 382
8.328 LOCEventRegisterReqResp Struct Reference . . . 382
8.328.1 Detailed Description . . . 382
8.328.2 Field Documentation . . . 384
8.328.2.1 eventRegister . . . 384
8.329 LOCExtPowerStateReqResp Struct Reference . . . 384
8.329.1 Detailed Description . . . 384
8.329.2 Field Documentation . . . 384
8.329.2.1 extPowerState . . . 384
8.330 LocGetServerReq Struct Reference . . . 384
8.330.1 Detailed Description . . . 384
8.330.2 Field Documentation . . . 385
8.330.2.1 pServerAddrTypeMask . . . 385
8.330.2.2 serverType . . . 385
8.331 LocInjectPositionReq Struct Reference . . . 385
8.331.1 Detailed Description . . . 385
8.331.2 Field Documentation . . . 388
8.331.2.1 pAltitudeSrcInfo . . . 388
8.331.2.2 pAltitudeWrtEllipsoid . . . 388
8.331.2.3 pAltitudeWrtMeanSeaLevel . . . 388
8.331.2.4 pHorConfidence . . . 389
8.331.2.5 pHorReliability . . . 389
8.331.2.6 pHorUncCircular . . . 389
8.331.2.7 pLatitude . . . 389
8.331.2.8 pLongitude . . . 389
8.331.2.9 pPositionSrc . . . 389
8.331.2.10 pRawHorConfidence . . . 389
8.331.2.11 pRawHorUncCircular . . . 389
8.331.2.12 pTimestampAge . . . 389
8.331.2.13 pTimestampUtc . . . 389
8.331.2.14 pVertConfidence . . . 389
8.331.2.15 pVertReliability . . . 389

8.331.2.16 pVertUnc . . . 389
8.332 LocInjectSensorDataReq Struct Reference . . . 389
8.332.1 Detailed Description . . . 389
8.332.2 Field Documentation . . . 390
8.332.2.1 pAcceleroData . . . 390
8.332.2.2 pAcceleroTempData . . . 390
8.332.2.3 pAcceleroTimeSrc . . . 390
8.332.2.4 pGyroData . . . 390
8.332.2.5 pGyroTempData . . . 390
8.332.2.6 pGyroTimeSrc . . . 390
8.332.2.7 pOpaqueIdentifier . . . 390
8.333 LocSetCradleMountReq Struct Reference . . . 390
8.333.1 Detailed Description . . . 391
8.333.2 Field Documentation . . . 391
8.333.2.1 pConfidence . . . 391
8.333.2.2 state . . . 391
8.334 LOCStartReq Struct Reference . . . 391
8.334.1 Detailed Description . . . 391
8.334.2 Field Documentation . . . 392
8.334.2.1 pApplicationInfo . . . 392
8.334.2.2 pConfigAltitudeAssumed . . . 392
8.334.2.3 pHorizontalAccuracyLvl . . . 393
8.334.2.4 pIntermediateReportState . . . 393
8.334.2.5 pMinIntervalTime . . . 393
8.334.2.6 pRecurrenceType . . . 393
8.334.2.7 SessionId . . . 393
8.335 LOCStopReq Struct Reference . . . 393
8.335.1 Detailed Description . . . 393
8.335.2 Field Documentation . . . 393
8.335.2.1 sessionId . . . 393
8.336 LteCQIParm Struct Reference . . . 393
8.336.1 Detailed Description . . . 393
8.336.2 Field Documentation . . . 394
8.336.2.1 CQIValueCW0 . . . 394
8.336.2.2 CQIValueCW1 . . . 394
8.336.2.3 ValidityCW0 . . . 394
8.336.2.4 ValidityCW1 . . . 394
8.337 lteEARFCN Struct Reference . . . 394
8.337.1 Detailed Description . . . 394
8.337.2 Field Documentation . . . 394

8.337.2.1 earfcn0 . . . 395
8.337.2.2 earfcn1 . . . 395
8.337.2.3 status . . . 395
8.338 lteGsmCellInfo Struct Reference . . . 395
8.338.1 Detailed Description . . . 395
8.338.2 Field Documentation . . . 396
8.338.2.1 cellReselPriority . . . 396
8.338.2.2 cells_len . . . 396
8.338.2.3 GsmCellInfo . . . 396
8.338.2.4 nccPermitted . . . 396
8.338.2.5 threshGsmHigh . . . 396
8.338.2.6 threshGsmLow . . . 396
8.339 LTEInfo Struct Reference . . . 396
8.339.1 Detailed Description . . . 396
8.339.2 Field Documentation . . . 398
8.339.2.1 band . . . 398
8.339.2.2 bandwidth . . . 398
8.339.2.3 emmConnState . . . 398
8.339.2.4 emmState . . . 398
8.339.2.5 emmSubState . . . 398
8.339.2.6 RXChan . . . 398
8.339.2.7 TXChan . . . 398
8.340 LTEInfoInterfreq Struct Reference . . . 398
8.340.1 Detailed Description . . . 398
8.340.2 Field Documentation . . . 398
8.340.2.1 freqsLen . . . 398
8.340.2.2 InfoInterfreq . . . 398
8.340.2.3 ueInIdle . . . 398
8.341 LTEInfoIntrafreq Struct Reference . . . 398
8.341.1 Detailed Description . . . 399
8.341.2 Field Documentation . . . 400
8.341.2.1 CellParams . . . 400
8.341.2.2 cellReselPriority . . . 400
8.341.2.3 cellsLen . . . 400
8.341.2.4 earfcn . . . 400
8.341.2.5 globalCellId . . . 400
8.341.2.6 plmn . . . 400
8.341.2.7 servingCellId . . . 400
8.341.2.8 sIntraSearch . . . 400
8.341.2.9 sNonIntraSearch . . . 400

8.341.2.10 tac . . . 400
8.341.2.11 threshServingLow . . . 400
8.341.2.12 ueInIdle . . . 401
8.342 LTEInfoNeighboringGSM Struct Reference . . . 401
8.342.1 Detailed Description . . . 401
8.342.2 Field Documentation . . . 401
8.342.2.1 freqsLen . . . 401
8.342.2.2 LteGsmCellInfo . . . 401
8.342.2.3 ueInIdle . . . 401
8.343 LTEInfoNeighboringWCDMA Struct Reference . . . 401
8.343.1 Detailed Description . . . 401
8.343.2 Field Documentation . . . 402
8.343.2.1 freqsLen . . . 402
8.343.2.2 LTEWCDMACellInfo . . . 402
8.343.2.3 ueInIdle . . . 402
8.344 LteNasReleaseInfo_s Struct Reference . . . 402
8.344.1 Detailed Description . . . 402
8.344.2 Field Documentation . . . 402
8.344.2.1 nas_major . . . 402
8.344.2.2 nas_minor . . . 402
8.344.2.3 nas_release . . . 403
8.345 ltePCI Struct Reference . . . 403
8.345.1 Detailed Description . . . 403
8.345.2 Field Documentation . . . 403
8.345.2.1 earfcn . . . 403
8.345.2.2 pci . . . 403
8.345.2.3 status . . . 403
8.346 lteRsrpinformation Struct Reference . . . 403
8.346.1 Detailed Description . . . 403
8.346.2 Field Documentation . . . 404
8.346.2.1 rsrplevel . . . 404
8.347 LTERSRPThresh Struct Reference . . . 404
8.347.1 Detailed Description . . . 404
8.347.2 Field Documentation . . . 404
8.347.2.1 LTERSRPThreshListLen . . . 404
8.347.2.2 pLTERSRPThreshList . . . 404
8.348 LTERSRQThresh Struct Reference . . . 404
8.348.1 Detailed Description . . . 404
8.348.2 Field Documentation . . . 405
8.348.2.1 LTERSRQThreshListLen . . . 405

8.348.2.2 pLTERSRQThreshList . . . 405
8.349 LTERSSIThresh Struct Reference . . . 405
8.349.1 Detailed Description . . . 405
8.349.2 Field Documentation . . . 405
8.349.2.1 LTERSSIThreshListLen . . . 405
8.349.2.2 pLTERSSIThreshList . . . 405
8.350 LteSccRxInfoResp Struct Reference . . . 405
8.350.1 Detailed Description . . . 405
8.350.2 Field Documentation . . . 405
8.350.2.1 pSccRxInfo . . . 406
8.351 LTESigRptCfg Struct Reference . . . 406
8.351.1 Detailed Description . . . 406
8.351.2 Field Documentation . . . 406
8.351.2.1 avgPeriod . . . 406
8.351.2.2 rptRate . . . 406
8.352 LTESigRptConfig Struct Reference . . . 406
8.352.1 Detailed Description . . . 407
8.352.2 Field Documentation . . . 407
8.352.2.1 avgPeriod . . . 407
8.352.2.2 rptRate . . . 407
8.353 lteSnrinformation Struct Reference . . . 407
8.353.1 Detailed Description . . . 407
8.353.2 Field Documentation . . . 408
8.353.2.1 snrlevel . . . 408
8.354 LTESNRThresh Struct Reference . . . 408
8.354.1 Detailed Description . . . 408
8.354.2 Field Documentation . . . 408
8.354.2.1 LTESNRThresListLen . . . 408
8.354.2.2 pLTESNRThresList . . . 408
8.355 LTESNRThreshold Struct Reference . . . 408
8.355.1 Detailed Description . . . 409
8.355.2 Field Documentation . . . 409
8.355.2.1 LTESNRThreshListLen . . . 409
8.355.2.2 pLTESNRThreshList . . . 409
8.356 LTESSInfo Struct Reference . . . 409
8.356.1 Detailed Description . . . 409
8.356.2 Field Documentation . . . 410
8.356.2.1 rsrp . . . 410
8.356.2.2 rsrq . . . 410
8.356.2.3 rssi . . . 410

8.356.2.4 snr . . . 410
8.357 lteSSInfo Struct Reference . . . 410
8.357.1 Detailed Description . . . 410
8.357.2 Field Documentation . . . 410
8.357.2.1 rsrp . . . 410
8.357.2.2 rsrq . . . 410
8.357.2.3 rssi . . . 410
8.357.2.4 snr . . . 410
8.358 LTESysInfo Struct Reference . . . 411
8.358.1 Detailed Description . . . 411
8.358.2 Field Documentation . . . 412
8.358.2.1 cellId . . . 412
8.358.2.2 cellIdValid . . . 412
8.358.2.3 lac . . . 412
8.358.2.4 lacValid . . . 412
8.358.2.5 MCC . . . 413
8.358.2.6 MNC . . . 413
8.358.2.7 networkIdValid . . . 413
8.358.2.8 regRejectInfoValid . . . 413
8.358.2.9 rejCause . . . 413
8.358.2.10 rejectSrvDomain . . . 413
8.358.2.11 sysInfoLTE . . . 413
8.358.2.12 tac . . . 413
8.358.2.13 tacValid . . . 413
8.359 lteWcdmaCellInfo Struct Reference . . . 413
8.359.1 Detailed Description . . . 413
8.359.2 Field Documentation . . . 414
8.359.2.1 cellReselPriority . . . 414
8.359.2.2 cellsLen . . . 414
8.359.2.3 threshXhigh . . . 414
8.359.2.4 threshXlow . . . 414
8.359.2.5 uarfcn . . . 414
8.359.2.6 WCDMACellInfo . . . 414
8.360 messageModeTlv Struct Reference . . . 414
8.360.1 Detailed Description . . . 414
8.360.2 Field Documentation . . . 414
8.360.2.1 MessageModeInfo . . . 414
8.360.2.2 TlvPresent . . . 414
8.361 messageWaitingInfoContent Struct Reference . . . 414
8.361.1 Detailed Description . . . 415

8.361.2 Field Documentation . . . 415
8.361.2.1 activeInd . . . 415
8.361.2.2 msgCount . . . 415
8.361.2.3 msgType . . . 415
8.362 minBasedIMSI Struct Reference . . . 415
8.362.1 Detailed Description . . . 415
8.362.2 Field Documentation . . . 416
8.362.2.1 imsiM1112 . . . 416
8.362.2.2 imsiMS1 . . . 416
8.362.2.3 imsiMS2 . . . 416
8.362.2.4 mccM . . . 416
8.363 mitigationDevList Struct Reference . . . 416
8.363.1 Detailed Description . . . 416
8.363.2 Field Documentation . . . 416
8.363.2.1 maxMitigationLevel . . . 417
8.363.2.2 mitigationDevId . . . 417
8.363.2.3 mitigationDevIdLen . . . 417
8.364 MNRInfo Struct Reference . . . 417
8.364.1 Detailed Description . . . 417
8.364.2 Field Documentation . . . 417
8.364.2.1 mcc . . . 417
8.364.2.2 mnc . . . 417
8.364.2.3 rat . . . 417
8.365 ModifyProfileIn Struct Reference . . . 417
8.365.1 Detailed Description . . . 417
8.365.2 Field Documentation . . . 418
8.365.2.1 curProfile . . . 418
8.365.2.2 pProfileID . . . 418
8.365.2.3 pProfileType . . . 418
8.366 ModifyProfileOut Struct Reference . . . 418
8.366.1 Detailed Description . . . 418
8.366.2 Field Documentation . . . 418
8.366.2.1 pExtErrorCode . . . 418
8.367 msgWaitingInfo Struct Reference . . . 418
8.367.1 Detailed Description . . . 419
8.367.2 Field Documentation . . . 419
8.367.2.1 msgWaitInfo . . . 419
8.367.2.2 numInstances . . . 419
8.368 namName Struct Reference . . . 419
8.368.1 Detailed Description . . . 419

8.368.2 Field Documentation . . . 419
8.368.2.1 namName . . . 419
8.368.2.2 namNameLen . . . 419
8.369 nas_acqOrderPref Struct Reference . . . 420
8.369.1 Detailed Description . . . 420
8.369.2 Field Documentation . . . 420
8.369.2.1 acqOrdeLen . . . 420
8.369.2.2 pAcqOrder . . . 420
8.370 nas_ActPilotPNElement Struct Reference . . . 420
8.370.1 Detailed Description . . . 420
8.370.2 Field Documentation . . . 421
8.370.2.1 ActSetPilotPN . . . 421
8.370.2.2 ActSetPilotPNStrength . . . 421
8.371 nas_AddCDMASysInfo Struct Reference . . . 421
8.371.1 Detailed Description . . . 421
8.371.2 Field Documentation . . . 421
8.371.2.1 geoSysIdx . . . 421
8.371.2.2 regPrd . . . 421
8.372 nas_AddSysInfo Struct Reference . . . 421
8.372.1 Detailed Description . . . 421
8.372.2 Field Documentation . . . 422
8.372.2.1 cellBroadcastCap . . . 422
8.372.2.2 geoSysIdx . . . 422
8.373 nas_CallBarringSysInfo Struct Reference . . . 422
8.373.1 Detailed Description . . . 422
8.373.2 Field Documentation . . . 422
8.373.2.1 csBarStatus . . . 422
8.373.2.2 psBarStatus . . . 423
8.374 nas_callBarStatus Struct Reference . . . 423
8.374.1 Detailed Description . . . 423
8.374.2 Field Documentation . . . 423
8.374.2.1 csBarStatus . . . 423
8.374.2.2 psBarStatus . . . 423
8.375 nas_CDMAChannel Struct Reference . . . 423
8.375.1 Detailed Description . . . 424
8.375.2 Field Documentation . . . 424
8.375.2.1 priChA . . . 424
8.375.2.2 priChB . . . 424
8.375.2.3 secChA . . . 424
8.375.2.4 secChB . . . 424

8.376 nas_CDMAECIOThresh Struct Reference . . . 424
8.376.1 Detailed Description . . . 424
8.376.2 Field Documentation . . . 425
8.376.2.1 CDMAECIOThreshListLen . . . 425
8.376.2.2 pCDMAECIOThreshList . . . 425
8.377 nas_CDMAInfo Struct Reference . . . 425
8.377.1 Detailed Description . . . 425
8.377.2 Field Documentation . . . 425
8.377.2.1 baseId . . . 425
8.377.2.2 baseLat . . . 426
8.377.2.3 baseLong . . . 426
8.377.2.4 nid . . . 426
8.377.2.5 refpn . . . 426
8.377.2.6 sid . . . 426
8.378 nas_CDMARSSIThresh Struct Reference . . . 426
8.378.1 Detailed Description . . . 426
8.378.2 Field Documentation . . . 426
8.378.2.1 CDMARSSIThreshListLen . . . 426
8.378.2.2 pCDMARSSIThreshList . . . 426
8.379 nas_CDMASysInfo Struct Reference . . . 426
8.379.1 Detailed Description . . . 427
8.379.2 Field Documentation . . . 429
8.379.2.1 baseId . . . 429
8.379.2.2 baseLat . . . 429
8.379.2.3 baseLong . . . 429
8.379.2.4 bsInfoValid . . . 429
8.379.2.5 bsPRev . . . 429
8.379.2.6 bsPRevValid . . . 429
8.379.2.7 ccsSupported . . . 429
8.379.2.8 ccsSupportedValid . . . 429
8.379.2.9 cdmaSysIdValid . . . 429
8.379.2.10 isSysPrlMatch . . . 429
8.379.2.11 isSysPrlMatchValid . . . 429
8.379.2.12 MCC . . . 430
8.379.2.13 MNC . . . 430
8.379.2.14 networkID . . . 430
8.379.2.15 networkIdValid . . . 430
8.379.2.16 packetZone . . . 430
8.379.2.17 packetZoneValid . . . 430
8.379.2.18 pRevInUse . . . 430

8.379.2.19 pRevInUseValid . . . 430
8.379.2.20 sysInfoCDMA . . . 430
8.379.2.21 systemID . . . 430
8.380 nas_CDMASysInfoExt Struct Reference . . . 430
8.380.1 Detailed Description . . . 430
8.380.2 Field Documentation . . . 430
8.380.2.1 imsi_11_12 . . . 430
8.380.2.2 MCC . . . 430
8.381 nas_cellParams Struct Reference . . . 430
8.381.1 Detailed Description . . . 431
8.381.2 Field Documentation . . . 431
8.381.2.1 pci . . . 431
8.381.2.2 rsrp . . . 431
8.381.2.3 rsrq . . . 431
8.381.2.4 rssi . . . 431
8.381.2.5 srxlev . . . 431
8.382 nas_CommInfo Struct Reference . . . 431
8.382.1 Detailed Description . . . 432
8.382.2 Field Documentation . . . 433
8.382.2.1 imsRegState . . . 433
8.382.2.2 modemMode . . . 433
8.382.2.3 psState . . . 433
8.382.2.4 systemMode . . . 433
8.382.2.5 temperature . . . 433
8.383 nas_CSGID Struct Reference . . . 433
8.383.1 Detailed Description . . . 433
8.383.2 Field Documentation . . . 434
8.383.2.1 id . . . 434
8.383.2.2 mcc . . . 434
8.383.2.3 mnc . . . 434
8.383.2.4 mncPcsDigits . . . 434
8.383.2.5 rat . . . 434
8.384 nas_currentPLMN Struct Reference . . . 434
8.384.1 Detailed Description . . . 434
8.384.2 Field Documentation . . . 435
8.384.2.1 MCC . . . 435
8.384.2.2 MNC . . . 435
8.384.2.3 netDescr . . . 435
8.384.2.4 netDescrLength . . . 435
8.385 nas_dataSrvCapabilities Struct Reference . . . 435

8.385.1 Detailed Description . . . 435
8.385.2 Field Documentation . . . 436
8.385.2.1 dataCapabilities . . . 436
8.385.2.2 dataCapabilitiesLen . . . 436
8.386 nas_DataStatusDetail Struct Reference . . . 436
8.386.1 Detailed Description . . . 436
8.386.2 Field Documentation . . . 437
8.386.2.1 IPAddress . . . 437
8.386.2.2 LastErrCode . . . 437
8.387 nas_detailSvcInfo Struct Reference . . . 438
8.387.1 Detailed Description . . . 438
8.387.2 Field Documentation . . . 439
8.387.2.1 hdrHybrid . . . 439
8.387.2.2 hdrSrvStatus . . . 439
8.387.2.3 isSysForbidden . . . 439
8.387.2.4 srvCapability . . . 439
8.387.2.5 srvStatus . . . 439
8.388 nas_DeviceConfigDetail Struct Reference . . . 439
8.388.1 Detailed Description . . . 439
8.388.2 Field Documentation . . . 440
8.388.2.1 Chipset . . . 440
8.388.2.2 HWVersion . . . 440
8.388.2.3 QLIC . . . 440
8.388.2.4 Technology . . . 440
8.389 nas_dirNum Struct Reference . . . 440
8.389.1 Detailed Description . . . 440
8.389.2 Field Documentation . . . 440
8.389.2.1 dirNum . . . 440
8.389.2.2 dirNumLen . . . 440
8.390 nas_DRCParams Struct Reference . . . 440
8.390.1 Detailed Description . . . 441
8.390.2 Field Documentation . . . 441
8.390.2.1 DRCCover . . . 441
8.390.2.2 DRCValue . . . 441
8.391 nas_ecioListElement Struct Reference . . . 441
8.391.1 Detailed Description . . . 441
8.391.2 Field Documentation . . . 441
8.391.2.1 ecio . . . 442
8.391.2.2 radioIf . . . 442
8.392 nas_ECIOThresh Struct Reference . . . 442

8.392.1 Detailed Description . . . 442
8.392.2 Field Documentation . . . 442
8.392.2.1 ECIOThresListLen . . . 442
8.392.2.2 pECIOThresList . . . 442
8.393 nas_errorRateListElement Struct Reference . . . 442
8.393.1 Detailed Description . . . 442
8.393.2 Field Documentation . . . 443
8.393.2.1 errorRate . . . 443
8.393.2.2 radioIf . . . 443
8.394 nas_GERANInfo Struct Reference . . . 443
8.394.1 Detailed Description . . . 444
8.394.2 Field Documentation . . . 445
8.394.2.1 arfcn . . . 445
8.394.2.2 bsic . . . 445
8.394.2.3 cellID . . . 445
8.394.2.4 insNmrCellInfo . . . 445
8.394.2.5 lac . . . 445
8.394.2.6 nmrInst . . . 445
8.394.2.7 plmn . . . 445
8.394.2.8 rxLev . . . 445
8.394.2.9 timingAdvance . . . 445
8.395 nas_geranInstInfo Struct Reference . . . 445
8.395.1 Detailed Description . . . 445
8.395.2 Field Documentation . . . 445
8.395.2.1 geranArfcn . . . 445
8.395.2.2 geranBsicBcc . . . 446
8.395.2.3 geranBsicNcc . . . 446
8.395.2.4 geranRssi . . . 446
8.396 nas_gsmCellInfo Struct Reference . . . 446
8.396.1 Detailed Description . . . 446
8.396.2 Field Documentation . . . 446
8.396.2.1 arfcn . . . 447
8.396.2.2 band1900 . . . 447
8.396.2.3 bsicId . . . 447
8.396.2.4 cellIdValid . . . 447
8.396.2.5 rssi . . . 447
8.396.2.6 srxlev . . . 447
8.397 nas_GSMRSSIThresh Struct Reference . . . 447
8.397.1 Detailed Description . . . 447
8.397.2 Field Documentation . . . 447

8.397.2.1 GSMRSSIThreshListLen . . . 447
8.397.2.2 pGSMRSSIThreshList . . . 447
8.398 nas_GSMSrvStatusInfo Struct Reference . . . 447
8.398.1 Detailed Description . . . 447
8.398.2 Field Documentation . . . 448
8.398.2.1 isPrefDataPath . . . 448
8.398.2.2 srvStatus . . . 448
8.398.2.3 trueSrvStatus . . . 448
8.399 nas_GSMSysInfo Struct Reference . . . 448
8.399.1 Detailed Description . . . 449
8.399.2 Field Documentation . . . 450
8.399.2.1 cellId . . . 450
8.399.2.2 cellIdValid . . . 450
8.399.2.3 dtmSupp . . . 450
8.399.2.4 dtmSuppValid . . . 451
8.399.2.5 egprsSupp . . . 451
8.399.2.6 egprsSuppValid . . . 451
8.399.2.7 lac . . . 451
8.399.2.8 lacValid . . . 451
8.399.2.9 MCC . . . 451
8.399.2.10 MNC . . . 451
8.399.2.11 networkIdValid . . . 451
8.399.2.12 regRejectInfoValid . . . 451
8.399.2.13 rejCause . . . 451
8.399.2.14 rejectSrvDomain . . . 451
8.399.2.15 sysInfoGSM . . . 451
8.400 nas_HDRECIOThresh Struct Reference . . . 451
8.400.1 Detailed Description . . . 451
8.400.2 Field Documentation . . . 451
8.400.2.1 HDRECIOThreshListLen . . . 451
8.400.2.2 pHDRECIOThreshList . . . 451
8.401 nas_HDRIOThresh Struct Reference . . . 451
8.401.1 Detailed Description . . . 452
8.401.2 Field Documentation . . . 452
8.401.2.1 HDRIOThreshListLen . . . 452
8.401.2.2 pHDRIOThreshList . . . 452
8.402 nas_HDRRSSIThresh Struct Reference . . . 452
8.402.1 Detailed Description . . . 452
8.402.2 Field Documentation . . . 452
8.402.2.1 HDRRSSIThreshListLen . . . 452

8.402.2.2 pHDRRSSIThreshList . . . 453
8.403 nas_HDRSINRThresh Struct Reference . . . 453
8.403.1 Detailed Description . . . 453
8.403.2 Field Documentation . . . 453
8.403.2.1 HDRSINRThresListLen . . . 453
8.403.2.2 pHDRSINRThresList . . . 453
8.404 nas_HDRSINRThreshold Struct Reference . . . 453
8.404.1 Detailed Description . . . 453
8.404.2 Field Documentation . . . 454
8.404.2.1 HDRSINRThreshListLen . . . 454
8.404.2.2 pHDRSINRThreshList . . . 454
8.405 nas_HDRSysInfo Struct Reference . . . 454
8.405.1 Detailed Description . . . 454
8.405.2 Field Documentation . . . 456
8.405.2.1 hdrActiveProt . . . 456
8.405.2.2 hdrActiveProtValid . . . 456
8.405.2.3 hdrPersonality . . . 456
8.405.2.4 hdrPersonalityValid . . . 456
8.405.2.5 is856SysId . . . 456
8.405.2.6 is856SysIdValid . . . 456
8.405.2.7 isSysPrlMatch . . . 456
8.405.2.8 isSysPrlMatchValid . . . 456
8.405.2.9 sysInfoHDR . . . 456
8.406 nas_homeSIDNID Struct Reference . . . 456
8.406.1 Detailed Description . . . 456
8.406.2 Field Documentation . . . 456
8.406.2.1 numInstances . . . 456
8.406.2.2 SidNid . . . 456
8.407 nas_infoInterFreq Struct Reference . . . 456
8.407.1 Detailed Description . . . 457
8.407.2 Field Documentation . . . 457
8.407.2.1 cell_resel_priority . . . 457
8.407.2.2 cellInterFreqParams . . . 457
8.407.2.3 cells_len . . . 457
8.407.2.4 earfcn . . . 457
8.407.2.5 threshXHigh . . . 458
8.407.2.6 threshXLow . . . 458
8.408 nas_IOThresh Struct Reference . . . 458
8.408.1 Detailed Description . . . 458
8.408.2 Field Documentation . . . 458

8.408.2.1 IOThresListLen . . . 458
8.408.2.2 pIOThresList . . . 458
8.409 nas_lteEARFCN Struct Reference . . . 458
8.409.1 Detailed Description . . . 458
8.409.2 Field Documentation . . . 459
8.409.2.1 earfcn0 . . . 459
8.409.2.2 earfcn1 . . . 459
8.409.2.3 status . . . 459
8.410 nas_lteGsmCellInfo Struct Reference . . . 459
8.410.1 Detailed Description . . . 459
8.410.2 Field Documentation . . . 460
8.410.2.1 cellReselPriority . . . 460
8.410.2.2 cells_len . . . 460
8.410.2.3 GsmCellInfo . . . 460
8.410.2.4 nccPermitted . . . 460
8.410.2.5 threshGsmHigh . . . 460
8.410.2.6 threshGsmLow . . . 460
8.411 nas_LTEInfo Struct Reference . . . 460
8.411.1 Detailed Description . . . 460
8.411.2 Field Documentation . . . 462
8.411.2.1 band . . . 462
8.411.2.2 bandwidth . . . 462
8.411.2.3 emmConnState . . . 462
8.411.2.4 emmState . . . 462
8.411.2.5 emmSubState . . . 462
8.411.2.6 RXChan . . . 462
8.411.2.7 TXChan . . . 462
8.412 nas_LTEInfoInterfreq Struct Reference . . . 463
8.412.1 Detailed Description . . . 463
8.412.2 Field Documentation . . . 463
8.412.2.1 freqsLen . . . 463
8.412.2.2 InfoInterfreq . . . 463
8.412.2.3 ueInIdle . . . 463
8.413 nas_LTEInfoIntrafreq Struct Reference . . . 463
8.413.1 Detailed Description . . . 464
8.413.2 Field Documentation . . . 465
8.413.2.1 CellParams . . . 465
8.413.2.2 cellReselPriority . . . 465
8.413.2.3 cellsLen . . . 465
8.413.2.4 earfcn . . . 465

8.413.2.5 globalCellId . . . 465
8.413.2.6 plmn . . . 465
8.413.2.7 servingCellId . . . 465
8.413.2.8 sIntraSearch . . . 465
8.413.2.9 sNonIntraSearch . . . 465
8.413.2.10 tac . . . 465
8.413.2.11 threshServingLow . . . 465
8.413.2.12 ueInIdle . . . 465
8.414 nas_LTEInfoNeighboringGSM Struct Reference . . . 465
8.414.1 Detailed Description . . . 466
8.414.2 Field Documentation . . . 466
8.414.2.1 freqsLen . . . 466
8.414.2.2 LteGsmCellInfo . . . 466
8.414.2.3 ueInIdle . . . 466
8.415 nas_LTEInfoNeighboringWCDMA Struct Reference . . . 466
8.415.1 Detailed Description . . . 466
8.415.2 Field Documentation . . . 467
8.415.2.1 freqsLen . . . 467
8.415.2.2 LTEWCDMACellInfo . . . 467
8.415.2.3 ueInIdle . . . 467
8.416 nas_ltePCI Struct Reference . . . 467
8.416.1 Detailed Description . . . 467
8.416.2 Field Documentation . . . 467
8.416.2.1 earfcn . . . 467
8.416.2.2 pci . . . 467
8.416.2.3 status . . . 467
8.417 nas_lteRsrpinformation Struct Reference . . . 467
8.417.1 Detailed Description . . . 467
8.417.2 Field Documentation . . . 468
8.417.2.1 rsrplevel . . . 468
8.418 nas_LTERSRPThresh Struct Reference . . . 468
8.418.1 Detailed Description . . . 468
8.418.2 Field Documentation . . . 468
8.418.2.1 LTERSRPThreshListLen . . . 468
8.418.2.2 pLTERSRPThreshList . . . 468
8.419 nas_LTERSRQThresh Struct Reference . . . 468
8.419.1 Detailed Description . . . 468
8.419.2 Field Documentation . . . 469
8.419.2.1 LTERSRQThreshListLen . . . 469
8.419.2.2 pLTERSRQThreshList . . . 469

8.420 nas_LTERSSIThresh Struct Reference . . . 469
8.420.1 Detailed Description . . . 469
8.420.2 Field Documentation . . . 469
8.420.2.1 LTERSSIThreshListLen . . . 469
8.420.2.2 pLTERSSIThreshList . . . 469
8.421 nas_LTESigRptCfg Struct Reference . . . 469
8.421.1 Detailed Description . . . 470
8.421.2 Field Documentation . . . 470
8.421.2.1 avgPeriod . . . 470
8.421.2.2 rptRate . . . 470
8.422 nas_LTESigRptConfig Struct Reference . . . 470
8.422.1 Detailed Description . . . 470
8.422.2 Field Documentation . . . 471
8.422.2.1 avgPeriod . . . 471
8.422.2.2 rptRate . . . 471
8.423 nas_lteSnrinformation Struct Reference . . . 471
8.423.1 Detailed Description . . . 471
8.423.2 Field Documentation . . . 471
8.423.2.1 snrlevel . . . 472
8.424 nas_LTESNRThresh Struct Reference . . . 472
8.424.1 Detailed Description . . . 472
8.424.2 Field Documentation . . . 472
8.424.2.1 LTESNRThresListLen . . . 472
8.424.2.2 pLTESNRThresList . . . 472
8.425 nas_LTESNRThreshold Struct Reference . . . 472
8.425.1 Detailed Description . . . 472
8.425.2 Field Documentation . . . 473
8.425.2.1 LTESNRThreshListLen . . . 473
8.425.2.2 pLTESNRThreshList . . . 473
8.426 nas_LTESysInfo Struct Reference . . . 473
8.426.1 Detailed Description . . . 473
8.426.2 Field Documentation . . . 475
8.426.2.1 cellId . . . 475
8.426.2.2 cellIdValid . . . 475
8.426.2.3 lac . . . 475
8.426.2.4 lacValid . . . 475
8.426.2.5 MCC . . . 475
8.426.2.6 MNC . . . 475
8.426.2.7 networkIdValid . . . 475
8.426.2.8 regRejectInfoValid . . . 475

8.426.2.9 rejCause . . . 475
8.426.2.10 rejectSrvDomain . . . 475
8.426.2.11 sysInfoLTE . . . 475
8.426.2.12 tac . . . 475
8.426.2.13 tacValid . . . 475
8.427 nas_lteWcdmaCellInfo Struct Reference . . . 475
8.427.1 Detailed Description . . . 476
8.427.2 Field Documentation . . . 476
8.427.2.1 cellReselPriority . . . 476
8.427.2.2 cellsLen . . . 476
8.427.2.3 threshXhigh . . . 476
8.427.2.4 threshXlow . . . 476
8.427.2.5 uarfcn . . . 476
8.427.2.6 WCDMACellInfo . . . 476
8.428 nas_minBasedIMSI Struct Reference . . . 476
8.428.1 Detailed Description . . . 477
8.428.2 Field Documentation . . . 477
8.428.2.1 imsiM1112 . . . 477
8.428.2.2 imsiMS1 . . . 477
8.428.2.3 imsiMS2 . . . 477
8.428.2.4 mccM . . . 477
8.429 nas_MNRInfo Struct Reference . . . 477
8.429.1 Detailed Description . . . 477
8.429.2 Field Documentation . . . 478
8.429.2.1 mcc . . . 478
8.429.2.2 mnc . . . 478
8.429.2.3 rat . . . 478
8.430 nas_namName Struct Reference . . . 478
8.430.1 Detailed Description . . . 478
8.430.2 Field Documentation . . . 478
8.430.2.1 namName . . . 478
8.430.2.2 namNameLen . . . 478
8.431 nas_netSelectionPref Struct Reference . . . 478
8.431.1 Detailed Description . . . 479
8.431.2 Field Documentation . . . 479
8.431.2.1 mcc . . . 479
8.431.2.2 mnc . . . 479
8.431.2.3 netReg . . . 479
8.432 nas_NetworkStat1x Struct Reference . . . 479
8.432.1 Detailed Description . . . 479

8.432.2 Field Documentation . . . 481
8.432.2.1 ActSetCnt . . . 481
8.432.2.2 NeighborSetCnt . . . 481
8.432.2.3 pActPilotPNElements . . . 481
8.432.2.4 pNeighborSetPilotPN . . . 481
8.432.2.5 RX_EC_IO . . . 481
8.432.2.6 RX_PWR . . . 481
8.432.2.7 SO . . . 481
8.432.2.8 State . . . 481
8.432.2.9 TX_PWR . . . 481
8.433 nas_NetworkStatEVDO Struct Reference . . . 481
8.433.1 Detailed Description . . . 481
8.433.2 Field Documentation . . . 482
8.433.2.1 MACIndex . . . 483
8.433.2.2 PER . . . 483
8.433.2.3 PilotEnergy . . . 483
8.433.2.4 pSectorID . . . 483
8.433.2.5 RX_PWR . . . 483
8.433.2.6 SectorIDLen . . . 483
8.433.2.7 SNR . . . 483
8.433.2.8 State . . . 483
8.434 nas_nmrCellInfo Struct Reference . . . 483
8.434.1 Detailed Description . . . 483
8.434.2 Field Documentation . . . 484
8.434.2.1 nmrArfcn . . . 484
8.434.2.2 nmrBsic . . . 484
8.434.2.3 nmrCellID . . . 484
8.434.2.4 nmrLac . . . 484
8.434.2.5 nmrPlmn . . . 484
8.434.2.6 nmrRxLev . . . 484
8.435 nas_operatorNameString Struct Reference . . . 484
8.435.1 Detailed Description . . . 484
8.435.2 Field Documentation . . . 484
8.435.2.1 PLMNName . . . 484
8.436 nas_OperatorPLMNData Struct Reference . . . 484
8.436.1 Detailed Description . . . 485
8.436.2 Field Documentation . . . 485
8.436.2.1 lac1 . . . 485
8.436.2.2 lac2 . . . 485
8.436.2.3 mcc . . . 485

8.436.2.4 mnc . . . 485
8.436.2.5 PLMNRecID . . . 485
8.437 nas_operatorPLMNList Struct Reference . . . 485
8.437.1 Detailed Description . . . 486
8.437.2 Field Documentation . . . 486
8.437.2.1 numInstance . . . 486
8.437.2.2 PLMNData . . . 486
8.438 nas_PhyCaAggPcellInfo Struct Reference . . . 486
8.438.1 Detailed Description . . . 486
8.438.2 Field Documentation . . . 487
8.438.2.1 dl_bw_value . . . 487
8.438.2.2 freq . . . 487
8.438.2.3 iLTEbandValue . . . 487
8.438.2.4 pci . . . 487
8.438.2.5 TlvPresent . . . 487
8.439 nas_PhyCaAggScellDlBw Struct Reference . . . 487
8.439.1 Detailed Description . . . 487
8.439.2 Field Documentation . . . 487
8.439.2.1 dl_bw_value . . . 487
8.439.2.2 TlvPresent . . . 487
8.440 nas_PhyCaAggScellIndex Struct Reference . . . 487
8.440.1 Detailed Description . . . 487
8.440.2 Field Documentation . . . 488
8.440.2.1 scell_idx . . . 488
8.440.2.2 TlvPresent . . . 488
8.441 nas_PhyCaAggScellIndType Struct Reference . . . 488
8.441.1 Detailed Description . . . 488
8.441.2 Field Documentation . . . 488
8.441.2.1 freq . . . 488
8.441.2.2 pci . . . 488
8.441.2.3 scell_state . . . 488
8.441.2.4 TlvPresent . . . 488
8.442 nas_PhyCaAggScellInfo Struct Reference . . . 489
8.442.1 Detailed Description . . . 489
8.442.2 Field Documentation . . . 491
8.442.2.1 dl_bw_value . . . 491
8.442.2.2 freq . . . 491
8.442.2.3 iLTEbandValue . . . 491
8.442.2.4 pci . . . 491
8.442.2.5 scell_state . . . 491

8.442.2.6 TlvPresent . . . 491
8.443 nas_PilotSetData Struct Reference . . . 491
8.443.1 Detailed Description . . . 491
8.443.2 Field Documentation . . . 491
8.443.2.1 NumPilots . . . 491
8.443.2.2 pPilotSetInfo . . . 491
8.444 nas_PilotSetParams Struct Reference . . . 492
8.444.1 Detailed Description . . . 492
8.444.2 Field Documentation . . . 492
8.444.2.1 PilotPN . . . 492
8.444.2.2 PilotStrength . . . 492
8.444.2.3 PilotType . . . 492
8.445 nas_PLMNNetworkName Struct Reference . . . 492
8.445.1 Detailed Description . . . 492
8.445.2 Field Documentation . . . 493
8.445.2.1 numInstance . . . 493
8.445.2.2 PLMNNetName . . . 493
8.446 nas_PLMNNetworkNameData Struct Reference . . . 493
8.446.1 Detailed Description . . . 493
8.446.2 Field Documentation . . . 494
8.446.2.1 codingScheme . . . 494
8.446.2.2 countryInitials . . . 494
8.446.2.3 longName . . . 494
8.446.2.4 longNameLen . . . 494
8.446.2.5 longNameSpareBits . . . 494
8.446.2.6 shortName . . . 494
8.446.2.7 shortNameLen . . . 494
8.446.2.8 shortNameSpareBits . . . 494
8.447 nas_protocolSubtypeElement Struct Reference . . . 494
8.447.1 Detailed Description . . . 495
8.447.2 Field Documentation . . . 495
8.447.2.1 AccessMac . . . 496
8.447.2.2 AuthProt . . . 496
8.447.2.3 ControlMac . . . 496
8.447.2.4 EncryptProt . . . 496
8.447.2.5 ForwardMac . . . 496
8.447.2.6 IdleState . . . 496
8.447.2.7 KeyExchange . . . 496
8.447.2.8 MultDisc . . . 496
8.447.2.9 PhysicalLayer . . . 496

8.447.2.10 ReverseMac . . . 496
8.447.2.11 SecProt . . . 496
8.447.2.12 VirtStream . . . 496
8.448 nas_qaQmi3Gpp2TimeZone Struct Reference . . . 496
8.448.1 Detailed Description . . . 496
8.448.2 Field Documentation . . . 496
8.448.2.1 daylightSavings . . . 497
8.448.2.2 leapSeconds . . . 497
8.448.2.3 localTimeOffset . . . 497
8.449 nas_QmiNas3GppNetworkInfo Struct Reference . . . 497
8.449.1 Detailed Description . . . 497
8.449.2 Field Documentation . . . 498
8.449.2.1 Desription . . . 498
8.449.2.2 Forbidden . . . 498
8.449.2.3 InUse . . . 498
8.449.2.4 MCC . . . 498
8.449.2.5 MNC . . . 498
8.449.2.6 Preferred . . . 498
8.449.2.7 Roaming . . . 498
8.450 nas_QmiNas3GppNetworkRAT Struct Reference . . . 498
8.450.1 Detailed Description . . . 498
8.450.2 Field Documentation . . . 498
8.450.2.1 MCC . . . 499
8.450.2.2 MNC . . . 499
8.450.2.3 RAT . . . 499
8.451 nas_QmisNasPcsDigit Struct Reference . . . 499
8.451.1 Detailed Description . . . 499
8.451.2 Field Documentation . . . 499
8.451.2.1 includes_pcs_digit . . . 499
8.451.2.2 MCC . . . 499
8.451.2.3 MNC . . . 499
8.452 nas_RejectReasonTlv Struct Reference . . . 499
8.452.1 Detailed Description . . . 499
8.452.2 Field Documentation . . . 500
8.452.2.1 rejectCause . . . 500
8.452.2.2 serviceDomain . . . 500
8.452.2.3 TlvPresent . . . 500
8.453 nas_RFInfoTlv Struct Reference . . . 500
8.453.1 Detailed Description . . . 500
8.453.2 Field Documentation . . . 501

8.453.2.1 activeBandClass . . . 501
8.453.2.2 activeChannel . . . 501
8.453.2.3 radioInterface . . . 501
8.453.2.4 radioInterfaceSize . . . 501
8.453.2.5 TlvPresent . . . 501
8.454 nas_roamIndList Struct Reference . . . 501
8.454.1 Detailed Description . . . 501
8.454.2 Field Documentation . . . 502
8.454.2.1 numInstances . . . 502
8.454.2.2 radioInterface . . . 502
8.454.2.3 roamIndicator . . . 502
8.455 nas_RSRPThresh Struct Reference . . . 502
8.455.1 Detailed Description . . . 502
8.455.2 Field Documentation . . . 502
8.455.2.1 pRSRPThresList . . . 502
8.455.2.2 RSRPThresListLen . . . 502
8.456 nas_rsrqInformation Struct Reference . . . 502
8.456.1 Detailed Description . . . 502
8.456.2 Field Documentation . . . 503
8.456.2.1 radioIf . . . 503
8.456.2.2 rsrq . . . 503
8.457 nas_RSRQThresh Struct Reference . . . 503
8.457.1 Detailed Description . . . 503
8.457.2 Field Documentation . . . 503
8.457.2.1 pRSRQThresList . . . 503
8.457.2.2 RSRQThresListLen . . . 503
8.458 nas_RSSIThresh Struct Reference . . . 503
8.458.1 Detailed Description . . . 504
8.458.2 Field Documentation . . . 504
8.458.2.1 pRSSIThresList . . . 504
8.458.2.2 RSSIThresListLen . . . 504
8.459 nas_rxInfo Struct Reference . . . 504
8.459.1 Detailed Description . . . 504
8.459.2 Field Documentation . . . 505
8.459.2.1 ecio . . . 505
8.459.2.2 isRadioTuned . . . 505
8.459.2.3 phase . . . 505
8.459.2.4 rscp . . . 505
8.459.2.5 rsrp . . . 505
8.459.2.6 rxPower . . . 505

8.460 nas_RxSigInfo Struct Reference . . . 505
8.460.1 Detailed Description . . . 505
8.460.2 Field Documentation . . . 506
8.460.2.1 isRadioTuned . . . 506
8.460.2.2 rsrp . . . 506
8.460.2.3 rxChainIndex . . . 506
8.460.2.4 rxPower . . . 506
8.461 nas_rxSignalStrengthListElement Struct Reference . . . 506
8.461.1 Detailed Description . . . 506
8.461.2 Field Documentation . . . 507
8.461.2.1 radioIf . . . 507
8.461.2.2 rxSignalStrength . . . 507
8.462 nas_SccRxInfo Struct Reference . . . 507
8.462.1 Detailed Description . . . 507
8.462.2 Field Documentation . . . 508
8.462.2.1 numInstances . . . 508
8.462.2.2 rsrq . . . 508
8.462.2.3 sigInfo . . . 508
8.462.2.4 snr . . . 508
8.462.2.5 TlvPresent . . . 508
8.463 nas_serviceProviderName Struct Reference . . . 508
8.463.1 Detailed Description . . . 508
8.463.2 Field Documentation . . . 509
8.463.2.1 displayCondition . . . 509
8.463.2.2 spn . . . 509
8.463.2.3 spnLength . . . 509
8.464 nas_servSystem Struct Reference . . . 509
8.464.1 Detailed Description . . . 509
8.464.2 Field Documentation . . . 510
8.464.2.1 csAttachState . . . 510
8.464.2.2 numRadioInterfaces . . . 510
8.464.2.3 psAttachState . . . 510
8.464.2.4 radioInterface . . . 510
8.464.2.5 regState . . . 510
8.464.2.6 selNetwork . . . 510
8.465 nas_sidNid Struct Reference . . . 510
8.465.1 Detailed Description . . . 510
8.465.2 Field Documentation . . . 511
8.465.2.1 nid . . . 511
8.465.2.2 sid . . . 511

8.466 nas_SignalStrengthTlv Struct Reference . . . 511
8.466.1 Detailed Description . . . 511
8.466.2 Field Documentation . . . 511
8.466.2.1 radioInterface . . . 511
8.466.2.2 signalStrength . . . 511
8.466.2.3 TlvPresent . . . 511
8.467 nas_SLQSSignalStrengthsIndReq Struct Reference . . . 511
8.467.1 Detailed Description . . . 512
8.467.2 Field Documentation . . . 513
8.467.2.1 ecioDelta . . . 513
8.467.2.2 ecioThresholdList . . . 513
8.467.2.3 ecioThresholdListLen . . . 513
8.467.2.4 ioDelta . . . 513
8.467.2.5 lteRsrpDelta . . . 513
8.467.2.6 lteSnrDelta . . . 513
8.467.2.7 rsrqDelta . . . 513
8.467.2.8 rxSignalStrengthDelta . . . 513
8.467.2.9 sinrDelta . . . 513
8.467.2.10 sinrThresholdList . . . 513
8.467.2.11 sinrThresholdListLen . . . 513
8.468 nas_SLQSSignalStrengthsInformation Struct Reference . . . 513
8.468.1 Detailed Description . . . 513
8.468.2 Field Documentation . . . 514
8.468.2.1 ecioInfo . . . 514
8.468.2.2 errorRateInfo . . . 514
8.468.2.3 io . . . 514
8.468.2.4 lteRsrpinfo . . . 514
8.468.2.5 lteSnrinfo . . . 514
8.468.2.6 rsrqInfo . . . 514
8.468.2.7 rxSignalStrengthInfo . . . 514
8.468.2.8 sinr . . . 514
8.469 nas_SLQSSignalStrengthsTlv Struct Reference . . . 514
8.469.1 Detailed Description . . . 514
8.469.2 Field Documentation . . . 514
8.469.2.1 sSLQSSignalStrengthsInfo . . . 514
8.469.2.2 TlvPresent . . . 514
8.470 nas_SrvStatusInfo Struct Reference . . . 515
8.470.1 Detailed Description . . . 515
8.470.2 Field Documentation . . . 515
8.470.2.1 isPrefDataPath . . . 515

8.470.2.2 srvStatus . . . 515
8.471 nas_sysInfoCommon Struct Reference . . . 515
8.471.1 Detailed Description . . . 515
8.471.2 Field Documentation . . . 517
8.471.2.1 isSysForbidden . . . 517
8.471.2.2 isSysForbiddenValid . . . 517
8.471.2.3 roamStatus . . . 517
8.471.2.4 roamStatusValid . . . 517
8.471.2.5 srvCapability . . . 517
8.471.2.6 srvCapabilityValid . . . 517
8.471.2.7 srvDomain . . . 517
8.471.2.8 srvDomainValid . . . 517
8.472 nas_TDSCDMAECIOThresh Struct Reference . . . 518
8.472.1 Detailed Description . . . 518
8.472.2 Field Documentation . . . 518
8.472.2.1 pTDSCDMAECIOThreshList . . . 518
8.472.2.2 TDSCDMAECIOThreshListLen . . . 518
8.473 nas_TDSCDMARSCPThresh Struct Reference . . . 518
8.473.1 Detailed Description . . . 518
8.473.2 Field Documentation . . . 518
8.473.2.1 pTDSCDMARSCPThreshList . . . 518
8.473.2.2 TDSCDMARSCPThreshListLen . . . 519
8.474 nas_TDSCDMARSSIThresh Struct Reference . . . 519
8.474.1 Detailed Description . . . 519
8.474.2 Field Documentation . . . 519
8.474.2.1 pTDSCDMARSSIThreshList . . . 519
8.474.2.2 TDSCDMARSSIThreshListLen . . . 519
8.475 nas_TDSCDMASINRCONFThresh Struct Reference . . . 519
8.475.1 Detailed Description . . . 519
8.475.2 Field Documentation . . . 519
8.475.2.1 pTDSCDMASINRCONFThreshList . . . 520
8.475.2.2 TDSCDMASINRCONFThreshListLen . . . 520
8.476 nas_TDSCDMASINRThresh Struct Reference . . . 520
8.476.1 Detailed Description . . . 520
8.476.2 Field Documentation . . . 520
8.476.2.1 pTDSCDMASINRThreshList . . . 520
8.476.2.2 TDSCDMASINRThreshListLen . . . 520
8.477 nas_timeInfo Struct Reference . . . 520
8.477.1 Detailed Description . . . 520
8.477.2 Field Documentation . . . 522

8.477.2.1 day . . . 522
8.477.2.2 dayLtSavingAdj . . . 522
8.477.2.3 dayOfWeek . . . 522
8.477.2.4 hour . . . 522
8.477.2.5 minute . . . 522
8.477.2.6 month . . . 522
8.477.2.7 radioInterface . . . 522
8.477.2.8 second . . . 522
8.477.2.9 timeZone . . . 522
8.477.2.10 tlvPresent . . . 522
8.477.2.11 year . . . 522
8.478 nas_trueIMSI Struct Reference . . . 522
8.478.1 Detailed Description . . . 522
8.478.2 Field Documentation . . . 523
8.478.2.1 imsiT1112 . . . 523
8.478.2.2 imsiTaddrNum . . . 523
8.478.2.3 imsiTS1 . . . 523
8.478.2.4 imsiTS2 . . . 523
8.478.2.5 mccT . . . 523
8.479 nas_txInfo Struct Reference . . . 523
8.479.1 Detailed Description . . . 523
8.479.2 Field Documentation . . . 523
8.479.2.1 isInTraffic . . . 523
8.479.2.2 txPower . . . 523
8.480 nas_UMTSInfo Struct Reference . . . 523
8.480.1 Detailed Description . . . 524
8.480.2 Field Documentation . . . 525
8.480.2.1 cellID . . . 525
8.480.2.2 ecio . . . 525
8.480.2.3 geranInst . . . 525
8.480.2.4 GeranInstInfo . . . 525
8.480.2.5 lac . . . 525
8.480.2.6 plmn . . . 525
8.480.2.7 psc . . . 525
8.480.2.8 rscp . . . 525
8.480.2.9 uarfcn . . . 525
8.480.2.10 umtsInst . . . 525
8.480.2.11 UMTSInstInfo . . . 525
8.481 nas_UMTSinstInfo Struct Reference . . . 525
8.481.1 Detailed Description . . . 525

8.481.2 Field Documentation . . . 526
8.481.2.1 umtsEcio . . . 526
8.481.2.2 umtsPsc . . . 526
8.481.2.3 umtsRscp . . . 526
8.481.2.4 umtsUarfcn . . . 526
8.482 nas_umtsLTENbrCell Struct Reference . . . 526
8.482.1 Detailed Description . . . 526
8.482.2 Field Documentation . . . 527
8.482.2.1 cellIsTDD . . . 527
8.482.2.2 earfcn . . . 527
8.482.2.3 pci . . . 527
8.482.2.4 rsrp . . . 527
8.482.2.5 rsrq . . . 527
8.482.2.6 srxlev . . . 527
8.483 nas_UniversalTime Struct Reference . . . 527
8.483.1 Detailed Description . . . 527
8.483.2 Field Documentation . . . 528
8.483.2.1 day . . . 528
8.483.2.2 dayOfWeek . . . 528
8.483.2.3 hour . . . 528
8.483.2.4 minute . . . 528
8.483.2.5 month . . . 528
8.483.2.6 second . . . 528
8.483.2.7 year . . . 528
8.484 nas_wcdmaCellInfo Struct Reference . . . 528
8.484.1 Detailed Description . . . 528
8.484.2 Field Documentation . . . 529
8.484.2.1 cpich_ecno . . . 529
8.484.2.2 cpich_rscp . . . 529
8.484.2.3 psc . . . 529
8.484.2.4 srxlev . . . 529
8.485 nas_WCDMAECIOThresh Struct Reference . . . 529
8.485.1 Detailed Description . . . 529
8.485.2 Field Documentation . . . 529
8.485.2.1 pWCDMAECIOThreshList . . . 529
8.485.2.2 WCDMAECIOThreshListLen . . . 529
8.486 nas_WCDMAInfoLTENeighborCell Struct Reference . . . 529
8.486.1 Detailed Description . . . 530
8.486.2 Field Documentation . . . 530
8.486.2.1 UMTSLTENbrCell . . . 530

8.486.2.2 umtsLTENbrCellLen . . . 530
8.486.2.3 wcdmaRRCState . . . 530
8.487 nas_WCDMARSSIThresh Struct Reference . . . 530
8.487.1 Detailed Description . . . 530
8.487.2 Field Documentation . . . 531
8.487.2.1 pWCDMARSSIThreshList . . . 531
8.487.2.2 WCDMARSSIThreshListLen . . . 531
8.488 nas_WCDMASysInfo Struct Reference . . . 531
8.488.1 Detailed Description . . . 531
8.488.2 Field Documentation . . . 533
8.488.2.1 cellId . . . 534
8.488.2.2 cellIdValid . . . 534
8.488.2.3 hsCallStatus . . . 534
8.488.2.4 hsCallStatusValid . . . 534
8.488.2.5 hsInd . . . 534
8.488.2.6 hsIndValid . . . 534
8.488.2.7 lac . . . 534
8.488.2.8 lacValid . . . 534
8.488.2.9 MCC . . . 534
8.488.2.10 MNC . . . 534
8.488.2.11 networkIdValid . . . 534
8.488.2.12 psc . . . 534
8.488.2.13 pscValid . . . 534
8.488.2.14 regRejectInfoValid . . . 534
8.488.2.15 rejCause . . . 534
8.488.2.16 rejectSrvDomain . . . 534
8.488.2.17 sysInfoWCDMA . . . 534
8.489 nas_wcdmaUARFCN Struct Reference . . . 534
8.489.1 Detailed Description . . . 534
8.489.2 Field Documentation . . . 535
8.489.2.1 status . . . 535
8.489.2.2 uarfcn . . . 535
8.490 NASBandPreferenceTlv Struct Reference . . . 535
8.490.1 Detailed Description . . . 535
8.490.2 Field Documentation . . . 535
8.490.2.1 band_pref . . . 535
8.490.2.2 TlvPresent . . . 535
8.491 nasCellLocationInfoResp Struct Reference . . . 535
8.491.1 Detailed Description . . . 535
8.491.2 Field Documentation . . . 536

8.491.2.1 pCDMAInfo . . . 536
8.491.2.2 pGERANInfo . . . 536
8.491.2.3 pLTEInfoInterfreq . . . 536
8.491.2.4 pLTEInfoIntrafreq . . . 536
8.491.2.5 pLTEInfoNeighboringGSM . . . 536
8.491.2.6 pLTEInfoNeighboringWCDMA . . . 536
8.491.2.7 pUMTSCellID . . . 536
8.491.2.8 pUMTSInfo . . . 536
8.491.2.9 pWCDMAInfoLTENeighborCell . . . 536
8.492 NASEmergencyModeTlv Struct Reference . . . 536
8.492.1 Detailed Description . . . 537
8.492.2 Field Documentation . . . 537
8.492.2.1 EmerMode . . . 537
8.492.2.2 TlvPresent . . . 537
8.493 nasGet3GPP2SubscriptionInfoReq Struct Reference . . . 537
8.493.1 Detailed Description . . . 537
8.493.2 Field Documentation . . . 537
8.493.2.1 namID . . . 537
8.494 nasGet3GPP2SubscriptionInfoResp Struct Reference . . . 537
8.494.1 Detailed Description . . . 537
8.494.2 Field Documentation . . . 538
8.494.2.1 pCDMAChannel . . . 538
8.494.2.2 pDirNum . . . 538
8.494.2.3 pHomeSIDNID . . . 538
8.494.2.4 pMinBasedIMSI . . . 538
8.494.2.5 pNAMNameInfo . . . 538
8.494.2.6 pTrueIMSI . . . 538
8.495 nasGetHDRColorCodeResp Struct Reference . . . 538
8.495.1 Detailed Description . . . 538
8.495.2 Field Documentation . . . 539
8.495.2.1 pColorCode . . . 539
8.496 nasGetLTECphyCa Struct Reference . . . 539
8.496.1 Field Documentation . . . 539
8.496.1.1 sPhyCaAggPcellInfo . . . 539
8.496.1.2 sPhyCaAggScellDlBw . . . 539
8.496.1.3 sPhyCaAggScellIndex . . . 539
8.496.1.4 sPhyCaAggScellIndType . . . 539
8.496.1.5 sPhyCaAggScellInfo . . . 539
8.497 NasGetLTECphyCaInfo Struct Reference . . . 539
8.497.1 Detailed Description . . . 539

8.497.2 Field Documentation . . . 540
8.497.2.1 PhyCaAggPcellInfo . . . 540
8.497.2.2 PhyCaAggScellDlBw . . . 540
8.497.2.3 PhyCaAggScellIndex . . . 540
8.497.2.4 PhyCaAggScellIndType . . . 540
8.497.2.5 PhyCaAggScellInfo . . . 540
8.498 nasGetLTECphyCaResp Struct Reference . . . 540
8.498.1 Field Documentation . . . 540
8.498.1.1 pPhyCaAggPcellInfo . . . 540
8.498.1.2 pPhyCaAggScellDlBw . . . 540
8.498.1.3 pPhyCaAggScellIndex . . . 540
8.498.1.4 pPhyCaAggScellIndType . . . 540
8.498.1.5 pPhyCaAggScellInfo . . . 540
8.499 nasGetSigInfoResp Struct Reference . . . 540
8.499.1 Detailed Description . . . 541
8.499.2 Field Documentation . . . 541
8.499.2.1 pCDMASSInfo . . . 541
8.499.2.2 pGSMSSInfo . . . 541
8.499.2.3 pHDRSSInfo . . . 541
8.499.2.4 pLTESSInfo . . . 541
8.499.2.5 pTDSCDMASigInfoExt . . . 541
8.499.2.6 pTDSCDMASigInfoRscp . . . 541
8.499.2.7 pWCDMASSInfo . . . 541
8.500 nasGetSysInfoResp Struct Reference . . . 541
8.500.1 Detailed Description . . . 542
8.500.2 Field Documentation . . . 543
8.500.2.1 pAddCDMASysInfo . . . 543
8.500.2.2 pAddGSMSysInfo . . . 543
8.500.2.3 pAddHDRSysInfo . . . 544
8.500.2.4 pAddLTESysInfo . . . 544
8.500.2.5 pAddWCDMASysInfo . . . 544
8.500.2.6 pCDMASrvStatusInfo . . . 544
8.500.2.7 pCDMASysInfo . . . 544
8.500.2.8 pGSMCallBarringSysInfo . . . 544
8.500.2.9 pGSMCipherDomainSysInfo . . . 544
8.500.2.10 pGSMSrvStatusInfo . . . 544
8.500.2.11 pGSMSysInfo . . . 544
8.500.2.12 pHDRSrvStatusInfo . . . 544
8.500.2.13 pHDRSysInfo . . . 544
8.500.2.14 pLTESrvStatusInfo . . . 544

8.500.2.15 pLTESysInfo . . . 544
8.500.2.16 pLTEVoiceSupportSysInfo . . . 544
8.500.2.17 pWCDMACallBarringSysInfo . . . 544
8.500.2.18 pWCDMACipherDomainSysInfo . . . 544
8.500.2.19 pWCDMASrvStatusInfo . . . 544
8.500.2.20 pWCDMASysInfo . . . 544
8.501 nasGetTxRxInfoReq Struct Reference . . . 544
8.501.1 Detailed Description . . . 544
8.501.2 Field Documentation . . . 545
8.501.2.1 radio_if . . . 545
8.502 nasGetTxRxInfoResp Struct Reference . . . 545
8.502.1 Detailed Description . . . 545
8.502.2 Field Documentation . . . 545
8.502.2.1 pRXChain0Info . . . 545
8.502.2.2 pRXChain1Info . . . 545
8.502.2.3 pTXInfo . . . 545
8.503 NASGWAcqOrderPrefTlv Struct Reference . . . 545
8.503.1 Detailed Description . . . 546
8.503.2 Field Documentation . . . 546
8.503.2.1 GWAcqOrderPref . . . 546
8.503.2.2 TlvPresent . . . 546
8.504 nasIndicationRegisterReq Struct Reference . . . 546
8.504.1 Detailed Description . . . 546
8.504.2 Field Documentation . . . 549
8.504.2.1 pDDTMInd . . . 549
8.504.2.2 pDualStandByPrefInd . . . 549
8.504.2.3 pErrorRateInd . . . 549
8.504.2.4 pHDRNewUATIAssInd . . . 549
8.504.2.5 pHDRSessionCloseInd . . . 549
8.504.2.6 pLTECphyCa . . . 549
8.504.2.7 pManagedRoamingInd . . . 549
8.504.2.8 pNetworkRejectInd . . . 549
8.504.2.9 pNetworkTimeInd . . . 549
8.504.2.10 pServingSystemInd . . . 549
8.504.2.11 pSignalStrengthInd . . . 549
8.504.2.12 pSubscriptionInfoInd . . . 549
8.504.2.13 pSuppressSysInfoInd . . . 549
8.504.2.14 pSysInfoInd . . . 549
8.504.2.15 pSystemSelectionInd . . . 549
8.505 nasInitNetworkReg Struct Reference . . . 549

8.505.1 Detailed Description . . . 549
8.505.2 Field Documentation . . . 550
8.505.2.1 pChangeDuration . . . 550
8.505.2.2 pMncPcsDigitStatus . . . 550
8.505.2.3 pMNRInfo . . . 550
8.505.2.4 regAction . . . 550
8.506 NASLTEBandPreferenceTlv Struct Reference . . . 550
8.506.1 Detailed Description . . . 550
8.506.2 Field Documentation . . . 550
8.506.2.1 LTEBandPref . . . 550
8.506.2.2 TlvPresent . . . 550
8.507 NASLteNasReleaseInfoTlv Struct Reference . . . 550
8.507.1 Detailed Description . . . 551
8.507.2 Field Documentation . . . 551
8.507.2.1 nas_major . . . 551
8.507.2.2 nas_minor . . . 551
8.507.2.3 nas_release . . . 551
8.507.2.4 TlvPresent . . . 551
8.508 NASModePreferenceTlv Struct Reference . . . 551
8.508.1 Detailed Description . . . 551
8.508.2 Field Documentation . . . 551
8.508.2.1 ModePref . . . 552
8.508.2.2 TlvPresent . . . 552
8.509 NASNetSelPreferenceTlv Struct Reference . . . 552
8.509.1 Detailed Description . . . 552
8.509.2 Field Documentation . . . 552
8.509.2.1 NetSelPref . . . 552
8.509.2.2 TlvPresent . . . 552
8.510 nasNetworkReject Struct Reference . . . 552
8.510.1 Detailed Description . . . 552
8.510.2 Field Documentation . . . 555
8.510.2.1 pCsgId . . . 555
8.510.2.2 pPlmnId . . . 555
8.510.2.3 radioIf . . . 555
8.510.2.4 rejectCause . . . 555
8.510.2.5 serviceDomain . . . 555
8.511 nasNetworkTime Struct Reference . . . 555
8.511.1 Detailed Description . . . 555
8.511.2 Field Documentation . . . 556
8.511.2.1 pDayltSavAdj . . . 556

8.511.2.2 pRadioInterface . . . 556
8.511.2.3 pTimeZone . . . 556
8.511.2.4 universalTime . . . 556
8.512 nasOperatorNameResp Struct Reference . . . 556
8.512.1 Detailed Description . . . 556
8.512.2 Field Documentation . . . 557
8.512.2.1 pNITZInformation . . . 557
8.512.2.2 pOperatorNameString . . . 557
8.512.2.3 pOperatorPLMNList . . . 557
8.512.2.4 pPLMNNetworkName . . . 557
8.512.2.5 pSrvcProviderName . . . 557
8.513 NASOTAMessageTlv Struct Reference . . . 557
8.513.1 Detailed Description . . . 557
8.513.2 Field Documentation . . . 558
8.513.2.1 data_buf . . . 558
8.513.2.2 data_len . . . 558
8.513.2.3 message_type . . . 558
8.513.2.4 TlvPresent . . . 558
8.514 NASPhyCaAggPcellInfo Struct Reference . . . 558
8.514.1 Detailed Description . . . 558
8.514.2 Field Documentation . . . 558
8.514.2.1 dl_bw_value . . . 558
8.514.2.2 freq . . . 558
8.514.2.3 iLTEbandValue . . . 558
8.514.2.4 pci . . . 559
8.514.2.5 TlvPresent . . . 559
8.515 NASPhyCaAggScellDlBw Struct Reference . . . 559
8.515.1 Detailed Description . . . 559
8.515.2 Field Documentation . . . 559
8.515.2.1 dl_bw_value . . . 559
8.515.2.2 TlvPresent . . . 559
8.516 NASPhyCaAggScellIndex Struct Reference . . . 559
8.516.1 Detailed Description . . . 559
8.516.2 Field Documentation . . . 559
8.516.2.1 scell_idx . . . 560
8.516.2.2 TlvPresent . . . 560
8.517 NASPhyCaAggScellIndType Struct Reference . . . 560
8.517.1 Detailed Description . . . 560
8.517.2 Field Documentation . . . 560
8.517.2.1 freq . . . 560

8.517.2.2 pci . . . 560
8.517.2.3 scell_state . . . 560
8.517.2.4 TlvPresent . . . 560
8.518 NASPhyCaAggScellInfo Struct Reference . . . 560
8.518.1 Detailed Description . . . 561
8.518.2 Field Documentation . . . 561
8.518.2.1 dl_bw_value . . . 561
8.518.2.2 freq . . . 561
8.518.2.3 iLTEbandValue . . . 561
8.518.2.4 pci . . . 561
8.518.2.5 scell_state . . . 561
8.518.2.6 TlvPresent . . . 561
8.519 nasPLMNNameReq Struct Reference . . . 561
8.519.1 Detailed Description . . . 562
8.519.2 Field Documentation . . . 562
8.519.2.1 mcc . . . 562
8.519.2.2 mnc . . . 562
8.519.2.3 pMncPcsStatus . . . 562
8.520 nasPLMNNameResp Struct Reference . . . 562
8.520.1 Detailed Description . . . 563
8.520.2 Field Documentation . . . 564
8.520.2.1 longName . . . 564
8.520.2.2 longNameCI . . . 564
8.520.2.3 longNameEn . . . 564
8.520.2.4 longNameLen . . . 564
8.520.2.5 longNameSB . . . 564
8.520.2.6 shortName . . . 564
8.520.2.7 shortNameCI . . . 564
8.520.2.8 shortNameEn . . . 564
8.520.2.9 shortNameLen . . . 564
8.520.2.10 shortNameSB . . . 564
8.520.2.11 spn . . . 564
8.520.2.12 spnEncoding . . . 565
8.520.2.13 spnLength . . . 565
8.521 NASPRLPreferenceTlv Struct Reference . . . 565
8.521.1 Detailed Description . . . 565
8.521.2 Field Documentation . . . 565
8.521.2.1 PRLPref . . . 565
8.521.2.2 TlvPresent . . . 565
8.522 NASQmiCbkNasSwiOTAMessageInd Struct Reference . . . 565

8.522.1 Detailed Description . . . 565
8.522.2 Field Documentation . . . 565
8.522.2.1 nasRelInfoTlv . . . 566
8.522.2.2 otaMsgTlv . . . 566
8.522.2.3 timeTlv . . . 566
8.523 NASQmiCbkNasSystemSelPrefInd Struct Reference . . . 566
8.523.1 Detailed Description . . . 566
8.523.2 Field Documentation . . . 567
8.523.2.1 BPTlv . . . 567
8.523.2.2 EMTlv . . . 567
8.523.2.3 GWAOPTlv . . . 567
8.523.2.4 LBPTlv . . . 567
8.523.2.5 MPTlv . . . 567
8.523.2.6 NSPTlv . . . 567
8.523.2.7 PRLPTlv . . . 567
8.523.2.8 RPTlv . . . 567
8.523.2.9 SDPTlv . . . 567
8.524 NASRoamPreferenceTlv Struct Reference . . . 567
8.524.1 Detailed Description . . . 567
8.524.2 Field Documentation . . . 567
8.524.2.1 RoamPref . . . 567
8.524.2.2 TlvPresent . . . 567
8.525 NASServDomainPrefTlv Struct Reference . . . 567
8.525.1 Detailed Description . . . 568
8.525.2 Field Documentation . . . 568
8.525.2.1 SrvDomainPref . . . 568
8.525.2.2 TlvPresent . . . 568
8.526 NASServingSystemInfo Struct Reference . . . 568
8.526.1 Detailed Description . . . 568
8.526.2 Field Documentation . . . 569
8.526.2.1 csAttachState . . . 569
8.526.2.2 hdrPersonality . . . 569
8.526.2.3 psAttachState . . . 569
8.526.2.4 radioInterfaceList . . . 569
8.526.2.5 radioInterfaceNo . . . 569
8.526.2.6 registrationState . . . 569
8.526.2.7 selectedNetwork . . . 569
8.527 nasSigInfo Struct Reference . . . 569
8.527.1 Detailed Description . . . 570
8.527.2 Field Documentation . . . 570

8.527.2.1 pCDMASigInfo . . . 570
8.527.2.2 pGSMSigInfo . . . 570
8.527.2.3 pHDRSigInfo . . . 570
8.527.2.4 pLTESigInfo . . . 570
8.527.2.5 pRscp . . . 570
8.527.2.6 pTDSCDMASigInfoExt . . . 570
8.527.2.7 pWCDMASigInfo . . . 570
8.528 nasSwiGetChannelLockResp Struct Reference . . . 570
8.528.1 Detailed Description . . . 571
8.528.2 Field Documentation . . . 571
8.528.2.1 pLteEARFCN . . . 571
8.528.2.2 pLtePCI . . . 571
8.528.2.3 pWcdmaUARFCN . . . 571
8.529 NasSwiIndReg Struct Reference . . . 571
8.529.1 Detailed Description . . . 571
8.529.2 Field Documentation . . . 572
8.529.2.1 gsmUmtsDl . . . 572
8.529.2.2 gsmUmtsUl . . . 572
8.529.2.3 lteEmmDl . . . 572
8.529.2.4 lteEmmUl . . . 572
8.529.2.5 lteEsmDl . . . 572
8.529.2.6 lteEsmUl . . . 572
8.529.2.7 pRankIndicatorInd . . . 572
8.529.2.8 pTimer . . . 572
8.530 nasSwiSetChannelLockReq Struct Reference . . . 572
8.530.1 Detailed Description . . . 572
8.530.2 Field Documentation . . . 573
8.530.2.1 pLteEARFCN . . . 573
8.530.2.2 pLtePCI . . . 573
8.530.2.3 pWcdmaUARFCN . . . 573
8.531 nasSysInfo Struct Reference . . . 573
8.531.1 Detailed Description . . . 573
8.531.2 Field Documentation . . . 575
8.531.2.1 pAddCDMASysInfo . . . 575
8.531.2.2 pAddGSMSysInfo . . . 575
8.531.2.3 pAddHDRSysInfo . . . 575
8.531.2.4 pAddLTESysInfo . . . 575
8.531.2.5 pAddWCDMASysInfo . . . 575
8.531.2.6 pCDMASrvStatusInfo . . . 575
8.531.2.7 pCDMASysInfo . . . 575

8.531.2.8 pGSMCallBarringSysInfo . . . 575
8.531.2.9 pGSMCipherDomainSysInfo . . . 575
8.531.2.10 pGSMSrvStatusInfo . . . 575
8.531.2.11 pGSMSysInfo . . . 575
8.531.2.12 pHDRSrvStatusInfo . . . 576
8.531.2.13 pHDRSysInfo . . . 576
8.531.2.14 pLTESrvStatusInfo . . . 576
8.531.2.15 pLTESysInfo . . . 576
8.531.2.16 pLTEVoiceSupportSysInfo . . . 576
8.531.2.17 pSysInfoNoChange . . . 576
8.531.2.18 pWCDMACallBarringSysInfo . . . 576
8.531.2.19 pWCDMACipherDomainSysInfo . . . 576
8.531.2.20 pWCDMASrvStatusInfo . . . 576
8.531.2.21 pWCDMASysInfo . . . 576
8.532 NASTimeInfoTlv Struct Reference . . . 576
8.532.1 Detailed Description . . . 576
8.532.2 Field Documentation . . . 576
8.532.2.1 time . . . 576
8.532.2.2 TlvPresent . . . 576
8.533 nasTimers Struct Reference . . . 576
8.533.1 Detailed Description . . . 577
8.533.2 Field Documentation . . . 577
8.533.2.1 t3396_apn . . . 577
8.533.2.2 t3396_plmn_id . . . 577
8.533.2.3 t3396_val . . . 577
8.534 netSelectionPref Struct Reference . . . 577
8.534.1 Detailed Description . . . 577
8.534.2 Field Documentation . . . 577
8.534.2.1 mcc . . . 578
8.534.2.2 mnc . . . 578
8.534.2.3 netReg . . . 578
8.535 NetStats Struct Reference . . . 578
8.535.1 Detailed Description . . . 578
8.535.2 Field Documentation . . . 578
8.535.2.1 rx_bytes . . . 578
8.535.2.2 rx_errors . . . 579
8.535.2.3 rx_overflows . . . 579
8.535.2.4 rx_packets . . . 579
8.535.2.5 tx_bytes . . . 579
8.535.2.6 tx_errors . . . 579

8.535.2.7 tx_overflows . . . 579
8.535.2.8 tx_packets . . . 579
8.536 NetworkDebugResp Struct Reference . . . 579
8.536.1 Detailed Description . . . 579
8.536.2 Field Documentation . . . 579
8.536.2.1 pDataStatusDetail . . . 579
8.536.2.2 pDeviceConfigDetail . . . 579
8.536.2.3 pNetworkStat1x . . . 580
8.536.2.4 pNetworkStatEVDO . . . 580
8.536.2.5 pObjectVer . . . 580
8.537 NetworkStat1x Struct Reference . . . 580
8.537.1 Detailed Description . . . 580
8.537.2 Field Documentation . . . 581
8.537.2.1 ActSetCnt . . . 581
8.537.2.2 NeighborSetCnt . . . 581
8.537.2.3 pActPilotPNElements . . . 581
8.537.2.4 pNeighborSetPilotPN . . . 582
8.537.2.5 RX_EC_IO . . . 582
8.537.2.6 RX_PWR . . . 582
8.537.2.7 SO . . . 582
8.537.2.8 State . . . 582
8.537.2.9 TX_PWR . . . 582
8.538 NetworkStatEVDO Struct Reference . . . 582
8.538.1 Detailed Description . . . 582
8.538.2 Field Documentation . . . 583
8.538.2.1 MACIndex . . . 583
8.538.2.2 PER . . . 583
8.538.2.3 PilotEnergy . . . 583
8.538.2.4 pSectorID . . . 583
8.538.2.5 RX_PWR . . . 583
8.538.2.6 SectorIDLen . . . 583
8.538.2.7 SNR . . . 583
8.538.2.8 State . . . 583
8.539 newMTMessageTlv Struct Reference . . . 583
8.539.1 Detailed Description . . . 584
8.539.2 Field Documentation . . . 584
8.539.2.1 MTMessageInfo . . . 584
8.539.2.2 TlvPresent . . . 584
8.540 newPwdData Struct Reference . . . 584
8.540.1 Detailed Description . . . 584

8.540.2 Field Documentation . . . 584
8.540.2.1 newPwd . . . 584
8.540.2.2 newPwdAgain . . . 584
8.541 nmrCellInfo Struct Reference . . . 584
8.541.1 Detailed Description . . . 585
8.541.2 Field Documentation . . . 585
8.541.2.1 nmrArfcn . . . 585
8.541.2.2 nmrBsic . . . 586
8.541.2.3 nmrCellID . . . 586
8.541.2.4 nmrLac . . . 586
8.541.2.5 nmrPlmn . . . 586
8.541.2.6 nmrRxLev . . . 586
8.542 NSSAudioCtrl Struct Reference . . . 586
8.542.1 Detailed Description . . . 586
8.542.2 Field Documentation . . . 586
8.542.2.1 downLink . . . 586
8.542.2.2 upLink . . . 586
8.543 NWProfile Struct Reference . . . 586
8.543.1 Detailed Description . . . 586
8.543.2 Field Documentation . . . 587
8.543.2.1 pProfSz . . . 587
8.543.2.2 pProfValues . . . 587
8.543.2.3 tech . . . 587
8.544 omaDmConfigTlv Struct Reference . . . 587
8.544.1 Detailed Description . . . 587
8.544.2 Field Documentation . . . 587
8.544.2.1 alertmsg . . . 587
8.544.2.2 alertmsglength . . . 587
8.544.2.3 state . . . 587
8.544.2.4 userInputReq . . . 587
8.544.2.5 userInputTimeout . . . 588
8.545 omaDmConfigTlvExt Struct Reference . . . 588
8.545.1 Detailed Description . . . 588
8.545.2 Field Documentation . . . 590
8.545.2.1 alertmsg . . . 590
8.545.2.2 alertmsglength . . . 590
8.545.2.3 state . . . 590
8.545.2.4 userInputReq . . . 590
8.545.2.5 userInputTimeout . . . 590
8.546 omaDmFotaTlv Struct Reference . . . 590

8.546.1 Detailed Description . . . 590
8.546.2 Field Documentation . . . 591
8.546.2.1 description . . . 591
8.546.2.2 descriptionlength . . . 591
8.546.2.3 fwdloadsize . . . 592
8.546.2.4 fwloadComplete . . . 592
8.546.2.5 namelength . . . 592
8.546.2.6 package_name . . . 592
8.546.2.7 sessionType . . . 592
8.546.2.8 severity . . . 592
8.546.2.9 state . . . 592
8.546.2.10 updateCompleteStatus . . . 592
8.546.2.11 userInputReq . . . 592
8.546.2.12 userInputTimeout . . . 592
8.546.2.13 version . . . 592
8.546.2.14 versionlength . . . 592
8.547 omaDmFotaTlvExt Struct Reference . . . 592
8.547.1 Detailed Description . . . 592
8.547.2 Field Documentation . . . 594
8.547.2.1 description . . . 594
8.547.2.2 descriptionlength . . . 594
8.547.2.3 fumoResultCode . . . 594
8.547.2.4 namelength . . . 594
8.547.2.5 package_name . . . 594
8.547.2.6 packageSize . . . 594
8.547.2.7 receivedBytes . . . 594
8.547.2.8 reserved . . . 594
8.547.2.9 state . . . 594
8.547.2.10 userInputTimeout . . . 594
8.547.2.11 version . . . 594
8.547.2.12 versionlength . . . 594
8.548 omaDmNotificationsTlv Struct Reference . . . 594
8.548.1 Detailed Description . . . 594
8.548.2 Field Documentation . . . 595
8.548.2.1 notification . . . 595
8.548.2.2 sessionStatus . . . 595
8.549 operatorNameString Struct Reference . . . 595
8.549.1 Detailed Description . . . 595
8.549.2 Field Documentation . . . 595
8.549.2.1 PLMNName . . . 596

8.550 OperatorPLMNData Struct Reference . . . 596
8.550.1 Detailed Description . . . 596
8.550.2 Field Documentation . . . 596
8.550.2.1 lac1 . . . 596
8.550.2.2 lac2 . . . 596
8.550.2.3 mcc . . . 596
8.550.2.4 mnc . . . 596
8.550.2.5 PLMNRecID . . . 596
8.551 operatorPLMNList Struct Reference . . . 596
8.551.1 Detailed Description . . . 597
8.551.2 Field Documentation . . . 597
8.551.2.1 numInstance . . . 597
8.551.2.2 PLMNData . . . 597
8.552 pack_dms_ActivateAutomatic_t Struct Reference . . . 597
8.552.1 Detailed Description . . . 597
8.552.2 Field Documentation . . . 597
8.552.2.1 actCode . . . 597
8.553 pack_dms_GetCustFeaturesV2_t Struct Reference . . . 597
8.553.1 Detailed Description . . . 598
8.553.2 Field Documentation . . . 598
8.553.2.1 cust_id . . . 598
8.553.2.2 list_type . . . 598
8.553.2.3 Tlvresult . . . 598
8.554 pack_dms_ResetToFactoryDefaults_t Struct Reference . . . 598
8.554.1 Detailed Description . . . 598
8.554.2 Field Documentation . . . 598
8.554.2.1 spc . . . 598
8.555 pack_dms_SetActivationStatusCallback_t Struct Reference . . . 598
8.555.1 Detailed Description . . . 599
8.555.2 Field Documentation . . . 599
8.555.2.1 activationState . . . 599
8.556 pack_dms_SetCrashAction_t Struct Reference . . . 599
8.556.1 Detailed Description . . . 599
8.556.2 Field Documentation . . . 599
8.556.2.1 crashAction . . . 599
8.557 pack_dms_SetCustFeature_t Struct Reference . . . 599
8.557.1 Detailed Description . . . 600
8.557.2 Field Documentation . . . 601
8.557.2.1 DHCPRelayEnabled . . . 601
8.557.2.2 DisableIMSI . . . 601

8.557.2.3 GpsEnable . . . 601
8.557.2.4 GPSLPM . . . 601
8.557.2.5 GPSSel . . . 601
8.557.2.6 IPFamSupport . . . 601
8.557.2.7 IsVoiceEnabled . . . 601
8.557.2.8 RMAutoConnect . . . 601
8.557.2.9 SMSSupport . . . 601
8.558 pack_dms_SetCustFeaturesV2_t Struct Reference . . . 601
8.558.1 Detailed Description . . . 602
8.558.2 Field Documentation . . . 602
8.558.2.1 cust_id . . . 602
8.558.2.2 cust_value . . . 602
8.558.2.3 Tlvresult . . . 602
8.558.2.4 value_length . . . 602
8.559 pack_dms_SetEventReport_t Struct Reference . . . 602
8.559.1 Detailed Description . . . 602
8.559.2 Field Documentation . . . 602
8.559.2.1 mode . . . 602
8.560 pack_dms_SetPower_t Struct Reference . . . 603
8.560.1 Detailed Description . . . 603
8.560.2 Field Documentation . . . 603
8.560.2.1 mode . . . 603
8.560.2.2 Tlvresult . . . 603
8.561 pack_dms_SetUSBComp_t Struct Reference . . . 603
8.561.1 Detailed Description . . . 603
8.561.2 Field Documentation . . . 604
8.561.2.1 Tlvresult . . . 604
8.561.2.2 USBComp . . . 604
8.562 pack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference . . . 604
8.562.1 Detailed Description . . . 604
8.562.2 Field Documentation . . . 605
8.562.2.1 resetInfoInd . . . 605
8.563 pack_dms_SLQSSwiGetCrashInfo_t Struct Reference . . . 605
8.563.1 Detailed Description . . . 605
8.563.2 Field Documentation . . . 605
8.563.2.1 clear . . . 605
8.564 pack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference . . . 605
8.564.1 Detailed Description . . . 605
8.564.2 Field Documentation . . . 605
8.564.2.1 pDestSMSContent . . . 605

8.564.2.2 pDestSMSNum . . . 606
8.565 pack_dms_SLQSSwiSetHostDevInfo_t Struct Reference . . . 606
8.565.1 Detailed Description . . . 606
8.565.2 Field Documentation . . . 606
8.565.2.1 manString . . . 606
8.565.2.2 modelString . . . 606
8.565.2.3 plasmaIDString . . . 606
8.565.2.4 swVerString . . . 606
8.566 pack_dms_SLQSSwiSetOSInfo_t Struct Reference . . . 606
8.566.1 Detailed Description . . . 607
8.566.2 Field Documentation . . . 607
8.566.2.1 nameString . . . 607
8.566.2.2 versionString . . . 607
8.567 pack_dms_UIMChangePIN_t Struct Reference . . . 607
8.567.1 Detailed Description . . . 607
8.567.2 Field Documentation . . . 607
8.567.2.1 id . . . 607
8.567.2.2 newValue . . . 607
8.567.2.3 oldValue . . . 607
8.568 pack_dms_UIMGetControlKeyStatus_t Struct Reference . . . 608
8.568.1 Detailed Description . . . 608
8.568.2 Field Documentation . . . 608
8.568.2.1 facility . . . 608
8.569 pack_dms_UIMGetICCID_t Struct Reference . . . 608
8.569.1 Detailed Description . . . 608
8.569.2 Field Documentation . . . 608
8.569.2.1 Tlvresult . . . 608
8.570 pack_dms_UIMSetControlKeyProtection_t Struct Reference . . . 608
8.570.1 Detailed Description . . . 609
8.570.2 Field Documentation . . . 609
8.570.2.1 facility . . . 609
8.570.2.2 facilityCk . . . 609
8.570.2.3 facilityState . . . 609
8.571 pack_dms_UIMSetPINProtection_t Struct Reference . . . 609
8.571.1 Detailed Description . . . 609
8.571.2 Field Documentation . . . 610
8.571.2.1 bEnable . . . 610
8.571.2.2 id . . . 610
8.571.2.3 value . . . 610
8.572 pack_dms_UIMUnblockControlKey_t Struct Reference . . . 610

8.572.1 Detailed Description . . . 610
8.572.2 Field Documentation . . . 610
8.572.2.1 facility . . . 610
8.572.2.2 facilityCk . . . 610
8.573pack_dms_UIMUnblockPIN_t Struct Reference . . . 610
8.573.1 Detailed Description . . . 611
8.573.2 Field Documentation . . . 611
8.573.2.1 id . . . 611
8.573.2.2 newPin . . . 611
8.573.2.3 pukValue . . . 611
8.574pack_dms_UIMVerifyPIN_t Struct Reference . . . 611
8.574.1 Detailed Description . . . 611
8.574.2 Field Documentation . . . 611
8.574.2.1 id . . . 611
8.574.2.2 value . . . 611
8.575pack_fms_GetImagesPreference_t Struct Reference . . . 612
8.575.1 Detailed Description . . . 612
8.575.2 Field Documentation . . . 612
8.575.2.1 Tlvresult . . . 612
8.576pack_fms_GetStoredImages_t Struct Reference . . . 612
8.576.1 Detailed Description . . . 612
8.576.2 Field Documentation . . . 612
8.576.2.1 Tlvresult . . . 612
8.577pack_fms_SetImagesPreference_t Struct Reference . . . 612
8.577.1 Detailed Description . . . 613
8.577.2 Field Documentation . . . 613
8.577.2.1 bForceDownload . . . 613
8.577.2.2 imageListSize . . . 613
8.577.2.3 modemindex . . . 613
8.577.2.4 pImageList . . . 613
8.577.2.5 Tlvresult . . . 613
8.578pack_loc_Delete_Assist_Data_t Struct Reference . . . 613
8.578.1 Detailed Description . . . 613
8.578.2 Field Documentation . . . 614
8.578.2.1 pBdsSVInfo . . . 614
8.578.2.2 pCellDb . . . 614
8.578.2.3 pClkInfo . . . 614
8.578.2.4 pGnssData . . . 614
8.578.2.5 pSVInfo . . . 614
8.578.2.6 Tlvresult . . . 614

8.579 pack_loc_EventRegister_t Struct Reference . . . 614
8.579.1 Detailed Description . . . 614
8.579.2 Field Documentation . . . 616
8.579.2.1 eventRegister . . . 616
8.579.2.2 Tlvresult . . . 616
8.580 pack_loc_SetExtPowerState_t Struct Reference . . . 616
8.580.1 Detailed Description . . . 616
8.580.2 Field Documentation . . . 616
8.580.2.1 extPowerState . . . 616
8.580.2.2 Tlvresult . . . 616
8.581 pack_loc_SetOperationMode_t Struct Reference . . . 616
8.581.1 Detailed Description . . . 617
8.581.2 Field Documentation . . . 617
8.581.2.1 mode . . . 617
8.581.2.2 Tlvresult . . . 617
8.582 pack_loc_SLQSLOCGetBestAvailPos_t Struct Reference . . . 617
8.582.1 Detailed Description . . . 617
8.582.2 Field Documentation . . . 617
8.582.2.1 Tlvresult . . . 617
8.582.2.2 xid . . . 618
8.583 pack_loc_SLQSLOCGetServer_t Struct Reference . . . 618
8.583.1 Detailed Description . . . 618
8.583.2 Field Documentation . . . 618
8.583.2.1 has_serverAddrTypeMask . . . 618
8.583.2.2 serverAddrTypeMask . . . 618
8.583.2.3 serverType . . . 618
8.584 pack_loc_SLQSLOCInjectPosition_t Struct Reference . . . 618
8.584.1 Detailed Description . . . 619
8.584.2 Field Documentation . . . 622
8.584.2.1 altitudeSrcInfo . . . 622
8.584.2.2 altitudeWrtEllipsoid . . . 622
8.584.2.3 altitudeWrtMeanSeaLevel . . . 622
8.584.2.4 has_altitudeSrcInfo . . . 622
8.584.2.5 has_altitudeWrtEllipsoid . . . 622
8.584.2.6 has_altitudeWrtMeanSeaLevel . . . 622
8.584.2.7 has_horConfidence . . . 622
8.584.2.8 has_horReliability . . . 622
8.584.2.9 has_horUncCircular . . . 622
8.584.2.10 has_latitude . . . 622
8.584.2.11 has_longitude . . . 622

8.584.2.12 has_positionSrc . . . 622
8.584.2.13 has_rawHorConfidence . . . 622
8.584.2.14 has_rawHorUncCircular . . . 622
8.584.2.15 has_timestampAge . . . 622
8.584.2.16 has_timestampUtc . . . 622
8.584.2.17 has_vertConfidence . . . 622
8.584.2.18 has_vertRelicability . . . 623
8.584.2.19 has_vertUnc . . . 623
8.584.2.20 horConfidence . . . 623
8.584.2.21 horReliability . . . 623
8.584.2.22 horUncCircular . . . 623
8.584.2.23 latitude . . . 623
8.584.2.24 longitude . . . 623
8.584.2.25 positionSrc . . . 623
8.584.2.26 rawHorConfidence . . . 623
8.584.2.27 rawHorUncCircular . . . 623
8.584.2.28 timestampAge . . . 623
8.584.2.29 timestampUtc . . . 623
8.584.2.30 vertConfidence . . . 623
8.584.2.31 vertReliability . . . 623
8.584.2.32 vertUnc . . . 623
8.585 pack_loc_SLQSLOCInjectSensorData_t Struct Reference . . . 623
8.585.1 Detailed Description . . . 623
8.585.2 Field Documentation . . . 624
8.585.2.1 acceleroData . . . 624
8.585.2.2 acceleroTimeSrc . . . 624
8.585.2.3 accelTemp . . . 624
8.585.2.4 gyroData . . . 624
8.585.2.5 gyroTemp . . . 625
8.585.2.6 gyroTimeSrc . . . 625
8.585.2.7 has_acceleroTimeSrc . . . 625
8.585.2.8 has_accelTemp . . . 625
8.585.2.9 has_accleroData . . . 625
8.585.2.10 has_gyroData . . . 625
8.585.2.11 has_gyroTemp . . . 625
8.585.2.12 has_gyroTimeSrc . . . 625
8.585.2.13 has_opaqueId . . . 625
8.585.2.14 opaqueId . . . 625
8.586 pack_loc_SLQSLOCInjectUTCTime_t Struct Reference . . . 625
8.586.1 Detailed Description . . . 625

8.586.2 Field Documentation . . . 625
8.586.2.1 timeMsec . . . 625
8.586.2.2 timeUncMsec . . . 625
8.587 pack_loc_SLQSLOCSetCradleMountConfig_t Struct Reference . . . 625
8.587.1 Detailed Description . . . 626
8.587.2 Field Documentation . . . 626
8.587.2.1 confidence . . . 626
8.587.2.2 has_confidence . . . 626
8.587.2.3 state . . . 626
8.588 pack_loc_Start_t Struct Reference . . . 626
8.588.1 Detailed Description . . . 626
8.588.2 Field Documentation . . . 627
8.588.2.1 pApplicationInfo . . . 627
8.588.2.2 pConfigAltitudeAssumed . . . 628
8.588.2.3 pHorizontalAccuracyLvl . . . 628
8.588.2.4 pIntermediateReportState . . . 628
8.588.2.5 pMinIntervalTime . . . 628
8.588.2.6 pRecurrenceType . . . 628
8.588.2.7 SessionId . . . 628
8.588.2.8 Tlvresult . . . 628
8.589 pack_loc_Stop_t Struct Reference . . . 628
8.589.1 Detailed Description . . . 628
8.589.2 Field Documentation . . . 628
8.589.2.1 SessionId . . . 628
8.589.2.2 Tlvresult . . . 628
8.590 pack_nas_InitiateDomainAttach_t Struct Reference . . . 628
8.590.1 Detailed Description . . . 628
8.590.2 Field Documentation . . . 629
8.590.2.1 action . . . 629
8.591 pack_nas_SetACCOLC_t Struct Reference . . . 629
8.591.1 Detailed Description . . . 629
8.591.2 Field Documentation . . . 629
8.591.2.1 accolc . . . 629
8.591.2.2 spc . . . 629
8.592 pack_nas_SetCDMANetworkParameters_t Struct Reference . . . 629
8.592.1 Detailed Description . . . 629
8.592.2 Field Documentation . . . 630
8.592.2.1 pApplication . . . 630
8.592.2.2 pBroadcast . . . 630
8.592.2.3 pCustomSCP . . . 631

8.592.2.4 pForceRev0 . . . 631
8.592.2.5 pProtocol . . . 631
8.592.2.6 pRoaming . . . 631
8.592.2.7 pSPC . . . 631
8.593 pack_nas_SetNetworkPreference_t Struct Reference . . . 631
8.593.1 Detailed Description . . . 631
8.593.2 Field Documentation . . . 631
8.593.2.1 Duration . . . 631
8.593.2.2 TechnologyPref . . . 632
8.593.2.3 Tlvresult . . . 632
8.594 pack_nas_SLQSConfigSigInfo_t Struct Reference . . . 632
8.594.1 Detailed Description . . . 632
8.594.2 Field Documentation . . . 633
8.594.2.1 pECIOThresh . . . 633
8.594.2.2 pHDRSINRThresh . . . 633
8.594.2.3 pIOThresh . . . 633
8.594.2.4 pLTESigRptCfg . . . 633
8.594.2.5 pLTESNRThresh . . . 633
8.594.2.6 pRSRPThresh . . . 633
8.594.2.7 pRSRQThresh . . . 633
8.594.2.8 pRSSIThresh . . . 633
8.594.2.9 pTDSCDMASINRCONFThresh . . . 633
8.595 pack_nas_SLQSGetPLMNName_t Struct Reference . . . 633
8.595.1 Detailed Description . . . 633
8.595.2 Field Documentation . . . 634
8.595.2.1 mcc . . . 634
8.595.2.2 mnc . . . 634
8.595.2.3 pMncPcsStatus . . . 634
8.596 pack_nas_SLQSInitiateNetworkRegistration_t Struct Reference . . . 634
8.596.1 Detailed Description . . . 634
8.596.2 Field Documentation . . . 635
8.596.2.1 pChangeDuration . . . 635
8.596.2.2 pMncPcsDigitStatus . . . 635
8.596.2.3 pMNRInfo . . . 635
8.596.2.4 regAction . . . 635
8.597 pack_nas_SLQSNasConfigSigInfo2_t Struct Reference . . . 635
8.597.1 Detailed Description . . . 636
8.597.2 Field Documentation . . . 638
8.597.2.1 pCDMAECIODelta . . . 638
8.597.2.2 pCDMAECIOThresh . . . 638

8.597.2.3 pCDMARSSIDelta . . . 638
8.597.2.4 pCDMARSSIThresh . . . 638
8.597.2.5 pGSMRSSIDelta . . . 638
8.597.2.6 pGSMRSSIThresh . . . 638
8.597.2.7 pHDRECIODelta . . . 638
8.597.2.8 pHDRECIOThresh . . . 639
8.597.2.9 pHDRIODelta . . . 639
8.597.2.10 pHDRIOThresh . . . 639
8.597.2.11 pHDRRSSIDelta . . . 639
8.597.2.12 pHDRRSSIThresh . . . 639
8.597.2.13 pHDRSINRDelta . . . 639
8.597.2.14 pHDRSINRThresh . . . 639
8.597.2.15 pLTERSRPDelta . . . 639
8.597.2.16 pLTERSRPThresh . . . 639
8.597.2.17 pLTERSRQDelta . . . 639
8.597.2.18 pLTERSRQThresh . . . 639
8.597.2.19 pLTERSSIDelta . . . 639
8.597.2.20 pLTERSSIThresh . . . 639
8.597.2.21 pLTESigRptConfig . . . 639
8.597.2.22 pLTESNRDelta . . . 639
8.597.2.23 pLTESNRThresh . . . 639
8.597.2.24 pTDSCDMAECIODelta . . . 639
8.597.2.25 pTDSCDMAECIOThresh . . . 639
8.597.2.26 pTDSCDMARSCPDelta . . . 639
8.597.2.27 pTDSCDMARSCPThresh . . . 639
8.597.2.28 pTDSCDMARSSIDelta . . . 639
8.597.2.29 pTDSCDMARSSIThresh . . . 639
8.597.2.30 pTDSCDMASINRDelta . . . 639
8.597.2.31 pTDSCDMASINRThresh . . . 639
8.597.2.32 pWCDMAECIODelta . . . 639
8.597.2.33 pWCDMAECIOThresh . . . 639
8.597.2.34 pWCDMARSSIDelta . . . 639
8.597.2.35 pWCDMARSSIThresh . . . 639
8.598 pack_nas_SLQSNasGet3GPP2Subscription_t Struct Reference . . . 640
8.598.1 Detailed Description . . . 640
8.598.2 Field Documentation . . . 640
8.598.2.1 namID . . . 640
8.599 pack_nas_SLQSNasGetTxRxInfo_t Struct Reference . . . 640
8.599.1 Detailed Description . . . 640
8.599.2 Field Documentation . . . 640

8.599.2.1 radio_if . . . 640
8.600 pack_nas_SLQSNasIndicationRegisterExt_t Struct Reference . . . 640
8.600.1 Detailed Description . . . 641
8.600.2 Field Documentation . . . 643
8.600.2.1 pDDTMInd . . . 643
8.600.2.2 pDualStandByPrefInd . . . 643
8.600.2.3 pErrorRateInd . . . 643
8.600.2.4 pHDRNewUATIAssInd . . . 643
8.600.2.5 pHDRSessionCloseInd . . . 643
8.600.2.6 pLTECphyCa . . . 643
8.600.2.7 pManagedRoamingInd . . . 643
8.600.2.8 pNetworkRejectInd . . . 643
8.600.2.9 pNetworkTimeInd . . . 643
8.600.2.10 pServingSystemInd . . . 643
8.600.2.11 pSignalStrengthInd . . . 643
8.600.2.12 pSubscriptionInfoInd . . . 643
8.600.2.13 pSuppressSysInfoInd . . . 643
8.600.2.14 pSysInfoInd . . . 643
8.600.2.15 pSystemSelectionInd . . . 643
8.601 pack_nas_SLQSNasSwiIndicationRegister_t Struct Reference . . . 643
8.601.1 Detailed Description . . . 644
8.601.2 Field Documentation . . . 644
8.601.2.1 gsmUmtsDl . . . 644
8.601.2.2 gsmUmtsUl . . . 644
8.601.2.3 lteEmmDl . . . 644
8.601.2.4 lteEmmUl . . . 644
8.601.2.5 lteEsmDl . . . 644
8.601.2.6 lteEsmUl . . . 644
8.601.2.7 pRankIndicatorInd . . . 644
8.601.2.8 pTimer . . . 645
8.602 pack_nas_SLQSNASSwiSetChannelLock_t Struct Reference . . . 645
8.602.1 Detailed Description . . . 645
8.602.2 Field Documentation . . . 645
8.602.2.1 pLteEARFCN . . . 645
8.602.2.2 pLtePCI . . . 645
8.602.2.3 pWcdmaUARFCN . . . 645
8.603 pack_nas_SLQSSetBandPreference_t Struct Reference . . . 645
8.603.1 Detailed Description . . . 645
8.603.2 Field Documentation . . . 646
8.603.2.1 bandPref . . . 647

8.604 pack_nas_SLQSSetSignalStrengthsCallback_t Struct Reference . . . 647
8.604.1 Detailed Description . . . 647
8.604.2 Field Documentation . . . 647
8.604.2.1 bEnable . . . 647
8.604.2.2 pSigIndReq . . . 647
8.605 pack_nas_SLQSSetSysSelectionPref_t Struct Reference . . . 647
8.605.1 Detailed Description . . . 647
8.605.2 Field Documentation . . . 652
8.605.2.1 pAcqOrderPref . . . 652
8.605.2.2 pBandPref . . . 652
8.605.2.3 pChgDuration . . . 652
8.605.2.4 pCSGID . . . 652
8.605.2.5 pEmerMode . . . 652
8.605.2.6 pGWAcqOrderPref . . . 652
8.605.2.7 pLTEBandPref . . . 652
8.605.2.8 pMNCInCPCSDigStat . . . 652
8.605.2.9 pModePref . . . 652
8.605.2.10 pNetSelPref . . . 652
8.605.2.11 pPRLPref . . . 652
8.605.2.12 pRAT . . . 652
8.605.2.13 pRoamPref . . . 652
8.605.2.14 pSrvDomainPref . . . 652
8.605.2.15 pSrvRegRestriction . . . 653
8.605.2.16 pTdscdmaBandPref . . . 653
8.606 pack_nas_SLQSSwiPSDetach_t Struct Reference . . . 653
8.606.1 Detailed Description . . . 653
8.606.2 Field Documentation . . . 653
8.606.2.1 pDetachAction . . . 653
8.607 pack_pds_PDSInjectTimeReference_t Struct Reference . . . 653
8.607.1 Detailed Description . . . 653
8.607.2 Field Documentation . . . 653
8.607.2.1 systemDiscontinuities . . . 653
8.607.2.2 systemTime . . . 653
8.608 pack_pds_ResetPDSData_t Struct Reference . . . 654
8.608.1 Detailed Description . . . 654
8.608.2 Field Documentation . . . 654
8.608.2.1 pCellDataMask . . . 654
8.608.2.2 pGPSDataMask . . . 654
8.609 pack_pds_SetEventReportCallback_t Struct Reference . . . 654
8.609.1 Detailed Description . . . 655

8.609.2 Field Documentation . . . 655
8.609.2.1 posDataNmea . . . 655
8.609.2.2 rptPosData . . . 655
8.610pack_pds_SetPDSDefaults_t Struct Reference . . . 655
8.610.1 Detailed Description . . . 655
8.610.2 Field Documentation . . . 656
8.610.2.1 accuracy . . . 656
8.610.2.2 interval . . . 656
8.610.2.3 operation . . . 656
8.610.2.4 timeout . . . 656
8.611pack_pds_SetPDSState_t Struct Reference . . . 656
8.611.1 Detailed Description . . . 656
8.611.2 Field Documentation . . . 656
8.611.2.1 enable . . . 656
8.612pack_pds_SetPortAutomaticTracking_t Struct Reference . . . 656
8.612.1 Detailed Description . . . 656
8.612.2 Field Documentation . . . 657
8.612.2.1 bAuto . . . 657
8.613pack_pds_SetServiceAutomaticTracking_t Struct Reference . . . 657
8.613.1 Detailed Description . . . 657
8.613.2 Field Documentation . . . 657
8.613.2.1 bAuto . . . 657
8.614pack_pds_SetXTRAAutomaticDownload_t Struct Reference . . . 657
8.614.1 Detailed Description . . . 657
8.614.2 Field Documentation . . . 658
8.614.2.1 bEnabled . . . 658
8.614.2.2 interval . . . 658
8.615pack_pds_SetXTRANetwork_t Struct Reference . . . 658
8.615.1 Detailed Description . . . 658
8.615.2 Field Documentation . . . 658
8.615.2.1 preference . . . 658
8.616pack_pds_SLQSGetAGPSConfig_t Struct Reference . . . 658
8.616.1 Detailed Description . . . 658
8.616.2 Field Documentation . . . 659
8.616.2.1 pNetworkMode . . . 659
8.617pack_pds_SLQSPDSInjectAbsoluteTimeReference_t Struct Reference . . . 659
8.617.1 Detailed Description . . . 659
8.617.2 Field Documentation . . . 659
8.617.2.1 forceFlag . . . 659
8.617.2.2 timeBase . . . 659

8.617.2.3 timeMsec . . . 659
8.617.2.4 timeUncMsec . . . 659
8.618 pack_pds_SLQSPDSInjectPositionData_t Struct Reference . . . 659
8.618.1 Detailed Description . . . 660
8.618.2 Field Documentation . . . 661
8.618.2.1 pAltitudeWrtEllipsoid . . . 661
8.618.2.2 pAltitudeWrtSealevel . . . 661
8.618.2.3 pHorizontalConfidence . . . 661
8.618.2.4 pHorizontalUncCircular . . . 661
8.618.2.5 pLatitude . . . 661
8.618.2.6 pLongitude . . . 661
8.618.2.7 pPositionSource . . . 661
8.618.2.8 pTimeStamp . . . 661
8.618.2.9 pTimeType . . . 661
8.618.2.10 pVerticalConfidence . . . 661
8.618.2.11 pVerticalUnc . . . 661
8.619 pack_pds_SLQSSetAGPSConfig_t Struct Reference . . . 661
8.619.1 Detailed Description . . . 662
8.619.2 Field Documentation . . . 662
8.619.2.1 pNetworkMode . . . 662
8.619.2.2 pServerAddress . . . 662
8.619.2.3 pServerPort . . . 662
8.619.2.4 pServerURL . . . 662
8.619.2.5 pServerURLLength . . . 662
8.620 pack_pds_SLQSSetPositionMethodState_t Struct Reference . . . 662
8.620.1 Detailed Description . . . 662
8.620.2 Field Documentation . . . 663
8.620.2.1 pWifiState . . . 663
8.620.2.2 pXtraDataState . . . 663
8.620.2.3 pXtraTimeState . . . 663
8.621 pack_pds_StartPDSTrackingSessionExt_t Struct Reference . . . 663
8.621.1 Detailed Description . . . 663
8.621.2 Field Documentation . . . 664
8.621.2.1 fixAccuracy . . . 664
8.621.2.2 fixCount . . . 664
8.621.2.3 fixInterval . . . 664
8.621.2.4 fixTimeout . . . 664
8.621.2.5 sessionControl . . . 664
8.621.2.6 sessionOperation . . . 664
8.621.2.7 sessionServerOption . . . 664

8.621.2.8 sessionType . . . 664
8.622 pack_qmi_t Struct Reference . . . 664
8.622.1 Detailed Description . . . 665
8.622.2 Field Documentation . . . 665
8.622.2.1 msgid . . . 665
8.622.2.2 svc . . . 665
8.622.2.3 timeout . . . 665
8.622.2.4 xid . . . 665
8.623 pack_qos_SLQSQosSwiReadApnExtraParams_t Struct Reference . . . 665
8.623.1 Detailed Description . . . 665
8.623.2 Field Documentation . . . 665
8.623.2.1 apnId . . . 665
8.624 pack_qos_SLQSQosSwiReadDataStats_t Struct Reference . . . 665
8.624.1 Detailed Description . . . 665
8.624.2 Field Documentation . . . 666
8.624.2.1 apnId . . . 666
8.625 pack_qos_SLQSSetQosEventCallback_t Struct Reference . . . 666
8.625.1 Detailed Description . . . 666
8.625.2 Field Documentation . . . 666
8.625.2.1 enable . . . 666
8.626 pack_rms_SetSMSWake_t Struct Reference . . . 666
8.626.1 Detailed Description . . . 666
8.626.2 Field Documentation . . . 667
8.626.2.1 enabled . . . 667
8.626.2.2 wake_mask . . . 667
8.627 pack_sar_SLQSSetRfSarState_t Struct Reference . . . 667
8.627.1 Detailed Description . . . 667
8.627.2 Field Documentation . . . 667
8.627.2.1 RfSarState . . . 667
8.628 pack_sms_SaveSMS_t Struct Reference . . . 667
8.628.1 Detailed Description . . . 667
8.628.2 Field Documentation . . . 668
8.628.2.1 messageFormat . . . 668
8.628.2.2 messageSize . . . 668
8.628.2.3 pMessage . . . 668
8.628.2.4 storageType . . . 668
8.629 pack_sms_SendSMS_t Struct Reference . . . 668
8.629.1 Detailed Description . . . 668
8.629.2 Field Documentation . . . 669
8.629.2.1 messageFormat . . . 669

8.629.2.2 messageSize . . . 669
8.629.2.3 pLinktimer . . . 669
8.629.2.4 pMessage . . . 669
8.630 pack_sms_SetNewSMSCallback_t Struct Reference . . . 669
8.630.1 Detailed Description . . . 669
8.630.2 Field Documentation . . . 669
8.630.2.1 status . . . 669
8.631 pack_sms_SetSMSCAddress_t Struct Reference . . . 669
8.631.1 Detailed Description . . . 669
8.631.2 Field Documentation . . . 670
8.631.2.1 pSMSCAddress . . . 670
8.631.2.2 pSMSCType . . . 670
8.632 pack_sms_SLQSDeleteSMS_t Struct Reference . . . 670
8.632.1 Detailed Description . . . 670
8.632.2 Field Documentation . . . 670
8.632.2.1 pMessageIndex . . . 671
8.632.2.2 pMessageMode . . . 671
8.632.2.3 pMessageTag . . . 671
8.632.2.4 storageType . . . 671
8.633 pack_sms_SLQSGetSMS_t Struct Reference . . . 671
8.633.1 Detailed Description . . . 671
8.633.2 Field Documentation . . . 671
8.633.2.1 messageIndex . . . 671
8.633.2.2 pMessageMode . . . 671
8.633.2.3 storageType . . . 671
8.634 pack_sms_SLQSGetSmsBroadcastConfig_t Struct Reference . . . 671
8.634.1 Detailed Description . . . 671
8.634.2 Field Documentation . . . 672
8.634.2.1 mode . . . 672
8.635 pack_sms_SLQSGetSMSList_t Struct Reference . . . 672
8.635.1 Detailed Description . . . 672
8.635.2 Field Documentation . . . 672
8.635.2.1 pMessageMode . . . 672
8.635.2.2 pRequestedTag . . . 672
8.635.2.3 storageType . . . 672
8.636 pack_sms_SLQSModifySMSStatus_t Struct Reference . . . 673
8.636.1 Detailed Description . . . 673
8.636.2 Field Documentation . . . 673
8.636.2.1 messageIndex . . . 673
8.636.2.2 messageTag . . . 673

8.636.2.3 pMessageMode . . . 673
8.636.2.4 storageType . . . 673
8.637 pack_sms_SLQSSendAsyncSMS_t Struct Reference . . . 673
8.637.1 Detailed Description . . . 673
8.637.2 Field Documentation . . . 674
8.637.2.1 pSendSmsParams . . . 674
8.638 pack_sms_SLQSSetIndicationRegister_t Struct Reference . . . 674
8.638.1 Detailed Description . . . 674
8.638.2 Field Documentation . . . 674
8.638.2.1 pSetIndicationRegReq . . . 674
8.639 pack_sms_SLQSSetSmsBroadcastActivation_t Struct Reference . . . 674
8.639.1 Detailed Description . . . 674
8.639.2 Field Documentation . . . 675
8.639.2.1 broadcastActivate . . . 675
8.639.2.2 mode . . . 675
8.640 pack_sms_SLQSSetSmsBroadcastConfig_t Struct Reference . . . 675
8.640.1 Detailed Description . . . 675
8.640.2 Field Documentation . . . 675
8.640.2.1 mode . . . 675
8.640.2.2 pBroadcastConfig . . . 675
8.640.2.3 pCDMABroadcastConfig . . . 675
8.641 pack_sms_SLQSSetSmsStorage_t Struct Reference . . . 675
8.641.1 Detailed Description . . . 675
8.641.2 Field Documentation . . . 676
8.641.2.1 smsStorage . . . 676
8.642 pack_sms_SLQSSmsGetMaxStorageSize_t Struct Reference . . . 676
8.642.1 Detailed Description . . . 676
8.642.2 Field Documentation . . . 676
8.642.2.1 pMaxStorageSizeReq . . . 676
8.643 pack_sms_SLQSSmsSetRoutes_t Struct Reference . . . 676
8.643.1 Detailed Description . . . 676
8.643.2 Field Documentation . . . 677
8.643.2.1 pSetRoutesReq . . . 677
8.644 pack_swiloc_SwiLocSetAutoStart_t Struct Reference . . . 677
8.644.1 Detailed Description . . . 677
8.644.2 Field Documentation . . . 678
8.644.2.1 fix_rate . . . 678
8.644.2.2 fix_type . . . 678
8.644.2.3 function . . . 678
8.644.2.4 max_dist . . . 678

8.644.2.5 max_time . . . 678
8.644.2.6 set_fix_rate . . . 678
8.644.2.7 set_fix_type . . . 678
8.644.2.8 set_function . . . 678
8.644.2.9 set_max_dist . . . 678
8.644.2.10 set_max_time . . . 678
8.645 pack_swioma_SLQSOMADMCancelSession_t Struct Reference . . . 678
8.645.1 Detailed Description . . . 678
8.645.2 Field Documentation . . . 679
8.645.2.1 sessionType . . . 679
8.646 pack_swioma_SLQSOMADMGetSessionInfo_t Struct Reference . . . 679
8.646.1 Detailed Description . . . 679
8.646.2 Field Documentation . . . 679
8.646.2.1 SessionType . . . 679
8.647 pack_swioma_SLQSOMADMSendSelection_t Struct Reference . . . 679
8.647.1 Detailed Description . . . 679
8.647.2 Field Documentation . . . 680
8.647.2.1 pDeferTime . . . 680
8.647.2.2 pRejectReason . . . 680
8.647.2.3 selection . . . 680
8.648 pack_swioma_SLQSOMADMSetSettings_t Struct Reference . . . 680
8.648.1 Detailed Description . . . 680
8.648.2 Field Documentation . . . 681
8.648.2.1 FOTAdownload . . . 681
8.648.2.2 FOTAUpdate . . . 681
8.648.2.3 pAutosdm . . . 681
8.648.2.4 pFwAutoCheck . . . 681
8.649 pack_swioma_SLQSOMADMStartSession_t Struct Reference . . . 681
8.649.1 Detailed Description . . . 681
8.649.2 Field Documentation . . . 681
8.649.2.1 sessionType . . . 681
8.650 pack_uim_ChangePin_t Struct Reference . . . 681
8.650.1 Detailed Description . . . 682
8.650.2 Field Documentation . . . 682
8.650.2.1 changePIN . . . 682
8.650.2.2 EncryptedPIN1 . . . 682
8.650.2.3 pIndicationToken . . . 682
8.650.2.4 pKeyReferenceID . . . 682
8.650.2.5 sessionInfo . . . 682
8.650.2.6 Tlvresult . . . 682

8.651 pack_uim_ReadTransparent_t Struct Reference . . . 682
8.651.1 Detailed Description . . . 683
8.651.2 Field Documentation . . . 683
8.651.2.1 fileIndex . . . 683
8.651.2.2 pEncryptData . . . 683
8.651.2.3 pIndicationToken . . . 683
8.651.2.4 readTransparent . . . 683
8.651.2.5 sessionInfo . . . 683
8.651.2.6 Tlvresult . . . 683
8.652 pack_uim_SetPinProtection_t Struct Reference . . . 684
8.652.1 Detailed Description . . . 684
8.652.2 Field Documentation . . . 684
8.652.2.1 EncryptedPIN1 . . . 684
8.652.2.2 pIndicationToken . . . 684
8.652.2.3 pinProtection . . . 684
8.652.2.4 pKeyReferenceID . . . 684
8.652.2.5 sessionInfo . . . 684
8.652.2.6 Tlvresult . . . 685
8.653 pack_uim_SLQSUIMAuthenticate_t Struct Reference . . . 685
8.653.1 Detailed Description . . . 685
8.653.2 Field Documentation . . . 685
8.653.2.1 authData . . . 685
8.653.2.2 pIndicationToken . . . 685
8.653.2.3 sessionInfo . . . 685
8.654 pack_uim_SLQSUIMDepersonalization_t Struct Reference . . . 685
8.654.1 Detailed Description . . . 685
8.654.2 Field Documentation . . . 686
8.654.2.1 depersonilisationInfo . . . 686
8.655 pack_uim_SLQSUIMEventRegister_t Struct Reference . . . 686
8.655.1 Detailed Description . . . 686
8.655.2 Field Documentation . . . 686
8.655.2.1 eventMask . . . 686
8.656 pack_uim_SLQSUIMGetConfiguration_t Struct Reference . . . 686
8.656.1 Detailed Description . . . 686
8.656.2 Field Documentation . . . 686
8.656.2.1 pConfigurationMask . . . 686
8.657 pack_uim_SLQSUIMGetFileAttributes_t Struct Reference . . . 687
8.657.1 Detailed Description . . . 687
8.657.2 Field Documentation . . . 687
8.657.2.1 fileIndex . . . 687

8.657.2.2 pIndicationToken . . . 687
8.657.2.3 sessionInfo . . . 687
8.658 pack_uim_SLQSUIMPowerDown_t Struct Reference . . . 687
8.658.1 Detailed Description . . . 687
8.658.2 Field Documentation . . . 688
8.658.2.1 slot . . . 688
8.659 pack_uim_SLQSUIMPowerUp_t Struct Reference . . . 688
8.659.1 Detailed Description . . . 688
8.659.2 Field Documentation . . . 688
8.659.2.1 pIgnoreHotSwapSwitch . . . 688
8.659.2.2 slot . . . 688
8.660 pack_uim_SLQSUIMRefreshComplete_t Struct Reference . . . 688
8.660.1 Detailed Description . . . 688
8.660.2 Field Documentation . . . 689
8.660.2.1 refreshComplete . . . 689
8.660.2.2 sessionInfo . . . 689
8.661 pack_uim_SLQSUIMRefreshGetLastEvent_t Struct Reference . . . 689
8.661.1 Detailed Description . . . 689
8.661.2 Field Documentation . . . 689
8.661.2.1 sessionInfo . . . 689
8.662 pack_uim_SLQSUIMRefreshOK_t Struct Reference . . . 689
8.662.1 Detailed Description . . . 689
8.662.2 Field Documentation . . . 690
8.662.2.1 OKtoRefresh . . . 690
8.662.2.2 sessionInfo . . . 690
8.663 pack_uim_SLQSUIMRefreshRegister_t Struct Reference . . . 690
8.663.1 Detailed Description . . . 690
8.663.2 Field Documentation . . . 690
8.663.2.1 regRefresh . . . 690
8.663.2.2 sessionInfo . . . 690
8.664 pack_uim_SLQSUIMSwitchSlot_t Struct Reference . . . 690
8.664.1 Detailed Description . . . 690
8.664.2 Field Documentation . . . 691
8.664.2.1 bLogicalSlot . . . 691
8.664.2.2 ulPhysicalSlot . . . 691
8.665 pack_uim_UnblockPin_t Struct Reference . . . 691
8.665.1 Detailed Description . . . 691
8.665.2 Field Documentation . . . 692
8.665.2.1 EncryptedPIN1 . . . 692
8.665.2.2 pIndicationToken . . . 692

8.665.2.3 pinProtection . . . 692
8.665.2.4 pKeyReferenceID . . . 692
8.665.2.5 sessionInfo . . . 692
8.665.2.6 Tlvresult . . . 692
8.666 pack_uim_VerifyPin_t Struct Reference . . . 692
8.666.1 Detailed Description . . . 692
8.666.2 Field Documentation . . . 693
8.666.2.1 pEncryptedPIN1 . . . 693
8.666.2.2 pIndicationToken . . . 693
8.666.2.3 pKeyReferenceID . . . 693
8.666.2.4 sessionInfo . . . 693
8.666.2.5 Tlvresult . . . 693
8.666.2.6 verifyPIN . . . 693
8.667 pack_voice_AnswerUSSD_t Struct Reference . . . 693
8.667.1 Detailed Description . . . 693
8.667.2 Field Documentation . . . 694
8.667.2.1 pInfo . . . 694
8.668 pack_voice_OriginateUSSD_t Struct Reference . . . 694
8.668.1 Detailed Description . . . 694
8.668.2 Field Documentation . . . 694
8.668.2.1 pInfo . . . 694
8.669 pack_voice_SLQSOriginateUSSD_t Struct Reference . . . 694
8.669.1 Detailed Description . . . 694
8.669.2 Field Documentation . . . 695
8.669.2.1 ussData . . . 695
8.669.2.2 ussDCS . . . 695
8.669.2.3 ussLen . . . 695
8.670 pack_voice_SLQSVoiceALSSelectLine_t Struct Reference . . . 695
8.670.1 Detailed Description . . . 695
8.670.2 Field Documentation . . . 695
8.670.2.1 lineValue . . . 695
8.671 pack_voice_SLQSVoiceALSSetLineSwitching_t Struct Reference . . . 695
8.671.1 Detailed Description . . . 695
8.671.2 Field Documentation . . . 695
8.671.2.1 switchOption . . . 695
8.672 pack_voice_SLQSVoiceAnswerCall_t Struct Reference . . . 696
8.672.1 Detailed Description . . . 696
8.672.2 Field Documentation . . . 696
8.672.2.1 pCallId . . . 696
8.673 pack_voice_SLQSVoiceBindSubscription_t Struct Reference . . . 696

8.673.1 Detailed Description . . . 696
8.673.2 Field Documentation . . . 696
8.673.2.1 subsType . . . 696
8.674 pack_voice_SLQSVoiceBurstDTMF_t Struct Reference . . . 696
8.674.1 Detailed Description . . . 697
8.674.2 Field Documentation . . . 697
8.674.2.1 BurstDTMFInfo . . . 697
8.674.2.2 pBurstDTMFLengths . . . 697
8.675 pack_voice_SLQSVoiceDialCall_t Struct Reference . . . 697
8.675.1 Detailed Description . . . 697
8.675.2 Field Documentation . . . 698
8.675.2.1 callNumber . . . 698
8.675.2.2 pCallPartySubAdd . . . 698
8.675.2.3 pCallType . . . 698
8.675.2.4 pCLIRType . . . 698
8.675.2.5 pCUGInfo . . . 698
8.675.2.6 pEmergencyCategory . . . 699
8.675.2.7 pSvcType . . . 699
8.675.2.8 pUUSInFo . . . 699
8.676 pack_voice_SLQSVoiceEndCall_t Struct Reference . . . 699
8.676.1 Detailed Description . . . 699
8.676.2 Field Documentation . . . 699
8.676.2.1 pCallId . . . 699
8.677 pack_voice_SLQSVoiceGetCallBarring_t Struct Reference . . . 699
8.677.1 Detailed Description . . . 699
8.677.2 Field Documentation . . . 700
8.677.2.1 pSvcClass . . . 700
8.677.2.2 reason . . . 700
8.678 pack_voice_SLQSVoiceGetCallForwardingStatus_t Struct Reference . . . 700
8.678.1 Detailed Description . . . 700
8.678.2 Field Documentation . . . 701
8.678.2.1 pSvcClass . . . 701
8.678.2.2 Reason . . . 701
8.679 pack_voice_SLQSVoiceGetCallInfo_t Struct Reference . . . 701
8.679.1 Detailed Description . . . 701
8.679.2 Field Documentation . . . 701
8.679.2.1 callID . . . 701
8.680 pack_voice_SLQSVoiceGetCallWaiting_t Struct Reference . . . 701
8.680.1 Detailed Description . . . 702
8.680.2 Field Documentation . . . 702

8.680.2.1 pSvcClass . . . 702
8.681 pack_voice_SLQSVoiceGetConfig_t Struct Reference . . . 702
8.681.1 Detailed Description . . . 702
8.681.2 Field Documentation . . . 703
8.681.2.1 pAirTimer . . . 703
8.681.2.2 pAMRStatus . . . 703
8.681.2.3 pAutoAnswer . . . 703
8.681.2.4 pNamID . . . 703
8.681.2.5 pPrefVoicePrivacy . . . 703
8.681.2.6 pPrefVoiceSO . . . 703
8.681.2.7 pRoamTimer . . . 703
8.681.2.8 pTTYMode . . . 703
8.681.2.9 pVoiceDomainPref . . . 703
8.682 pack_voice_SLQSVoiceIndicationRegister_t Struct Reference . . . 704
8.682.1 Detailed Description . . . 704
8.682.2 Field Documentation . . . 704
8.682.2.1 pRegDTMFEvents . . . 704
8.682.2.2 pRegVoicePrivacyEvents . . . 704
8.682.2.3 pSuppsNotifEvents . . . 704
8.683 pack_voice_SLQSVoiceManageCalls_t Struct Reference . . . 704
8.683.1 Detailed Description . . . 705
8.683.2 Field Documentation . . . 705
8.683.2.1 pCallID . . . 705
8.683.2.2 SUPSType . . . 705
8.684 pack_voice_SLQSVoiceOrigUSSDNoWait_t Struct Reference . . . 705
8.684.1 Detailed Description . . . 705
8.684.2 Field Documentation . . . 706
8.684.2.1 USSInformation . . . 706
8.685 pack_voice_SLQSVoiceSendFlash_t Struct Reference . . . 706
8.685.1 Detailed Description . . . 706
8.685.2 Field Documentation . . . 706
8.685.2.1 pCallID . . . 706
8.685.2.2 pFlashPayLd . . . 706
8.685.2.3 pFlashType . . . 706
8.686 pack_voice_SLQSVoiceSetCallBarringPassword_t Struct Reference . . . 706
8.686.1 Detailed Description . . . 707
8.686.2 Field Documentation . . . 707
8.686.2.1 newPasswd . . . 707
8.686.2.2 newPasswdAgain . . . 707
8.686.2.3 oldPasswd . . . 707

8.686.2.4 Reason . . . 707
8.687 pack_voice_SLQSVoiceSetConfig_t Struct Reference . . . 707
8.687.1 Detailed Description . . . 708
8.687.2 Field Documentation . . . 709
8.687.2.1 pAirTimerConfig . . . 709
8.687.2.2 pAutoAnswer . . . 709
8.687.2.3 pPrefVoiceDomain . . . 709
8.687.2.4 pPrefVoiceSO . . . 709
8.687.2.5 pRoamTimerConfig . . . 709
8.687.2.6 pTTYMode . . . 709
8.688 pack_voice_SLQSVoiceSetPreferredPrivacy_t Struct Reference . . . 709
8.688.1 Detailed Description . . . 709
8.688.2 Field Documentation . . . 709
8.688.2.1 privacyPref . . . 709
8.689 pack_voice_SLQSVoiceSetSUPSService_t Struct Reference . . . 709
8.689.1 Detailed Description . . . 709
8.689.2 Field Documentation . . . 711
8.689.2.1 pCallBarringPasswd . . . 711
8.689.2.2 pCallForwardingNumber . . . 711
8.689.2.3 pCallFwdTypeAndPlan . . . 711
8.689.2.4 pServiceClass . . . 711
8.689.2.5 pTimerVal . . . 711
8.689.2.6 reason . . . 711
8.689.2.7 voiceSvc . . . 711
8.690 pack_voice_SLQSVoiceStartContDTMF_t Struct Reference . . . 711
8.690.1 Detailed Description . . . 711
8.690.2 Field Documentation . . . 712
8.690.2.1 DTMFdigit . . . 712
8.690.2.2 pCallID . . . 712
8.691 pack_voice_SLQSVoiceStopContDTMF_t Struct Reference . . . 712
8.691.1 Detailed Description . . . 712
8.691.2 Field Documentation . . . 712
8.691.2.1 callID . . . 712
8.692 pack_wds_DHCPv4ClientLeaseChange_t Struct Reference . . . 712
8.692.1 Detailed Description . . . 712
8.692.2 Field Documentation . . . 713
8.692.2.1 pEnableNotification . . . 713
8.693 pack_wds_GetDefaultProfile_t Struct Reference . . . 713
8.693.1 Detailed Description . . . 713
8.693.2 Field Documentation . . . 713

8.693.2.1 profiletype . . . 713
8.694 pack_wds_GetDefaultProfileNum_t Struct Reference . . . 713
8.694.1 Detailed Description . . . 713
8.694.2 Field Documentation . . . 713
8.694.2.1 family . . . 713
8.694.2.2 type . . . 714
8.695 pack_wds_GetDormancyState_t Struct Reference . . . 714
8.695.1 Detailed Description . . . 714
8.696 pack_wds_GetLastMobileIPError_t Struct Reference . . . 714
8.696.1 Detailed Description . . . 714
8.697 pack_wds_GetMobileIP_t Struct Reference . . . 714
8.697.1 Detailed Description . . . 714
8.698 pack_wds_GetMobileIPProfile_t Struct Reference . . . 714
8.698.1 Detailed Description . . . 714
8.698.2 Field Documentation . . . 715
8.698.2.1 index . . . 715
8.699 pack_wds_GetPacketStatistics_t Struct Reference . . . 715
8.699.1 Detailed Description . . . 715
8.699.2 Field Documentation . . . 715
8.699.2.1 pStatMask . . . 715
8.700 pack_wds_GetPacketStatus_t Struct Reference . . . 715
8.700.1 Detailed Description . . . 715
8.700.2 Field Documentation . . . 716
8.700.2.1 statmask . . . 716
8.701 pack_wds_GetSessionDuration_t Struct Reference . . . 716
8.701.1 Detailed Description . . . 716
8.702 pack_wds_RMSetTransferStatistics_t Struct Reference . . . 716
8.702.1 Detailed Description . . . 716
8.702.2 Field Documentation . . . 716
8.702.2.1 RmTrasnferStaticsReq . . . 716
8.703 pack_wds_SetAutoconnect_t Struct Reference . . . 717
8.703.1 Detailed Description . . . 717
8.703.2 Field Documentation . . . 717
8.703.2.1 acroamsetting . . . 717
8.703.2.2 acsetting . . . 717
8.704 pack_wds_SetDefaultProfile_t Struct Reference . . . 717
8.704.1 Detailed Description . . . 717
8.704.2 Field Documentation . . . 718
8.704.2.1 authentication . . . 719
8.704.2.2 ipAddress . . . 719

8.704.2.3 pApnname . . . 719
8.704.2.4 pdpType . . . 719
8.704.2.5 pName . . . 719
8.704.2.6 pPassword . . . 719
8.704.2.7 primaryDNS . . . 719
8.704.2.8 profileType . . . 719
8.704.2.9 pUsername . . . 719
8.704.2.10 secondaryDNS . . . 719
8.705 pack_wds_SetDefaultProfileNum_t Struct Reference . . . 719
8.705.1 Detailed Description . . . 719
8.705.2 Field Documentation . . . 719
8.705.2.1 family . . . 719
8.705.2.2 index . . . 719
8.705.2.3 type . . . 719
8.706 pack_wds_SetMobileIP_t Struct Reference . . . 720
8.706.1 Detailed Description . . . 720
8.706.2 Field Documentation . . . 720
8.706.2.1 mode . . . 720
8.707 pack_wds_SetMobileIPParameters_t Struct Reference . . . 720
8.707.1 Detailed Description . . . 720
8.707.2 Field Documentation . . . 721
8.707.2.1 pHA2002bis . . . 721
8.707.2.2 pHAAuthenticator . . . 721
8.707.2.3 pMode . . . 721
8.707.2.4 pReRegPeriod . . . 721
8.707.2.5 pReRegTraffic . . . 721
8.707.2.6 pRetryInterval . . . 721
8.707.2.7 pRetryLimit . . . 721
8.707.2.8 pSPC . . . 721
8.708 pack_wds_SetMobileIPProfile_t Struct Reference . . . 721
8.708.1 Detailed Description . . . 722
8.708.2 Field Documentation . . . 722
8.708.2.1 index . . . 723
8.708.2.2 pAAASPI . . . 723
8.708.2.3 pAddress . . . 723
8.708.2.4 pEnabled . . . 723
8.708.2.5 pHASPI . . . 723
8.708.2.6 pMNAAA . . . 723
8.708.2.7 pMNHA . . . 723
8.708.2.8 pNAI . . . 723

8.708.2.9 pPrimaryHA . . . 723
8.708.2.10 pRevTunneling . . . 723
8.708.2.11 pSecondaryHA . . . 723
8.708.2.12 spc . . . 723
8.709 pack_wds_SLQSCreateProfile_t Struct Reference . . . 723
8.709.1 Detailed Description . . . 723
8.709.2 Field Documentation . . . 724
8.709.2.1 pCurProfile . . . 724
8.709.2.2 pProfileId . . . 724
8.709.2.3 pProfileType . . . 724
8.710 pack_wds_SLQSDeleteProfile_t Struct Reference . . . 724
8.710.1 Detailed Description . . . 724
8.710.2 Field Documentation . . . 724
8.710.2.1 profileIndex . . . 724
8.710.2.2 profileType . . . 724
8.711 pack_wds_SLQSGetCurrDataSystemStat_t Struct Reference . . . 724
8.711.1 Detailed Description . . . 724
8.712 pack_wds_SLQSGetDataBearerTechnology_t Struct Reference . . . 725
8.712.1 Detailed Description . . . 725
8.713 pack_wds_SLQSGetDUNCallInfo_t Struct Reference . . . 725
8.713.1 Detailed Description . . . 725
8.713.2 Field Documentation . . . 726
8.713.2.1 Mask . . . 726
8.713.2.2 pReportChannelRate . . . 726
8.713.2.3 pReportConnStatus . . . 726
8.713.2.4 pReportDataBearerTech . . . 726
8.713.2.5 pReportDormStatus . . . 726
8.713.2.6 pTransferStatInd . . . 726
8.714 pack_wds_SLQSGetProfileSettings_t Struct Reference . . . 726
8.714.1 Detailed Description . . . 726
8.714.2 Field Documentation . . . 727
8.714.2.1 ProfileId . . . 727
8.714.2.2 ProfileType . . . 727
8.715 pack_wds_SLQSGetRuntimeSettings_t Struct Reference . . . 727
8.715.1 Detailed Description . . . 727
8.715.2 Field Documentation . . . 728
8.715.2.1 pReqSettings . . . 728
8.716 pack_wds_SLQSModifyProfile_t Struct Reference . . . 728
8.716.1 Detailed Description . . . 728
8.716.2 Field Documentation . . . 729

8.716.2.1 curProfile . . . 729
8.716.2.2 pProfileId . . . 729
8.716.2.3 pProfileType . . . 729
8.717 pack_wds_SLQSSet3GPPConfigItem_t Struct Reference . . . 729
8.717.1 Detailed Description . . . 729
8.717.2 Field Documentation . . . 730
8.717.2.1 LTEAttachProfileListLen . . . 730
8.717.2.2 p3gppRelease . . . 730
8.717.2.3 pDefaultPDNEnabled . . . 730
8.717.2.4 pLTEAttachProfile . . . 730
8.717.2.5 pLTEAttachProfileList . . . 730
8.717.2.6 pProfileList . . . 731
8.718 pack_wds_SLQSSetIPFamilyPreference_t Struct Reference . . . 731
8.718.1 Detailed Description . . . 731
8.718.2 Field Documentation . . . 731
8.718.2.1 IPFamilyPreference . . . 731
8.719 pack_wds_SLQSSetWdsEventCallback_t Struct Reference . . . 731
8.719.1 Detailed Description . . . 731
8.719.2 Field Documentation . . . 732
8.719.2.1 currentDataBearer . . . 732
8.719.2.2 dataBearer . . . 732
8.719.2.3 dataBearerTechExt . . . 732
8.719.2.4 dataSystemStatus . . . 732
8.719.2.5 dormancyStatus . . . 732
8.719.2.6 interval . . . 732
8.719.2.7 mobileIP . . . 732
8.719.2.8 transferStats . . . 732
8.720 pack_wds_SLQSSGetDHCPv4ClientConfig_t Struct Reference . . . 732
8.720.1 Detailed Description . . . 732
8.720.2 Field Documentation . . . 732
8.720.2.1 pProfileId . . . 732
8.721 pack_wds_SLQSSSetDHCPv4ClientConfig_t Struct Reference . . . 732
8.721.1 Detailed Description . . . 732
8.721.2 Field Documentation . . . 733
8.721.2.1 pHwConfig . . . 733
8.721.2.2 pProfileId . . . 733
8.721.2.3 pRequestOptionList . . . 733
8.722 pack_wds_SLQSSSetLoopback_t Struct Reference . . . 733
8.722.1 Detailed Description . . . 733
8.722.2 Field Documentation . . . 733

8.722.2.1 loopbackMode . . . 733
8.722.2.2 loopbackMultiplier . . . 733
8.723 pack_wds_SLQSStartDataSession_t Struct Reference . . . 733
8.723.1 Detailed Description . . . 734
8.723.2 Field Documentation . . . 734
8.723.2.1 pAuth . . . 734
8.723.2.2 pPass . . . 734
8.723.2.3 pprofileid3gpp . . . 734
8.723.2.4 pprofileid3gpp2 . . . 734
8.723.2.5 pTech . . . 734
8.723.2.6 pUser . . . 735
8.724 pack_wds_SLQSStopDataSession_t Struct Reference . . . 735
8.724.1 Detailed Description . . . 735
8.724.2 Field Documentation . . . 735
8.724.2.1 psid . . . 735
8.725 pack_wds_SLQSWdsSetEventReport_t Struct Reference . . . 735
8.725.1 Detailed Description . . . 735
8.725.2 Field Documentation . . . 736
8.725.2.1 pCurrChannelRateInd . . . 736
8.725.2.2 pCurrDataBearerTechInd . . . 736
8.725.2.3 pCurrPrefDataSysInd . . . 736
8.725.2.4 pDataBearerTechInd . . . 736
8.725.2.5 pDataCallStatusChangeInd . . . 736
8.725.2.6 pDataSystemStatusChangeInd . . . 737
8.725.2.7 pDormancyStatusInd . . . 737
8.725.2.8 pEVDOPageMonPerChangeInd . . . 737
8.725.2.9 pMIPStatusInd . . . 737
8.725.2.10 pTransferStatInd . . . 737
8.726 pack_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference . . . 737
8.726.1 Detailed Description . . . 737
8.726.2 Field Documentation . . . 737
8.726.2.1 contextId . . . 737
8.726.2.2 contextType . . . 737
8.727 PackCreateProfileOut Struct Reference . . . 737
8.727.1 Detailed Description . . . 737
8.727.2 Field Documentation . . . 738
8.727.2.1 ExtErrorCode . . . 738
8.727.2.2 ProfileIndex . . . 738
8.727.2.3 ProfileType . . . 738
8.728 packgetDyingGaspCfg Struct Reference . . . 738

8.728.1 Detailed Description . . . 738
8.728.2 Field Documentation . . . 738
8.728.2.1 pDestSMSContent . . . 738
8.728.2.2 pDestSMSNum . . . 738
8.729 packgetDyingGaspStatistics Struct Reference . . . 738
8.729.1 Detailed Description . . . 739
8.729.2 Field Documentation . . . 739
8.729.2.1 pSMSAttemptedFlag . . . 739
8.729.2.2 pTimeStamp . . . 739
8.730 PCMparams Struct Reference . . . 739
8.730.1 Detailed Description . . . 739
8.730.2 Field Documentation . . . 739
8.730.2.1 iFaceTab . . . 739
8.730.2.2 iFaceTabLen . . . 739
8.731 PCSCFFQDNAddress Struct Reference . . . 740
8.731.1 Detailed Description . . . 740
8.731.2 Field Documentation . . . 740
8.731.2.1 fqdnAddr . . . 740
8.731.2.2 fqdnLen . . . 740
8.732 PCSCFFQDNAddressList Struct Reference . . . 740
8.732.1 Detailed Description . . . 740
8.732.2 Field Documentation . . . 740
8.732.2.1 numInstances . . . 740
8.732.2.2 pcsfFQDNAddress . . . 740
8.733 PCSCFIPv4ServerAddressList Struct Reference . . . 741
8.733.1 Detailed Description . . . 741
8.733.2 Field Documentation . . . 741
8.733.2.1 numInstances . . . 741
8.733.2.2 pscsfIPv4Addr . . . 741
8.734 PDSPositionData Struct Reference . . . 741
8.734.1 Detailed Description . . . 741
8.734.2 Field Documentation . . . 743
8.734.2.1 pAltitudeWrtEllipsoid . . . 743
8.734.2.2 pAltitudeWrtSealevel . . . 743
8.734.2.3 pHorizontalConfidence . . . 743
8.734.2.4 pHorizontalUncCircular . . . 743
8.734.2.5 pLatitude . . . 743
8.734.2.6 pLongitude . . . 743
8.734.2.7 pPositionSource . . . 743
8.734.2.8 pTimeStamp . . . 743

8.734.2.9 pTimeType . . . 743
8.734.2.10 pVerticalConfidence . . . 743
8.734.2.11 pVerticalUnc . . . 743
8.735 PDSPosMethodStateReq Struct Reference . . . 743
8.735.1 Detailed Description . . . 743
8.735.2 Field Documentation . . . 744
8.735.2.1 pWifiState . . . 744
8.735.2.2 pXtraDataState . . . 744
8.735.2.3 pXtraTimeState . . . 744
8.736 peerNumberInfo Struct Reference . . . 744
8.736.1 Detailed Description . . . 744
8.736.2 Field Documentation . . . 745
8.736.2.1 callID . . . 745
8.736.2.2 number . . . 745
8.736.2.3 numLen . . . 745
8.736.2.4 numPI . . . 745
8.736.2.5 numPlan . . . 745
8.736.2.6 numSI . . . 746
8.736.2.7 numType . . . 746
8.737 personalizationStatus Struct Reference . . . 746
8.737.1 Detailed Description . . . 746
8.737.2 Field Documentation . . . 746
8.737.2.1 feature . . . 746
8.737.2.2 numFeatures . . . 746
8.737.2.3 unblockLeft . . . 746
8.737.2.4 verifyLeft . . . 746
8.738 PhyCaAggPcellInfo Struct Reference . . . 747
8.738.1 Detailed Description . . . 747
8.738.2 Field Documentation . . . 747
8.738.2.1 dl_bw_value . . . 747
8.738.2.2 freq . . . 747
8.738.2.3 iLTEbandValue . . . 747
8.738.2.4 pci . . . 747
8.738.2.5 TlvPresent . . . 747
8.739 PhyCaAggScellDlBw Struct Reference . . . 747
8.739.1 Detailed Description . . . 748
8.739.2 Field Documentation . . . 748
8.739.2.1 dl_bw_value . . . 748
8.739.2.2 TlvPresent . . . 748
8.740 PhyCaAggScellIndex Struct Reference . . . 748

8.740.1 Detailed Description . . . 748
8.740.2 Field Documentation . . . 748
8.740.2.1 scell_idx . . . 748
8.740.2.2 TlvPresent . . . 748
8.741 PhyCaAggScellIndType Struct Reference . . . 748
8.741.1 Detailed Description . . . 749
8.741.2 Field Documentation . . . 749
8.741.2.1 freq . . . 749
8.741.2.2 pci . . . 749
8.741.2.3 scell_state . . . 749
8.741.2.4 TlvPresent . . . 749
8.742 PhyCaAggScellInfo Struct Reference . . . 749
8.742.1 Detailed Description . . . 749
8.742.2 Field Documentation . . . 752
8.742.2.1 dl_bw_value . . . 752
8.742.2.2 freq . . . 752
8.742.2.3 iLTEbandValue . . . 752
8.742.2.4 pci . . . 752
8.742.2.5 scell_state . . . 752
8.742.2.6 TlvPresent . . . 752
8.743 PilotSetData Struct Reference . . . 752
8.743.1 Detailed Description . . . 752
8.743.2 Field Documentation . . . 752
8.743.2.1 NumPilots . . . 752
8.743.2.2 pPilotSetInfo . . . 752
8.744 PilotSetParams Struct Reference . . . 753
8.744.1 Detailed Description . . . 753
8.744.2 Field Documentation . . . 753
8.744.2.1 PilotPN . . . 753
8.744.2.2 PilotStrength . . . 753
8.744.2.3 PilotType . . . 753
8.745 pktErrRate Struct Reference . . . 753
8.745.1 Detailed Description . . . 753
8.745.2 Field Documentation . . . 753
8.745.2.1 exponent . . . 753
8.745.2.2 multiplier . . . 754
8.746 PlmnID Struct Reference . . . 754
8.746.1 Detailed Description . . . 754
8.746.2 Field Documentation . . . 754
8.746.2.1 mcc . . . 754

8.746.2.2 mnc . . . 754
8.746.2.3 pcsDigit . . . 754
8.746.2.4 TlvPresent . . . 754
8.747 PLMNNetworkName Struct Reference . . . 754
8.747.1 Detailed Description . . . 755
8.747.2 Field Documentation . . . 755
8.747.2.1 numInstance . . . 755
8.747.2.2 PLMNNetName . . . 755
8.748 PLMNNetworkNameData Struct Reference . . . 755
8.748.1 Detailed Description . . . 755
8.748.2 Field Documentation . . . 756
8.748.2.1 codingScheme . . . 756
8.748.2.2 countryInitials . . . 756
8.748.2.3 longName . . . 756
8.748.2.4 longNameLen . . . 756
8.748.2.5 longNameSpareBits . . . 756
8.748.2.6 shortName . . . 756
8.748.2.7 shortNameLen . . . 757
8.748.2.8 shortNameSpareBits . . . 757
8.749 Port Struct Reference . . . 757
8.749.1 Detailed Description . . . 757
8.749.2 Field Documentation . . . 757
8.749.2.1 port . . . 757
8.749.2.2 range . . . 757
8.750 precisionDilution_s Struct Reference . . . 757
8.750.1 Detailed Description . . . 757
8.750.2 Field Documentation . . . 758
8.750.2.1 HDOP . . . 758
8.750.2.2 PDOP . . . 758
8.750.2.3 VDOP . . . 758
8.751 PrefImageList Struct Reference . . . 758
8.751.1 Detailed Description . . . 758
8.751.2 Field Documentation . . . 758
8.751.2.1 listEntries . . . 758
8.751.2.2 listSize . . . 758
8.752 prefVoiceSO Struct Reference . . . 758
8.752.1 Detailed Description . . . 759
8.752.2 Field Documentation . . . 760
8.752.2.1 evrcCapability . . . 760
8.752.2.2 homeOrigVoiceSO . . . 760

8.752.2.3 homePageVoiceSO . . . 760
8.752.2.4 namID . . . 760
8.752.2.5 roamOrigVoiceSO . . . 760
8.753 Profile3GPP Struct Reference . . . 760
8.753.1 Detailed Description . . . 761
8.753.2 Field Documentation . . . 765
8.753.2.1 pAddrAllocPref . . . 765
8.753.2.2 pAPNClass . . . 765
8.753.2.3 pAPNDisabledFlag . . . 765
8.753.2.4 pAPNName . . . 765
8.753.2.5 pAPNnameSize . . . 765
8.753.2.6 pAuthenticationPref . . . 765
8.753.2.7 pGPRSMinimumQoS . . . 765
8.753.2.8 pGPRSRequestedQos . . . 765
8.753.2.9 pImCnFlag . . . 765
8.753.2.10 pIPv4AddrPref . . . 765
8.753.2.11 pIPv6AddPref . . . 765
8.753.2.12 pPassword . . . 765
8.753.2.13 pPasswordSize . . . 765
8.753.2.14 pPcscfAddrUsingDhcp . . . 765
8.753.2.15 pPcscfAddrUsingPCO . . . 765
8.753.2.16 pPDNInactivTimeout . . . 765
8.753.2.17 pPdpAccessConFlag . . . 765
8.753.2.18 pPdpContext . . . 765
8.753.2.19 pPdpDataCompType . . . 765
8.753.2.20 pPdpHdrCompType . . . 765
8.753.2.21 pPDPtype . . . 765
8.753.2.22 pPriDNSIPv4AddPref . . . 765
8.753.2.23 pPriDNSIPv6addpref . . . 765
8.753.2.24 pPrimaryID . . . 766
8.753.2.25 pProfilename . . . 766
8.753.2.26 pProfilenameSize . . . 766
8.753.2.27 pQosClassID . . . 766
8.753.2.28 pSecDNSIPv4AddPref . . . 766
8.753.2.29 pSecDNSIPv6addpref . . . 766
8.753.2.30 pSecondaryFlag . . . 766
8.753.2.31 pTFTID1Params . . . 766
8.753.2.32 pTFTID2Params . . . 766
8.753.2.33 pUMTSMinQoS . . . 766
8.753.2.34 pUMTSMinQosSigInd . . . 766

8.753.2.35 pUMTSReqQoS . . . 766
8.753.2.36 pUMTSReqQoSSigInd . . . 766
8.753.2.37 pUsername . . . 766
8.753.2.38 pUsernameSize . . . 766
8.754 Profile3GPP2 Struct Reference . . . 766
8.754.1 Detailed Description . . . 767
8.754.2 Field Documentation . . . 770
8.754.2.1 pAllowLinger . . . 770
8.754.2.2 pAPNClass3GPP2 . . . 770
8.754.2.3 pAPNEnabled3GPP2 . . . 770
8.754.2.4 pApnString . . . 770
8.754.2.5 pApnStringSize . . . 770
8.754.2.6 pAppPriority . . . 770
8.754.2.7 pAppType . . . 770
8.754.2.8 pAuthPassword . . . 770
8.754.2.9 pAuthPasswordSize . . . 770
8.754.2.10 pAuthProtocol . . . 771
8.754.2.11 pAuthRetryCount . . . 771
8.754.2.12 pAuthTimeout . . . 771
8.754.2.13 pDataMode . . . 771
8.754.2.14 pDataRate . . . 771
8.754.2.15 pIpcpAckTimeout . . . 771
8.754.2.16 pIpcpCreqRetryCount . . . 771
8.754.2.17 pIsPcscfAddressNedded . . . 771
8.754.2.18 pLcpAckTimeout . . . 771
8.754.2.19 pLcpCreqRetryCount . . . 771
8.754.2.20 pNegoDnsSrvrPref . . . 771
8.754.2.21 pPDNInactivTimeout3GPP2 . . . 771
8.754.2.22 pPdnType . . . 771
8.754.2.23 pPppSessCloseTimer1x . . . 771
8.754.2.24 pPppSessCloseTimerDO . . . 771
8.754.2.25 pPrimaryV4DnsAddress . . . 771
8.754.2.26 pPriV6DnsAddress . . . 771
8.754.2.27 pRATType . . . 771
8.754.2.28 pSecondaryV4DnsAddress . . . 771
8.754.2.29 pSecV6DnsAddress . . . 771
8.754.2.30 pUserId . . . 771
8.754.2.31 pUserIdSize . . . 771
8.755 ProfileIdentifier Struct Reference . . . 771
8.755.1 Detailed Description . . . 771

8.755.2 Field Documentation . . . 772
8.755.2.1 profileIndex . . . 772
8.755.2.2 profileType . . . 772
8.756 protocolSubtypeElement Struct Reference . . . 772
8.756.1 Detailed Description . . . 772
8.756.2 Field Documentation . . . 773
8.756.2.1 AccessMac . . . 773
8.756.2.2 AuthProt . . . 773
8.756.2.3 ControlMac . . . 773
8.756.2.4 EncryptProt . . . 773
8.756.2.5 ForwardMac . . . 773
8.756.2.6 IdleState . . . 773
8.756.2.7 KeyExchange . . . 773
8.756.2.8 MultDisc . . . 773
8.756.2.9 PhysicalLayer . . . 773
8.756.2.10 ReverseMac . . . 773
8.756.2.11 SecProt . . . 773
8.756.2.12 VirtStream . . . 773
8.757 PSDetachReq Struct Reference . . . 773
8.757.1 Detailed Description . . . 774
8.757.2 Field Documentation . . . 774
8.757.2.1 pDetachAction . . . 774
8.758 qaQmi3Gpp2TimeZone Struct Reference . . . 774
8.758.1 Detailed Description . . . 774
8.758.2 Field Documentation . . . 774
8.758.2.1 daylightSavings . . . 774
8.758.2.2 leapSeconds . . . 774
8.758.2.3 localTimeOffset . . . 775
8.759 qaQmiInterfaceInfo Struct Reference . . . 775
8.759.1 Detailed Description . . . 775
8.759.2 Field Documentation . . . 775
8.759.2.1 qaQmiinstanceid . . . 775
8.759.2.2 qaQmisvctype . . . 775
8.759.2.3 v4sessionId . . . 775
8.759.2.4 v6sessionId . . . 775
8.760 qaQmiServingSystemParam Struct Reference . . . 775
8.760.1 Detailed Description . . . 776
8.760.2 Field Documentation . . . 778
8.760.2.1 BasestationID . . . 778
8.760.2.2 BasestationLatitude . . . 778

8.760.2.3 BasestationLongitude . . . 778
8.760.2.4 CallBarStatus . . . 778
8.760.2.5 CDMA_P_Rev . . . 778
8.760.2.6 CDMASystemInfoExt . . . 778
8.760.2.7 CellID . . . 778
8.760.2.8 concSvcInfo . . . 778
8.760.2.9 CurrentPLMN . . . 779
8.760.2.10 DataSrvCapabilities . . . 779
8.760.2.11 defaultRoamInd . . . 779
8.760.2.12 DetailedSvcInfo . . . 779
8.760.2.13 DTMInd . . . 779
8.760.2.14 Gpp2TimeZone . . . 779
8.760.2.15 GppNetworkDSTAdjustment . . . 779
8.760.2.16 GppTimeZone . . . 779
8.760.2.17 hdrPersonality . . . 779
8.760.2.18 Lac . . . 779
8.760.2.19 NetworkID . . . 779
8.760.2.20 PRLInd . . . 779
8.760.2.21 roamIndicatorVal . . . 779
8.760.2.22 RoamingIndicatorList . . . 779
8.760.2.23 ServingSystem . . . 779
8.760.2.24 SystemID . . . 779
8.760.2.25 trackAreaCode . . . 779
8.761 QmiCbkCatEventStatusReportInd Struct Reference . . . 779
8.761.1 Field Documentation . . . 779
8.761.1.1 CCETlv . . . 779
8.761.1.2 event_Index . . . 779
8.762 QmiCbkLocBestAvailPosInd Struct Reference . . . 779
8.762.1 Detailed Description . . . 780
8.762.2 Field Documentation . . . 784
8.762.2.1 pAltitudeWrtEllipsoid . . . 784
8.762.2.2 pAltitudeWrtMeanSeaLevel . . . 784
8.762.2.3 pGpsTime . . . 784
8.762.2.4 pHeading . . . 784
8.762.2.5 pHeadingUnc . . . 784
8.762.2.6 pHorCirConf . . . 784
8.762.2.7 pHorEllpConf . . . 784
8.762.2.8 pHorReliability . . . 784
8.762.2.9 pHorUncCircular . . . 784
8.762.2.10 pHorUncEllipseOrientAzimuth . . . 784

8.762.2.11 pHorUncEllipseSemiMajor . . . 784
8.762.2.12 pHorUncEllipseSemiMinor . . . 785
8.762.2.13 pLatitude . . . 785
8.762.2.14 pLongitude . . . 785
8.762.2.15 pMagneticDeviation . . . 785
8.762.2.16 pPrecisionDilution . . . 785
8.762.2.17 pSensorDataUsage . . . 785
8.762.2.18 pSpeedHorizontal . . . 785
8.762.2.19 pSpeedUnc . . . 785
8.762.2.20 pSpeedVertical . . . 785
8.762.2.21 pSpeedVerticalUnc . . . 785
8.762.2.22 pSvUsedforFix . . . 785
8.762.2.23 pTechnologyMask . . . 785
8.762.2.24 pTimeSrc . . . 785
8.762.2.25 pTimestampUtc . . . 785
8.762.2.26 pTimeUnc . . . 785
8.762.2.27 pVertConfidence . . . 785
8.762.2.28 pVertReliability . . . 785
8.762.2.29 pVertUnc . . . 785
8.762.2.30 pXid . . . 785
8.762.2.31 status . . . 785
8.763 QmiCbkLocCradleMountInd Struct Reference . . . 785
8.763.1 Detailed Description . . . 785
8.763.2 Field Documentation . . . 786
8.763.2.1 cradleMountConfigStatus . . . 786
8.764 QmiCbkLocEngineStateInd Struct Reference . . . 786
8.764.1 Detailed Description . . . 786
8.764.2 Field Documentation . . . 786
8.764.2.1 engineState . . . 786
8.765 QmiCbkLocEventTimeSyncInd Struct Reference . . . 786
8.765.1 Detailed Description . . . 787
8.765.2 Field Documentation . . . 787
8.765.2.1 timeSyncRefCounter . . . 787
8.766 QmiCbkLocGetServerInd Struct Reference . . . 787
8.766.1 Detailed Description . . . 787
8.766.2 Field Documentation . . . 788
8.766.2.1 pIPv4AddrInfo . . . 788
8.766.2.2 pIPv6AddrInfo . . . 788
8.766.2.3 pURL . . . 788
8.766.2.4 serverStatus . . . 788

8.766.2.5 serverType . . . 788
8.767 QmiCbkLocInjectPositionInd Struct Reference . . . 788
8.767.1 Detailed Description . . . 788
8.767.2 Field Documentation . . . 789
8.767.2.1 status . . . 789
8.768 QmiCbkLocInjectSensorDataInd Struct Reference . . . 789
8.768.1 Detailed Description . . . 789
8.768.2 Field Documentation . . . 790
8.768.2.1 injectSensorDataStatus . . . 790
8.768.2.2 pAccelSamplesAccepted . . . 790
8.768.2.3 pAccelTempSamplesAccepted . . . 790
8.768.2.4 pGyroSamplesAccepted . . . 790
8.768.2.5 pGyroTempSamplesAccepted . . . 790
8.768.2.6 pOpaqueIdentifier . . . 790
8.769 QmiCbkLocInjectTimeInd Struct Reference . . . 790
8.769.1 Detailed Description . . . 791
8.769.2 Field Documentation . . . 791
8.769.2.1 injectTimeSyncStatus . . . 791
8.770 QmiCbkLocInjectUTCTimeInd Struct Reference . . . 791
8.770.1 Detailed Description . . . 791
8.770.2 Field Documentation . . . 792
8.770.2.1 status . . . 792
8.771 QmiCbkLocPositionReportInd Struct Reference . . . 792
8.771.1 Detailed Description . . . 792
8.771.2 Field Documentation . . . 796
8.771.2.1 pAltitudeAssumed . . . 796
8.771.2.2 pAltitudeWrtEllipsoid . . . 796
8.771.2.3 pAltitudeWrtMeanSeaLevel . . . 796
8.771.2.4 pFixId . . . 796
8.771.2.5 pGpsTime . . . 796
8.771.2.6 pHeading . . . 797
8.771.2.7 pHeadingUnc . . . 797
8.771.2.8 pHorConfidence . . . 797
8.771.2.9 pHorReliability . . . 797
8.771.2.10 pHorUncCircular . . . 797
8.771.2.11 pHorUncEllipseOrientAzimuth . . . 797
8.771.2.12 pHorUncEllipseSemiMajor . . . 797
8.771.2.13 pHorUncEllipseSemiMinor . . . 797
8.771.2.14 pLatitude . . . 797
8.771.2.15 pLeapSeconds . . . 797

8.771.2.16 pLongitude . . . 797
8.771.2.17 pMagneticDeviation . . . 797
8.771.2.18 pPrecisionDilution . . . 797
8.771.2.19 pSensorDataUsage . . . 797
8.771.2.20 pSpeedHorizontal . . . 797
8.771.2.21 pSpeedUnc . . . 797
8.771.2.22 pSpeedVertical . . . 797
8.771.2.23 pSvUsedforFix . . . 797
8.771.2.24 pTechnologyMask . . . 797
8.771.2.25 pTimeSrc . . . 797
8.771.2.26 pTimestampUtc . . . 797
8.771.2.27 pTimeUnc . . . 797
8.771.2.28 pVertConfidence . . . 797
8.771.2.29 pVertReliability . . . 797
8.771.2.30 pVertUnc . . . 797
8.771.2.31 sessionId . . . 797
8.771.2.32 sessionStatus . . . 797
8.772 QmiCbkLocSensorStreamingInd Struct Reference . . . 797
8.772.1 Detailed Description . . . 798
8.772.2 Field Documentation . . . 798
8.772.2.1 pAccelAcceptReady . . . 798
8.772.2.2 pAccelTempAcceptReady . . . 798
8.772.2.3 pGyroAcceptReady . . . 798
8.772.2.4 pGyroTempAcceptReady . . . 798
8.773 QmiCbkLocSetExtPowerConfigInd Struct Reference . . . 798
8.773.1 Detailed Description . . . 798
8.773.2 Field Documentation . . . 799
8.773.2.1 status . . . 799
8.774 QmiCbkNasLTECphyCaInfo Struct Reference . . . 799
8.774.1 Detailed Description . . . 799
8.774.2 Field Documentation . . . 800
8.774.2.1 sPhyCaAggPcellInfo . . . 800
8.774.2.2 sPhyCaAggScellDlBw . . . 800
8.774.2.3 sPhyCaAggScellIndex . . . 800
8.774.2.4 sPhyCaAggScellIndType . . . 800
8.774.2.5 sPhyCaAggScellInfo . . . 800
8.775 QmiCbkSwiOmaDmEventStatusReportInd Struct Reference . . . 800
8.775.1 Field Documentation . . . 800
8.775.1.1 SITlv . . . 800
8.776 QmiCbkSwiOmaDmEventStatusReportIndExt Struct Reference . . . 800

8.776.1 Field Documentation . . . 800
8.776.1.1 SITlv . . . 800
8.777 QmiCbkTmdMitiLvlRptInd Struct Reference . . . 800
8.777.1 Detailed Description . . . 800
8.777.2 Field Documentation . . . 801
8.777.2.1 currentMitigationLvl . . . 801
8.777.2.2 MitigationDevInfo . . . 801
8.778 QmiCbkWdsStatisticsIndState Struct Reference . . . 801
8.778.1 Detailed Description . . . 801
8.778.2 Field Documentation . . . 801
8.778.2.1 RxDropConutTlv . . . 801
8.778.2.2 RxOkByteCountTlv . . . 801
8.778.2.3 RxOkConutTlv . . . 801
8.778.2.4 TxDropConutTlv . . . 801
8.778.2.5 TxOkByteCountTlv . . . 801
8.778.2.6 TxOkConutTlv . . . 802
8.779 qmifwinfo_s Struct Reference . . . 802
8.779.1 Detailed Description . . . 802
8.779.2 Field Documentation . . . 802
8.779.2.1 dev . . . 802
8.779.2.2 g . . . 802
8.779.2.3 s . . . 802
8.780 QmiNas3GppNetworkInfo Struct Reference . . . 802
8.780.1 Detailed Description . . . 803
8.780.2 Field Documentation . . . 803
8.780.2.1 pDesription . . . 804
8.780.2.2 pForbidden . . . 804
8.780.2.3 pInUse . . . 804
8.780.2.4 pMCC . . . 804
8.780.2.5 pMNC . . . 804
8.780.2.6 pPreferred . . . 804
8.780.2.7 pRoaming . . . 804
8.781 QmiNasGetRFBandInfoResp Struct Reference . . . 804
8.781.1 Field Documentation . . . 804
8.781.1.1 pInstancesSize . . . 804
8.781.1.2 pRFBandInfoElements . . . 804
8.781.1.3 results . . . 804
8.782 QmiNasPerformNetworkScanResp Struct Reference . . . 804
8.782.1 Field Documentation . . . 804
8.782.1.1 pInstances . . . 804

8.782.1.2 pInstanceSize . . . 804
8.782.1.3 results . . . 804
8.783 qmiSmsMessageList Struct Reference . . . 804
8.783.1 Detailed Description . . . 805
8.783.2 Field Documentation . . . 805
8.783.2.1 messageIndex . . . 805
8.783.2.2 messageTag . . . 805
8.784 qmiWDSDataBearerTechnology Struct Reference . . . 805
8.784.1 Detailed Description . . . 805
8.784.2 Field Documentation . . . 806
8.784.2.1 currentNetwork . . . 806
8.784.2.2 ratMask . . . 806
8.784.2.3 soMask . . . 806
8.785 QmiWdsIpAddressInfo Struct Reference . . . 806
8.785.1 Detailed Description . . . 806
8.785.2 Field Documentation . . . 807
8.785.2.1 pIPAddressV4 . . . 807
8.785.2.2 pIPAddressV6 . . . 807
8.785.2.3 pIPv6prefixlen . . . 807
8.786 qmiWdsRunTimeSettings Struct Reference . . . 807
8.786.1 Detailed Description . . . 808
8.786.2 Field Documentation . . . 810
8.786.2.1 pAPNName . . . 810
8.786.2.2 pAuthentication . . . 810
8.786.2.3 pDomainList . . . 810
8.786.2.4 pGPRSGrantedQoS . . . 810
8.786.2.5 pGWAddressV4 . . . 810
8.786.2.6 pIMCNflag . . . 810
8.786.2.7 pIPAddressV4 . . . 810
8.786.2.8 pIPFamilyPreference . . . 810
8.786.2.9 pIPV6AddrInfo . . . 810
8.786.2.10 pIPV6GWAddrInfo . . . 810
8.786.2.11 pMtu . . . 810
8.786.2.12 pPCSCFAddrPCO . . . 810
8.786.2.13 pPCSCFFQDNAddrList . . . 810
8.786.2.14 pPDPType . . . 810
8.786.2.15 pPrimaryDNSV4 . . . 810
8.786.2.16 pPrimaryDNSV6 . . . 810
8.786.2.17 pProfileID . . . 810
8.786.2.18 pProfileName . . . 811

8.786.2.19 pSecondaryDNSV4 . . . 811
8.786.2.20 pSecondaryDNSV6 . . . 811
8.786.2.21 pServerAddrList . . . 811
8.786.2.22 pSubnetMaskV4 . . . 811
8.786.2.23 pTechnology . . . 811
8.786.2.24 pUMTSGrantedQoS . . . 811
8.786.2.25 pUsername . . . 811
8.787 qmuxIPTable Struct Reference . . . 811
8.787.1 Field Documentation . . . 811
8.787.1.1 ipAddress . . . 811
8.787.1.2 muxID . . . 811
8.788 QosClassID Struct Reference . . . 811
8.788.1 Detailed Description . . . 811
8.788.2 Field Documentation . . . 812
8.788.2.1 gDlBitRate . . . 812
8.788.2.2 gUlBitRate . . . 812
8.788.2.3 maxDlBitRate . . . 812
8.788.2.4 maxUlBitRate . . . 812
8.788.2.5 QCI . . . 812
8.789 QosEventInfo Struct Reference . . . 812
8.789.1 Detailed Description . . . 812
8.789.2 Field Documentation . . . 813
8.789.2.1 pDataBearer . . . 813
8.789.2.2 pPacketsCountRX . . . 813
8.789.2.3 pPacketsCountTX . . . 813
8.789.2.4 pTotalBytesRX . . . 813
8.789.2.5 pTotalBytesTX . . . 814
8.790 QosFlowInfo Struct Reference . . . 814
8.790.1 Detailed Description . . . 814
8.790.2 Field Documentation . . . 814
8.790.2.1 pBearerID . . . 814
8.790.2.2 pQFlowState . . . 814
8.790.2.3 pRxQFilter . . . 814
8.790.2.4 pRxQFlowGranted . . . 814
8.790.2.5 pTxQFilter . . . 814
8.790.2.6 pTxQFlowGranted . . . 815
8.791 QosFlowInfoState Struct Reference . . . 815
8.791.1 Detailed Description . . . 815
8.791.2 Field Documentation . . . 815
8.791.2.1 id . . . 815

8.791.2.2 isNewFlow . . . 815
8.791.2.3 state . . . 815
8.792 QosMap Struct Reference . . . 815
8.792.1 Detailed Description . . . 815
8.792.2 Field Documentation . . . 816
8.792.2.1 dscp . . . 816
8.792.2.2 qos_id . . . 816
8.792.2.3 state . . . 816
8.793 RankIndicatorInd Struct Reference . . . 816
8.793.1 Field Documentation . . . 816
8.793.1.1 Count1 . . . 816
8.793.1.2 Count2 . . . 816
8.794 readResult Struct Reference . . . 816
8.794.1 Detailed Description . . . 816
8.794.2 Field Documentation . . . 817
8.794.2.1 content . . . 817
8.794.2.2 contentLen . . . 817
8.795 readTransparentInfo Struct Reference . . . 817
8.795.1 Detailed Description . . . 817
8.795.2 Field Documentation . . . 817
8.795.2.1 length . . . 817
8.795.2.2 offset . . . 817
8.796 redirNumInfo Struct Reference . . . 817
8.796.1 Detailed Description . . . 817
8.796.2 Field Documentation . . . 819
8.796.2.1 number . . . 819
8.796.2.2 numLen . . . 819
8.796.2.3 numPlan . . . 819
8.796.2.4 numType . . . 819
8.796.2.5 PI . . . 819
8.796.2.6 reason . . . 819
8.796.2.7 SI . . . 819
8.797 registerRefresh Struct Reference . . . 819
8.797.1 Detailed Description . . . 819
8.797.2 Field Documentation . . . 820
8.797.2.1 arrfileInfo . . . 820
8.797.2.2 numFiles . . . 820
8.797.2.3 registerFlag . . . 820
8.797.2.4 voteForInit . . . 820
8.798 remainingRetries Struct Reference . . . 820

8.798.1 Detailed Description . . . 820
8.798.2 Field Documentation . . . 820
8.798.2.1 unblockLeft . . . 820
8.798.2.2 verifyLeft . . . 820
8.799 remoteParty Name Struct Reference . . . 820
8.799.1 Detailed Description . . . 821
8.799.2 Field Documentation . . . 821
8.799.2.1 callerName . . . 821
8.799.2.2 codingScheme . . . 821
8.799.2.3 nameLen . . . 821
8.799.2.4 namePI . . . 821
8.800 remotePartyNum Struct Reference . . . 821
8.800.1 Detailed Description . . . 821
8.800.2 Field Documentation . . . 822
8.800.2.1 numLen . . . 822
8.800.2.2 presentationInd . . . 822
8.800.2.3 remPartyNumber . . . 822
8.801 ReqFieldsList Struct Reference . . . 822
8.801.1 Detailed Description . . . 822
8.801.2 Field Documentation . . . 822
8.801.2.1 requestFields . . . 822
8.801.2.2 requestFieldsLen . . . 823
8.802 RespFieldsList Struct Reference . . . 823
8.802.1 Detailed Description . . . 823
8.802.2 Field Documentation . . . 823
8.802.2.1 responseFields . . . 823
8.802.2.2 responseFieldsLen . . . 823
8.803 RFBandInfoElements Struct Reference . . . 823
8.803.1 Detailed Description . . . 823
8.803.2 Field Documentation . . . 824
8.803.2.1 activeBandClass . . . 824
8.803.2.2 activeBandClass . . . 824
8.803.2.3 activeChannel . . . 824
8.803.2.4 activeChannel . . . 824
8.803.2.5 radioInterface . . . 824
8.803.2.6 radioInterface . . . 824
8.804 rmTrasnferStaticsReq Struct Reference . . . 824
8.804.1 Detailed Description . . . 824
8.804.2 Field Documentation . . . 825
8.804.2.1 bResetStatistics . . . 825

8.804.2.2 ulMask . . . 825
8.805 roamIndList Struct Reference . . . 825
8.805.1 Detailed Description . . . 825
8.805.2 Field Documentation . . . 826
8.805.2.1 numInstances . . . 826
8.805.2.2 radioInterface . . . 826
8.805.2.3 roamIndicator . . . 826
8.806 RoamingInfo Struct Reference . . . 826
8.806.1 Field Documentation . . . 826
8.806.1.1 roaming_ind . . . 826
8.806.1.2 TlvPresent . . . 826
8.807 roamTimer Struct Reference . . . 826
8.807.1 Detailed Description . . . 826
8.807.2 Field Documentation . . . 827
8.807.2.1 namID . . . 827
8.807.2.2 roamTimerValue . . . 827
8.808 RSRPThresh Struct Reference . . . 827
8.808.1 Detailed Description . . . 827
8.808.2 Field Documentation . . . 827
8.808.2.1 pRSRPThresList . . . 827
8.808.2.2 RSRPThresListLen . . . 827
8.809 rsrqInformation Struct Reference . . . 827
8.809.1 Detailed Description . . . 827
8.809.2 Field Documentation . . . 828
8.809.2.1 radioIf . . . 828
8.809.2.2 rsrq . . . 828
8.810 RSRQThresh Struct Reference . . . 828
8.810.1 Detailed Description . . . 828
8.810.2 Field Documentation . . . 828
8.810.2.1 pRSRQThresList . . . 828
8.810.2.2 RSRQThresListLen . . . 828
8.811 RSSIThresh Struct Reference . . . 828
8.811.1 Detailed Description . . . 829
8.811.2 Field Documentation . . . 829
8.811.2.1 pRSSIThresList . . . 829
8.811.2.2 RSSIThresListLen . . . 829
8.812 RXAGCList Struct Reference . . . 829
8.812.1 Detailed Description . . . 829
8.812.2 Field Documentation . . . 830
8.812.2.1 pRXAIG . . . 830

8.812.2.2 pRXComprSlope . . . 830
8.812.2.3 pRXComprThres . . . 830
8.812.2.4 pRXExpSlope . . . 830
8.812.2.5 pRXExpThres . . . 830
8.812.2.6 pRXStaticGain . . . 830
8.813 RXAVCList Struct Reference . . . 830
8.813.1 Detailed Description . . . 830
8.813.2 Field Documentation . . . 830
8.813.2.1 pAVRXAVCHeadroom . . . 831
8.813.2.2 pAVRXAVCSens . . . 831
8.814 rxInfo Struct Reference . . . 831
8.814.1 Detailed Description . . . 831
8.814.2 Field Documentation . . . 831
8.814.2.1 ecio . . . 831
8.814.2.2 isRadioTuned . . . 831
8.814.2.3 phase . . . 832
8.814.2.4 rscp . . . 832
8.814.2.5 rsrp . . . 832
8.814.2.6 rxPower . . . 832
8.815 RXPCMIIRFltr Struct Reference . . . 832
8.815.1 Detailed Description . . . 832
8.815.2 Field Documentation . . . 833
8.815.2.1 pFlag . . . 833
8.815.2.2 pStage0Val . . . 833
8.815.2.3 pStage1Val . . . 833
8.815.2.4 pStage2Val . . . 833
8.815.2.5 pStage3Val . . . 833
8.815.2.6 pStage4Val . . . 833
8.815.2.7 pStageCnt . . . 833
8.816 RxSigInfo Struct Reference . . . 833
8.816.1 Detailed Description . . . 834
8.816.2 Field Documentation . . . 834
8.816.2.1 isRadioTuned . . . 834
8.816.2.2 rsrp . . . 834
8.816.2.3 rxChainIndex . . . 834
8.816.2.4 rxPower . . . 834
8.817 rxSignalStrengthListElement Struct Reference . . . 834
8.817.1 Detailed Description . . . 834
8.817.2 Field Documentation . . . 835
8.817.2.1 radioIf . . . 835

8.817.2.2 rxSignalStrength . . . 835
8.818 sApnExtraParams Struct Reference . . . 835
8.818.1 Detailed Description . . . 835
8.818.2 Field Documentation . . . 836
8.818.2.1 ambr_dl . . . 836
8.818.2.2 ambr_dl_ext . . . 836
8.818.2.3 ambr_dl_ext2 . . . 836
8.818.2.4 ambr_ul . . . 836
8.818.2.5 ambr_ul_ext . . . 836
8.818.2.6 ambr_ul_ext2 . . . 836
8.818.2.7 apnId . . . 836
8.819 satelliteInfo Struct Reference . . . 836
8.819.1 Detailed Description . . . 837
8.819.2 Field Documentation . . . 838
8.819.2.1 azimuth . . . 838
8.819.2.2 elevation . . . 838
8.819.2.3 gnssSvId . . . 838
8.819.2.4 healthStatus . . . 838
8.819.2.5 snr . . . 838
8.819.2.6 svInfoMask . . . 838
8.819.2.7 svListLen . . . 838
8.819.2.8 svStatus . . . 838
8.819.2.9 system . . . 838
8.819.2.10 validMask . . . 838
8.820 SccRxInfo Struct Reference . . . 839
8.820.1 Detailed Description . . . 839
8.820.2 Field Documentation . . . 839
8.820.2.1 numInstances . . . 839
8.820.2.2 rsrq . . . 839
8.820.2.3 sigInfo . . . 839
8.820.2.4 snr . . . 839
8.820.2.5 TlvPresent . . . 839
8.821 sensorData Struct Reference . . . 839
8.821.1 Detailed Description . . . 840
8.821.2 Field Documentation . . . 841
8.821.2.1 flags . . . 841
8.821.2.2 sensorDataLen . . . 841
8.821.2.3 timeOfFirstSample . . . 841
8.821.2.4 timeOffset . . . 841
8.821.2.5 xAxis . . . 841

8.821.2.6 yAxis . . . 841
8.821.2.7 zAxis . . . 841
8.822 sensorData_t Struct Reference . . . 841
8.822.1 Detailed Description . . . 841
8.822.2 Field Documentation . . . 842
8.822.2.1 flags . . . 842
8.822.2.2 sensorDataLen . . . 842
8.822.2.3 timeOfFirstSample . . . 842
8.822.2.4 timeOffset . . . 842
8.822.2.5 xAxis . . . 842
8.822.2.6 yAxis . . . 842
8.822.2.7 zAxis . . . 842
8.823 sensorDataUsage_s Struct Reference . . . 843
8.823.1 Detailed Description . . . 843
8.823.2 Field Documentation . . . 843
8.823.2.1 aidingIndicatorMask . . . 843
8.823.2.2 usageMask . . . 843
8.824 serialNumbersInfo Struct Reference . . . 843
8.824.1 Detailed Description . . . 844
8.824.2 Field Documentation . . . 844
8.824.2.1 esnSize . . . 844
8.824.2.2 imeiSize . . . 844
8.824.2.3 imeiSvnSize . . . 844
8.824.2.4 meidSize . . . 844
8.824.2.5 pESNString . . . 844
8.824.2.6 pIMEIString . . . 844
8.824.2.7 pImeiSvnString . . . 844
8.824.2.8 pMEIDString . . . 844
8.825 serviceProviderName Struct Reference . . . 845
8.825.1 Detailed Description . . . 845
8.825.2 Field Documentation . . . 845
8.825.2.1 displayCondition . . . 845
8.825.2.2 spn . . . 845
8.825.2.3 spnLength . . . 845
8.826 ServingSystemInfo Struct Reference . . . 845
8.826.1 Detailed Description . . . 845
8.826.2 Field Documentation . . . 846
8.826.2.1 csAttachState . . . 846
8.826.2.2 hdrPersonality . . . 846
8.826.2.3 psAttachState . . . 846

8.826.2.4 radioInterfaceList . . . 847
8.826.2.5 radioInterfaceNo . . . 847
8.826.2.6 registrationState . . . 847
8.826.2.7 selectedNetwork . . . 847
8.827 servSystem Struct Reference . . . 847
8.827.1 Detailed Description . . . 847
8.827.2 Field Documentation . . . 848
8.827.2.1 csAttachState . . . 848
8.827.2.2 numRadioInterfaces . . . 848
8.827.2.3 psAttachState . . . 848
8.827.2.4 radioInterface . . . 848
8.827.2.5 regState . . . 848
8.827.2.6 selNetwork . . . 848
8.828 sessionInfo Union Reference . . . 848
8.828.1 Detailed Description . . . 848
8.828.2 Field Documentation . . . 849
8.828.2.1 omaDmConfig . . . 849
8.828.2.2 omaDmFota . . . 849
8.828.2.3 omaDmNotifications . . . 849
8.829 sessionInfoExt Union Reference . . . 849
8.829.1 Detailed Description . . . 849
8.829.2 Field Documentation . . . 849
8.829.2.1 omaDmConfig . . . 849
8.829.2.2 omaDmFota . . . 849
8.830 sessionInfoTlv Struct Reference . . . 849
8.830.1 Detailed Description . . . 849
8.830.2 Field Documentation . . . 849
8.830.2.1 sessionInfo . . . 849
8.830.2.2 sessionType . . . 849
8.830.2.3 TlvPresent . . . 849
8.831 sessionInfoTlvExt Struct Reference . . . 850
8.831.1 Detailed Description . . . 850
8.831.2 Field Documentation . . . 850
8.831.2.1 sessionInfo . . . 850
8.831.2.2 sessionType . . . 850
8.831.2.3 TlvPresent . . . 850
8.832 SetAudioPathConfigReq Struct Reference . . . 850
8.832.1 Detailed Description . . . 850
8.832.2 Field Documentation . . . 851
8.832.2.1 pCodecSTGain . . . 851

8.832.2.2 pDTMFTXGain . . . 852
8.832.2.3 pECMode . . . 852
8.832.2.4 pNSEnable . . . 852
8.832.2.5 Profile . . . 852
8.832.2.6 pRXAGCList . . . 852
8.832.2.7 pRXAVCAGCSwitch . . . 852
8.832.2.8 pRXAVCList . . . 852
8.832.2.9 pRXPCMIIRFltr . . . 852
8.832.2.10 pTXAGCList . . . 852
8.832.2.11 pTXAVCSwitch . . . 852
8.832.2.12 pTXGain . . . 852
8.832.2.13 pTXPCMIIRFltr . . . 852
8.833 SetAudioProfileReq Struct Reference . . . 852
8.833.1 Detailed Description . . . 852
8.833.2 Field Documentation . . . 853
8.833.2.1 EarMute . . . 853
8.833.2.2 Generator . . . 853
8.833.2.3 MicMute . . . 853
8.833.2.4 Profile . . . 853
8.833.2.5 Volume . . . 853
8.834 SetAudioVolTLBConfigReq Struct Reference . . . 853
8.834.1 Detailed Description . . . 853
8.834.2 Field Documentation . . . 854
8.834.2.1 Generator . . . 854
8.834.2.2 Item . . . 854
8.834.2.3 Profile . . . 854
8.834.2.4 Volume . . . 854
8.834.2.5 VolValue . . . 854
8.835 SetAudioVolTLBConfigResp Struct Reference . . . 854
8.835.1 Detailed Description . . . 854
8.835.2 Field Documentation . . . 854
8.835.2.1 ResCode . . . 854
8.836 setCustomSettingV2 Struct Reference . . . 855
8.836.1 Detailed Description . . . 855
8.836.2 Field Documentation . . . 855
8.836.2.1 cust_id . . . 855
8.836.2.2 cust_value . . . 855
8.836.2.3 value_length . . . 855
8.837 setDyingGaspCfg Struct Reference . . . 855
8.837.1 Detailed Description . . . 855

8.837.2 Field Documentation . . . 856
8.837.2.1 pDestSMSContent . . . 856
8.837.2.2 pDestSMSNum . . . 856
8.838 SetIMSSMSConfigReq Struct Reference . . . 856
8.838.1 Detailed Description . . . 856
8.838.2 Field Documentation . . . 856
8.838.2.1 pPhoneCtxtURI . . . 856
8.838.2.2 pPhoneCtxtURILen . . . 856
8.838.2.3 pSMSFormat . . . 856
8.838.2.4 pSMSOverIPNwInd . . . 856
8.839 SetIMSSMSConfigResp Struct Reference . . . 856
8.839.1 Detailed Description . . . 857
8.839.2 Field Documentation . . . 857
8.839.2.1 pSettingResp . . . 857
8.840 SetIMSUserConfigReq Struct Reference . . . 857
8.840.1 Detailed Description . . . 857
8.840.2 Field Documentation . . . 857
8.840.2.1 pIMSDomain . . . 857
8.840.2.2 pIMSDomainLen . . . 857
8.841 SetIMSUserConfigResp Struct Reference . . . 857
8.841.1 Detailed Description . . . 858
8.841.2 Field Documentation . . . 858
8.841.2.1 pSettingResp . . . 858
8.842 SetIMSVoIPConfigReq Struct Reference . . . 858
8.842.1 Detailed Description . . . 858
8.842.2 Field Documentation . . . 860
8.842.2.1 pAmrMode . . . 860
8.842.2.2 pAmrOctetAligned . . . 860
8.842.2.3 pAmrWbEnable . . . 860
8.842.2.4 pAmrWBMode . . . 860
8.842.2.5 pAmrWBOctetAligned . . . 860
8.842.2.6 pMinSessionExpiryTimer . . . 860
8.842.2.7 pRingBackTimer . . . 860
8.842.2.8 pRingingTimer . . . 860
8.842.2.9 pRTPRTCPInactTimer . . . 860
8.842.2.10 pScrAmrEnable . . . 860
8.842.2.11 pScrAmrWbEnable . . . 860
8.842.2.12 pSessionExpiryTimer . . . 860
8.843 SetIMSVoIPConfigResp Struct Reference . . . 860
8.843.1 Detailed Description . . . 860

8.843.2 Field Documentation . . . 861
8.843.2.1 pSettingResp . . . 861
8.844 SetM2MAudioAVCFGReq Struct Reference . . . 861
8.844.1 Detailed Description . . . 861
8.844.2 Field Documentation . . . 861
8.844.2.1 Device . . . 861
8.844.2.2 PIFACEId . . . 861
8.844.2.3 pPCMParams . . . 861
8.844.2.4 Profile . . . 861
8.845 SetM2MAudioLPBKReq Struct Reference . . . 862
8.845.1 Detailed Description . . . 862
8.845.2 Field Documentation . . . 862
8.845.2.1 Enable . . . 862
8.846 SetM2MAudioProfileReq Struct Reference . . . 862
8.846.1 Detailed Description . . . 862
8.846.2 Field Documentation . . . 863
8.846.2.1 pCwtMute . . . 863
8.846.2.2 pEarMute . . . 863
8.846.2.3 pGenerator . . . 863
8.846.2.4 pMicMute . . . 863
8.846.2.5 Profile . . . 863
8.846.2.6 pVolume . . . 863
8.847 SetM2MAudioVolumeReq Struct Reference . . . 863
8.847.1 Detailed Description . . . 863
8.847.2 Field Documentation . . . 864
8.847.2.1 Generator . . . 864
8.847.2.2 Level . . . 864
8.847.2.3 Profile . . . 864
8.848 SetM2MAVMuteReq Struct Reference . . . 864
8.848.1 Detailed Description . . . 864
8.848.2 Field Documentation . . . 864
8.848.2.1 EarMute . . . 864
8.848.2.2 MicMute . . . 864
8.848.2.3 pCwtMute . . . 865
8.848.2.4 Profile . . . 865
8.849 SetM2MSpkrGainReq Struct Reference . . . 865
8.849.1 Detailed Description . . . 865
8.849.2 Field Documentation . . . 865
8.849.2.1 Profile . . . 865
8.849.2.2 Value . . . 865

8.850 setPINProtection Struct Reference . . . 865
8.850.1 Detailed Description . . . 865
8.850.2 Field Documentation . . . 866
8.850.2.1 pinID . . . 866
8.850.2.2 pinLength . . . 866
8.850.2.3 pinOperation . . . 866
8.850.2.4 pinValue . . . 866
8.851 SetRegMgrConfigReq Struct Reference . . . 866
8.851.1 Detailed Description . . . 866
8.851.2 Field Documentation . . . 867
8.851.2.1 pCSCFPortName . . . 867
8.851.2.2 pCSCFPortNameLen . . . 867
8.851.2.3 pIMSTestMode . . . 867
8.851.2.4 pPriCSCFPort . . . 867
8.852 SetRegMgrConfigResp Struct Reference . . . 867
8.852.1 Detailed Description . . . 867
8.852.2 Field Documentation . . . 867
8.852.2.1 pSettingResp . . . 867
8.853 setSignalStrengthInfo Struct Reference . . . 867
8.853.1 Detailed Description . . . 868
8.853.2 Field Documentation . . . 871
8.853.2.1 pCDMAECIODelta . . . 871
8.853.2.2 pCDMAECIOThresh . . . 871
8.853.2.3 pCDMARSSIDelta . . . 871
8.853.2.4 pCDMARSSIThresh . . . 871
8.853.2.5 pGSMRSSIDelta . . . 871
8.853.2.6 pGSMRSSIThresh . . . 871
8.853.2.7 pHDRECIODelta . . . 871
8.853.2.8 pHDRECIOThresh . . . 871
8.853.2.9 pHDRIODelta . . . 871
8.853.2.10 pHDRIOThresh . . . 871
8.853.2.11 pHDRRSSIDelta . . . 871
8.853.2.12 pHDRRSSIThresh . . . 871
8.853.2.13 pHDRSINRDelta . . . 871
8.853.2.14 pHDRSINRThresh . . . 871
8.853.2.15 pLTERSRPDelta . . . 871
8.853.2.16 pLTERSRPThresh . . . 871
8.853.2.17 pLTERSRQDelta . . . 871
8.853.2.18 pLTERSRQThresh . . . 872
8.853.2.19 pLTERSSIDelta . . . 872

8.853.2.20 pLTERSSIThresh . . . 872
8.853.2.21 pLTESigRptConfig . . . 872
8.853.2.22 pLTESNRDelta . . . 872
8.853.2.23 pLTESNRThresh . . . 872
8.853.2.24 pTDSCDMAECIODelta . . . 872
8.853.2.25 pTDSCDMAECIOThresh . . . 872
8.853.2.26 pTDSCDMARSCPDelta . . . 872
8.853.2.27 pTDSCDMARSCPThresh . . . 872
8.853.2.28 pTDSCDMARSSIDelta . . . 872
8.853.2.29 pTDSCDMARSSIThresh . . . 872
8.853.2.30 pTDSCDMASINRDelta . . . 872
8.853.2.31 pTDSCDMASINRThresh . . . 872
8.853.2.32 pWCDMAECIODelta . . . 872
8.853.2.33 pWCDMAECIOThresh . . . 872
8.853.2.34 pWCDMARSSIDelta . . . 872
8.853.2.35 pWCDMARSSIThresh . . . 872
8.854 SetSIPConfigReq Struct Reference . . . 872
8.854.1 Detailed Description . . . 872
8.854.2 Field Documentation . . . 873
8.854.2.1 pSigCompEnabled . . . 873
8.854.2.2 pSIPLocalPort . . . 873
8.854.2.3 pSubscribeTimer . . . 873
8.854.2.4 pTimerSIPReg . . . 873
8.854.2.5 pTimerT1 . . . 873
8.854.2.6 pTimerT2 . . . 873
8.854.2.7 pTimerTf . . . 873
8.855 SetSIPConfigResp Struct Reference . . . 873
8.855.1 Detailed Description . . . 873
8.855.2 Field Documentation . . . 874
8.855.2.1 pSettingResp . . . 874
8.856 sGetDeviceSeriesResult Struct Reference . . . 874
8.856.1 Detailed Description . . . 874
8.856.2 Field Documentation . . . 874
8.856.2.1 eDevice . . . 874
8.856.2.2 uResult . . . 874
8.857 sidNid Struct Reference . . . 874
8.857.1 Detailed Description . . . 874
8.857.2 Field Documentation . . . 875
8.857.2.1 nid . . . 875
8.857.2.2 sid . . . 875

8.858 sigInfo Struct Reference . . . 875
8.858.1 Detailed Description . . . 875
8.858.2 Field Documentation . . . 876
8.858.2.1 pECIOThresh . . . 876
8.858.2.2 pHDRSINRThresh . . . 876
8.858.2.3 pIOThresh . . . 876
8.858.2.4 pLTESigRptCfg . . . 876
8.858.2.5 pLTESNRThresh . . . 876
8.858.2.6 pRSRPThresh . . . 876
8.858.2.7 pRSRQThresh . . . 876
8.858.2.8 pRSSIThresh . . . 876
8.858.2.9 pTDSCDMASINRCONFThresh . . . 876
8.859 signalInfo Struct Reference . . . 876
8.859.1 Detailed Description . . . 876
8.859.2 Field Documentation . . . 877
8.859.2.1 alertPitch . . . 877
8.859.2.2 signal . . . 877
8.859.2.3 signalType . . . 877
8.860 SignalStrengthDataType Struct Reference . . . 877
8.860.1 Field Documentation . . . 877
8.860.1.1 thresholds . . . 877
8.860.1.2 thresholdsSize . . . 877
8.861 slot_t Struct Reference . . . 877
8.861.1 Detailed Description . . . 877
8.861.2 Field Documentation . . . 878
8.861.2.1 bICCID . . . 878
8.861.2.2 bICCIDLength . . . 878
8.861.2.3 bLogicalSlot . . . 878
8.861.2.4 uPhyCardStatus . . . 878
8.861.2.5 uPhySlotStatus . . . 878
8.862 slotInf Struct Reference . . . 878
8.862.1 Detailed Description . . . 878
8.862.2 Field Documentation . . . 879
8.862.2.1 AppStatus . . . 879
8.862.2.2 cardState . . . 879
8.862.2.3 errorState . . . 879
8.862.2.4 numApp . . . 879
8.862.2.5 upinRetries . . . 880
8.862.2.6 upinState . . . 880
8.862.2.7 upukRetries . . . 880

8.863 slotInfo Struct Reference . . . 880
8.863.1 Detailed Description . . . 880
8.863.2 Field Documentation . . . 881
8.863.2.1 AppStatus . . . 881
8.863.2.2 cardState . . . 881
8.863.2.3 errorState . . . 881
8.863.2.4 numApp . . . 881
8.863.2.5 upinRetries . . . 881
8.863.2.6 upinState . . . 881
8.863.2.7 upukRetries . . . 881
8.864 slots_t Struct Reference . . . 881
8.864.1 Detailed Description . . . 881
8.864.2 Field Documentation . . . 882
8.864.2.1 uimSlotStatus . . . 882
8.865 slqsautoconnect Struct Reference . . . 882
8.865.1 Detailed Description . . . 882
8.865.2 Field Documentation . . . 882
8.865.2.1 acroamsetting . . . 882
8.865.2.2 acsetting . . . 882
8.865.2.3 action . . . 882
8.866 SLQSDeleteProfileParams Struct Reference . . . 882
8.866.1 Detailed Description . . . 883
8.866.2 Field Documentation . . . 883
8.866.2.1 profileIndex . . . 883
8.866.2.2 profileType . . . 883
8.867 slqsfwinfo_s Struct Reference . . . 883
8.867.1 Detailed Description . . . 883
8.867.2 Field Documentation . . . 884
8.867.2.1 appversion_str . . . 884
8.867.2.2 bootversion_str . . . 884
8.867.2.3 carrier_str . . . 884
8.867.2.4 cur_carr_name . . . 884
8.867.2.5 cur_carr_rev . . . 884
8.867.2.6 modelid_str . . . 884
8.867.2.7 packageid_str . . . 884
8.867.2.8 priversion_str . . . 884
8.867.2.9 sku_str . . . 884
8.868 SlqsNas3GppNetworkInfo Struct Reference . . . 884
8.868.1 Detailed Description . . . 885
8.868.2 Field Documentation . . . 885

8.868.2.1 Description . . . 885
8.868.2.2 Forbidden . . . 885
8.868.2.3 InUse . . . 885
8.868.2.4 MCC . . . 885
8.868.2.5 MNC . . . 885
8.868.2.6 Preferred . . . 886
8.868.2.7 Roaming . . . 886
8.869 SlqsNasPcsDigit Struct Reference . . . 886
8.869.1 Detailed Description . . . 886
8.869.2 Field Documentation . . . 886
8.869.2.1 includes_pcs_digit . . . 886
8.869.2.2 MCC . . . 886
8.869.2.3 MNC . . . 886
8.870 slqssendasyncsmsparams_s Struct Reference . . . 886
8.870.1 Detailed Description . . . 887
8.870.2 Field Documentation . . . 888
8.870.2.1 messageFormat . . . 888
8.870.2.2 messageSize . . . 888
8.870.2.3 pFollowOnDC . . . 888
8.870.2.4 pForceOnDC . . . 888
8.870.2.5 pLinktimer . . . 888
8.870.2.6 pMessage . . . 888
8.870.2.7 pRetryMessage . . . 888
8.870.2.8 pRetryMessageId . . . 888
8.870.2.9 pServiceOption . . . 888
8.870.2.10 pSmsOnIms . . . 888
8.870.2.11 pUserData . . . 888
8.871 slqssendsmsparams_s Struct Reference . . . 888
8.871.1 Detailed Description . . . 889
8.871.2 Field Documentation . . . 889
8.871.2.1 messageFailureCode . . . 889
8.871.2.2 messageFormat . . . 889
8.871.2.3 messageID . . . 889
8.871.2.4 messageSize . . . 889
8.871.2.5 pLinktimer . . . 889
8.871.2.6 pMessage . . . 889
8.871.2.7 pSmsOnIms . . . 889
8.872 slqsSessionStateInfo Struct Reference . . . 890
8.872.1 Detailed Description . . . 890
8.872.2 Field Documentation . . . 890

8.872.2.1 pQmiInterfaceInfo . . . 890
8.872.2.2 reconfiguration_required . . . 890
8.872.2.3 sessionEndReason . . . 890
8.872.2.4 state . . . 890
8.873 slqsSignalStrengthInfo Struct Reference . . . 890
8.873.1 Detailed Description . . . 891
8.873.2 Field Documentation . . . 892
8.873.2.1 ecioList . . . 892
8.873.2.2 ecioListLen . . . 892
8.873.2.3 errorRateList . . . 892
8.873.2.4 errorRateListLen . . . 892
8.873.2.5 Io . . . 892
8.873.2.6 ltersrp . . . 892
8.873.2.7 ltesnr . . . 892
8.873.2.8 rsrqInfo . . . 892
8.873.2.9 rxSignalStrengthList . . . 892
8.873.2.10 rxSignalStrengthListLen . . . 892
8.873.2.11 signalStrengthReqMask . . . 893
8.873.2.12 sinr . . . 893
8.874 SLQSSignalStrengthsIndReq Struct Reference . . . 893
8.874.1 Detailed Description . . . 893
8.874.2 Field Documentation . . . 894
8.874.2.1 ecioDelta . . . 894
8.874.2.2 ecioThresholdList . . . 894
8.874.2.3 ecioThresholdListLen . . . 894
8.874.2.4 ioDelta . . . 894
8.874.2.5 lteRsrpDelta . . . 894
8.874.2.6 lteSnrDelta . . . 894
8.874.2.7 rsrqDelta . . . 894
8.874.2.8 rxSignalStrengthDelta . . . 894
8.874.2.9 sinrDelta . . . 894
8.874.2.10 sinrThresholdList . . . 894
8.874.2.11 sinrThresholdListLen . . . 894
8.875 SLQSSignalStrengthsInformation Struct Reference . . . 895
8.875.1 Detailed Description . . . 895
8.875.2 Field Documentation . . . 895
8.875.2.1 ecioInfo . . . 896
8.875.2.2 errorRateInfo . . . 896
8.875.2.3 io . . . 896
8.875.2.4 lteRsrpinfo . . . 896

8.875.2.5 lteSnrinfo . . . 896
8.875.2.6 rsrqInfo . . . 896
8.875.2.7 rxSignalStrengthInfo . . . 896
8.875.2.8 sinr . . . 896
8.876 slqsWdsEventInfo Struct Reference . . . 896
8.876.1 Detailed Description . . . 896
8.876.2 Field Documentation . . . 897
8.876.2.1 pDataBearer . . . 897
8.876.2.2 pDormancyStatus . . . 897
8.876.2.3 pPacketsCountRX . . . 897
8.876.2.4 pPacketsCountTX . . . 897
8.876.2.5 pQmiInterfaceInfo . . . 898
8.876.2.6 pTotalBytesRX . . . 898
8.876.2.7 pTotalBytesTX . . . 898
8.877 sms_BroadcastConfig Struct Reference . . . 898
8.877.1 Detailed Description . . . 898
8.877.2 Field Documentation . . . 898
8.877.2.1 fromServiceId . . . 898
8.877.2.2 selected . . . 898
8.877.2.3 toServiceId . . . 898
8.878 sms_CDMABroadcastConfig Struct Reference . . . 898
8.878.1 Detailed Description . . . 898
8.878.2 Field Documentation . . . 899
8.878.2.1 language . . . 899
8.878.2.2 selected . . . 899
8.878.2.3 serviceCategory . . . 899
8.879 sms_getIndicationReg Struct Reference . . . 899
8.879.1 Detailed Description . . . 899
8.879.2 Field Documentation . . . 900
8.879.2.1 pRegCallStatInfoEvt . . . 900
8.879.2.2 pRegTransLayerInfoEvt . . . 900
8.879.2.3 pRegTransNWRegInfoEvt . . . 900
8.880 sms_getMsgWaitingInfo Struct Reference . . . 900
8.880.1 Detailed Description . . . 900
8.880.2 Field Documentation . . . 900
8.880.2.1 msgWaitInfo . . . 900
8.880.2.2 numInstances . . . 900
8.881 sms_getTransLayerInfo Struct Reference . . . 900
8.881.1 Detailed Description . . . 901
8.881.2 Field Documentation . . . 901

8.881.2.1 pRegInd . . . 901
8.881.2.2 pTransLayerInfo . . . 901
8.882 sms_getTransNWRegInfo Struct Reference . . . 901
8.882.1 Detailed Description . . . 901
8.882.2 Field Documentation . . . 902
8.882.2.1 pRegStatus . . . 902
8.883 sms_maxStorageSizeReq Struct Reference . . . 902
8.883.1 Detailed Description . . . 902
8.883.2 Field Documentation . . . 902
8.883.2.1 pMessageMode . . . 902
8.883.2.2 storageType . . . 902
8.884 sms_maxStorageSizeResp Struct Reference . . . 902
8.884.1 Detailed Description . . . 902
8.884.2 Field Documentation . . . 903
8.884.2.1 freeSlots . . . 903
8.884.2.2 maxStorageSize . . . 903
8.885 sms_messageWaitingInfoContent Struct Reference . . . 903
8.885.1 Detailed Description . . . 903
8.885.2 Field Documentation . . . 903
8.885.2.1 activeInd . . . 903
8.885.2.2 msgCount . . . 903
8.885.2.3 msgType . . . 903
8.886 sms_msgProtocolResp Struct Reference . . . 904
8.886.1 Detailed Description . . . 904
8.886.2 Field Documentation . . . 904
8.886.2.1 msgProtocol . . . 904
8.887 sms_qaQmi3GPP2BroadcastCfgInfo Struct Reference . . . 904
8.887.1 Detailed Description . . . 904
8.887.2 Field Documentation . . . 905
8.887.2.1 activated_ind . . . 905
8.887.2.2 CDMABroadcastConfig . . . 905
8.887.2.3 num_instances . . . 905
8.888 sms_qaQmi3GPPBroadcastCfgInfo Struct Reference . . . 905
8.888.1 Detailed Description . . . 905
8.888.2 Field Documentation . . . 905
8.888.2.1 activated_ind . . . 905
8.888.2.2 broadcastConfig . . . 905
8.888.2.3 num_instances . . . 905
8.889 sms_routeEntry Struct Reference . . . 905
8.889.1 Detailed Description . . . 906

8.889.2 Field Documentation . . . 906
8.889.2.1 messageClass . . . 906
8.889.2.2 messageType . . . 906
8.889.2.3 receiptAction . . . 906
8.889.2.4 routeStorage . . . 906
8.890 sms_sendAsyncsmsParams Struct Reference . . . 906
8.890.1 Detailed Description . . . 907
8.890.2 Field Documentation . . . 908
8.890.2.1 messageFormat . . . 908
8.890.2.2 messageSize . . . 908
8.890.2.3 pFollowOnDC . . . 908
8.890.2.4 pForceOnDC . . . 908
8.890.2.5 pLinktimer . . . 908
8.890.2.6 pMessage . . . 908
8.890.2.7 pRetryMessage . . . 908
8.890.2.8 pRetryMessageId . . . 908
8.890.2.9 pServiceOption . . . 908
8.890.2.10 pSmsOnIms . . . 908
8.890.2.11 pUserData . . . 908
8.891 sms_setIndicationReg Struct Reference . . . 909
8.891.1 Detailed Description . . . 909
8.891.2 Field Documentation . . . 909
8.891.2.1 pRegCallStatInfoEvt . . . 909
8.891.2.2 pRegTransLayerInfoEvt . . . 909
8.891.2.3 pRegTransNWRegInfoEvt . . . 909
8.892 sms_setRoutesReq Struct Reference . . . 909
8.892.1 Detailed Description . . . 910
8.892.2 Field Documentation . . . 910
8.892.2.1 numOfRoutes . . . 910
8.892.2.2 pTransferStatusReport . . . 910
8.892.2.3 routeList . . . 910
8.893 sms_transLayerInfo Struct Reference . . . 910
8.893.1 Detailed Description . . . 910
8.893.2 Field Documentation . . . 910
8.893.2.1 TransCap . . . 911
8.893.2.2 TransType . . . 911
8.894 SMSAsyncRawSend_s Struct Reference . . . 911
8.894.1 Detailed Description . . . 911
8.894.2 Field Documentation . . . 912
8.894.2.1 alphaIDLen . . . 912

8.894.2.2 causeCode . . . 912
8.894.2.3 errorClass . . . 912
8.894.2.4 messageID . . . 912
8.894.2.5 msgDelFailureCause . . . 912
8.894.2.6 msgDelFailureType . . . 912
8.894.2.7 pAlphaID . . . 912
8.894.2.8 RPCause . . . 912
8.894.2.9 sendStatus . . . 912
8.894.2.10TPCause . . . 912
8.894.2.11userData . . . 912
8.895SMSCAddress Struct Reference . . . 912
8.895.1 Detailed Description . . . 913
8.895.2 Field Documentation . . . 913
8.895.2.1 data . . . 913
8.895.2.2 length . . . 913
8.896sMSCAddressInfo Struct Reference . . . 913
8.896.1 Detailed Description . . . 913
8.896.2 Field Documentation . . . 913
8.896.2.1 data . . . 913
8.896.2.2 length . . . 913
8.897sMSCAddressTlv Struct Reference . . . 913
8.897.1 Detailed Description . . . 914
8.897.2 Field Documentation . . . 914
8.897.2.1 SMSCInfo . . . 914
8.897.2.2 TlvPresent . . . 914
8.898SMSEtwsMessage Struct Reference . . . 914
8.898.1 Detailed Description . . . 914
8.898.2 Field Documentation . . . 914
8.898.2.1 data . . . 914
8.898.2.2 length . . . 914
8.898.2.3 notificationType . . . 914
8.899sMSEtwsMessageInfo Struct Reference . . . 915
8.899.1 Detailed Description . . . 915
8.899.2 Field Documentation . . . 915
8.899.2.1 data . . . 915
8.899.2.2 length . . . 915
8.899.2.3 notificationType . . . 915
8.900sMSEtwsMessageTlv Struct Reference . . . 915
8.900.1 Detailed Description . . . 915
8.900.2 Field Documentation . . . 916

8.900.2.1 EtwsMessageInfo . . . 916
8.900.2.2 TlvPresent . . . 916
8.901 SMSEtwsPlmn Struct Reference . . . 916
8.901.1 Detailed Description . . . 916
8.901.2 Field Documentation . . . 916
8.901.2.1 mobileCountryCode . . . 916
8.901.2.2 mobileNetworkCode . . . 916
8.902 sMSEtwsPlmnInfo Struct Reference . . . 916
8.902.1 Detailed Description . . . 916
8.902.2 Field Documentation . . . 917
8.902.2.1 mobileCountryCode . . . 917
8.902.2.2 mobileNetworkCode . . . 917
8.903 SMSEventInfo_s Struct Reference . . . 917
8.903.1 Detailed Description . . . 917
8.903.2 Field Documentation . . . 918
8.903.2.1 pEtwsMessageInfo . . . 918
8.903.2.2 pEtwsPlmnInfo . . . 918
8.903.2.3 pMessageModeInfo . . . 918
8.903.2.4 pMTMessageInfo . . . 918
8.903.2.5 pSMSCAddressInfo . . . 918
8.903.2.6 pSMSOnIMSInfo . . . 918
8.903.2.7 pTransferRouteMTMessageInfo . . . 918
8.903.2.8 smsEventType . . . 918
8.904 smsMaxStorageSizeReq Struct Reference . . . 918
8.904.1 Detailed Description . . . 918
8.904.2 Field Documentation . . . 918
8.904.2.1 pMessageMode . . . 919
8.904.2.2 storageType . . . 919
8.905 smsMaxStorageSizeResp Struct Reference . . . 919
8.905.1 Detailed Description . . . 919
8.905.2 Field Documentation . . . 919
8.905.2.1 freeSlots . . . 919
8.905.2.2 maxStorageSize . . . 919
8.906 SMSMemoryInfo Struct Reference . . . 919
8.906.1 Detailed Description . . . 919
8.906.2 Field Documentation . . . 920
8.906.2.1 messageMode . . . 920
8.906.2.2 storageType . . . 920
8.907 SMSMessageMode Struct Reference . . . 920
8.907.1 Detailed Description . . . 920

8.907.2 Field Documentation . . . 920
8.907.2.1 messageMode . . . 920
8.908 sMSMessageModeInfo Struct Reference . . . 920
8.908.1 Detailed Description . . . 920
8.908.2 Field Documentation . . . 920
8.908.2.1 messageMode . . . 920
8.909 smsMsgprotocolResp Struct Reference . . . 920
8.909.1 Detailed Description . . . 921
8.909.2 Field Documentation . . . 921
8.909.2.1 msgProtocol . . . 921
8.910 SMSMTMessage Struct Reference . . . 921
8.910.1 Detailed Description . . . 921
8.910.2 Field Documentation . . . 921
8.910.2.1 messageIndex . . . 921
8.910.2.2 storageType . . . 921
8.911 sMSMTMessageInfo Struct Reference . . . 921
8.911.1 Detailed Description . . . 922
8.911.2 Field Documentation . . . 922
8.911.2.1 messageIndex . . . 922
8.911.2.2 storageType . . . 922
8.912 SMSOnIMS Struct Reference . . . 922
8.912.1 Detailed Description . . . 922
8.912.2 Field Documentation . . . 922
8.912.2.1 smsOnIMS . . . 922
8.913 sMSOnIMSInfo Struct Reference . . . 922
8.913.1 Detailed Description . . . 922
8.913.2 Field Documentation . . . 923
8.913.2.1 smsOnIMS . . . 923
8.914 sMSOnIMSTlv Struct Reference . . . 923
8.914.1 Detailed Description . . . 923
8.914.2 Field Documentation . . . 923
8.914.2.1 IMSInfo . . . 923
8.914.2.2 TlvPresent . . . 923
8.915 smsRouteEntry Struct Reference . . . 923
8.915.1 Detailed Description . . . 923
8.915.2 Field Documentation . . . 924
8.915.2.1 messageClass . . . 924
8.915.2.2 messageType . . . 924
8.915.2.3 receiptAction . . . 924
8.915.2.4 routeStorage . . . 924

8.916 smsSetRoutesReq Struct Reference . . . 924
8.916.1 Detailed Description . . . 924
8.916.2 Field Documentation . . . 925
8.916.2.1 numOfRoutes . . . 925
8.916.2.2 pTransferStatusReport . . . 925
8.916.2.3 routeList . . . 925
8.917 SMSTransferRouteMTMessage Struct Reference . . . 925
8.917.1 Detailed Description . . . 925
8.917.2 Field Documentation . . . 926
8.917.2.1 ackIndicator . . . 926
8.917.2.2 data . . . 926
8.917.2.3 format . . . 926
8.917.2.4 length . . . 926
8.917.2.5 transactionID . . . 926
8.918 sMSTransferRouteMTMessageInfo Struct Reference . . . 926
8.918.1 Detailed Description . . . 926
8.918.2 Field Documentation . . . 926
8.918.2.1 ackIndicator . . . 926
8.918.2.2 data . . . 926
8.918.2.3 format . . . 926
8.918.2.4 length . . . 927
8.918.2.5 transactionID . . . 927
8.919 sQosFlowStat Struct Reference . . . 927
8.919.1 Detailed Description . . . 927
8.919.2 Field Documentation . . . 927
8.919.2.1 bearerId . . . 927
8.919.2.2 tx_bytes . . . 927
8.919.2.3 tx_bytes_drp . . . 927
8.919.2.4 tx_pkt . . . 927
8.919.2.5 tx_pkt_drp . . . 927
8.920 sQosStat Struct Reference . . . 927
8.920.1 Detailed Description . . . 928
8.920.2 Field Documentation . . . 928
8.920.2.1 apnId . . . 928
8.920.2.2 numQosFlow . . . 928
8.920.2.3 qosFlow . . . 928
8.920.2.4 total_rx_bytes . . . 929
8.920.2.5 total_rx_pkt . . . 929
8.920.2.6 total_tx_bytes . . . 929
8.920.2.7 total_tx_bytes_drp . . . 929

8.920.2.8 total_tx_pkt . . . 929
8.920.2.9 total_tx_pkt_drp . . . 929
8.921 SrvStatusInfo Struct Reference . . . 929
8.921.1 Detailed Description . . . 929
8.921.2 Field Documentation . . . 929
8.921.2.1 isPrefDataPath . . . 929
8.921.2.2 srvStatus . . . 929
8.922 ssdatasession_params Struct Reference . . . 929
8.922.1 Detailed Description . . . 930
8.922.2 Field Documentation . . . 931
8.922.2.1 action . . . 931
8.922.2.2 failureReason . . . 932
8.922.2.3 failureReasonv4 . . . 932
8.922.2.4 failureReasonv6 . . . 932
8.922.2.5 instanceId . . . 932
8.922.2.6 ipfamily . . . 932
8.922.2.7 pAuthentication . . . 932
8.922.2.8 pPassword . . . 932
8.922.2.9 pProfileId3GPP . . . 932
8.922.2.10 pProfileId3GPP2 . . . 932
8.922.2.11 pTechnology . . . 932
8.922.2.12 pUsername . . . 932
8.922.2.13 rcv4 . . . 932
8.922.2.14 rcv6 . . . 932
8.922.2.15 sessionId . . . 932
8.922.2.16 v4sessionId . . . 932
8.922.2.17 v6sessionId . . . 932
8.922.2.18 verbFailReason . . . 932
8.922.2.19 verbFailReasonType . . . 932
8.923 SupportedMsgList Struct Reference . . . 932
8.923.1 Detailed Description . . . 932
8.923.2 Field Documentation . . . 933
8.923.2.1 supportedMsgLen . . . 933
8.923.2.2 supportedMsgs . . . 933
8.924 SUPSInfo Struct Reference . . . 933
8.924.1 Detailed Description . . . 933
8.924.2 Field Documentation . . . 933
8.924.2.1 isModByCC . . . 933
8.924.2.2 svcType . . . 933
8.925 SV Struct Reference . . . 933

8.925.1 Detailed Description . . . 934
8.925.2 Field Documentation . . . 934
8.925.2.1 id . . . 934
8.925.2.2 mask . . . 934
8.925.2.3 system . . . 934
8.926 SVInfo Struct Reference . . . 934
8.926.1 Detailed Description . . . 934
8.926.2 Field Documentation . . . 935
8.926.2.1 len . . . 935
8.926.2.2 pSV . . . 935
8.927 svUsedforFix_s Struct Reference . . . 935
8.927.1 Detailed Description . . . 935
8.927.2 Field Documentation . . . 935
8.927.2.1 gnssSvUsedList . . . 935
8.927.2.2 gnssSvUsedList_len . . . 935
8.928 SWI_STRUCT_CarrierImage Struct Reference . . . 935
8.928.1 Detailed Description . . . 936
8.928.2 Field Documentation . . . 936
8.928.2.1 m_FwBuildId . . . 936
8.928.2.2 m_FwImageId . . . 936
8.928.2.3 m_nCarrierId . . . 936
8.928.2.4 m_nFolderId . . . 936
8.928.2.5 m_nStorage . . . 936
8.928.2.6 m_PriBuildId . . . 937
8.928.2.7 m_PriImageId . . . 937
8.929 SwiLocGetAutoStartResp Struct Reference . . . 937
8.929.1 Detailed Description . . . 937
8.929.2 Field Documentation . . . 938
8.929.2.1 fix_rate . . . 938
8.929.2.2 fix_rate_reported . . . 938
8.929.2.3 fix_type . . . 938
8.929.2.4 fix_type_reported . . . 938
8.929.2.5 function . . . 938
8.929.2.6 function_reported . . . 938
8.929.2.7 max_dist . . . 938
8.929.2.8 max_dist_reported . . . 938
8.929.2.9 max_time . . . 938
8.929.2.10 max_time_reported . . . 938
8.930 SwiLocSetAutoStartReq Struct Reference . . . 938
8.930.1 Detailed Description . . . 939

8.930.2 Field Documentation . . . 939
8.930.2.1 fix_rate . . . 940
8.930.2.2 fix_type . . . 940
8.930.2.3 function . . . 940
8.930.2.4 max_dist . . . 940
8.930.2.5 max_time . . . 940
8.930.2.6 set_fix_rate . . . 940
8.930.2.7 set_fix_type . . . 940
8.930.2.8 set_function . . . 940
8.930.2.9 set_max_dist . . . 940
8.930.2.10 set_max_time . . . 940
8.931 swiModemStatusResp Struct Reference . . . 940
8.931.1 Detailed Description . . . 940
8.931.2 Field Documentation . . . 940
8.931.2.1 commonInfo . . . 940
8.931.2.2 pLTEInfo . . . 940
8.932 SwiOTAMsg_s Struct Reference . . . 940
8.932.1 Detailed Description . . . 941
8.932.2 Field Documentation . . . 941
8.932.2.1 data . . . 941
8.932.2.2 data_len . . . 941
8.932.2.3 pLteNasRelInfo . . . 941
8.932.2.4 pTime . . . 941
8.932.2.5 type . . . 941
8.933 swiPDPRuntimeSettingsReq Struct Reference . . . 941
8.933.1 Detailed Description . . . 941
8.933.2 Field Documentation . . . 942
8.933.2.1 contextId . . . 942
8.933.2.2 contextType . . . 942
8.934 swiPDPRuntimeSettingsResp Struct Reference . . . 942
8.934.1 Detailed Description . . . 942
8.934.2 Field Documentation . . . 944
8.934.2.1 pAPNName . . . 944
8.934.2.2 pBearerId . . . 944
8.934.2.3 pContextId . . . 944
8.934.2.4 pIPv4Address . . . 944
8.934.2.5 pIPv4GWAddress . . . 944
8.934.2.6 pIPv6Address . . . 944
8.934.2.7 pIPv6GWAddress . . . 944
8.934.2.8 pPrDNSIPv4Address . . . 944

8.934.2.9 pPrDNSIPv6Address . . . 944
8.934.2.10 pPrPCSCFIPv4Address . . . 944
8.934.2.11 pPrPCSCFIPv6Address . . . 944
8.934.2.12 pSeDNSIPv4Address . . . 944
8.934.2.13 pSeDNSIPv6Address . . . 944
8.934.2.14 pSePCSCFIPv4Address . . . 944
8.934.2.15 pSePCSCFIPv6Address . . . 944
8.935 swiQosFilter Struct Reference . . . 944
8.935.1 Detailed Description . . . 945
8.935.2 Field Documentation . . . 946
8.935.2.1 index . . . 946
8.935.2.2 pEspSpi . . . 946
8.935.2.3 pId . . . 946
8.935.2.4 pIPv4DstAddr . . . 946
8.935.2.5 pIPv4SrcAddr . . . 946
8.935.2.6 pIPv6DstAddr . . . 946
8.935.2.7 pIPv6Label . . . 946
8.935.2.8 pIPv6SrcAddr . . . 946
8.935.2.9 pIPv6TrafCls . . . 946
8.935.2.10 pNxtHdrProto . . . 946
8.935.2.11 pPrecedence . . . 946
8.935.2.12 pTCPDstPort . . . 946
8.935.2.13 pTCPSrcPort . . . 946
8.935.2.14 pTos . . . 946
8.935.2.15 pTranDstPort . . . 946
8.935.2.16 pTranSrcPort . . . 946
8.935.2.17 pUDPDstPort . . . 946
8.935.2.18 pUDPSrcPort . . . 947
8.935.2.19 version . . . 947
8.936 swiQosFlow Struct Reference . . . 947
8.936.1 Detailed Description . . . 947
8.936.2 Field Documentation . . . 949
8.936.2.1 index . . . 949
8.936.2.2 p3GPP2Pri . . . 949
8.936.2.3 p3GPPImCn . . . 949
8.936.2.4 p3GPPResResidualBER . . . 949
8.936.2.5 p3GPPSigInd . . . 949
8.936.2.6 p3GPPTraHdlPri . . . 949
8.936.2.7 pDataRate . . . 949
8.936.2.8 pJitter . . . 949

8.936.2.9 pLatency . . . 949
8.936.2.10 pLteQci . . . 949
8.936.2.11 pMaxAllowedPktSz . . . 950
8.936.2.12 pMinPolicedPktSz . . . 950
8.936.2.13 pPktErrRate . . . 950
8.936.2.14 pProfileId3GPP2 . . . 950
8.936.2.15 pTokenBucket . . . 950
8.936.2.16 pTrafficClass . . . 950
8.937 swiQosGranted Struct Reference . . . 950
8.937.1 Detailed Description . . . 950
8.937.2 Field Documentation . . . 950
8.937.2.1 pRxFlow . . . 950
8.937.2.2 pTxFlow . . . 950
8.938 swiQosIds Struct Reference . . . 950
8.938.1 Detailed Description . . . 950
8.938.2 Field Documentation . . . 950
8.938.2.1 pIds . . . 951
8.938.2.2 sz . . . 951
8.939 swiQosModifyReq Struct Reference . . . 951
8.939.1 Detailed Description . . . 951
8.939.2 Field Documentation . . . 951
8.939.2.1 id . . . 951
8.939.2.2 pRxFilter . . . 951
8.939.2.3 pRxFlow . . . 951
8.939.2.4 pTxFilter . . . 951
8.939.2.5 pTxFlow . . . 951
8.940 swiQosReq Struct Reference . . . 951
8.940.1 Detailed Description . . . 951
8.940.2 Field Documentation . . . 952
8.940.2.1 index . . . 952
8.940.2.2 pRxFilter . . . 952
8.940.2.3 pRxFlow . . . 952
8.940.2.4 pTxFilter . . . 952
8.940.2.5 pTxFlow . . . 952
8.941 swiRMTrasnferStaticsReq Struct Reference . . . 952
8.941.1 Detailed Description . . . 952
8.941.2 Field Documentation . . . 953
8.941.2.1 bResetStatistics . . . 953
8.941.2.2 ulMask . . . 953
8.942 sysInfoCommon Struct Reference . . . 953

8.942.1 Detailed Description . . . 953
8.942.2 Field Documentation . . . 955
8.942.2.1 isSysForbidden . . . 955
8.942.2.2 isSysForbiddenValid . . . 955
8.942.2.3 roamStatus . . . 955
8.942.2.4 roamStatusValid . . . 955
8.942.2.5 srvCapability . . . 955
8.942.2.6 srvCapabilityValid . . . 955
8.942.2.7 srvDomain . . . 955
8.942.2.8 srvDomainValid . . . 955
8.943t_gpsTime Struct Reference . . . 955
8.943.1 Field Documentation . . . 955
8.943.1.1 gpsTimeOfWeekMs . . . 955
8.943.1.2 gpsWeek . . . 955
8.944t_sensor Struct Reference . . . 955
8.944.1 Field Documentation . . . 955
8.944.1.1 aidingIndicatorMask . . . 955
8.944.1.2 usageMask . . . 955
8.945t_Sv Struct Reference . . . 956
8.945.1 Field Documentation . . . 956
8.945.1.1 entries . . . 956
8.945.1.2 len . . . 956
8.946TDSCDMAECIOThresh Struct Reference . . . 956
8.946.1 Detailed Description . . . 956
8.946.2 Field Documentation . . . 956
8.946.2.1 pTDSCDMAECIOThreshList . . . 956
8.946.2.2 TDSCDMAECIOThreshListLen . . . 956
8.947TDSCDMARSCPThresh Struct Reference . . . 956
8.947.1 Detailed Description . . . 956
8.947.2 Field Documentation . . . 957
8.947.2.1 pTDSCDMARSCPThreshList . . . 957
8.947.2.2 TDSCDMARSCPThreshListLen . . . 957
8.948TDSCDMARSSIThresh Struct Reference . . . 957
8.948.1 Detailed Description . . . 957
8.948.2 Field Documentation . . . 957
8.948.2.1 pTDSCDMARSSIThreshList . . . 957
8.948.2.2 TDSCDMARSSIThreshListLen . . . 957
8.949TDSCDMASigInfoExt Struct Reference . . . 957
8.949.1 Detailed Description . . . 958
8.949.2 Field Documentation . . . 958

8.949.2.1 ecio . . . 958
8.949.2.2 rscp . . . 958
8.949.2.3 rssi . . . 958
8.949.2.4 sinr . . . 958
8.950 tdscdmaSigInfoExt Struct Reference . . . 958
8.950.1 Detailed Description . . . 958
8.950.2 Field Documentation . . . 959
8.950.2.1 ecio . . . 959
8.950.2.2 rscp . . . 959
8.950.2.3 rssi . . . 959
8.950.2.4 sinr . . . 959
8.951 TDSCDMASINRCONFThresh Struct Reference . . . 959
8.951.1 Detailed Description . . . 959
8.951.2 Field Documentation . . . 959
8.951.2.1 pTDSCDMASINRCONFThreshList . . . 959
8.951.2.2 TDSCDMASINRCONFThreshListLen . . . 959
8.952 TDSCDMASINRThresh Struct Reference . . . 959
8.952.1 Detailed Description . . . 959
8.952.2 Field Documentation . . . 960
8.952.2.1 pTDSCDMASINRThreshList . . . 960
8.952.2.2 TDSCDMASINRThreshListLen . . . 960
8.953 tempData_t Struct Reference . . . 960
8.953.1 Detailed Description . . . 960
8.953.2 Field Documentation . . . 961
8.953.2.1 temperature . . . 961
8.953.2.2 temperatureDataLen . . . 961
8.953.2.3 timeOfFirstSample . . . 961
8.953.2.4 timeOffset . . . 961
8.953.2.5 timeSource . . . 961
8.954 tempratureData Struct Reference . . . 961
8.954.1 Detailed Description . . . 961
8.954.2 Field Documentation . . . 962
8.954.2.1 temperature . . . 962
8.954.2.2 temperatureDataLen . . . 962
8.954.2.3 timeOfFirstSample . . . 962
8.954.2.4 timeOffset . . . 962
8.954.2.5 timeSource . . . 962
8.955 TFTIDParams Struct Reference . . . 962
8.955.1 Detailed Description . . . 962
8.955.2 Field Documentation . . . 963

8.955.2.1 destPortRangeEnd . . . 963
8.955.2.2 destPortRangeStart . . . 963
8.955.2.3 eValid . . . 963
8.955.2.4 filterId . . . 964
8.955.2.5 flowLabel . . . 964
8.955.2.6 IPSECSPI . . . 964
8.955.2.7 ipVersion . . . 964
8.955.2.8 nextHeader . . . 964
8.955.2.9 pSourceIP . . . 964
8.955.2.10 sourceIPMask . . . 964
8.955.2.11 srcPortRangeEnd . . . 964
8.955.2.12 srcPortRangeStart . . . 964
8.955.2.13 tosMask . . . 964
8.956 timeInfo Struct Reference . . . 964
8.956.1 Detailed Description . . . 964
8.956.2 Field Documentation . . . 965
8.956.2.1 day . . . 965
8.956.2.2 dayLtSavingAdj . . . 965
8.956.2.3 dayOfWeek . . . 965
8.956.2.4 hour . . . 965
8.956.2.5 minute . . . 965
8.956.2.6 month . . . 966
8.956.2.7 radioInterface . . . 966
8.956.2.8 second . . . 966
8.956.2.9 timeZone . . . 966
8.956.2.10 TlvPresent . . . 966
8.956.2.11 year . . . 966
8.957 TmdDeRegNotMitigationLvlReq Struct Reference . . . 966
8.957.1 Detailed Description . . . 966
8.957.2 Field Documentation . . . 966
8.957.2.1 mitigationDevID . . . 966
8.957.2.2 mitigationDevIDLen . . . 966
8.958 TmdGetMitigationDevListResp Struct Reference . . . 966
8.958.1 Detailed Description . . . 966
8.958.2 Field Documentation . . . 967
8.958.2.1 pMitigationDevList . . . 967
8.958.2.2 pMitigationDevListLen . . . 967
8.959 TmdGetMitigationLvlReq Struct Reference . . . 967
8.959.1 Detailed Description . . . 967
8.959.2 Field Documentation . . . 967

8.959.2.1 mitigationDevID . . . 967
8.959.2.2 mitigationDevIDLen . . . 967
8.960 TmdGetMitigationLvlResp Struct Reference . . . 967
8.960.1 Detailed Description . . . 968
8.960.2 Field Documentation . . . 968
8.960.2.1 pCurrentmitigationLvl . . . 968
8.960.2.2 pReqMitigationLvl . . . 968
8.961 TmdMitigationLvlIndReq Struct Reference . . . 968
8.961.1 Detailed Description . . . 968
8.961.2 Field Documentation . . . 968
8.961.2.1 mitigationDevID . . . 968
8.961.2.2 mitigationDevIDLen . . . 968
8.962 TmdRegNotMitigationLvlReq Struct Reference . . . 968
8.962.1 Detailed Description . . . 969
8.962.2 Field Documentation . . . 969
8.962.2.1 mitigationDevID . . . 969
8.962.2.2 mitigationDevIDLen . . . 969
8.963 tokenBucket Struct Reference . . . 969
8.963.1 Detailed Description . . . 969
8.963.2 Field Documentation . . . 969
8.963.2.1 bucketSz . . . 969
8.963.2.2 peakRate . . . 969
8.963.2.3 tokenRate . . . 969
8.964 Tos Struct Reference . . . 969
8.964.1 Detailed Description . . . 970
8.964.2 Field Documentation . . . 970
8.964.2.1 mask . . . 970
8.964.2.2 val . . . 970
8.965 transferRouteMessageTlv Struct Reference . . . 970
8.965.1 Detailed Description . . . 970
8.965.2 Field Documentation . . . 970
8.965.2.1 TlvPresent . . . 970
8.965.2.2 TransferRouteMTMessageInfo . . . 971
8.966 TransferStatInd Struct Reference . . . 971
8.966.1 Detailed Description . . . 971
8.966.2 Field Documentation . . . 971
8.966.2.1 StatsMask . . . 971
8.966.2.2 StatsPeriod . . . 971
8.967 TransferStatsDataType Struct Reference . . . 971
8.967.1 Field Documentation . . . 971

8.967.1.1 interval . . . 971
8.968 TrStatInd Struct Reference . . . 971
8.968.1 Detailed Description . . . 972
8.968.2 Field Documentation . . . 972
8.968.2.1 statsMask . . . 972
8.968.2.2 statsPeriod . . . 972
8.969 trueIMSI Struct Reference . . . 972
8.969.1 Detailed Description . . . 972
8.969.2 Field Documentation . . . 973
8.969.2.1 imsiT1112 . . . 973
8.969.2.2 imsiTaddrNum . . . 973
8.969.2.3 imsiTS1 . . . 973
8.969.2.4 imsiTS2 . . . 973
8.969.2.5 mccT . . . 973
8.970 TXAGCList Struct Reference . . . 973
8.970.1 Detailed Description . . . 973
8.970.2 Field Documentation . . . 974
8.970.2.1 pTXAIG . . . 974
8.970.2.2 pTXComprSlope . . . 974
8.970.2.3 pTXComprThres . . . 974
8.970.2.4 pTXExpSlope . . . 974
8.970.2.5 pTXExpThres . . . 974
8.970.2.6 pTXStaticGain . . . 974
8.971 txInfo Struct Reference . . . 974
8.971.1 Detailed Description . . . 974
8.971.2 Field Documentation . . . 974
8.971.2.1 isInTraffic . . . 974
8.971.2.2 txPower . . . 975
8.972 TXPCMIIRFltr Struct Reference . . . 975
8.972.1 Detailed Description . . . 975
8.972.2 Field Documentation . . . 976
8.972.2.1 pFlag . . . 976
8.972.2.2 pStage0Val . . . 976
8.972.2.3 pStage1Val . . . 976
8.972.2.4 pStage2Val . . . 976
8.972.2.5 pStage3Val . . . 976
8.972.2.6 pStage4Val . . . 976
8.972.2.7 pStageCnt . . . 976
8.973 uim_appStatus Struct Reference . . . 976
8.973.1 Detailed Description . . . 977

8.973.2 Field Documentation . . . 978
8.973.2.1 aidLength . . . 979
8.973.2.2 aidVal . . . 979
8.973.2.3 appState . . . 979
8.973.2.4 appType . . . 979
8.973.2.5 persoFeature . . . 979
8.973.2.6 persoRetries . . . 979
8.973.2.7 persoState . . . 979
8.973.2.8 persoUnblockRetries . . . 979
8.973.2.9 pin1Retries . . . 979
8.973.2.10 pin1State . . . 979
8.973.2.11 pin2Retries . . . 979
8.973.2.12 pin2State . . . 979
8.973.2.13 puk1Retries . . . 979
8.973.2.14 puk2Retries . . . 979
8.973.2.15 univPin . . . 979
8.974 uim_authenticateResult Struct Reference . . . 979
8.974.1 Detailed Description . . . 979
8.974.2 Field Documentation . . . 979
8.974.2.1 content . . . 979
8.974.2.2 contentLen . . . 980
8.975 uim_authenticationData Struct Reference . . . 980
8.975.1 Detailed Description . . . 980
8.975.2 Field Documentation . . . 980
8.975.2.1 context . . . 980
8.975.2.2 data . . . 980
8.975.2.3 dataLen . . . 981
8.976 uim_cardResult Struct Reference . . . 981
8.976.1 Detailed Description . . . 981
8.976.2 Field Documentation . . . 981
8.976.2.1 sw1 . . . 981
8.976.2.2 sw2 . . . 981
8.977 uim_cardStatus Struct Reference . . . 981
8.977.1 Detailed Description . . . 981
8.977.2 Field Documentation . . . 982
8.977.2.1 index1xPri . . . 982
8.977.2.2 index1xSec . . . 982
8.977.2.3 indexGwPri . . . 982
8.977.2.4 indexGwSec . . . 982
8.977.2.5 numSlot . . . 982

8.977.2.6 SlotInfo . . . 982
8.978 uim_changeUIMPIN Struct Reference . . . 982
8.978.1 Detailed Description . . . 983
8.978.2 Field Documentation . . . 983
8.978.2.1 oldPINLen . . . 983
8.978.2.2 oldPINVal . . . 983
8.978.2.3 pinID . . . 983
8.978.2.4 pinLen . . . 983
8.978.2.5 pinVal . . . 983
8.979 uim_depersonalizationInformation Struct Reference . . . 983
8.979.1 Detailed Description . . . 983
8.979.2 Field Documentation . . . 984
8.979.2.1 ckLen . . . 984
8.979.2.2 ckVal . . . 984
8.979.2.3 feature . . . 984
8.979.2.4 operation . . . 984
8.980 uim_encryptedPIN1 Struct Reference . . . 984
8.980.1 Detailed Description . . . 984
8.980.2 Field Documentation . . . 985
8.980.2.1 pin1Len . . . 985
8.980.2.2 pin1Val . . . 985
8.981 uim_fileAttributes Struct Reference . . . 985
8.981.1 Detailed Description . . . 985
8.981.2 Field Documentation . . . 988
8.981.2.1 fileID . . . 988
8.981.2.2 fileSize . . . 988
8.981.2.3 fileType . . . 988
8.981.2.4 rawLen . . . 988
8.981.2.5 rawValue . . . 988
8.981.2.6 recordCount . . . 988
8.981.2.7 recordSize . . . 988
8.981.2.8 secActivate . . . 988
8.981.2.9 secActivateMask . . . 988
8.981.2.10 secDeactivate . . . 988
8.981.2.11 secDeactivateMask . . . 988
8.981.2.12 secIncrease . . . 988
8.981.2.13 secIncreaseMask . . . 988
8.981.2.14 secRead . . . 988
8.981.2.15 secReadMask . . . 988
8.981.2.16 secWrite . . . 988

8.981.2.17 secWriteMask . . . 988
8.982 uim_fileInfo Struct Reference . . . 988
8.982.1 Detailed Description . . . 988
8.982.2 Field Documentation . . . 989
8.982.2.1 fileID . . . 989
8.982.2.2 path . . . 989
8.982.2.3 pathLen . . . 989
8.983 uim_hotSwapStatus Struct Reference . . . 989
8.983.1 Detailed Description . . . 989
8.983.2 Field Documentation . . . 989
8.983.2.1 hotSwap . . . 989
8.983.2.2 hotSwapLength . . . 989
8.984 uim_personalizationStatus Struct Reference . . . 989
8.984.1 Detailed Description . . . 990
8.984.2 Field Documentation . . . 990
8.984.2.1 feature . . . 990
8.984.2.2 numFeatures . . . 990
8.984.2.3 unblockLeft . . . 990
8.984.2.4 verifyLeft . . . 990
8.985 uim_readResult Struct Reference . . . 990
8.985.1 Detailed Description . . . 991
8.985.2 Field Documentation . . . 991
8.985.2.1 content . . . 991
8.985.2.2 contentLen . . . 991
8.986 uim_readTransparentInfo Struct Reference . . . 991
8.986.1 Detailed Description . . . 991
8.986.2 Field Documentation . . . 991
8.986.2.1 length . . . 991
8.986.2.2 offset . . . 991
8.987 uim_refreshevent Struct Reference . . . 991
8.987.1 Detailed Description . . . 992
8.987.2 Field Documentation . . . 993
8.987.2.1 aid . . . 993
8.987.2.2 aidLength . . . 993
8.987.2.3 arrfileInfo . . . 993
8.987.2.4 mode . . . 993
8.987.2.5 numOfFiles . . . 993
8.987.2.6 sessionType . . . 993
8.987.2.7 stage . . . 993
8.988 uim_registerRefresh Struct Reference . . . 993

8.988.1 Detailed Description . . . 993
8.988.2 Field Documentation . . . 994
8.988.2.1 arrfileInfo . . . 994
8.988.2.2 numFiles . . . 994
8.988.2.3 registerFlag . . . 994
8.988.2.4 voteForInit . . . 994
8.989uim_remainingRetries Struct Reference . . . 994
8.989.1 Detailed Description . . . 994
8.989.2 Field Documentation . . . 994
8.989.2.1 unblockLeft . . . 994
8.989.2.2 verifyLeft . . . 995
8.990uim_sessionInformation Struct Reference . . . 995
8.990.1 Detailed Description . . . 995
8.990.2 Field Documentation . . . 995
8.990.2.1 aid . . . 995
8.990.2.2 aidLength . . . 995
8.990.2.3 sessionType . . . 995
8.991uim_setPINProtection Struct Reference . . . 995
8.991.1 Detailed Description . . . 996
8.991.2 Field Documentation . . . 996
8.991.2.1 pinID . . . 996
8.991.2.2 pinLength . . . 996
8.991.2.3 pinOperation . . . 996
8.991.2.4 pinValue . . . 996
8.992uim_slotInfo Struct Reference . . . 996
8.992.1 Detailed Description . . . 997
8.992.2 Field Documentation . . . 998
8.992.2.1 AppStatus . . . 998
8.992.2.2 cardState . . . 998
8.992.2.3 errorState . . . 998
8.992.2.4 numApp . . . 998
8.992.2.5 upinRetries . . . 998
8.992.2.6 upinState . . . 998
8.992.2.7 upukRetries . . . 998
8.993uim_UIMSessionInformation Struct Reference . . . 998
8.993.1 Detailed Description . . . 998
8.993.2 Field Documentation . . . 999
8.993.2.1 aid . . . 999
8.993.2.2 aidLength . . . 999
8.993.2.3 sessionType . . . 999

8.994 uim_unblockUIMPIN Struct Reference . . . 999
8.994.1 Detailed Description . . . 999
8.994.2 Field Documentation . . . 999
8.994.2.1 newPINLen . . . 999
8.994.2.2 newPINVal . . . 999
8.994.2.3 pinID . . . 999
8.994.2.4 pukLen . . . 1000
8.994.2.5 pukVal . . . 1000
8.995 uim_verifyUIMPIN Struct Reference . . . 1000
8.995.1 Detailed Description . . . 1000
8.995.2 Field Documentation . . . 1000
8.995.2.1 pinID . . . 1000
8.995.2.2 pinLen . . . 1000
8.995.2.3 pinVal . . . 1000
8.996 UIMAuthenticateReq Struct Reference . . . 1000
8.996.1 Detailed Description . . . 1000
8.996.2 Field Documentation . . . 1001
8.996.2.1 authData . . . 1001
8.996.2.2 pIndicationToken . . . 1001
8.996.2.3 sessionInfo . . . 1001
8.997 UIMAuthenticateResp Struct Reference . . . 1001
8.997.1 Detailed Description . . . 1001
8.997.2 Field Documentation . . . 1002
8.997.2.1 pAuthenticateResult . . . 1002
8.997.2.2 pCardResult . . . 1002
8.997.2.3 pIndicationToken . . . 1002
8.998 UIMChangePinReq Struct Reference . . . 1002
8.998.1 Detailed Description . . . 1002
8.998.2 Field Documentation . . . 1002
8.998.2.1 changePIN . . . 1002
8.998.2.2 pIndicationToken . . . 1002
8.998.2.3 pKeyReferenceID . . . 1002
8.998.2.4 sessionInfo . . . 1002
8.999 UIMDepersonalizationReq Struct Reference . . . 1003
8.999.1 Detailed Description . . . 1003
8.999.2 Field Documentation . . . 1003
8.999.2.1 depersonilisationInfo . . . 1003
8.1000 UIMDepersonalizationResp Struct Reference . . . 1003
8.1000.1 Detailed Description . . . 1003
8.1000.2 Field Documentation . . . 1003

8.1000.2.1 pRemainingRetries . . . 1003
8.1001 UIMEventRegisterReqResp Struct Reference . . . 1003
8.1001.1 Detailed Description . . . 1004
8.1001.2 Field Documentation . . . 1004
8.1001.2.1 eventMask . . . 1004
8.1002 UIMGetCardStatusResp Struct Reference . . . 1004
8.1002.1 Detailed Description . . . 1004
8.1002.2 Field Documentation . . . 1004
8.1002.2.1 pCardStatus . . . 1004
8.1002.2.2 pHotSwapStatus . . . 1004
8.1003 UIMGetConfigurationReq Struct Reference . . . 1004
8.1003.1 Detailed Description . . . 1005
8.1003.2 Field Documentation . . . 1005
8.1003.2.1 pConfigurationMask . . . 1005
8.1004 UIMGetConfigurationResp Struct Reference . . . 1005
8.1004.1 Detailed Description . . . 1005
8.1004.2 Field Documentation . . . 1006
8.1004.2.1 pAutoSelection . . . 1006
8.1004.2.2 pHaltSubscription . . . 1006
8.1004.2.3 pPersonalizationStatus . . . 1006
8.1005 UIMGetFileAttributesReq Struct Reference . . . 1006
8.1005.1 Detailed Description . . . 1006
8.1005.2 Field Documentation . . . 1006
8.1005.2.1 fileIndex . . . 1006
8.1005.2.2 pIndicationToken . . . 1006
8.1005.2.3 sessionInfo . . . 1006
8.1006 UIMGetFileAttributesResp Struct Reference . . . 1006
8.1006.1 Detailed Description . . . 1007
8.1006.2 Field Documentation . . . 1007
8.1006.2.1 pCardResult . . . 1007
8.1006.2.2 pFileAttributes . . . 1007
8.1006.2.3 pIndicationToken . . . 1007
8.1007 UIMGetSlotsStatusResp Struct Reference . . . 1007
8.1007.1 Detailed Description . . . 1007
8.1007.2 Field Documentation . . . 1007
8.1007.2.1 pNumberOfPhySlot . . . 1007
8.1007.2.2 pUimSlotsStatus . . . 1008
8.1008 UIMPinResp Struct Reference . . . 1008
8.1008.1 Detailed Description . . . 1008
8.1008.2 Field Documentation . . . 1008

8.1008.2.1pEncryptedPIN1 . . . 1008
8.1008.2.2pIndicationToken . . . 1008
8.1008.2.3pRemainingRetries . . . 1008
8.1009UIMPowerDownReq Struct Reference . . . 1008
8.1009.1Detailed Description . . . 1008
8.1009.2Field Documentation . . . 1009
8.1009.2.1slot . . . 1009
8.1010UIMPowerUpReq Struct Reference . . . 1009
8.1010.1Detailed Description . . . 1009
8.1010.2Field Documentation . . . 1009
8.1010.2.1pIgnoreHotSwapSwitch . . . 1009
8.1010.2.2slot . . . 1009
8.1011UIMReadTransparentReq Struct Reference . . . 1009
8.1011.1Detailed Description . . . 1010
8.1011.2Field Documentation . . . 1010
8.1011.2.1fileIndex . . . 1010
8.1011.2.2pEncryptData . . . 1010
8.1011.2.3pIndicationToken . . . 1010
8.1011.2.4readTransparent . . . 1010
8.1011.2.5sessionInfo . . . 1010
8.1012UIMReadTransparentResp Struct Reference . . . 1010
8.1012.1Detailed Description . . . 1010
8.1012.2Field Documentation . . . 1011
8.1012.2.1pCardResult . . . 1011
8.1012.2.2pEncryptedData . . . 1011
8.1012.2.3pIndicationToken . . . 1011
8.1012.2.4pReadResult . . . 1011
8.1013UIMRefreshCompleteReq Struct Reference . . . 1011
8.1013.1Detailed Description . . . 1011
8.1013.2Field Documentation . . . 1012
8.1013.2.1refreshComplete . . . 1012
8.1013.2.2sessionInfo . . . 1012
8.1014UIMRefreshEvent Struct Reference . . . 1012
8.1014.1Detailed Description . . . 1012
8.1014.2Field Documentation . . . 1013
8.1014.2.1aid . . . 1013
8.1014.2.2aidLength . . . 1013
8.1014.2.3arrfileInfo . . . 1013
8.1014.2.4mode . . . 1013
8.1014.2.5numOfFiles . . . 1013

8.1014.2.6 sessionType . . . 1013
8.1014.2.7 stage . . . 1013
8.1015 UIMRefreshGetLastEventReq Struct Reference . . . 1013
8.1015.1 Detailed Description . . . 1014
8.1015.2 Field Documentation . . . 1014
8.1015.2.1 sessionInfo . . . 1014
8.1016 UIMRefreshGetLastEventResp Struct Reference . . . 1014
8.1016.1 Detailed Description . . . 1014
8.1016.2 Field Documentation . . . 1014
8.1016.2.1 pRefreshEvent . . . 1014
8.1017 UIMRefreshOKReq Struct Reference . . . 1014
8.1017.1 Detailed Description . . . 1014
8.1017.2 Field Documentation . . . 1015
8.1017.2.1 OKtoRefresh . . . 1015
8.1017.2.2 sessionInfo . . . 1015
8.1018 UIMRefreshRegisterReq Struct Reference . . . 1015
8.1018.1 Detailed Description . . . 1015
8.1018.2 Field Documentation . . . 1015
8.1018.2.1 regRefresh . . . 1015
8.1018.2.2 sessionInfo . . . 1015
8.1019 UIMSessionInformation Struct Reference . . . 1015
8.1019.1 Detailed Description . . . 1016
8.1019.2 Field Documentation . . . 1016
8.1019.2.1 aid . . . 1016
8.1019.2.2 aidLength . . . 1016
8.1019.2.3 sessionType . . . 1016
8.1020 UIMSetPinProtectionReq Struct Reference . . . 1016
8.1020.1 Detailed Description . . . 1016
8.1020.2 Field Documentation . . . 1017
8.1020.2.1 pIndicationToken . . . 1017
8.1020.2.2 pinProtection . . . 1017
8.1020.2.3 pKeyReferenceID . . . 1017
8.1020.2.4 sessionInfo . . . 1017
8.1021 UIMSlotsStatus Struct Reference . . . 1017
8.1021.1 Detailed Description . . . 1017
8.1021.2 Field Documentation . . . 1017
8.1021.2.1 uimSlotStatus . . . 1017
8.1022 UIMSlotStatus Struct Reference . . . 1017
8.1022.1 Detailed Description . . . 1018
8.1022.2 Field Documentation . . . 1018

8.1022.2.1 bICCID . . . 1018
8.1022.2.2 bICCIDLength . . . 1018
8.1022.2.3 bLogicalSlot . . . 1018
8.1022.2.4 uPhyCardStatus . . . 1018
8.1022.2.5 uPhySlotStatus . . . 1018
8.1023 UIMSlotStatusChangeInfo Struct Reference . . . 1019
8.1023.1 Detailed Description . . . 1019
8.1023.2 Field Documentation . . . 1019
8.1023.2.1 bNumberOfPhySlots . . . 1019
8.1023.2.2 slotsstatusChange . . . 1019
8.1024 UIMStatusChangeInfo Struct Reference . . . 1019
8.1024.1 Detailed Description . . . 1019
8.1024.2 Field Documentation . . . 1019
8.1024.2.1 statusChange . . . 1019
8.1025 UIMSwitchSlotReq Struct Reference . . . 1019
8.1025.1 Detailed Description . . . 1020
8.1025.2 Field Documentation . . . 1020
8.1025.2.1 bLogicalSlot . . . 1020
8.1025.2.2 ulPhysicalSlot . . . 1020
8.1026 UIMUnblockPinReq Struct Reference . . . 1020
8.1026.1 Detailed Description . . . 1020
8.1026.2 Field Documentation . . . 1021
8.1026.2.1 pIndicationToken . . . 1021
8.1026.2.2 pKeyReferenceID . . . 1021
8.1026.2.3 sessionInfo . . . 1021
8.1026.2.4 unblockPIN . . . 1021
8.1027 UIMVerifyPinReq Struct Reference . . . 1021
8.1027.1 Detailed Description . . . 1021
8.1027.2 Field Documentation . . . 1022
8.1027.2.1 pEncryptedPIN1 . . . 1022
8.1027.2.2 pIndicationToken . . . 1022
8.1027.2.3 pKeyReferenceID . . . 1022
8.1027.2.4 sessionInfo . . . 1022
8.1027.2.5 verifyPIN . . . 1022
8.1028 UMTSInfo Struct Reference . . . 1022
8.1028.1 Detailed Description . . . 1022
8.1028.2 Field Documentation . . . 1023
8.1028.2.1 cellID . . . 1023
8.1028.2.2 ecio . . . 1023
8.1028.2.3 geranInst . . . 1023

8.1028.2.4 GeranInstInfo . . . 1023
8.1028.2.5 lac . . . 1023
8.1028.2.6 plmn . . . 1024
8.1028.2.7 psc . . . 1024
8.1028.2.8 rscp . . . 1024
8.1028.2.9 uarfcn . . . 1024
8.1028.2.10 umtsInst . . . 1024
8.1028.2.11 UMTSInstInfo . . . 1024
8.1029 UMTSinstInfo Struct Reference . . . 1024
8.1029.1 Detailed Description . . . 1024
8.1029.2 Field Documentation . . . 1024
8.1029.2.1 umtsEcio . . . 1024
8.1029.2.2 umtsPsc . . . 1024
8.1029.2.3 umtsRscp . . . 1024
8.1029.2.4 umtsUarfcn . . . 1024
8.1030 umtsLTENbrCell Struct Reference . . . 1024
8.1030.1 Detailed Description . . . 1025
8.1030.2 Field Documentation . . . 1025
8.1030.2.1 cellIsTDD . . . 1025
8.1030.2.2 earfcn . . . 1025
8.1030.2.3 pci . . . 1025
8.1030.2.4 rsrp . . . 1025
8.1030.2.5 rsrq . . . 1025
8.1030.2.6 srxlev . . . 1025
8.1031 UMTSMinQoS Struct Reference . . . 1025
8.1031.1 Detailed Description . . . 1026
8.1031.2 Field Documentation . . . 1027
8.1031.2.1 deliveryErrSDU . . . 1027
8.1031.2.2 grntDownlinkBitrate . . . 1028
8.1031.2.3 grntUplinkBitrate . . . 1028
8.1031.2.4 maxDownlinkBitrate . . . 1028
8.1031.2.5 maxSDUSize . . . 1028
8.1031.2.6 maxUplinkBitrate . . . 1028
8.1031.2.7 qosDeliveryOrder . . . 1028
8.1031.2.8 resBerRatio . . . 1028
8.1031.2.9 sduErrorRatio . . . 1028
8.1031.2.10 trafficClass . . . 1028
8.1031.2.11 trafficPriority . . . 1028
8.1031.2.12 transferDelay . . . 1028
8.1032 UMTSQoS Struct Reference . . . 1028

8.1032.1 Detailed Description . . . 1028
8.1032.2 Field Documentation . . . 1030
8.1032.2.1 deliveryErrSDU . . . 1030
8.1032.2.2 grntDownlinkBitrate . . . 1030
8.1032.2.3 grntUplinkBitrate . . . 1030
8.1032.2.4 maxDownlinkBitrate . . . 1030
8.1032.2.5 maxSDUSize . . . 1030
8.1032.2.6 maxUplinkBitrate . . . 1030
8.1032.2.7 qosDeliveryOrder . . . 1030
8.1032.2.8 resBerRatio . . . 1030
8.1032.2.9 sduErrorRatio . . . 1030
8.1032.2.10 trafficClass . . . 1030
8.1032.2.11 trafficPriority . . . 1030
8.1032.2.12 transferDelay . . . 1030
8.1033 UMTSReqQoSSigInd Struct Reference . . . 1030
8.1033.1 Detailed Description . . . 1031
8.1033.2 Field Documentation . . . 1031
8.1033.2.1 SigInd . . . 1031
8.1033.2.2 UMTSReqQoS . . . 1031
8.1034 unblockUIMPIN Struct Reference . . . 1031
8.1034.1 Detailed Description . . . 1031
8.1034.2 Field Documentation . . . 1032
8.1034.2.1 newPINLen . . . 1032
8.1034.2.2 newPINVal . . . 1032
8.1034.2.3 pinID . . . 1032
8.1034.2.4 pukLen . . . 1032
8.1034.2.5 pukVal . . . 1032
8.1035 UniversalTime Struct Reference . . . 1032
8.1035.1 Detailed Description . . . 1032
8.1035.2 Field Documentation . . . 1033
8.1035.2.1 day . . . 1033
8.1035.2.2 dayOfWeek . . . 1033
8.1035.2.3 hour . . . 1033
8.1035.2.4 minute . . . 1033
8.1035.2.5 month . . . 1033
8.1035.2.6 second . . . 1033
8.1035.2.7 year . . . 1033
8.1036 unpack_dms_GetActivationState_t Struct Reference . . . 1033
8.1036.1 Detailed Description . . . 1033
8.1036.2 Field Documentation . . . 1034

8.1036.2.1 state . . . 1034
8.1037 unpack_dms_GetBandCapability_t Struct Reference . . . 1034
8.1037.1 Detailed Description . . . 1034
8.1037.2 Field Documentation . . . 1035
8.1037.2.1 BandCapability . . . 1036
8.1037.2.2 Tlvresult . . . 1036
8.1038 unpack_dms_GetCrashAction_t Struct Reference . . . 1036
8.1038.1 Detailed Description . . . 1036
8.1038.2 Field Documentation . . . 1036
8.1038.2.1 DevCrashState . . . 1036
8.1038.2.2 Tlvresult . . . 1036
8.1039 unpack_dms_GetCustFeature_t Struct Reference . . . 1036
8.1039.1 Detailed Description . . . 1037
8.1039.2 Field Documentation . . . 1038
8.1039.2.1 DHCPRelayEnabled . . . 1038
8.1039.2.2 DisableIMSI . . . 1038
8.1039.2.3 GpsEnable . . . 1038
8.1039.2.4 GPSLPM . . . 1038
8.1039.2.5 GPSSel . . . 1038
8.1039.2.6 IPFamSupport . . . 1038
8.1039.2.7 IsVoiceEnabled . . . 1039
8.1039.2.8 RMAutoConnect . . . 1039
8.1039.2.9 SMSSupport . . . 1039
8.1039.2.10 Tlvresult . . . 1039
8.1040 unpack_dms_GetCustFeaturesV2_t Struct Reference . . . 1039
8.1040.1 Detailed Description . . . 1039
8.1040.2 Field Documentation . . . 1039
8.1040.2.1 GetCustomFeatureV2 . . . 1039
8.1040.2.2 Tlvresult . . . 1039
8.1041 unpack_dms_GetDeviceCap_t Struct Reference . . . 1039
8.1041.1 Detailed Description . . . 1039
8.1041.2 Field Documentation . . . 1040
8.1041.2.1 DataServiceCapability . . . 1040
8.1041.2.2 MaxRXChannelRate . . . 1040
8.1041.2.3 MaxTXChannelRate . . . 1040
8.1041.2.4 RadioIfaces . . . 1040
8.1041.2.5 RadioIfacesSize . . . 1040
8.1041.2.6 SimCapability . . . 1040
8.1041.2.7 Tlvresult . . . 1040
8.1042 unpack_dms_GetDeviceCapabilities_t Struct Reference . . . 1040

8.1042.1 Detailed Description . . . 1041
8.1042.2 Field Documentation . . . 1041
8.1042.2.1 dataServiceCaCapability . . . 1041
8.1042.2.2 maxRxChannelRate . . . 1041
8.1042.2.3 maxTxChannelRate . . . 1042
8.1042.2.4 RadioIfaces . . . 1042
8.1042.2.5 radioIfacesSize . . . 1042
8.1042.2.6 simCapability . . . 1042
8.1043 qmi_dms_GetDeviceHardwareRev_t Struct Reference . . . 1042
8.1043.1 Detailed Description . . . 1042
8.1043.2 Field Documentation . . . 1042
8.1043.2.1 String . . . 1042
8.1043.2.2 stringSize . . . 1042
8.1043.2.3 Tlvresult . . . 1042
8.1044 qmi_dms_GetDeviceMfr_t Struct Reference . . . 1042
8.1044.1 Detailed Description . . . 1043
8.1044.2 Field Documentation . . . 1043
8.1044.2.1 String . . . 1043
8.1044.2.2 stringSize . . . 1043
8.1044.2.3 Tlvresult . . . 1043
8.1045 qmi_dms_GetDeviceSerialNumbers_t Struct Reference . . . 1043
8.1045.1 Detailed Description . . . 1043
8.1045.2 Field Documentation . . . 1044
8.1045.2.1 esnSize . . . 1044
8.1045.2.2 ESNString . . . 1044
8.1045.2.3 meiSize . . . 1044
8.1045.2.4 MEIString . . . 1044
8.1045.2.5 meiSvnSize . . . 1044
8.1045.2.6 meiSvnString . . . 1044
8.1045.2.7 meidSize . . . 1044
8.1045.2.8 MEIDString . . . 1044
8.1045.2.9 Tlvresult . . . 1044
8.1046 qmi_dms_GetFirmwareInfo_t Struct Reference . . . 1044
8.1046.1 Detailed Description . . . 1045
8.1046.2 Field Documentation . . . 1046
8.1046.2.1 appversion_str . . . 1046
8.1046.2.2 bootversion_str . . . 1046
8.1046.2.3 carrier_str . . . 1046
8.1046.2.4 cur_carr_name . . . 1046
8.1046.2.5 cur_carr_rev . . . 1046

8.1046.2.6 modelid_str . . . 1046
8.1046.2.7 packageid_str . . . 1046
8.1046.2.8 priversion_str . . . 1046
8.1046.2.9 sku_str . . . 1046
8.1046.2.10 Tlvresult . . . 1046
8.1047 Unpack_dms_GetFirmwareRevision_t Struct Reference . . . 1046
8.1047.1 Detailed Description . . . 1046
8.1047.2 Field Documentation . . . 1047
8.1047.2.1 amssSize . . . 1047
8.1047.2.2 AMSSString . . . 1047
8.1047.2.3 PRIString . . . 1047
8.1047.2.4 Tlvresult . . . 1047
8.1048 Unpack_dms_GetFirmwareRevisions_t Struct Reference . . . 1047
8.1048.1 Detailed Description . . . 1047
8.1048.2 Field Documentation . . . 1048
8.1048.2.1 amssSize . . . 1048
8.1048.2.2 AMSSString . . . 1048
8.1048.2.3 bootSize . . . 1048
8.1048.2.4 BootString . . . 1048
8.1048.2.5 priSize . . . 1048
8.1048.2.6 PRIString . . . 1048
8.1048.2.7 Tlvresult . . . 1048
8.1049 Unpack_dms_GetFSN_t Struct Reference . . . 1048
8.1049.1 Detailed Description . . . 1048
8.1049.2 Field Documentation . . . 1049
8.1049.2.1 String . . . 1049
8.1049.2.2 Tlvresult . . . 1049
8.1050 Unpack_dms_GetHardwareRevision_t Struct Reference . . . 1049
8.1050.1 Detailed Description . . . 1049
8.1050.2 Field Documentation . . . 1049
8.1050.2.1 hwVer . . . 1049
8.1051 Unpack_dms_GetIMSI_t Struct Reference . . . 1049
8.1051.1 Detailed Description . . . 1049
8.1051.2 Field Documentation . . . 1049
8.1051.2.1 imsi . . . 1049
8.1051.2.2 Tlvresult . . . 1049
8.1052 Unpack_dms_GetManufacturer_t Struct Reference . . . 1050
8.1052.1 Detailed Description . . . 1050
8.1052.2 Field Documentation . . . 1050
8.1052.2.1 manufacturer . . . 1050

8.1052.2.2 Tlvresult . . . 1050
8.1053 npack_dms_GetModelID_t Struct Reference . . . 1050
8.1053.1 Detailed Description . . . 1050
8.1053.2 Field Documentation . . . 1050
8.1053.2.1 modelid . . . 1050
8.1053.2.2 Tlvresult . . . 1050
8.1054 npack_dms_GetNetworkTime_t Struct Reference . . . 1051
8.1054.1 Detailed Description . . . 1051
8.1054.2 Field Documentation . . . 1051
8.1054.2.1 source . . . 1051
8.1054.2.2 timestamp . . . 1051
8.1054.2.3 Tlvresult . . . 1051
8.1055 npack_dms_GetOfflineReason_t Struct Reference . . . 1051
8.1055.1 Detailed Description . . . 1051
8.1055.2 Field Documentation . . . 1052
8.1055.2.1 pbPlatform . . . 1052
8.1055.2.2 pReasonMask . . . 1052
8.1055.2.3 Tlvresult . . . 1052
8.1056 npack_dms_GetPower_t Struct Reference . . . 1052
8.1056.1 Detailed Description . . . 1052
8.1056.2 Field Documentation . . . 1053
8.1056.2.1 HardwareControlledMode . . . 1053
8.1056.2.2 OfflineReason . . . 1053
8.1056.2.3 OperationMode . . . 1053
8.1056.2.4 Tlvresult . . . 1053
8.1057 npack_dms_GetPRLVersion_t Struct Reference . . . 1054
8.1057.1 Detailed Description . . . 1054
8.1057.2 Field Documentation . . . 1054
8.1057.2.1 Tlvresult . . . 1054
8.1057.2.2 u16PRLVersion . . . 1054
8.1057.2.3 u8PRLPreference . . . 1054
8.1058 npack_dms_GetSerialNumbers_t Struct Reference . . . 1054
8.1058.1 Detailed Description . . . 1054
8.1058.2 Field Documentation . . . 1055
8.1058.2.1 esn . . . 1055
8.1058.2.2 mei_no . . . 1055
8.1058.2.3 meisv_svn . . . 1055
8.1058.2.4 meid . . . 1055
8.1059 npack_dms_GetUSBComp_t Struct Reference . . . 1055
8.1059.1 Detailed Description . . . 1055

8.1059.2 Field Documentation . . . 1057
8.1059.2.1 NumSupUSBComps . . . 1057
8.1059.2.2 SupUSBComps . . . 1057
8.1059.2.3 Tlvresult . . . 1057
8.1059.2.4 USBComp . . . 1057
8.1060 npack_dms_GetVoiceNumber_t Struct Reference . . . 1057
8.1060.1 Detailed Description . . . 1058
8.1060.2 Field Documentation . . . 1058
8.1060.2.1 MIN . . . 1058
8.1060.2.2 minSize . . . 1058
8.1060.2.3 Tlvresult . . . 1058
8.1060.2.4 VoiceNumber . . . 1058
8.1060.2.5 voiceNumberSize . . . 1058
8.1061 npack_dms_ResetToFactoryDefaults_t Struct Reference . . . 1058
8.1061.1 Detailed Description . . . 1058
8.1061.2 Field Documentation . . . 1059
8.1061.2.1 Tlvresult . . . 1059
8.1062 npack_dms_SetActivationStatusCallback_t Struct Reference . . . 1059
8.1062.1 Detailed Description . . . 1059
8.1062.2 Field Documentation . . . 1059
8.1062.2.1 Tlvresult . . . 1059
8.1063 npack_dms_SetCrashAction_t Struct Reference . . . 1059
8.1063.1 Detailed Description . . . 1059
8.1063.2 Field Documentation . . . 1059
8.1063.2.1 notused . . . 1059
8.1064 npack_dms_SetCustFeature_t Struct Reference . . . 1059
8.1064.1 Detailed Description . . . 1060
8.1064.2 Field Documentation . . . 1060
8.1064.2.1 Tlvresult . . . 1060
8.1065 npack_dms_SetCustFeaturesV2_t Struct Reference . . . 1060
8.1065.1 Detailed Description . . . 1060
8.1065.2 Field Documentation . . . 1060
8.1065.2.1 Tlvresult . . . 1060
8.1066 npack_dms_SetEventReport_ind_t Struct Reference . . . 1060
8.1066.1 Detailed Description . . . 1060
8.1066.2 Field Documentation . . . 1061
8.1066.2.1 ActivationStatusTlv . . . 1061
8.1066.2.2 OperatingModeTlv . . . 1061
8.1066.2.3 Tlvresult . . . 1061
8.1067 npack_dms_SetEventReport_t Struct Reference . . . 1061

8.1067.1 Detailed Description . . . 1061
8.1067.2 Field Documentation . . . 1061
8.1067.2.1 TlvResult . . . 1061
8.1068 unpack_dms_SetFirmwarePreference_t Struct Reference . . . 1061
8.1068.1 Detailed Description . . . 1061
8.1068.2 Field Documentation . . . 1061
8.1068.2.1 TlvResult . . . 1062
8.1069 unpack_dms_SetPower_t Struct Reference . . . 1062
8.1069.1 Detailed Description . . . 1062
8.1069.2 Field Documentation . . . 1062
8.1069.2.1 TlvResult . . . 1062
8.1070 unpack_dms_SetUSBComp_t Struct Reference . . . 1062
8.1070.1 Detailed Description . . . 1062
8.1070.2 Field Documentation . . . 1062
8.1070.2.1 TlvResult . . . 1062
8.1071 unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t Struct Reference . . . 1062
8.1071.1 Detailed Description . . . 1063
8.1071.2 Field Documentation . . . 1063
8.1071.2.1 source . . . 1063
8.1071.2.2 TlvResult . . . 1063
8.1071.2.3 type . . . 1063
8.1072 unpack_dms_SLQSDmsSwiGetResetInfo_t Struct Reference . . . 1063
8.1072.1 Detailed Description . . . 1063
8.1072.2 Field Documentation . . . 1064
8.1072.2.1 source . . . 1064
8.1072.2.2 TlvResult . . . 1064
8.1072.2.3 type . . . 1064
8.1073 unpack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference . . . 1064
8.1073.1 Detailed Description . . . 1064
8.1073.2 Field Documentation . . . 1065
8.1073.2.1 TlvResult . . . 1065
8.1074 unpack_dms_SLQSGetBandCapability_t Struct Reference . . . 1065
8.1074.1 Detailed Description . . . 1065
8.1074.2 Field Documentation . . . 1068
8.1074.2.1 bandCapability . . . 1068
8.1074.2.2 is_LteBandCapability_Available . . . 1068
8.1074.2.3 is_TdsBandCapability_Available . . . 1068
8.1074.2.4 LteBandCapability . . . 1068
8.1074.2.5 TdsBandCapability . . . 1068
8.1075 unpack_dms_SLQSGetERIFile_t Struct Reference . . . 1068

8.1075.1 Detailed Description . . . 1068
8.1075.2 Field Documentation . . . 1068
8.1075.2.1 eriFile . . . 1068
8.1075.2.2 Tlvresult . . . 1068
8.1076 unpack_dms_SLQSSwiClearDyingGaspStatistics_t Struct Reference . . . 1068
8.1076.1 Detailed Description . . . 1069
8.1076.2 Field Documentation . . . 1069
8.1076.2.1 Tlvresult . . . 1069
8.1077 unpack_dms_SLQSSwiGetCrashInfo_t Struct Reference . . . 1069
8.1077.1 Detailed Description . . . 1069
8.1077.2 Field Documentation . . . 1069
8.1077.2.1 crashInfoParam . . . 1069
8.1077.2.2 Tlvresult . . . 1069
8.1078 unpack_dms_SLQSSwiGetDyingGaspCfg_t Struct Reference . . . 1069
8.1078.1 Detailed Description . . . 1070
8.1078.2 Field Documentation . . . 1070
8.1078.2.1 pGetDyingGaspCfg . . . 1070
8.1078.2.2 Tlvresult . . . 1070
8.1079 unpack_dms_SLQSSwiGetDyingGaspStatistics_t Struct Reference . . . 1070
8.1079.1 Detailed Description . . . 1070
8.1079.2 Field Documentation . . . 1070
8.1079.2.1 pGetDyingGaspStatistics . . . 1070
8.1079.2.2 Tlvresult . . . 1070
8.1080 unpack_dms_SLQSSwiGetFirmwareCurr_t Struct Reference . . . 1070
8.1080.1 Detailed Description . . . 1071
8.1080.2 Field Documentation . . . 1071
8.1080.2.1 carrier . . . 1071
8.1080.2.2 fwvers . . . 1071
8.1080.2.3 numEntries . . . 1071
8.1080.2.4 pCurrImgInfo . . . 1071
8.1080.2.5 pkgver . . . 1071
8.1080.2.6 priver . . . 1071
8.1081 unpack_dms_SLQSSwiGetFwUpdateStatus_t Struct Reference . . . 1072
8.1081.1 Detailed Description . . . 1072
8.1081.2 Field Documentation . . . 1073
8.1081.2.1 imgType . . . 1073
8.1081.2.2 logString . . . 1073
8.1081.2.3 refData . . . 1073
8.1081.2.4 refString . . . 1073
8.1081.2.5 ResCode . . . 1073

8.1081.2.6 Tlvresult . . . 1073
8.1082 npack_dms_SLQSSwiGetHostDevInfo_t Struct Reference . . . 1073
8.1082.1 Detailed Description . . . 1073
8.1082.2 Field Documentation . . . 1074
8.1082.2.1 manString . . . 1074
8.1082.2.2 modelString . . . 1074
8.1082.2.3 plasmaIDString . . . 1074
8.1082.2.4 swVerString . . . 1074
8.1082.2.5 Tlvresult . . . 1074
8.1083 npack_dms_SLQSSwiGetOSInfo_t Struct Reference . . . 1074
8.1083.1 Detailed Description . . . 1074
8.1083.2 Field Documentation . . . 1074
8.1083.2.1 nameString . . . 1075
8.1083.2.2 Tlvresult . . . 1075
8.1083.2.3 versionString . . . 1075
8.1084 npack_dms_SLQSSwiGetSerialNoExt_t Struct Reference . . . 1075
8.1084.1 Detailed Description . . . 1075
8.1084.2 Field Documentation . . . 1075
8.1084.2.1 meidString . . . 1075
8.1084.2.2 Tlvresult . . . 1075
8.1085 npack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference . . . 1075
8.1085.1 Detailed Description . . . 1075
8.1085.2 Field Documentation . . . 1075
8.1085.2.1 Tlvresult . . . 1076
8.1086 npack_dms_SLQSSwiSetHostDevInfo_t Struct Reference . . . 1076
8.1086.1 Detailed Description . . . 1076
8.1086.2 Field Documentation . . . 1076
8.1086.2.1 Tlvresult . . . 1076
8.1087 npack_dms_SLQSSwiSetOSInfo_t Struct Reference . . . 1076
8.1087.1 Detailed Description . . . 1076
8.1087.2 Field Documentation . . . 1076
8.1087.2.1 Tlvresult . . . 1076
8.1088 npack_dms_SLQSUIMGetState_t Struct Reference . . . 1076
8.1088.1 Detailed Description . . . 1077
8.1088.2 Field Documentation . . . 1077
8.1088.2.1 state . . . 1077
8.1088.2.2 Tlvresult . . . 1077
8.1089 npack_dms_UIMGetControlKeyStatus_t Struct Reference . . . 1077
8.1089.1 Detailed Description . . . 1077
8.1089.2 Field Documentation . . . 1078

8.1089.2.1 facilityState . . . 1078
8.1089.2.2 TlvResult . . . 1078
8.1089.2.3 unblockRetriesLeft . . . 1078
8.1089.2.4 verifyRetriesLeft . . . 1078
8.1090 unpack_dms_UIMGetICCID_t Struct Reference . . . 1078
8.1090.1 Detailed Description . . . 1078
8.1090.2 Field Documentation . . . 1078
8.1090.2.1 String . . . 1078
8.1090.2.2 stringSize . . . 1078
8.1090.2.3 TlvResult . . . 1078
8.1091 unpack_dms_UIMGetPINStatus_t Struct Reference . . . 1078
8.1091.1 Detailed Description . . . 1079
8.1091.2 Field Documentation . . . 1080
8.1091.2.1 p1Status . . . 1080
8.1091.2.2 p1UnblockRetriesLeft . . . 1080
8.1091.2.3 p1VerifyRetriesLeft . . . 1080
8.1091.2.4 p2Status . . . 1080
8.1091.2.5 p2UnblockRetriesLeft . . . 1080
8.1091.2.6 p2VerifyRetriesLeft . . . 1080
8.1091.2.7 TlvResult . . . 1080
8.1092 unpack_dms_UIMSetControlKeyProtection_t Struct Reference . . . 1080
8.1092.1 Detailed Description . . . 1080
8.1092.2 Field Documentation . . . 1080
8.1092.2.1 TlvResult . . . 1080
8.1092.2.2 verifyRetriesLeft . . . 1080
8.1093 unpack_dms_UIMSetPINProtection_t Struct Reference . . . 1080
8.1093.1 Detailed Description . . . 1081
8.1093.2 Field Documentation . . . 1081
8.1093.2.1 TlvResult . . . 1081
8.1093.2.2 unblockRetriesLeft . . . 1081
8.1093.2.3 verifyRetriesLeft . . . 1081
8.1094 unpack_dms_UIMUnblockControlKey_t Struct Reference . . . 1081
8.1094.1 Detailed Description . . . 1081
8.1094.2 Field Documentation . . . 1082
8.1094.2.1 TlvResult . . . 1082
8.1094.2.2 unblockRetriesLeft . . . 1082
8.1095 unpack_fms_GetImagesPreference_t Struct Reference . . . 1082
8.1095.1 Detailed Description . . . 1082
8.1095.2 Field Documentation . . . 1082
8.1095.2.1 ImageListSize . . . 1082

8.1095.2.2 pImageList . . . 1082
8.1095.2.3 Tlvresult . . . 1082
8.1096 unpack_fms_GetStoredImages_t Struct Reference . . . 1082
8.1096.1 Detailed Description . . . 1082
8.1096.2 Field Documentation . . . 1083
8.1096.2.1 imageList . . . 1083
8.1096.2.2 imagelistSize . . . 1083
8.1096.2.3 Tlvresult . . . 1083
8.1097 unpack_fms_SetImagesPreference_t Struct Reference . . . 1083
8.1097.1 Detailed Description . . . 1083
8.1097.2 Field Documentation . . . 1083
8.1097.2.1 ImageTypes . . . 1083
8.1097.2.2 imageTypesSize . . . 1083
8.1097.2.3 Tlvresult . . . 1083
8.1098 unpack_loc_BestAvailPos_Ind_t Struct Reference . . . 1084
8.1098.1 Detailed Description . . . 1084
8.1098.2 Field Documentation . . . 1088
8.1098.2.1 pAltitudeWrtEllipsoid . . . 1088
8.1098.2.2 pAltitudeWrtMeanSeaLevel . . . 1088
8.1098.2.3 pGpsTime . . . 1088
8.1098.2.4 pHeading . . . 1088
8.1098.2.5 pHeadingUnc . . . 1088
8.1098.2.6 pHorCirConf . . . 1088
8.1098.2.7 pHorEllpConf . . . 1088
8.1098.2.8 pHorReliability . . . 1088
8.1098.2.9 pHorUncCircular . . . 1088
8.1098.2.10 pHorUncEllipseOrientAzimuth . . . 1088
8.1098.2.11 pHorUncEllipseSemiMajor . . . 1088
8.1098.2.12 pHorUncEllipseSemiMinor . . . 1089
8.1098.2.13 pLatitude . . . 1089
8.1098.2.14 pLongitude . . . 1089
8.1098.2.15 pMagneticDeviation . . . 1089
8.1098.2.16 pPrecisionDilution . . . 1089
8.1098.2.17 pSensorDataUsage . . . 1089
8.1098.2.18 pSpeedHorizontal . . . 1089
8.1098.2.19 pSpeedUnc . . . 1089
8.1098.2.20 pSpeedVertical . . . 1089
8.1098.2.21 pSpeedVerticalUnc . . . 1089
8.1098.2.22 pSvUsedforFix . . . 1089
8.1098.2.23 pTechnologyMask . . . 1089

8.1098.2.24 pTimeSrc . . . 1089
8.1098.2.25 pTimestampUtc . . . 1089
8.1098.2.26 pTimeUnc . . . 1089
8.1098.2.27 pVertConfidence . . . 1089
8.1098.2.28 pVertReliability . . . 1089
8.1098.2.29 pVertUnc . . . 1089
8.1098.2.30 pXid . . . 1089
8.1098.2.31 status . . . 1089
8.1098.2.32 tlvresult . . . 1089
8.1099 unpack_loc_Delete_Assist_Data_t Struct Reference . . . 1089
8.1099.1 Detailed Description . . . 1089
8.1099.2 Field Documentation . . . 1090
8.1099.2.1 Tlvresult . . . 1090
8.1100 unpack_loc_DeleteAssistData_Ind_t Struct Reference . . . 1090
8.1100.1 Detailed Description . . . 1090
8.1100.2 Field Documentation . . . 1090
8.1100.2.1 status . . . 1090
8.1100.2.2 Tlvresult . . . 1091
8.1101 unpack_loc_EngineState_Ind_t Struct Reference . . . 1091
8.1101.1 Detailed Description . . . 1091
8.1101.2 Field Documentation . . . 1091
8.1101.2.1 engineState . . . 1091
8.1101.2.2 Tlvresult . . . 1091
8.1102 unpack_loc_EventNMEA_Ind_t Struct Reference . . . 1091
8.1102.1 Detailed Description . . . 1091
8.1102.2 Field Documentation . . . 1092
8.1102.2.1 NMEAData . . . 1092
8.1102.2.2 Tlvresult . . . 1092
8.1103 unpack_loc_EventRegister_t Struct Reference . . . 1092
8.1103.1 Detailed Description . . . 1092
8.1103.2 Field Documentation . . . 1092
8.1103.2.1 Tlvresult . . . 1092
8.1104 unpack_loc_GetServer_Ind_t Struct Reference . . . 1092
8.1104.1 Detailed Description . . . 1092
8.1104.2 Field Documentation . . . 1093
8.1104.2.1 pIPv4AddrInfo . . . 1093
8.1104.2.2 pIPv6AddrInfo . . . 1093
8.1104.2.3 pURL . . . 1093
8.1104.2.4 serverStatus . . . 1093
8.1104.2.5 serverType . . . 1093

8.1104.2.6 Tlvresult . . . 1094
8.1105 npack_loc_GnssSvInfo_Ind_t Struct Reference . . . 1094
8.1105.1 Detailed Description . . . 1094
8.1105.2 Field Documentation . . . 1094
8.1105.2.1 altitudeAssumed . . . 1094
8.1105.2.2 pSatelliteInfo . . . 1094
8.1105.2.3 Tlvresult . . . 1094
8.1106 npack_loc_PositionRpt_Ind_t Struct Reference . . . 1094
8.1106.1 Detailed Description . . . 1095
8.1106.2 Field Documentation . . . 1100
8.1106.2.1 pAltitudeAssumed . . . 1100
8.1106.2.2 pAltitudeWrtEllipsoid . . . 1100
8.1106.2.3 pAltitudeWrtMeanSeaLevel . . . 1100
8.1106.2.4 pFixId . . . 1100
8.1106.2.5 pGpsTime . . . 1100
8.1106.2.6 pHeading . . . 1100
8.1106.2.7 pHeadingUnc . . . 1100
8.1106.2.8 pHorConfidence . . . 1100
8.1106.2.9 pHorReliability . . . 1100
8.1106.2.10 pHorUncCircular . . . 1100
8.1106.2.11 pHorUncEllipseOrientAzimuth . . . 1100
8.1106.2.12 pHorUncEllipseSemiMajor . . . 1100
8.1106.2.13 pHorUncEllipseSemiMinor . . . 1100
8.1106.2.14 pLatitude . . . 1100
8.1106.2.15 pLeapSeconds . . . 1100
8.1106.2.16 pLongitude . . . 1100
8.1106.2.17 pMagneticDeviation . . . 1100
8.1106.2.18 pPrecisionDilution . . . 1100
8.1106.2.19 pSensorDataUsage . . . 1100
8.1106.2.20 pSpeedHorizontal . . . 1100
8.1106.2.21 pSpeedUnc . . . 1100
8.1106.2.22 pSpeedVertical . . . 1100
8.1106.2.23 pSvUsedforFix . . . 1100
8.1106.2.24 pTechnologyMask . . . 1100
8.1106.2.25 pTimeSrc . . . 1100
8.1106.2.26 pTimestampUtc . . . 1101
8.1106.2.27 pTimeUnc . . . 1101
8.1106.2.28 pVertConfidence . . . 1101
8.1106.2.29 pVertReliability . . . 1101
8.1106.2.30 pVertUnc . . . 1101

8.1106.2.3.1 sessionId . . . 1101
8.1106.2.3.2 sessionStatus . . . 1101
8.1106.2.3.3 tlvresult . . . 1101
8.1107 unpack_loc_SetExtPowerConfig_Ind_t Struct Reference . . . 1101
8.1107.1 Detailed Description . . . 1101
8.1107.2 Field Documentation . . . 1101
8.1107.2.1 status . . . 1101
8.1107.2.2 Tlvresult . . . 1102
8.1108 unpack_loc_SetExtPowerState_t Struct Reference . . . 1102
8.1108.1 Detailed Description . . . 1102
8.1108.2 Field Documentation . . . 1102
8.1108.2.1 Tlvresult . . . 1102
8.1109 unpack_loc_SetOperationMode_Ind_t Struct Reference . . . 1102
8.1109.1 Detailed Description . . . 1102
8.1109.2 Field Documentation . . . 1103
8.1109.2.1 status . . . 1103
8.1109.2.2 Tlvresult . . . 1103
8.1110 unpack_loc_SetOperationMode_t Struct Reference . . . 1103
8.1110.1 Detailed Description . . . 1103
8.1110.2 Field Documentation . . . 1103
8.1110.2.1 Tlvresult . . . 1103
8.1111 unpack_loc_SLQSLOCGetBestAvailPos_t Struct Reference . . . 1103
8.1111.1 Detailed Description . . . 1103
8.1111.2 Field Documentation . . . 1103
8.1111.2.1 Tlvresult . . . 1103
8.1112 unpack_loc_Start_t Struct Reference . . . 1104
8.1112.1 Detailed Description . . . 1104
8.1112.2 Field Documentation . . . 1104
8.1112.2.1 Tlvresult . . . 1104
8.1113 unpack_loc_Stop_t Struct Reference . . . 1104
8.1113.1 Detailed Description . . . 1104
8.1113.2 Field Documentation . . . 1104
8.1113.2.1 Tlvresult . . . 1104
8.1114 unpack_nas_GetCDMANetworkParameters_t Struct Reference . . . 1104
8.1114.1 Detailed Description . . . 1105
8.1114.2 Field Documentation . . . 1106
8.1114.2.1 Application . . . 1106
8.1114.2.2 Broadcast . . . 1106
8.1114.2.3 CustomSCP . . . 1106
8.1114.2.4 ForceRev0 . . . 1106

8.1114.2.5 Protocol . . . 1106
8.1114.2.6 RegForeignNID . . . 1106
8.1114.2.7 RegForeignSID . . . 1106
8.1114.2.8 RegHomeSID . . . 1106
8.1114.2.9 Roaming . . . 1106
8.1114.2.10 SCI . . . 1107
8.1114.2.11 SCM . . . 1107
8.1115 npack_nas_GetHomeNetwork3GPP2_t Struct Reference . . . 1107
8.1115.1 Detailed Description . . . 1107
8.1115.2 Field Documentation . . . 1108
8.1115.2.1 nameSize . . . 1108
8.1115.2.2 pMCC . . . 1108
8.1115.2.3 pMNC . . . 1108
8.1115.2.4 pName . . . 1108
8.1115.2.5 pNID . . . 1109
8.1115.2.6 pNw2DescDisp . . . 1109
8.1115.2.7 pNw2DescEnc . . . 1109
8.1115.2.8 pNw2DescLen . . . 1109
8.1115.2.9 pNw2MCC . . . 1109
8.1115.2.10 pNw2MNC . . . 1109
8.1115.2.11 pNw2Name . . . 1109
8.1115.2.12 pSID . . . 1109
8.1116 npack_nas_GetHomeNetwork_t Struct Reference . . . 1109
8.1116.1 Detailed Description . . . 1109
8.1116.2 Field Documentation . . . 1110
8.1116.2.1 mcc . . . 1110
8.1116.2.2 mnc . . . 1110
8.1116.2.3 name . . . 1110
8.1116.2.4 nid . . . 1110
8.1116.2.5 sid . . . 1110
8.1117 npack_nas_GetNetworkPreference_t Struct Reference . . . 1110
8.1117.1 Detailed Description . . . 1110
8.1117.2 Field Documentation . . . 1111
8.1117.2.1 ActiveTechPref . . . 1111
8.1117.2.2 Duration . . . 1111
8.1117.2.3 PersistentTechPref . . . 1111
8.1117.2.4 Tlvresult . . . 1111
8.1118 npack_nas_GetRFInfo_t Struct Reference . . . 1111
8.1118.1 Detailed Description . . . 1111
8.1118.2 Field Documentation . . . 1112

8.1118.2.1 instancesSize . . . 1112
8.1118.2.2 RFBandInfoElements . . . 1112
8.1119 unpack_nas_GetServingNetwork_t Struct Reference . . . 1112
8.1119.1 Detailed Description . . . 1112
8.1119.2 Field Documentation . . . 1114
8.1119.2.1 CSDomain . . . 1114
8.1119.2.2 DataCaps . . . 1114
8.1119.2.3 DataCapsLen . . . 1114
8.1119.2.4 MCC . . . 1114
8.1119.2.5 MNC . . . 1114
8.1119.2.6 Name . . . 1114
8.1119.2.7 nameSize . . . 1114
8.1119.2.8 PSDomain . . . 1114
8.1119.2.9 RadioIfaces . . . 1114
8.1119.2.10 RadioIfacesSize . . . 1114
8.1119.2.11 RAN . . . 1114
8.1119.2.12 RegistrationState . . . 1114
8.1119.2.13 Roaming . . . 1114
8.1120 unpack_nas_GetServingNetworkCapabilities_t Struct Reference . . . 1114
8.1120.1 Detailed Description . . . 1115
8.1120.2 Field Documentation . . . 1115
8.1120.2.1 DataCaps . . . 1115
8.1120.2.2 DataCapsLen . . . 1115
8.1121 unpack_nas_GetSignalStrengths_t Struct Reference . . . 1115
8.1121.1 Detailed Description . . . 1115
8.1121.2 Field Documentation . . . 1115
8.1121.2.1 len . . . 1116
8.1121.2.2 radio . . . 1116
8.1121.2.3 rssi . . . 1116
8.1122 unpack_nas_PerformNetworkScan_t Struct Reference . . . 1116
8.1122.1 Detailed Description . . . 1116
8.1122.2 Field Documentation . . . 1117
8.1122.2.1 p3GppNetworkInfoInstances . . . 1117
8.1122.2.2 p3GppNetworkInstanceSize . . . 1117
8.1122.2.3 pPCSInstance . . . 1117
8.1122.2.4 pPCSInstanceSize . . . 1117
8.1122.2.5 pRATINstance . . . 1117
8.1122.2.6 pRATInstanceSize . . . 1117
8.1122.2.7 pScanResult . . . 1117
8.1123 unpack_nas_SetDataCapabilitiesCallback_ind_t Struct Reference . . . 1117

8.1123.1 Detailed Description . . . 1117
8.1123.2 Field Documentation . . . 1117
8.1123.2.1 dataCaps . . . 1117
8.1123.2.2 dataCapsSize . . . 1117
8.1124 npack_nas_SetEventReportInd_t Struct Reference . . . 1117
8.1124.1 Detailed Description . . . 1117
8.1124.2 Field Documentation . . . 1118
8.1124.2.1 RFTlv . . . 1118
8.1124.2.2 RRTlv . . . 1118
8.1124.2.3 SLQSSSTlv . . . 1118
8.1124.2.4 SSTlv . . . 1118
8.1125 npack_nas_SetNasLTECphyCaIndCallback_ind_t Struct Reference . . . 1118
8.1125.1 Detailed Description . . . 1118
8.1125.2 Field Documentation . . . 1118
8.1125.2.1 sPhyCaAggPcellInfo . . . 1118
8.1125.2.2 sPhyCaAggScellDlBw . . . 1119
8.1125.2.3 sPhyCaAggScellIndex . . . 1119
8.1125.2.4 sPhyCaAggScellIndType . . . 1119
8.1125.2.5 sPhyCaAggScellInfo . . . 1119
8.1126 npack_nas_SetNetworkPreference_t Struct Reference . . . 1119
8.1126.1 Detailed Description . . . 1119
8.1126.2 Field Documentation . . . 1119
8.1126.2.1 Tlvresult . . . 1119
8.1127 npack_nas_SetRoamingIndicatorCallback_ind_t Struct Reference . . . 1119
8.1127.1 Detailed Description . . . 1119
8.1127.2 Field Documentation . . . 1119
8.1127.2.1 roaming . . . 1119
8.1128 npack_nas_SetServingSystemCallback_ind_t Struct Reference . . . 1120
8.1128.1 Detailed Description . . . 1120
8.1128.2 Field Documentation . . . 1120
8.1128.2.1 SSInfo . . . 1120
8.1128.2.2 Tlvresult . . . 1120
8.1129 npack_nas_SLQSGetErrorRate_t Struct Reference . . . 1120
8.1129.1 Detailed Description . . . 1120
8.1129.2 Field Documentation . . . 1121
8.1129.2.1 pCDMAFrameErrRate . . . 1121
8.1129.2.2 pGSMBER . . . 1121
8.1129.2.3 pHDRPackErrRate . . . 1121
8.1129.2.4 pWCDMABER . . . 1121
8.1130 npack_nas_SlqsGetLTECphyCAInfo_t Struct Reference . . . 1121

8.1130.1 Detailed Description . . . 1121
8.1130.2 Field Documentation . . . 1122
8.1130.2.1 LTECphyCaInfo . . . 1122
8.1130.2.2 Tlvresult . . . 1122
8.1131 unpack_nas_SLQSGetNetworkTime_t Struct Reference . . . 1122
8.1131.1 Detailed Description . . . 1122
8.1131.2 Field Documentation . . . 1122
8.1131.2.1 p3GPP2TimeInfo . . . 1122
8.1131.2.2 p3GPPTimeInfo . . . 1122
8.1132 unpack_nas_SLQSGetOperatorNameData_t Struct Reference . . . 1122
8.1132.1 Detailed Description . . . 1122
8.1132.2 Field Documentation . . . 1123
8.1132.2.1 pNITZInformation . . . 1123
8.1132.2.2 pOperatorNameString . . . 1123
8.1132.2.3 pOperatorPLMNList . . . 1123
8.1132.2.4 pPLMNNetworkName . . . 1123
8.1132.2.5 pSrvcProviderName . . . 1123
8.1133 unpack_nas_SLQSGetPLMNName_t Struct Reference . . . 1123
8.1133.1 Detailed Description . . . 1124
8.1133.2 Field Documentation . . . 1125
8.1133.2.1 longName . . . 1125
8.1133.2.2 longNameCI . . . 1125
8.1133.2.3 longNameEn . . . 1125
8.1133.2.4 longNameLen . . . 1125
8.1133.2.5 longNameSB . . . 1125
8.1133.2.6 shortName . . . 1125
8.1133.2.7 shortNameCI . . . 1126
8.1133.2.8 shortNameEn . . . 1126
8.1133.2.9 shortNameLen . . . 1126
8.1133.2.10 shortNameSB . . . 1126
8.1133.2.11 spn . . . 1126
8.1133.2.12 spnEncoding . . . 1126
8.1133.2.13 spnLength . . . 1126
8.1134 unpack_nas_SLQSGetServingSystem_t Struct Reference . . . 1126
8.1134.1 Detailed Description . . . 1126
8.1134.2 Field Documentation . . . 1129
8.1134.2.1 BasestationID . . . 1129
8.1134.2.2 BasestationLatitude . . . 1129
8.1134.2.3 BasestationLongitude . . . 1129
8.1134.2.4 CallBarStatus . . . 1129

8.1134.2.5 CDMA_P_Rev . . . 1129
8.1134.2.6 CDMASystemInfoExt . . . 1129
8.1134.2.7 CellID . . . 1129
8.1134.2.8 ConcSvcInfo . . . 1129
8.1134.2.9 CurrentPLMN . . . 1129
8.1134.2.10 DataSrvCapabilities . . . 1129
8.1134.2.11 DefaultRoamInd . . . 1129
8.1134.2.12 DetailedSvcInfo . . . 1129
8.1134.2.13 DTMInd . . . 1129
8.1134.2.14 Hpp2TimeZone . . . 1129
8.1134.2.15 HppNetworkDSTAdjustment . . . 1129
8.1134.2.16 HppTimeZone . . . 1129
8.1134.2.17 HdrPersonality . . . 1129
8.1134.2.18 Mac . . . 1129
8.1134.2.19 NetworkID . . . 1130
8.1134.2.20 PRLInd . . . 1130
8.1134.2.21 RoamIndicatorVal . . . 1130
8.1134.2.22 RoamingIndicatorList . . . 1130
8.1134.2.23 ServingSystem . . . 1130
8.1134.2.24 SystemID . . . 1130
8.1134.2.25 TrackAreaCode . . . 1130
8.1135 npack_nas_SLQSGetSignalStrength_t Struct Reference . . . 1130
8.1135.1 Detailed Description . . . 1130
8.1135.2 Field Documentation . . . 1132
8.1135.2.1 ecioList . . . 1132
8.1135.2.2 ecioListLen . . . 1132
8.1135.2.3 errorRateList . . . 1132
8.1135.2.4 errorRateListLen . . . 1132
8.1135.2.5 io . . . 1132
8.1135.2.6 rsrp . . . 1132
8.1135.2.7 rsnr . . . 1132
8.1135.2.8 rsrqInfo . . . 1132
8.1135.2.9 rxSignalStrengthList . . . 1132
8.1135.2.10 rxSignalStrengthListLen . . . 1132
8.1135.2.11 signalStrengthReqMask . . . 1132
8.1135.2.12 snr . . . 1132
8.1136 npack_nas_SLQSGetSysInfo_t Struct Reference . . . 1132
8.1136.1 Detailed Description . . . 1133
8.1136.2 Field Documentation . . . 1135
8.1136.2.1 pAddCDMASysInfo . . . 1135

8.1136.2.2 pAddGSMSysInfo . . . 1135
8.1136.2.3 pAddHDRSysInfo . . . 1135
8.1136.2.4 pAddLTESysInfo . . . 1135
8.1136.2.5 pAddWCDMASysInfo . . . 1135
8.1136.2.6 pCDMASrvStatusInfo . . . 1135
8.1136.2.7 pCDMASysInfo . . . 1135
8.1136.2.8 pGSMCallBarringSysInfo . . . 1135
8.1136.2.9 pGSMCipherDomainSysInfo . . . 1135
8.1136.2.10 pGSMSrvStatusInfo . . . 1135
8.1136.2.11 pGSMSysInfo . . . 1135
8.1136.2.12 pHDRSrvStatusInfo . . . 1135
8.1136.2.13 pHDRSysInfo . . . 1135
8.1136.2.14 pLTESrvStatusInfo . . . 1135
8.1136.2.15 pLTESysInfo . . . 1135
8.1136.2.16 pLTEVoiceSupportSysInfo . . . 1135
8.1136.2.17 pWCDMACallBarringSysInfo . . . 1135
8.1136.2.18 pWCDMACipherDomainSysInfo . . . 1136
8.1136.2.19 pWCDMASrvStatusInfo . . . 1136
8.1136.2.20 pWCDMASysInfo . . . 1136
8.1137 unpack_nas_SLQSGetSysSelectionPref_t Struct Reference . . . 1136
8.1137.1 Detailed Description . . . 1136
8.1137.2 Field Documentation . . . 1140
8.1137.2.1 pBandPref . . . 1140
8.1137.2.2 pEmerMode . . . 1140
8.1137.2.3 pGWAcqOrderPref . . . 1140
8.1137.2.4 pLTEBandPref . . . 1140
8.1137.2.5 pModePref . . . 1140
8.1137.2.6 pNetSelPref . . . 1140
8.1137.2.7 pPRLPref . . . 1140
8.1137.2.8 pRoamPref . . . 1140
8.1137.2.9 pSrvDomainPref . . . 1140
8.1138 unpack_nas_SLQSNasGet3GPP2Subscription_t Struct Reference . . . 1140
8.1138.1 Detailed Description . . . 1141
8.1138.2 Field Documentation . . . 1141
8.1138.2.1 pCDMAChannel . . . 1141
8.1138.2.2 pDirNum . . . 1141
8.1138.2.3 pHomeSIDNID . . . 1141
8.1138.2.4 pMinBasedIMSI . . . 1141
8.1138.2.5 pNAMNameInfo . . . 1141
8.1138.2.6 pTrueIMSI . . . 1141

8.1139 unpack_nas_SLQSNasGetCellLocationInfo_t Struct Reference . . . 1141
8.1139.1 Detailed Description . . . 1142
8.1139.2 Field Documentation . . . 1142
8.1139.2.1 pCDMAInfo . . . 1142
8.1139.2.2 pGERANInfo . . . 1143
8.1139.2.3 pLTEInfoInterfreq . . . 1143
8.1139.2.4 pLTEInfoIntrafreq . . . 1143
8.1139.2.5 pLTEInfoNeighboringGSM . . . 1143
8.1139.2.6 pLTEInfoNeighboringWCDMA . . . 1143
8.1139.2.7 pUMTSCellID . . . 1143
8.1139.2.8 pUMTSInfo . . . 1143
8.1139.2.9 pWCDMAInfoLTENeighborCell . . . 1143
8.1140 unpack_nas_SLQSNasGetHDRColorCode_t Struct Reference . . . 1143
8.1140.1 Detailed Description . . . 1143
8.1140.2 Field Documentation . . . 1143
8.1140.2.1 pColorCode . . . 1143
8.1141 unpack_nas_SLQSNasGetSigInfo_t Struct Reference . . . 1143
8.1141.1 Detailed Description . . . 1144
8.1141.2 Field Documentation . . . 1144
8.1141.2.1 CDMASSInfo . . . 1144
8.1141.2.2 GSMSSInfo . . . 1144
8.1141.2.3 HDRSSInfo . . . 1144
8.1141.2.4 LTESSInfo . . . 1144
8.1141.2.5 WCDMASSInfo . . . 1144
8.1142 unpack_nas_SLQSNasGetTxRxInfo_t Struct Reference . . . 1144
8.1142.1 Detailed Description . . . 1144
8.1142.2 Field Documentation . . . 1145
8.1142.2.1 pRXChain0Info . . . 1145
8.1142.2.2 pRXChain1Info . . . 1145
8.1142.2.3 pTXInfo . . . 1145
8.1143 unpack_nas_SLQSNasNetworkTimeCallBack_ind_t Struct Reference . . . 1145
8.1143.1 Detailed Description . . . 1145
8.1143.2 Field Documentation . . . 1146
8.1143.2.1 pDayltSavAdj . . . 1146
8.1143.2.2 pRadioInterface . . . 1146
8.1143.2.3 pTimeZone . . . 1146
8.1143.2.4 universalTime . . . 1146
8.1144 unpack_nas_SLQSNasSigInfoCallback_ind_t Struct Reference . . . 1146
8.1144.1 Detailed Description . . . 1146
8.1144.2 Field Documentation . . . 1147

8.1144.2.1 pCDMASigInfo . . . 1147
8.1144.2.2 pGSMSigInfo . . . 1147
8.1144.2.3 pHDRSigInfo . . . 1147
8.1144.2.4 pLTESigInfo . . . 1147
8.1144.2.5 pRscp . . . 1147
8.1144.2.6 pTDSCDMASigInfoExt . . . 1147
8.1144.2.7 pWCDMASigInfo . . . 1147
8.1145 npack_nas_SLQSNASSwiGetChannelLock_t Struct Reference . . . 1147
8.1145.1 Detailed Description . . . 1147
8.1145.2 Field Documentation . . . 1148
8.1145.2.1 pLteEARFCN . . . 1148
8.1145.2.2 pLtePCI . . . 1148
8.1145.2.3 pWcdmaUARFCN . . . 1148
8.1146 npack_nas_SLQSNasSwiModemStatus_t Struct Reference . . . 1148
8.1146.1 Detailed Description . . . 1148
8.1146.2 Field Documentation . . . 1148
8.1146.2.1 commonInfo . . . 1148
8.1146.2.2 pLTEInfo . . . 1148
8.1147 npack_nas_SLQSNasSwiOTAMessageCallback_ind_t Struct Reference . . . 1148
8.1147.1 Detailed Description . . . 1149
8.1147.2 Field Documentation . . . 1149
8.1147.2.1 Info . . . 1149
8.1147.2.2 Tlvresult . . . 1149
8.1148 npack_nas_SLQSNasTimerCallback_ind_t Struct Reference . . . 1149
8.1148.1 Detailed Description . . . 1149
8.1148.2 Field Documentation . . . 1149
8.1148.2.1 t3396_apn . . . 1149
8.1148.2.2 t3396_plmn_id . . . 1149
8.1148.2.3 t3396_val . . . 1149
8.1149 npack_nas_SLQSSetSysSelectionPrefCallBack_ind_t Struct Reference . . . 1150
8.1149.1 Detailed Description . . . 1150
8.1149.2 Field Documentation . . . 1150
8.1149.2.1 Info . . . 1150
8.1149.2.2 Tlvresult . . . 1150
8.1150 npack_nas_SLQSSwiGetHDRPersonality_t Struct Reference . . . 1150
8.1150.1 Detailed Description . . . 1150
8.1150.2 Field Documentation . . . 1151
8.1150.2.1 pCurrentPersonality . . . 1151
8.1150.2.2 pPersonalityListLength . . . 1151
8.1150.2.3 pProtocolSubtypeElement . . . 1151

8.1151 unpack_nas_SLQSSwiGetHDRProtSubtype_t Struct Reference . . . 1151
8.1151.1 Detailed Description . . . 1151
8.1151.2 Field Documentation . . . 1151
8.1151.2.1 pAppSubType . . . 1151
8.1151.2.2 pCurrentPrsnlty . . . 1151
8.1151.2.3 pPersonalityListLength . . . 1151
8.1151.2.4 pProtoSubTypElmnt . . . 1151
8.1152 unpack_nas_SLQSSwiGetHRPDStats_t Struct Reference . . . 1152
8.1152.1 Detailed Description . . . 1152
8.1152.2 Field Documentation . . . 1152
8.1152.2.1 pDRCParams . . . 1152
8.1152.2.2 pPilotSetData . . . 1152
8.1152.2.3 pUATI . . . 1152
8.1153 unpack_nas_SLQSSwiGetLteCQI_t Struct Reference . . . 1152
8.1153.1 Detailed Description . . . 1152
8.1153.2 Field Documentation . . . 1153
8.1153.2.1 CQIValueCW0 . . . 1153
8.1153.2.2 CQIValueCW1 . . . 1153
8.1153.2.3 ValidityCW0 . . . 1153
8.1153.2.4 ValidityCW1 . . . 1153
8.1154 unpack_nas_SLQSSwiGetLteSccRxInfo_t Struct Reference . . . 1153
8.1154.1 Detailed Description . . . 1153
8.1154.2 Field Documentation . . . 1153
8.1154.2.1 pSccRxInfo . . . 1153
8.1155 unpack_nas_SLQSSwiNetworkDebug_t Struct Reference . . . 1154
8.1155.1 Detailed Description . . . 1154
8.1155.2 Field Documentation . . . 1154
8.1155.2.1 pDataStatusDetail . . . 1154
8.1155.2.2 pDeviceConfigDetail . . . 1154
8.1155.2.3 pNetworkStat1x . . . 1154
8.1155.2.4 pNetworkStatEVDO . . . 1154
8.1155.2.5 pObjectVer . . . 1154
8.1156 unpack_nas_SLQSSysInfoCallback_ind_t Struct Reference . . . 1154
8.1156.1 Detailed Description . . . 1155
8.1156.2 Field Documentation . . . 1157
8.1156.2.1 pAddCDMASysInfo . . . 1157
8.1156.2.2 pAddGSMSysInfo . . . 1157
8.1156.2.3 pAddHDRSysInfo . . . 1157
8.1156.2.4 pAddLTESysInfo . . . 1157
8.1156.2.5 pAddWCDMASysInfo . . . 1157

8.1156.2.6 pCDMASrvStatusInfo . . . 1157
8.1156.2.7 pCDMASysInfo . . . 1157
8.1156.2.8 pGSMCallBarringSysInfo . . . 1157
8.1156.2.9 pGSMCipherDomainSysInfo . . . 1157
8.1156.2.10 pGSMSrvStatusInfo . . . 1157
8.1156.2.11 pGSMSysInfo . . . 1157
8.1156.2.12 pHDRSrvStatusInfo . . . 1157
8.1156.2.13 pHDRSysInfo . . . 1157
8.1156.2.14 pLTESrvStatusInfo . . . 1157
8.1156.2.15 pLTESysInfo . . . 1157
8.1156.2.16 pLTEVoiceSupportSysInfo . . . 1157
8.1156.2.17 pSysInfoNoChange . . . 1157
8.1156.2.18 pWCDMACallBarringSysInfo . . . 1157
8.1156.2.19 pWCDMACipherDomainSysInfo . . . 1157
8.1156.2.20 pWCDMASrvStatusInfo . . . 1157
8.1156.2.21 pWCDMASysInfo . . . 1157
8.1157 unpack_omaDmConfigTlv_t Struct Reference . . . 1157
8.1157.1 Detailed Description . . . 1158
8.1157.2 Field Documentation . . . 1158
8.1157.2.1 alertmsg . . . 1158
8.1157.2.2 alertmsglength . . . 1158
8.1157.2.3 state . . . 1158
8.1157.2.4 userInputReq . . . 1158
8.1157.2.5 userInputTimeout . . . 1158
8.1158 unpack_omaDmFotaTlv_t Struct Reference . . . 1158
8.1158.1 Detailed Description . . . 1159
8.1158.2 Field Documentation . . . 1160
8.1158.2.1 description . . . 1160
8.1158.2.2 descriptionlength . . . 1160
8.1158.2.3 fwdloadsize . . . 1160
8.1158.2.4 fwloadComplete . . . 1160
8.1158.2.5 namelength . . . 1160
8.1158.2.6 package_name . . . 1160
8.1158.2.7 sessionType . . . 1160
8.1158.2.8 severity . . . 1160
8.1158.2.9 state . . . 1160
8.1158.2.10 updateCompleteStatus . . . 1160
8.1158.2.11 userInputReq . . . 1160
8.1158.2.12 userInputTimeout . . . 1160
8.1158.2.13 version . . . 1160

8.1158.2.1 versionlength . . . 1160
8.1159 unpack_omaDmNotificationsTlv_t Struct Reference . . . 1160
8.1159.1 Detailed Description . . . 1161
8.1159.2 Field Documentation . . . 1161
8.1159.2.1 notification . . . 1161
8.1159.2.2 sessionStatus . . . 1161
8.1160 unpack_pds_GetPDSDefaults_t Struct Reference . . . 1161
8.1160.1 Detailed Description . . . 1161
8.1160.2 Field Documentation . . . 1162
8.1160.2.1 pAccuracy . . . 1162
8.1160.2.2 pInterval . . . 1162
8.1160.2.3 pOperation . . . 1162
8.1160.2.4 pTimeout . . . 1162
8.1161 unpack_pds_GetPDSState_t Struct Reference . . . 1162
8.1161.1 Detailed Description . . . 1162
8.1161.2 Field Documentation . . . 1163
8.1161.2.1 pEnabledStatus . . . 1163
8.1161.2.2 pTrackingStatus . . . 1163
8.1162 unpack_pds_GetPortAutomaticTracking_t Struct Reference . . . 1163
8.1162.1 Detailed Description . . . 1163
8.1162.2 Field Documentation . . . 1163
8.1162.2.1 pbAuto . . . 1163
8.1163 unpack_pds_GetServiceAutomaticTracking_t Struct Reference . . . 1163
8.1163.1 Detailed Description . . . 1163
8.1163.2 Field Documentation . . . 1164
8.1163.2.1 pbAuto . . . 1164
8.1164 unpack_pds_GetXTRAAutomaticDownload_t Struct Reference . . . 1164
8.1164.1 Detailed Description . . . 1164
8.1164.2 Field Documentation . . . 1164
8.1164.2.1 pbEnabled . . . 1164
8.1164.2.2 pInterval . . . 1164
8.1165 unpack_pds_GetXTRANetwork_t Struct Reference . . . 1164
8.1165.1 Detailed Description . . . 1164
8.1165.2 Field Documentation . . . 1165
8.1165.2.1 pPreference . . . 1165
8.1166 unpack_pds_GetXTRAValidity_t Struct Reference . . . 1165
8.1166.1 Detailed Description . . . 1165
8.1166.2 Field Documentation . . . 1165
8.1166.2.1 pDuration . . . 1165
8.1166.2.2 pGPSWeek . . . 1165

8.1166.2.3 pGPSWeekOffset . . . 1165
8.1167 npack_pds_SetEventReport_Ind_t Struct Reference . . . 1165
8.1167.1 Detailed Description . . . 1166
8.1167.2 Field Documentation . . . 1167
8.1167.2.1 dLatitude . . . 1167
8.1167.2.2 dLongitude . . . 1167
8.1167.2.3 has_dLatitude . . . 1167
8.1167.2.4 has_dLongitude . . . 1167
8.1167.2.5 has_PositionDataNMEA . . . 1167
8.1167.2.6 has_posSrc . . . 1167
8.1167.2.7 has_SessionStatus . . . 1167
8.1167.2.8 PositionDataNMEA . . . 1167
8.1167.2.9 posSrc . . . 1167
8.1167.2.10 SessionStatus . . . 1167
8.1168 npack_pds_SetPdsState_Ind_t Struct Reference . . . 1167
8.1168.1 Detailed Description . . . 1167
8.1168.2 Field Documentation . . . 1168
8.1168.2.1 EnabledStatus . . . 1168
8.1168.2.2 TlvPresent . . . 1168
8.1168.2.3 TrackingStatus . . . 1168
8.1169 npack_pds_SLQSGetAGPSConfig_t Struct Reference . . . 1168
8.1169.1 Detailed Description . . . 1168
8.1169.2 Field Documentation . . . 1169
8.1169.2.1 pServerAddress . . . 1169
8.1169.2.2 pServerPort . . . 1169
8.1169.2.3 pServerURL . . . 1169
8.1169.2.4 pServerURLLength . . . 1169
8.1170 npack_pds_SLQSGetGPSStateInfo_t Struct Reference . . . 1169
8.1170.1 Detailed Description . . . 1169
8.1170.2 Field Documentation . . . 1172
8.1170.2.1 Altitude . . . 1172
8.1170.2.2 EngineState . . . 1172
8.1170.2.3 glo_almanac_sv_msk . . . 1172
8.1170.2.4 glo_ephemeris_sv_msk . . . 1172
8.1170.2.5 glo_health_sv_msk . . . 1172
8.1170.2.6 glo_visible_sv_msk . . . 1172
8.1170.2.7 gps_almanac_sv_msk . . . 1172
8.1170.2.8 gps_ephemeris_sv_msk . . . 1172
8.1170.2.9 gps_health_sv_msk . . . 1172
8.1170.2.10 gps_visible_sv_msk . . . 1172

8.1170.2.11 HorizontalUncertainty . . . 1172
8.1170.2.12 no_valid . . . 1172
8.1170.2.13 atitude . . . 1172
8.1170.2.14 ongitude . . . 1172
8.1170.2.15 bas_almanac_sv_msk . . . 1172
8.1170.2.16 bas_ephemeris_sv_msk . . . 1172
8.1170.2.17 bas_health_sv_msk . . . 1172
8.1170.2.18 bas_visible_sv_msk . . . 1172
8.1170.2.19 ime_uncert_ms . . . 1172
8.1170.2.20 imeStmp_gps_week . . . 1172
8.1170.2.21 imeStmp_tow_ms . . . 1172
8.1170.2.22 alidMask . . . 1173
8.1170.2.23 erticalUncertainty . . . 1173
8.1170.2.24 ra_start_gps_minutes . . . 1173
8.1170.2.25 ra_start_gps_week . . . 1173
8.1170.2.26 ra_valid_duration_hours . . . 1173
8.1171 npack_qmi_t Struct Reference . . . 1173
8.1171.1 Detailed Description . . . 1173
8.1171.2 Field Documentation . . . 1173
8.1171.2.1 msgid . . . 1173
8.1171.2.2 ype . . . 1173
8.1171.2.3 xid . . . 1173
8.1172 npack_qos_dataRate_t Struct Reference . . . 1173
8.1172.1 Detailed Description . . . 1173
8.1172.2 Field Documentation . . . 1174
8.1172.2.1 dataRateMax . . . 1174
8.1172.2.2 guaranteedRate . . . 1174
8.1173 npack_qos_IPv4Addr_t Struct Reference . . . 1174
8.1173.1 Detailed Description . . . 1174
8.1173.2 Field Documentation . . . 1174
8.1173.2.1 addr . . . 1174
8.1173.2.2 subnetMask . . . 1174
8.1174 npack_qos_IPv6Addr_t Struct Reference . . . 1174
8.1174.1 Detailed Description . . . 1174
8.1174.2 Field Documentation . . . 1174
8.1174.2.1 addr . . . 1175
8.1174.2.2 prefixLen . . . 1175
8.1175 npack_qos_IPv6TrafCls_t Struct Reference . . . 1175
8.1175.1 Detailed Description . . . 1175
8.1175.2 Field Documentation . . . 1175

8.1175.2.1 mask . . . 1175
8.1175.2.2 val . . . 1175
8.1176 npack_qos_pktErrRate_t Struct Reference . . . 1175
8.1176.1 Detailed Description . . . 1175
8.1176.2 Field Documentation . . . 1175
8.1176.2.1 exponent . . . 1175
8.1176.2.2 multiplier . . . 1176
8.1177 npack_qos_Port_t Struct Reference . . . 1176
8.1177.1 Detailed Description . . . 1176
8.1177.2 Field Documentation . . . 1176
8.1177.2.1 port . . . 1176
8.1177.2.2 range . . . 1176
8.1178 npack_qos_QosFlowInfo_t Struct Reference . . . 1176
8.1178.1 Detailed Description . . . 1176
8.1178.2 Field Documentation . . . 1177
8.1178.2.1 BearerID . . . 1177
8.1178.2.2 is_RxQFlowGranted_Available . . . 1177
8.1178.2.3 is_TxQFlowGranted_Available . . . 1177
8.1178.2.4 NumRxFilters . . . 1177
8.1178.2.5 NumTxFilters . . . 1177
8.1178.2.6 QFlowState . . . 1177
8.1178.2.7 RxQFilter . . . 1177
8.1178.2.8 RxQFlowGranted . . . 1177
8.1178.2.9 TxQFilter . . . 1177
8.1178.2.10 TxQFlowGranted . . . 1177
8.1179 npack_qos_QosFlowInfoState_t Struct Reference . . . 1178
8.1179.1 Detailed Description . . . 1178
8.1179.2 Field Documentation . . . 1178
8.1179.2.1 id . . . 1178
8.1179.2.2 isNewFlow . . . 1178
8.1179.2.3 state . . . 1178
8.1180 npack_qos_SLQSQosGetNetworkStatus_t Struct Reference . . . 1178
8.1180.1 Detailed Description . . . 1178
8.1180.2 Field Documentation . . . 1179
8.1180.2.1 NWQoSStatus . . . 1179
8.1181 npack_qos_SLQSQosSwiReadApnExtraParams_t Struct Reference . . . 1179
8.1181.1 Detailed Description . . . 1179
8.1181.2 Field Documentation . . . 1179
8.1181.2.1 ambr_dl . . . 1180
8.1181.2.2 ambr_dl_ext . . . 1180

8.1181.2.3 ambr_dl_ext2 . . . 1180
8.1181.2.4 ambr_ul . . . 1180
8.1181.2.5 ambr_ul_ext . . . 1180
8.1181.2.6 ambr_ul_ext2 . . . 1180
8.1181.2.7 apnId . . . 1180
8.1182 qmi_qos_SLQSQosSwiReadDataStats_t Struct Reference . . . 1180
8.1182.1 Detailed Description . . . 1180
8.1182.2 Field Documentation . . . 1181
8.1182.2.1 apnId . . . 1181
8.1182.2.2 numQosFlow . . . 1181
8.1182.2.3 qosFlow . . . 1181
8.1182.2.4 total_rx_bytes . . . 1181
8.1182.2.5 total_rx_pkt . . . 1181
8.1182.2.6 total_tx_bytes . . . 1181
8.1182.2.7 total_tx_bytes_drp . . . 1181
8.1182.2.8 total_tx_pkt . . . 1181
8.1182.2.9 total_tx_pkt_drp . . . 1181
8.1183 qmi_qos_SLQSSetQosEventCallback_ind_t Struct Reference . . . 1181
8.1183.1 Detailed Description . . . 1181
8.1183.2 Field Documentation . . . 1182
8.1183.2.1 NumFlows . . . 1182
8.1183.2.2 QosFlowInfo . . . 1182
8.1184 qmi_qos_SLQSSetQosNWStatusCallback_ind_t Struct Reference . . . 1182
8.1184.1 Detailed Description . . . 1182
8.1184.2 Field Documentation . . . 1182
8.1184.2.1 status . . . 1182
8.1185 qmi_qos_SLQSSetQosPriEventCallback_ind_t Struct Reference . . . 1182
8.1185.1 Detailed Description . . . 1182
8.1185.2 Field Documentation . . . 1182
8.1185.2.1 event . . . 1182
8.1186 qmi_qos_SLQSSetQosStatusCallback_ind_t Struct Reference . . . 1183
8.1186.1 Detailed Description . . . 1183
8.1186.2 Field Documentation . . . 1184
8.1186.2.1 event . . . 1184
8.1186.2.2 id . . . 1184
8.1186.2.3 reason . . . 1184
8.1186.2.4 status . . . 1184
8.1187 qmi_qos_swiQosFilter_t Struct Reference . . . 1184
8.1187.1 Detailed Description . . . 1185
8.1187.2 Field Documentation . . . 1186

8.1187.2.1 EspSpi . . . 1186
8.1187.2.2 Id . . . 1186
8.1187.2.3 Index . . . 1186
8.1187.2.4 IPv4DstAddr . . . 1186
8.1187.2.5 IPv4SrcAddr . . . 1186
8.1187.2.6 IPv4Tos . . . 1186
8.1187.2.7 IPv6DstAddr . . . 1187
8.1187.2.8 IPv6Label . . . 1187
8.1187.2.9 IPv6SrcAddr . . . 1187
8.1187.2.10 IPv6TrafCls . . . 1187
8.1187.2.11 is_EspSpi_Available . . . 1187
8.1187.2.12 is_Id_Available . . . 1187
8.1187.2.13 is_IPv4DstAddr_Available . . . 1187
8.1187.2.14 is_IPv4SrcAddr_Available . . . 1187
8.1187.2.15 is_IPv4Tos_Available . . . 1187
8.1187.2.16 is_IPv6DstAddr_Available . . . 1187
8.1187.2.17 is_IPv6Label_Available . . . 1187
8.1187.2.18 is_IPv6SrcAddr_Available . . . 1187
8.1187.2.19 is_IPv6TrafCls_Available . . . 1187
8.1187.2.20 is_NxtHdrProto_Available . . . 1187
8.1187.2.21 is_Precedence_Available . . . 1187
8.1187.2.22 is_TCPDstPort_Available . . . 1187
8.1187.2.23 is_TCPSrcPort_Available . . . 1187
8.1187.2.24 is_TranDstPort_Available . . . 1187
8.1187.2.25 is_TranSrcPort_Available . . . 1187
8.1187.2.26 is_UDPDstPort_Available . . . 1187
8.1187.2.27 is_UDPSrcPort_Available . . . 1187
8.1187.2.28 NxtHdrProto . . . 1187
8.1187.2.29 Precedence . . . 1187
8.1187.2.30 TCPDstPort . . . 1187
8.1187.2.31 TCPSrcPort . . . 1187
8.1187.2.32 TranDstPort . . . 1187
8.1187.2.33 TranSrcPort . . . 1187
8.1187.2.34 UDPDstPort . . . 1187
8.1187.2.35 UDPSrcPort . . . 1188
8.1187.2.36 Version . . . 1188
8.1188 unpack_qos_swiQosFlow_t Struct Reference . . . 1188
8.1188.1 Detailed Description . . . 1188
8.1188.2 Field Documentation . . . 1190
8.1188.2.1 DataRate . . . 1190

8.1188.2.2 index . . . 1190
8.1188.2.3 is_DataRate_Available . . . 1191
8.1188.2.4 is_Jitter_Available . . . 1191
8.1188.2.5 is_Latency_Available . . . 1191
8.1188.2.6 is_LteQci_Available . . . 1191
8.1188.2.7 is_MaxAllowedPktSz_Available . . . 1191
8.1188.2.8 is_MinPolicedPktSz_Available . . . 1191
8.1188.2.9 is_PktErrRate_Available . . . 1191
8.1188.2.10 is_ProfileId3GPP2_Available . . . 1191
8.1188.2.11 is_TokenBucket_Available . . . 1191
8.1188.2.12 is_TrafficClass_Available . . . 1191
8.1188.2.13 is_val_3GPP2Pri_Available . . . 1191
8.1188.2.14 is_val_3GPPImCn_Available . . . 1191
8.1188.2.15 is_val_3GPPResResidualBER_Available . . . 1191
8.1188.2.16 is_val_3GPPSigInd_Available . . . 1191
8.1188.2.17 is_val_3GPPTraHdlPri_Available . . . 1191
8.1188.2.18 Jitter . . . 1191
8.1188.2.19 Latency . . . 1191
8.1188.2.20 LteQci . . . 1191
8.1188.2.21 MaxAllowedPktSz . . . 1191
8.1188.2.22 MinPolicedPktSz . . . 1191
8.1188.2.23 PktErrRate . . . 1191
8.1188.2.24 ProfileId3GPP2 . . . 1191
8.1188.2.25 TokenBucket . . . 1191
8.1188.2.26 TrafficClass . . . 1191
8.1188.2.27 val_3GPP2Pri . . . 1191
8.1188.2.28 val_3GPPImCn . . . 1191
8.1188.2.29 val_3GPPResResidualBER . . . 1191
8.1188.2.30 val_3GPPSigInd . . . 1191
8.1188.2.31 val_3GPPTraHdlPri . . . 1192
8.1189 unpack_qos_tokenBucket_t Struct Reference . . . 1192
8.1189.1 Detailed Description . . . 1192
8.1189.2 Field Documentation . . . 1192
8.1189.2.1 bucketSz . . . 1192
8.1189.2.2 peakRate . . . 1192
8.1189.2.3 tokenRate . . . 1192
8.1190 unpack_qos_Tos_t Struct Reference . . . 1192
8.1190.1 Detailed Description . . . 1192
8.1190.2 Field Documentation . . . 1193
8.1190.2.1 mask . . . 1193

8.1190.2.2 val . . . 1193
8.1191 unpack_QosFlowStat_t Struct Reference . . . 1193
8.1191.1 Detailed Description . . . 1193
8.1191.2 Field Documentation . . . 1193
8.1191.2.1 bearerId . . . 1193
8.1191.2.2 tx_bytes . . . 1193
8.1191.2.3 tx_bytes_drp . . . 1193
8.1191.2.4 tx_pkt . . . 1193
8.1191.2.5 tx_pkt_drp . . . 1193
8.1192 unpack_rms_GetSMSWake_t Struct Reference . . . 1193
8.1192.1 Detailed Description . . . 1194
8.1192.2 Field Documentation . . . 1194
8.1192.2.1 enabled . . . 1194
8.1192.2.2 wake_mask . . . 1194
8.1193 unpack_RMTransferStatistics_ind_t Struct Reference . . . 1194
8.1193.1 Detailed Description . . . 1194
8.1193.2 Field Documentation . . . 1195
8.1193.2.1 RxDropConutTlv . . . 1195
8.1193.2.2 RxOkByteCountTlv . . . 1195
8.1193.2.3 RxOkConutTlv . . . 1195
8.1193.2.4 TxDropConutTlv . . . 1195
8.1193.2.5 TxOkByteCountTlv . . . 1195
8.1193.2.6 TxOkConutTlv . . . 1195
8.1194 unpack_sar_SLQSGetRfSarState_t Struct Reference . . . 1195
8.1194.1 Detailed Description . . . 1195
8.1194.2 Field Documentation . . . 1196
8.1194.2.1 pSarRFState . . . 1196
8.1195 unpack_sms_GetSMSCAddress_t Struct Reference . . . 1196
8.1195.1 Detailed Description . . . 1196
8.1195.2 Field Documentation . . . 1196
8.1195.2.1 addressSize . . . 1196
8.1195.2.2 pSMSCAddress . . . 1196
8.1195.2.3 pSMSCType . . . 1196
8.1195.2.4 typeSize . . . 1196
8.1196 unpack_sms_SaveSMS_t Struct Reference . . . 1196
8.1196.1 Detailed Description . . . 1196
8.1196.2 Field Documentation . . . 1197
8.1196.2.1 pMessageIndex . . . 1197
8.1197 unpack_sms_SendSMS_t Struct Reference . . . 1197
8.1197.1 Detailed Description . . . 1197

8.1197.2 Field Documentation . . . 1197
8.1197.2.1 messageFailureCode . . . 1197
8.1197.2.2 messageID . . . 1197
8.1198 qmi_wds_unpack_sms_SetNewSMSCallback_ind_t Struct Reference . . . 1197
8.1198.1 Detailed Description . . . 1197
8.1198.2 Field Documentation . . . 1198
8.1198.2.1 ETWSPLMNTlv . . . 1198
8.1198.2.2 ETWSTlv . . . 1198
8.1198.2.3 IMSTlv . . . 1198
8.1198.2.4 MMTlv . . . 1198
8.1198.2.5 NewMMTlv . . . 1198
8.1198.2.6 SMSCTlv . . . 1198
8.1198.2.7 TRMessageTlv . . . 1198
8.1199 qmi_wds_unpack_sms_SetNewSMSCallback_t Struct Reference . . . 1198
8.1199.1 Detailed Description . . . 1198
8.1200 qmi_wds_unpack_sms_SLQSDeleteSMS_t Struct Reference . . . 1199
8.1200.1 Detailed Description . . . 1199
8.1201 qmi_wds_unpack_sms_SLQSGetIndicationRegister_t Struct Reference . . . 1199
8.1201.1 Detailed Description . . . 1199
8.1201.2 Field Documentation . . . 1199
8.1201.2.1 pGetIndicationRegInfo . . . 1199
8.1202 qmi_wds_unpack_sms_SLQSGetMessageWaiting_t Struct Reference . . . 1199
8.1202.1 Detailed Description . . . 1199
8.1202.2 Field Documentation . . . 1200
8.1202.2.1 pGetMsgWaitingInfoResp . . . 1200
8.1203 qmi_wds_unpack_sms_SLQSGetSMS_t Struct Reference . . . 1200
8.1203.1 Detailed Description . . . 1200
8.1203.2 Field Documentation . . . 1200
8.1203.2.1 message . . . 1200
8.1203.2.2 messageFormat . . . 1200
8.1203.2.3 messageSize . . . 1200
8.1203.2.4 messageTag . . . 1201
8.1204 qmi_wds_unpack_sms_SLQSGetSmsBroadcastConfig_t Struct Reference . . . 1201
8.1204.1 Detailed Description . . . 1201
8.1204.2 Field Documentation . . . 1201
8.1204.2.1 pBroadcastConfig . . . 1201
8.1204.2.2 pCDMABroadcastConfig . . . 1201
8.1205 qmi_wds_unpack_sms_SLQSGetSMSList_t Struct Reference . . . 1201
8.1205.1 Detailed Description . . . 1201
8.1205.2 Field Documentation . . . 1202

8.1205.2.1 messageList . . . 1202
8.1205.2.2 messageListSize . . . 1202
8.1206 unpack_sms_SLQSGetTransLayerInfo_t Struct Reference . . . 1202
8.1206.1 Detailed Description . . . 1202
8.1206.2 Field Documentation . . . 1202
8.1206.2.1 pGetTransLayerInfo . . . 1202
8.1207 unpack_sms_SLQSGetTransNWRegInfo_t Struct Reference . . . 1202
8.1207.1 Detailed Description . . . 1202
8.1207.2 Field Documentation . . . 1202
8.1207.2.1 pGetTransNWRegInfo . . . 1202
8.1208 unpack_sms_SLQSModifySMSStatus_t Struct Reference . . . 1202
8.1208.1 Detailed Description . . . 1203
8.1209 unpack_sms_SLQSSmsGetMaxStorageSize_t Struct Reference . . . 1203
8.1209.1 Detailed Description . . . 1203
8.1209.2 Field Documentation . . . 1203
8.1209.2.1 pMaxStorageSizeResp . . . 1203
8.1210 unpack_sms_SLQSSmsGetMessageProtocol_t Struct Reference . . . 1203
8.1210.1 Detailed Description . . . 1203
8.1210.2 Field Documentation . . . 1203
8.1210.2.1 pMessageProtocol . . . 1203
8.1211 unpack_sms_SLQSSwiGetSMSStorage_t Struct Reference . . . 1204
8.1211.1 Detailed Description . . . 1204
8.1211.2 Field Documentation . . . 1204
8.1211.2.1 pSmsStorage . . . 1204
8.1212 unpack_sms_SLQSWmsMemoryFullCallBack_ind_t Struct Reference . . . 1204
8.1212.1 Detailed Description . . . 1204
8.1212.2 Field Documentation . . . 1204
8.1212.2.1 messageMode . . . 1204
8.1212.2.2 storageType . . . 1205
8.1213 unpack_swiloc_SwiLocGetAutoStart_t Struct Reference . . . 1205
8.1213.1 Detailed Description . . . 1205
8.1213.2 Field Documentation . . . 1206
8.1213.2.1 fix_rate . . . 1206
8.1213.2.2 fix_rate_reported . . . 1206
8.1213.2.3 fix_type . . . 1206
8.1213.2.4 fix_type_reported . . . 1206
8.1213.2.5 function . . . 1206
8.1213.2.6 function_reported . . . 1206
8.1213.2.7 max_dist . . . 1206
8.1213.2.8 max_dist_reported . . . 1206

8.1213.2.9 max_time . . . 1206
8.1213.2.10 max_time_reported . . . 1206
8.1214 qmi_qmanpack_swioma_SLQSOMADMAlertCallback_ind_t Struct Reference . . . 1206
8.1214.1 Detailed Description . . . 1206
8.1214.2 Field Documentation . . . 1207
8.1214.2.1 eventType . . . 1207
8.1214.2.2 SessionInfoConfig . . . 1207
8.1214.2.3 SessionInfoFota . . . 1207
8.1214.2.4 SessionInfoNotification . . . 1207
8.1215 qmi_qmanpack_swioma_SLQSOMADMGetSessionInfo_t Struct Reference . . . 1207
8.1215.1 Detailed Description . . . 1207
8.1215.2 Field Documentation . . . 1209
8.1215.2.1 Date . . . 1209
8.1215.2.2 DateLength . . . 1209
8.1215.2.3 PkgDescLength . . . 1209
8.1215.2.4 PkgDescription . . . 1209
8.1215.2.5 PkgName . . . 1209
8.1215.2.6 PkgNameLength . . . 1209
8.1215.2.7 RetryCount . . . 1209
8.1215.2.8 SessionState . . . 1209
8.1215.2.9 SessionType . . . 1209
8.1215.2.10 Severity . . . 1209
8.1215.2.11 Source . . . 1209
8.1215.2.12 SourceLength . . . 1209
8.1215.2.13 Status . . . 1210
8.1215.2.14 Time . . . 1210
8.1215.2.15 TimeLength . . . 1210
8.1215.2.16 UpdateCompleteStatus . . . 1210
8.1216 qmi_qmanpack_swioma_SLQSOMADMGetSettings_t Struct Reference . . . 1210
8.1216.1 Detailed Description . . . 1210
8.1216.2 Field Documentation . . . 1211
8.1216.2.1 Autosdm . . . 1211
8.1216.2.2 FOTAdownload . . . 1211
8.1216.2.3 FOTAUpdate . . . 1211
8.1216.2.4 FwAutoCheck . . . 1211
8.1216.2.5 OMADMEnabled . . . 1211
8.1217 qmi_qmanpack_swioma_SLQSOMADMStartSession_t Struct Reference . . . 1211
8.1217.1 Detailed Description . . . 1211
8.1217.2 Field Documentation . . . 1212
8.1217.2.1 FwAvailability . . . 1212

8.1218 unpack_uim_ChangePin_t Struct Reference . . . 1212
8.1218.1 Detailed Description . . . 1212
8.1218.2 Field Documentation . . . 1212
8.1218.2.1 pEncryptedPIN1 . . . 1212
8.1218.2.2 pIndicationToken . . . 1213
8.1218.2.3 pRemainingRetries . . . 1213
8.1218.2.4 Tlvresult . . . 1213
8.1219 unpack_uim_GetCardStatus_t Struct Reference . . . 1213
8.1219.1 Detailed Description . . . 1213
8.1219.2 Field Documentation . . . 1213
8.1219.2.1 pCardStatus . . . 1213
8.1219.2.2 pHotSwapStatus . . . 1213
8.1219.2.3 Tlvresult . . . 1213
8.1220 unpack_uim_ReadTransparent_t Struct Reference . . . 1213
8.1220.1 Detailed Description . . . 1214
8.1220.2 Field Documentation . . . 1214
8.1220.2.1 pCardResult . . . 1214
8.1220.2.2 pEncryptedData . . . 1214
8.1220.2.3 pIndicationToken . . . 1214
8.1220.2.4 pReadResult . . . 1214
8.1220.2.5 Tlvresult . . . 1214
8.1221 unpack_uim_SetPinProtection_t Struct Reference . . . 1214
8.1221.1 Detailed Description . . . 1214
8.1221.2 Field Documentation . . . 1215
8.1221.2.1 pEncryptedPIN1 . . . 1215
8.1221.2.2 pIndicationToken . . . 1215
8.1221.2.3 pRemainingRetries . . . 1215
8.1221.2.4 Tlvresult . . . 1215
8.1222 unpack_uim_SetUimSlotStatusChangeCallback_ind_t Struct Reference . . . 1215
8.1222.1 Detailed Description . . . 1215
8.1222.2 Field Documentation . . . 1215
8.1222.2.1 bNumberOfPhySlots . . . 1216
8.1222.2.2 slotsstatusChange . . . 1216
8.1223 unpack_uim_SLQSUIMAuthenticate_t Struct Reference . . . 1216
8.1223.1 Detailed Description . . . 1216
8.1223.2 Field Documentation . . . 1216
8.1223.2.1 pAuthenticateResult . . . 1216
8.1223.2.2 pCardResult . . . 1216
8.1223.2.3 pIndicationToken . . . 1216
8.1224 unpack_uim_SLQSUIMDepersonalization_t Struct Reference . . . 1216

8.1224.1 Detailed Description . . . 1216
8.1224.2 Field Documentation . . . 1217
8.1224.2.1 pRemainingRetries . . . 1217
8.1225 unpack_uim_SLQSUIMEventRegister_t Struct Reference . . . 1217
8.1225.1 Detailed Description . . . 1217
8.1225.2 Field Documentation . . . 1217
8.1225.2.1 eventMask . . . 1217
8.1226 unpack_uim_SLQSUIMGetConfiguration_t Struct Reference . . . 1217
8.1226.1 Detailed Description . . . 1217
8.1226.2 Field Documentation . . . 1218
8.1226.2.1 pAutoSelection . . . 1218
8.1226.2.2 pHaltSubscription . . . 1218
8.1226.2.3 pPersonalizationStatus . . . 1218
8.1227 unpack_uim_SLQSUIMGetFileAttributes_t Struct Reference . . . 1218
8.1227.1 Detailed Description . . . 1218
8.1227.2 Field Documentation . . . 1219
8.1227.2.1 pCardResult . . . 1219
8.1227.2.2 pFileAttributes . . . 1219
8.1227.2.3 pIndicationToken . . . 1219
8.1228 unpack_uim_SLQSUIMGetSlotsStatus_t Struct Reference . . . 1219
8.1228.1 Detailed Description . . . 1219
8.1228.2 Field Documentation . . . 1219
8.1228.2.1 pNumberOfPhySlot . . . 1219
8.1228.2.2 pUimSlotsStatus . . . 1219
8.1229 unpack_uim_SLQSUIMRefreshGetLastEvent_t Struct Reference . . . 1219
8.1229.1 Detailed Description . . . 1220
8.1229.2 Field Documentation . . . 1220
8.1229.2.1 pRefreshEvent . . . 1220
8.1230 unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t Struct Reference . . . 1220
8.1230.1 Detailed Description . . . 1220
8.1230.2 Field Documentation . . . 1220
8.1230.2.1 pCardStatus . . . 1220
8.1231 unpack_uim_UnblockPin_t Struct Reference . . . 1220
8.1231.1 Detailed Description . . . 1220
8.1231.2 Field Documentation . . . 1221
8.1231.2.1 pEncryptedPIN1 . . . 1221
8.1231.2.2 pIndicationToken . . . 1221
8.1231.2.3 pRemainingRetries . . . 1221
8.1231.2.4 Tlvresult . . . 1221
8.1232 unpack_uim_VerifyPin_t Struct Reference . . . 1221

8.1232.1 Detailed Description . . . 1221
8.1232.2 Field Documentation . . . 1222
8.1232.2.1 pEncryptedPIN1 . . . 1222
8.1232.2.2 pIndicationToken . . . 1222
8.1232.2.3 pRemainingRetries . . . 1222
8.1232.2.4 Tlvresult . . . 1222
8.1233 unpack_voice_SLQSOriginateUSSD_t Struct Reference . . . 1222
8.1233.1 Detailed Description . . . 1222
8.1233.2 Field Documentation . . . 1225
8.1233.2.1 pAlphaIDInfo . . . 1225
8.1233.2.2 pCallId . . . 1225
8.1233.2.3 pCcResultType . . . 1225
8.1233.2.4 pCCSuppsType . . . 1225
8.1233.2.5 pfailureCause . . . 1225
8.1233.2.6 pUSSDInfo . . . 1225
8.1234 unpack_voice_SLQSVoiceAnswerCall_t Struct Reference . . . 1225
8.1234.1 Detailed Description . . . 1225
8.1234.2 Field Documentation . . . 1226
8.1234.2.1 pCallId . . . 1226
8.1235 unpack_voice_SLQSVoiceBurstDTMF_t Struct Reference . . . 1226
8.1235.1 Detailed Description . . . 1226
8.1235.2 Field Documentation . . . 1226
8.1235.2.1 pCallID . . . 1226
8.1236 unpack_voice_SLQSVoiceDialCall_t Struct Reference . . . 1226
8.1236.1 Detailed Description . . . 1226
8.1236.2 Field Documentation . . . 1227
8.1236.2.1 pAlphaIDInfo . . . 1227
8.1236.2.2 pCallID . . . 1227
8.1236.2.3 pCCResultType . . . 1227
8.1236.2.4 pCCSUPSType . . . 1227
8.1237 unpack_voice_SLQSVoiceEndCall_t Struct Reference . . . 1227
8.1237.1 Detailed Description . . . 1227
8.1237.2 Field Documentation . . . 1228
8.1237.2.1 pCallId . . . 1228
8.1238 unpack_voice_SLQSVoiceGetAllCallInfo_t Struct Reference . . . 1228
8.1238.1 Detailed Description . . . 1228
8.1238.2 Field Documentation . . . 1230
8.1238.2.1 pArrAlertingPattern . . . 1230
8.1238.2.2 pArrAlertingType . . . 1230
8.1238.2.3 pArrAlphaID . . . 1230

8.1238.2.4 pArrCalledPartyNum . . . 1230
8.1238.2.5 pArrCallEndReason . . . 1230
8.1238.2.6 pArrCallInfo . . . 1230
8.1238.2.7 pArrConnectPartyNum . . . 1230
8.1238.2.8 pArrDiagInfo . . . 1230
8.1238.2.9 pArrRedirPartyNum . . . 1230
8.1238.2.10 pArrRemotePartyName . . . 1230
8.1238.2.11 pArrRemotePartyNum . . . 1230
8.1238.2.12 pArrSvcOption . . . 1230
8.1238.2.13 pArrUUSInfo . . . 1230
8.1238.2.14 pOTASPStatus . . . 1230
8.1238.2.15 pVoicePrivacy . . . 1230
8.1239 unpack_voice_SLQSVoiceGetCallBarring_t Struct Reference . . . 1230
8.1239.1 Detailed Description . . . 1231
8.1239.2 Field Documentation . . . 1232
8.1239.2.1 pAlphaIDInfo . . . 1232
8.1239.2.2 pCallID . . . 1232
8.1239.2.3 pCCResType . . . 1232
8.1239.2.4 pCCSUPSType . . . 1232
8.1239.2.5 pFailCause . . . 1232
8.1239.2.6 pSvcClass . . . 1232
8.1240 unpack_voice_SLQSVoiceGetCallForwardingStatus_t Struct Reference . . . 1232
8.1240.1 Detailed Description . . . 1232
8.1240.2 Field Documentation . . . 1233
8.1240.2.1 pAlphaIDInfo . . . 1233
8.1240.2.2 pCallID . . . 1233
8.1240.2.3 pCCResType . . . 1233
8.1240.2.4 pCCSUPSType . . . 1233
8.1240.2.5 pFailCause . . . 1233
8.1240.2.6 pGetCallFWExtInfo . . . 1233
8.1240.2.7 pGetCallFWInfo . . . 1233
8.1241 unpack_voice_SLQSVoiceGetCallInfo_t Struct Reference . . . 1233
8.1241.1 Detailed Description . . . 1234
8.1241.2 Field Documentation . . . 1236
8.1241.2.1 pAlertingPattern . . . 1236
8.1241.2.2 pAlertType . . . 1236
8.1241.2.3 pAlphaIDInfo . . . 1236
8.1241.2.4 pCallInfo . . . 1236
8.1241.2.5 pConnectNumInfo . . . 1236
8.1241.2.6 pDiagInfo . . . 1236

8.1241.2.7 pOTASPStatus . . . 1236
8.1241.2.8 pRemotePartyName . . . 1236
8.1241.2.9 pRemotePartyNum . . . 1236
8.1241.2.10 pSrvOpt . . . 1236
8.1241.2.11 pUUSInfo . . . 1236
8.1241.2.12 pVoicePrivacy . . . 1236
8.1242 nnpack_voice_SLQSVoiceGetCallWaiting_t Struct Reference . . . 1237
8.1242.1 Detailed Description . . . 1237
8.1242.2 Field Documentation . . . 1238
8.1242.2.1 pAlphaIDInfo . . . 1238
8.1242.2.2 pCallID . . . 1238
8.1242.2.3 pCCResType . . . 1238
8.1242.2.4 pCCSUPSType . . . 1238
8.1242.2.5 pFailCause . . . 1238
8.1242.2.6 pSvcClass . . . 1238
8.1243 nnpack_voice_SLQSVoiceGetCLIP_t Struct Reference . . . 1238
8.1243.1 Detailed Description . . . 1238
8.1243.2 Field Documentation . . . 1239
8.1243.2.1 pAlphaIDInfo . . . 1239
8.1243.2.2 pCallID . . . 1239
8.1243.2.3 pCCResType . . . 1239
8.1243.2.4 pCCSUPSType . . . 1239
8.1243.2.5 pCLIPResp . . . 1239
8.1243.2.6 pFailCause . . . 1239
8.1244 nnpack_voice_SLQSVoiceGetCLIR_t Struct Reference . . . 1239
8.1244.1 Detailed Description . . . 1240
8.1244.2 Field Documentation . . . 1240
8.1244.2.1 pAlphaIDInfo . . . 1240
8.1244.2.2 pCallID . . . 1240
8.1244.2.3 pCCResType . . . 1241
8.1244.2.4 pCCSUPSType . . . 1241
8.1244.2.5 pCLIRResp . . . 1241
8.1244.2.6 pFailCause . . . 1241
8.1245 nnpack_voice_SLQSVoiceGetCNAP_t Struct Reference . . . 1241
8.1245.1 Detailed Description . . . 1241
8.1245.2 Field Documentation . . . 1242
8.1245.2.1 pAlphaIDInfo . . . 1242
8.1245.2.2 pCallID . . . 1242
8.1245.2.3 pCCResType . . . 1242
8.1245.2.4 pCCSUPSType . . . 1242

8.1245.2.5 pCNAPResp . . . 1242
8.1245.2.6 pFailCause . . . 1242
8.1246 unpack_voice_SLQSVoiceGetCOLP_t Struct Reference . . . 1242
8.1246.1 Detailed Description . . . 1242
8.1246.2 Field Documentation . . . 1243
8.1246.2.1 pAlphaIDInfo . . . 1243
8.1246.2.2 pCallID . . . 1243
8.1246.2.3 pCCResType . . . 1243
8.1246.2.4 pCCSUPSType . . . 1243
8.1246.2.5 pCOLPResp . . . 1243
8.1246.2.6 pFailCause . . . 1243
8.1247 unpack_voice_SLQSVoiceGetCOLR_t Struct Reference . . . 1243
8.1247.1 Detailed Description . . . 1244
8.1247.2 Field Documentation . . . 1245
8.1247.2.1 pAlphaIDInfo . . . 1245
8.1247.2.2 pCallID . . . 1245
8.1247.2.3 pCCResType . . . 1245
8.1247.2.4 pCCSUPSType . . . 1245
8.1247.2.5 pCOLRResp . . . 1245
8.1247.2.6 pFailCause . . . 1245
8.1248 unpack_voice_SLQSVoiceGetConfig_t Struct Reference . . . 1245
8.1248.1 Detailed Description . . . 1245
8.1248.2 Field Documentation . . . 1247
8.1248.2.1 pAirTimerCnt . . . 1247
8.1248.2.2 pAutoAnswerStat . . . 1247
8.1248.2.3 pCurAMRConfig . . . 1247
8.1248.2.4 pCurPrefVoiceSO . . . 1247
8.1248.2.5 pCurrTTYMode . . . 1247
8.1248.2.6 pCurVoiceDomainPref . . . 1247
8.1248.2.7 pCurVoicePrivacyPref . . . 1247
8.1248.2.8 pRoamTimerCnt . . . 1247
8.1249 unpack_voice_SLQSVoiceManageCalls_t Struct Reference . . . 1247
8.1249.1 Detailed Description . . . 1247
8.1249.2 Field Documentation . . . 1247
8.1249.2.1 pFailCause . . . 1247
8.1250 unpack_voice_SLQSVoiceSendFlash_t Struct Reference . . . 1247
8.1250.1 Detailed Description . . . 1247
8.1250.2 Field Documentation . . . 1248
8.1250.2.1 pCallID . . . 1248
8.1251 unpack_voice_SLQSVoiceSetCallBarringPassword_t Struct Reference . . . 1248

8.1251.1 Detailed Description . . . 1248
8.1251.2 Field Documentation . . . 1249
8.1251.2.1 pAlphaIDInfo . . . 1249
8.1251.2.2 pCallID . . . 1249
8.1251.2.3 pCCResType . . . 1249
8.1251.2.4 pCCSUPSType . . . 1249
8.1251.2.5 pFailCause . . . 1249
8.1252 unpack_voice_SLQSVoiceSetConfig_t Struct Reference . . . 1249
8.1252.1 Detailed Description . . . 1249
8.1252.2 Field Documentation . . . 1250
8.1252.2.1 pAirTimerStatus . . . 1250
8.1252.2.2 pAutoAnsStatus . . . 1250
8.1252.2.3 pPrefVoiceSOStatus . . . 1251
8.1252.2.4 pRoamTimerStatus . . . 1251
8.1252.2.5 pTTYConfigStatus . . . 1251
8.1252.2.6 pVoiceDomainPrefStatus . . . 1251
8.1253 unpack_voice_SLQSVoiceSetSUPSService_t Struct Reference . . . 1251
8.1253.1 Detailed Description . . . 1251
8.1253.2 Field Documentation . . . 1252
8.1253.2.1 pAlphaIDInfo . . . 1252
8.1253.2.2 pCallID . . . 1252
8.1253.2.3 pCCResultType . . . 1252
8.1253.2.4 pCCSUPSType . . . 1252
8.1253.2.5 pFailCause . . . 1252
8.1254 unpack_voice_SLQSVoiceStartContDTMF_t Struct Reference . . . 1252
8.1254.1 Detailed Description . . . 1252
8.1254.2 Field Documentation . . . 1252
8.1254.2.1 pCallID . . . 1252
8.1255 unpack_voice_SLQSVoiceStopContDTMF_t Struct Reference . . . 1252
8.1255.1 Detailed Description . . . 1253
8.1255.2 Field Documentation . . . 1253
8.1255.2.1 callID . . . 1253
8.1256 unpack_wds_DHCPv4ClientLease_ind_t Struct Reference . . . 1253
8.1256.1 Detailed Description . . . 1253
8.1256.2 Field Documentation . . . 1253
8.1256.2.1 DHCPv4LeaseOptTlv . . . 1253
8.1256.2.2 DHCPv4LeaseStateTlv . . . 1253
8.1256.2.3 IPv4AddrTlv . . . 1253
8.1256.2.4 ProfileIdTlv . . . 1253
8.1257 unpack_wds_GetAutoconnect_t Struct Reference . . . 1253

8.1257.1 Detailed Description . . . 1254
8.1257.2 Field Documentation . . . 1254
8.1257.2.1 psetting . . . 1254
8.1258 unpack_wds_GetByteTotals_t Struct Reference . . . 1254
8.1258.1 Detailed Description . . . 1254
8.1258.2 Field Documentation . . . 1254
8.1258.2.1 pRXTotalBytes . . . 1254
8.1258.2.2 pTXTotalBytes . . . 1254
8.1259 unpack_wds_GetConnectionRate_t Struct Reference . . . 1254
8.1259.1 Detailed Description . . . 1255
8.1259.2 Field Documentation . . . 1255
8.1259.2.1 currentChannelRXRate . . . 1255
8.1259.2.2 currentChannelTXRate . . . 1255
8.1259.2.3 maxChannelRXRate . . . 1255
8.1259.2.4 maxChannelTXRate . . . 1255
8.1260 unpack_wds_GetDataBearerTechnology_t Struct Reference . . . 1255
8.1260.1 Detailed Description . . . 1255
8.1260.2 Field Documentation . . . 1256
8.1260.2.1 pDataBearer . . . 1256
8.1261 unpack_wds_GetDefaultProfile_t Struct Reference . . . 1256
8.1261.1 Detailed Description . . . 1256
8.1261.2 Field Documentation . . . 1258
8.1261.2.1 apnname . . . 1258
8.1261.2.2 apnsize . . . 1258
8.1261.2.3 auth . . . 1258
8.1261.2.4 ipaddr . . . 1258
8.1261.2.5 ipaddrv6 . . . 1258
8.1261.2.6 name . . . 1258
8.1261.2.7 namesize . . . 1258
8.1261.2.8 pdptype . . . 1258
8.1261.2.9 pridns . . . 1258
8.1261.2.10 pridnsv6 . . . 1258
8.1261.2.11 secdns . . . 1258
8.1261.2.12 secdnsv6 . . . 1258
8.1261.2.13 username . . . 1258
8.1261.2.14 usersize . . . 1258
8.1262 unpack_wds_GetDefaultProfileNum_t Struct Reference . . . 1258
8.1262.1 Detailed Description . . . 1259
8.1262.2 Field Documentation . . . 1259
8.1262.2.1 index . . . 1259

8.1263 npack_wds_GetDormancyState_t Struct Reference . . . 1259
8.1263.1 Detailed Description . . . 1259
8.1263.2 Field Documentation . . . 1259
8.1263.2.1 dormancyState . . . 1259
8.1264 npack_wds_GetLastMobileIPError_t Struct Reference . . . 1259
8.1264.1 Detailed Description . . . 1259
8.1264.2 Field Documentation . . . 1260
8.1264.2.1 error . . . 1260
8.1265 npack_wds_GetMobileIP_t Struct Reference . . . 1260
8.1265.1 Detailed Description . . . 1260
8.1265.2 Field Documentation . . . 1260
8.1265.2.1 mipMode . . . 1260
8.1266 npack_wds_GetMobileIPProfile_t Struct Reference . . . 1260
8.1266.1 Detailed Description . . . 1261
8.1266.2 Field Documentation . . . 1262
8.1266.2.1 AAASPI . . . 1262
8.1266.2.2 AAAState . . . 1262
8.1266.2.3 address . . . 1262
8.1266.2.4 enabled . . . 1262
8.1266.2.5 HASPI . . . 1262
8.1266.2.6 HAState . . . 1262
8.1266.2.7 NAI . . . 1262
8.1266.2.8 naiSize . . . 1262
8.1266.2.9 primaryHA . . . 1262
8.1266.2.10 revTunneling . . . 1262
8.1266.2.11 secondaryHA . . . 1262
8.1267 npack_wds_GetPacketStatistics_t Struct Reference . . . 1262
8.1267.1 Detailed Description . . . 1262
8.1267.2 Field Documentation . . . 1264
8.1267.2.1 pRXDroppedCount . . . 1264
8.1267.2.2 pRXOkBytesCount . . . 1264
8.1267.2.3 pRXOKBytesLastCall . . . 1264
8.1267.2.4 pRXPacketErrors . . . 1264
8.1267.2.5 pRXPacketOverflows . . . 1264
8.1267.2.6 pRXPacketSuccesses . . . 1264
8.1267.2.7 pTXDroppedCount . . . 1264
8.1267.2.8 pTXOkBytesCount . . . 1264
8.1267.2.9 pTXOKBytesLastCall . . . 1264
8.1267.2.10 pTXPacketErrors . . . 1264
8.1267.2.11 pTXPacketOverflows . . . 1264

8.1267.2.12 pTXPacketSuccesses . . . 1264
8.1268 npack_wds_GetPacketStatus_t Struct Reference . . . 1264
8.1268.1 Detailed Description . . . 1264
8.1268.2 Field Documentation . . . 1265
8.1268.2.1 rXDroppedCount . . . 1265
8.1268.2.2 rXOkBytesCount . . . 1266
8.1268.2.3 rXOKBytesLastCall . . . 1266
8.1268.2.4 rXPacketErrors . . . 1266
8.1268.2.5 rXPacketOverflows . . . 1266
8.1268.2.6 rXPacketSuccesses . . . 1266
8.1268.2.7 tXDroppedCount . . . 1266
8.1268.2.8 tXOkBytesCount . . . 1266
8.1268.2.9 tXOKBytesLastCall . . . 1266
8.1268.2.10 tXPacketErrors . . . 1266
8.1268.2.11 tXPacketOverflows . . . 1266
8.1268.2.12 tXPacketSuccesses . . . 1266
8.1269 npack_wds_GetSessionDuration_t Struct Reference . . . 1266
8.1269.1 Detailed Description . . . 1266
8.1269.2 Field Documentation . . . 1266
8.1269.2.1 callDuration . . . 1266
8.1270 npack_wds_GetSessionState_t Struct Reference . . . 1266
8.1270.1 Detailed Description . . . 1266
8.1270.2 Field Documentation . . . 1267
8.1270.2.1 connectionStatus . . . 1267
8.1271 npack_wds_RMSetTransferStatistics_t Struct Reference . . . 1267
8.1271.1 Detailed Description . . . 1267
8.1272 npack_wds_SetMobileIPProfile_t Struct Reference . . . 1267
8.1272.1 Detailed Description . . . 1267
8.1273 npack_wds_SLQSCreateProfile_t Struct Reference . . . 1267
8.1273.1 Detailed Description . . . 1267
8.1273.2 Field Documentation . . . 1268
8.1273.2.1 pCreateProfileOut . . . 1268
8.1273.2.2 pProfileID . . . 1268
8.1273.2.3 Tlvresult . . . 1268
8.1274 npack_wds_SLQSDeleteProfile_t Struct Reference . . . 1268
8.1274.1 Detailed Description . . . 1268
8.1274.2 Field Documentation . . . 1268
8.1274.2.1 extendedErrorCode . . . 1268
8.1275 npack_wds_SLQSGet3GPPConfigItem_t Struct Reference . . . 1268
8.1275.1 Detailed Description . . . 1269

8.1275.2 Field Documentation . . . 1270
8.1275.2.1 _3gppRelease . . . 1270
8.1275.2.2 defaultPDNEnabled . . . 1270
8.1275.2.3 LTEAttachProfile . . . 1270
8.1275.2.4 LTEAttachProfileList . . . 1270
8.1275.2.5 LTEAttachProfileListLen . . . 1270
8.1275.2.6 profileList . . . 1270
8.1276 npack_wds_SLQSGetCurrDataSystemStat_t Struct Reference . . . 1270
8.1276.1 Detailed Description . . . 1270
8.1276.2 Field Documentation . . . 1271
8.1276.2.1 currNetworkInfo . . . 1271
8.1276.2.2 networkInfoLen . . . 1271
8.1276.2.3 prefNetwork . . . 1271
8.1277 npack_wds_SLQSGetCurrentChannelRate_t Struct Reference . . . 1271
8.1277.1 Detailed Description . . . 1271
8.1277.2 Field Documentation . . . 1271
8.1277.2.1 current_channel_rx_rate . . . 1271
8.1277.2.2 current_channel_tx_rate . . . 1272
8.1277.2.3 max_channel_rx_rate . . . 1272
8.1277.2.4 max_channel_tx_rate . . . 1272
8.1278 npack_wds_SLQSGetDataBearerTechnology_t Struct Reference . . . 1272
8.1278.1 Detailed Description . . . 1272
8.1278.2 Field Documentation . . . 1272
8.1278.2.1 curDataBearerTechnology . . . 1272
8.1278.2.2 dataBearerMask . . . 1272
8.1278.2.3 lastCallDataBearerTechnology . . . 1272
8.1279 npack_wds_SLQSGetDUNCallInfo_t Struct Reference . . . 1272
8.1279.1 Detailed Description . . . 1273
8.1279.2 Field Documentation . . . 1275
8.1279.2.1 callEndReason . . . 1275
8.1279.2.2 channelRate . . . 1275
8.1279.2.3 connectionStatus . . . 1275
8.1279.2.4 dataBearerTech . . . 1275
8.1279.2.5 dormancyStatus . . . 1275
8.1279.2.6 lastCallDataBearerTech . . . 1275
8.1279.2.7 lastCallRXOKBytesCnt . . . 1275
8.1279.2.8 lastCallTXOKBytesCnt . . . 1275
8.1279.2.9 mdmCallDurationActive . . . 1275
8.1279.2.10 rxOKBytesCount . . . 1275
8.1279.2.11 txOKBytesCount . . . 1275

8.1280 unpack_wds_SLQSGetProfileSettings_t Struct Reference . . . 1275
8.1280.1 Detailed Description . . . 1276
8.1280.2 Field Documentation . . . 1276
8.1280.2.1 pProfileSettings . . . 1276
8.1280.2.2 ProfileType . . . 1276
8.1280.2.3 Tlvresult . . . 1276
8.1281 unpack_wds_SLQSGetRuntimeSettings_t Struct Reference . . . 1276
8.1281.1 Detailed Description . . . 1277
8.1281.2 Field Documentation . . . 1278
8.1281.2.1 APNName . . . 1278
8.1281.2.2 Authentication . . . 1278
8.1281.2.3 DomainList . . . 1278
8.1281.2.4 GPRSGrantedQoS . . . 1278
8.1281.2.5 GWAddressV4 . . . 1278
8.1281.2.6 IMCNflag . . . 1278
8.1281.2.7 IPFamilyPreference . . . 1278
8.1281.2.8 IPv4 . . . 1278
8.1281.2.9 IPV6AddrInfo . . . 1278
8.1281.2.10 IPV6GWAddrInfo . . . 1278
8.1281.2.11 Mtu . . . 1278
8.1281.2.12 PCSCFAddrPCO . . . 1278
8.1281.2.13 PCSCFFQDNAddrList . . . 1278
8.1281.2.14 PDPType . . . 1279
8.1281.2.15 PrimaryDNSV4 . . . 1279
8.1281.2.16 PrimaryDNSV6 . . . 1279
8.1281.2.17 ProfileID . . . 1279
8.1281.2.18 ProfileName . . . 1279
8.1281.2.19 SecondaryDNSV4 . . . 1279
8.1281.2.20 SecondaryDNSV6 . . . 1279
8.1281.2.21 ServerAddrList . . . 1279
8.1281.2.22 SubnetMaskV4 . . . 1279
8.1281.2.23 Technology . . . 1279
8.1281.2.24 UMTSGrantedQoS . . . 1279
8.1281.2.25 Username . . . 1279
8.1282 unpack_wds_SLQSModifyProfile_t Struct Reference . . . 1279
8.1282.1 Detailed Description . . . 1279
8.1282.2 Field Documentation . . . 1279
8.1282.2.1 pExtErrorCode . . . 1279
8.1283 unpack_wds_SLQSSetIPFamilyPreference_t Struct Reference . . . 1279
8.1283.1 Detailed Description . . . 1280

8.1283.2 Field Documentation . . . 1280
8.1283.2.1 Tlvresult . . . 1280
8.1284 npack_wds_SLQSSetPacketSrvStatusCallback_t Struct Reference . . . 1280
8.1284.1 Detailed Description . . . 1280
8.1284.2 Field Documentation . . . 1281
8.1284.2.1 bearerID . . . 1281
8.1284.2.2 conn_status . . . 1281
8.1284.2.3 ipFamily . . . 1281
8.1284.2.4 reconfigReqd . . . 1281
8.1284.2.5 sessionEndReason . . . 1281
8.1284.2.6 techName . . . 1282
8.1284.2.7 verboseSessnEndReason . . . 1282
8.1284.2.8 verboseSessnEndReasonType . . . 1282
8.1285 npack_wds_SLQSSetWdsEventCallback_ind_t Struct Reference . . . 1282
8.1285.1 Detailed Description . . . 1282
8.1285.2 Field Documentation . . . 1283
8.1285.2.1 currDBTechAvail . . . 1283
8.1285.2.2 currNWInfo . . . 1283
8.1285.2.3 dataSysStatAvail . . . 1283
8.1285.2.4 dBTechAvail . . . 1283
8.1285.2.5 dBtechExtAvail . . . 1283
8.1285.2.6 dBTechExtRatValue . . . 1283
8.1285.2.7 dBTechExtSoMask . . . 1283
8.1285.2.8 dBTechnology . . . 1283
8.1285.2.9 dBtechnologyExt . . . 1283
8.1285.2.10 dormancyStatAvail . . . 1283
8.1285.2.11 dormancyStatus . . . 1283
8.1285.2.12 ipstatAvail . . . 1283
8.1285.2.13 ipStatus . . . 1283
8.1285.2.14 netInfoLen . . . 1283
8.1285.2.15 prefNetwork . . . 1283
8.1285.2.16 ratMask . . . 1283
8.1285.2.17 rx_bytes . . . 1283
8.1285.2.18 rx_pkts . . . 1283
8.1285.2.19 soMask . . . 1283
8.1285.2.20 tx_bytes . . . 1283
8.1285.2.21 tx_pkts . . . 1283
8.1285.2.22 txferStatAvail . . . 1283
8.1286 npack_wds_SLQSSGetDHCPv4ClientConfig_t Struct Reference . . . 1284
8.1286.1 Detailed Description . . . 1284

8.1286.2 Field Documentation . . . 1284
8.1286.2.1 pHwConfig . . . 1284
8.1286.2.2 pRequestOptionList . . . 1284
8.1287 unpack_wds_SLQSSGetLoopback_t Struct Reference . . . 1284
8.1287.1 Detailed Description . . . 1284
8.1287.2 Field Documentation . . . 1285
8.1287.2.1 ByteLoopbackMode . . . 1285
8.1287.2.2 ByteLoopbackMultiplier . . . 1285
8.1288 unpack_wds_SLQSStartDataSession_t Struct Reference . . . 1285
8.1288.1 Detailed Description . . . 1285
8.1288.2 Field Documentation . . . 1285
8.1288.2.1 pFailureReason . . . 1285
8.1288.2.2 psid . . . 1285
8.1288.2.3 pVerboseFailReasonType . . . 1285
8.1288.2.4 pVerboseFailureReason . . . 1285
8.1289 unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference . . . 1285
8.1289.1 Detailed Description . . . 1286
8.1289.2 Field Documentation . . . 1287
8.1289.2.1 apnName . . . 1287
8.1289.2.2 bearerId . . . 1287
8.1289.2.3 contextId . . . 1287
8.1289.2.4 ipv4Address . . . 1287
8.1289.2.5 ipv4GWAddress . . . 1287
8.1289.2.6 ipv6Address . . . 1287
8.1289.2.7 ipv6GWAddress . . . 1287
8.1289.2.8 prDNSIPv4Address . . . 1288
8.1289.2.9 prDNSIPv6Address . . . 1288
8.1289.2.10 prPCSCFIPv4Address . . . 1288
8.1289.2.11 prPCSCFIPv6Address . . . 1288
8.1289.2.12 seDNSIPv4Address . . . 1288
8.1289.2.13 seDNSIPv6Address . . . 1288
8.1289.2.14 sePCSCFIPv4Address . . . 1288
8.1289.2.15 sePCSCFIPv6Address . . . 1288
8.1290 UnPackGetProfileSettingOut Struct Reference . . . 1288
8.1290.1 Detailed Description . . . 1288
8.1290.2 Field Documentation . . . 1288
8.1290.2.1 curProfile . . . 1288
8.1290.2.2 pExtErrCode . . . 1288
8.1291 unpackWdsProfileParam Union Reference . . . 1288
8.1291.1 Detailed Description . . . 1289

8.1291.2 Field Documentation . . . 1289
8.1291.2.1 SlqsProfile3GPP . . . 1289
8.1291.2.2 SlqsProfile3GPP2 . . . 1289
8.1292 rlAddr Struct Reference . . . 1289
8.1292.1 Detailed Description . . . 1289
8.1292.2 Field Documentation . . . 1289
8.1292.2.1 address . . . 1289
8.1292.2.2 TlvPresent . . . 1289
8.1293 USBCompConfig Struct Reference . . . 1289
8.1293.1 Detailed Description . . . 1290
8.1293.2 Field Documentation . . . 1290
8.1293.2.1 pUSBComp . . . 1290
8.1294 USBCompParams Struct Reference . . . 1290
8.1294.1 Detailed Description . . . 1290
8.1294.2 Field Documentation . . . 1292
8.1294.2.1 pNumSupUSBComps . . . 1292
8.1294.2.2 pSupUSBComps . . . 1292
8.1294.2.3 pUSBComp . . . 1292
8.1295 USSDNoWaitIndicationInfo Struct Reference . . . 1292
8.1295.1 Detailed Description . . . 1292
8.1295.2 Field Documentation . . . 1293
8.1295.2.1 pAlphaIdentifier . . . 1293
8.1295.2.2 pError . . . 1293
8.1295.2.3 pFailureCause . . . 1293
8.1295.2.4 pUSSDData . . . 1293
8.1296 USSDRespFNetwork Struct Reference . . . 1293
8.1296.1 Detailed Description . . . 1293
8.1296.2 Field Documentation . . . 1293
8.1296.2.1 pRespData . . . 1293
8.1296.2.2 pTypeCode . . . 1293
8.1297 USSInfo Struct Reference . . . 1293
8.1297.1 Detailed Description . . . 1294
8.1297.2 Field Documentation . . . 1294
8.1297.2.1 ussData . . . 1294
8.1297.2.2 ussDCS . . . 1294
8.1297.2.3 ussLen . . . 1294
8.1298 USSResp Struct Reference . . . 1294
8.1298.1 Field Documentation . . . 1294
8.1298.1.1 pAlphaIDInfo . . . 1294
8.1298.1.2 pCallId . . . 1294

8.1298.1.3 pCcResultType . . . 1294
8.1298.1.4 pCCSuppsType . . . 1294
8.1298.1.5 pfailureCause . . . 1295
8.1298.1.6 pUSSDInfo . . . 1295
8.1299 UUSInfo Struct Reference . . . 1295
8.1299.1 Detailed Description . . . 1295
8.1299.2 Field Documentation . . . 1296
8.1299.2.1 UUSData . . . 1296
8.1299.2.2 UUSDatalen . . . 1296
8.1299.2.3 UUSDcs . . . 1296
8.1299.2.4 UUSType . . . 1296
8.1300 VerifyUIMPIN Struct Reference . . . 1296
8.1300.1 Detailed Description . . . 1296
8.1300.2 Field Documentation . . . 1296
8.1300.2.1 pinID . . . 1296
8.1300.2.2 pinLen . . . 1296
8.1300.2.3 pinVal . . . 1296
8.1301 voice_airTimer Struct Reference . . . 1296
8.1301.1 Detailed Description . . . 1297
8.1301.2 Field Documentation . . . 1297
8.1301.2.1 airTimerValue . . . 1297
8.1301.2.2 namID . . . 1297
8.1302 voice_allCallsAlphaIDInfo Struct Reference . . . 1297
8.1302.1 Detailed Description . . . 1297
8.1302.2 Field Documentation . . . 1297
8.1302.2.1 AlphaIDInfo . . . 1297
8.1302.2.2 callID . . . 1297
8.1303 voice_allCallsDiagInfo Struct Reference . . . 1297
8.1303.1 Detailed Description . . . 1298
8.1303.2 Field Documentation . . . 1298
8.1303.2.1 callID . . . 1298
8.1303.2.2 DiagInfo . . . 1298
8.1304 voice_allCallsUUSInfo Struct Reference . . . 1298
8.1304.1 Detailed Description . . . 1298
8.1304.2 Field Documentation . . . 1298
8.1304.2.1 callID . . . 1298
8.1304.2.2 uusInfo . . . 1298
8.1305 voice_alphaIDInfo Struct Reference . . . 1298
8.1305.1 Detailed Description . . . 1299
8.1305.2 Field Documentation . . . 1299

8.1305.2.1 alphaDcs . . . 1299
8.1305.2.2 alphaLen . . . 1299
8.1305.2.3 alphaText . . . 1299
8.1306 voice_arrAlertingPattern Struct Reference . . . 1299
8.1306.1 Detailed Description . . . 1299
8.1306.2 Field Documentation . . . 1300
8.1306.2.1 alertingPattern . . . 1300
8.1306.2.2 callID . . . 1300
8.1306.2.3 numInstances . . . 1300
8.1307 voice_arrAlertingType Struct Reference . . . 1300
8.1307.1 Detailed Description . . . 1300
8.1307.2 Field Documentation . . . 1301
8.1307.2.1 AlertingType . . . 1301
8.1307.2.2 callID . . . 1301
8.1307.2.3 numInstances . . . 1301
8.1308 voice_arrAlphaID Struct Reference . . . 1301
8.1308.1 Detailed Description . . . 1301
8.1308.2 Field Documentation . . . 1301
8.1308.2.1 allCallsAlphaIDInfoArr . . . 1301
8.1308.2.2 numInstances . . . 1301
8.1309 voice_arrCalledPartyNum Struct Reference . . . 1301
8.1309.1 Detailed Description . . . 1301
8.1309.2 Field Documentation . . . 1302
8.1309.2.1 CalledPartyNum . . . 1302
8.1309.2.2 numInstances . . . 1302
8.1310 voice_arrCallEndReason Struct Reference . . . 1302
8.1310.1 Detailed Description . . . 1302
8.1310.2 Field Documentation . . . 1302
8.1310.2.1 callEndReason . . . 1302
8.1310.2.2 callID . . . 1302
8.1310.2.3 numInstances . . . 1302
8.1311 voice_arrCallInfo Struct Reference . . . 1302
8.1311.1 Detailed Description . . . 1303
8.1311.2 Field Documentation . . . 1303
8.1311.2.1 getAllCallInfo . . . 1303
8.1311.2.2 numInstances . . . 1303
8.1312 voice_arrConnectPartyNum Struct Reference . . . 1303
8.1312.1 Detailed Description . . . 1303
8.1312.2 Field Documentation . . . 1303
8.1312.2.1 ConnectedPartyNum . . . 1303

8.1312.2.2 numInstances . . . 1303
8.1313 voice_arrDiagInfo Struct Reference . . . 1304
8.1313.1 Detailed Description . . . 1304
8.1313.2 Field Documentation . . . 1304
8.1313.2.1 DiagInfo . . . 1304
8.1313.2.2 numInstances . . . 1304
8.1314 voice_arrRedirPartyNum Struct Reference . . . 1304
8.1314.1 Detailed Description . . . 1304
8.1314.2 Field Documentation . . . 1304
8.1314.2.1 numInstances . . . 1305
8.1314.2.2 RedirPartyNum . . . 1305
8.1315 voice_arrRemotePartyName Struct Reference . . . 1305
8.1315.1 Detailed Description . . . 1305
8.1315.2 Field Documentation . . . 1305
8.1315.2.1 GetAllCallRmtPtyName . . . 1305
8.1315.2.2 numInstances . . . 1305
8.1316 voice_arrRemotePartyNum Struct Reference . . . 1305
8.1316.1 Detailed Description . . . 1305
8.1316.2 Field Documentation . . . 1306
8.1316.2.1 numInstances . . . 1306
8.1316.2.2 RmtPtyNum . . . 1306
8.1317 voice_arrSvcOption Struct Reference . . . 1306
8.1317.1 Detailed Description . . . 1306
8.1317.2 Field Documentation . . . 1306
8.1317.2.1 callID . . . 1306
8.1317.2.2 numInstances . . . 1306
8.1317.2.3 srvOption . . . 1306
8.1318 voice_arrUUSInfo Struct Reference . . . 1306
8.1318.1 Detailed Description . . . 1306
8.1318.2 Field Documentation . . . 1307
8.1318.2.1 AllCallsUUSInfo . . . 1307
8.1318.2.2 numInstances . . . 1307
8.1319 voice_burstDTMFInfo Struct Reference . . . 1307
8.1319.1 Detailed Description . . . 1307
8.1319.2 Field Documentation . . . 1307
8.1319.2.1 digitCnt . . . 1307
8.1319.2.2 pCallID . . . 1307
8.1319.2.3 pDigitBuff . . . 1308
8.1320 voice_calledPartySubAdd Struct Reference . . . 1308
8.1320.1 Detailed Description . . . 1308

8.1320.2 Field Documentation . . . 1308
8.1320.2.1 extBit . . . 1308
8.1320.2.2 oddEvenInd . . . 1308
8.1320.2.3 subAddr . . . 1308
8.1320.2.4 subAddrLen . . . 1308
8.1320.2.5 subAddrType . . . 1308
8.1321 voice_callFwdTypeAndPlan Struct Reference . . . 1309
8.1321.1 Detailed Description . . . 1309
8.1321.2 Field Documentation . . . 1309
8.1321.2.1 numberPlan . . . 1309
8.1321.2.2 numberType . . . 1309
8.1322 voice_callFWExtInfo Struct Reference . . . 1309
8.1322.1 Detailed Description . . . 1310
8.1322.2 Field Documentation . . . 1311
8.1322.2.1 noReplyTimer . . . 1311
8.1322.2.2 number . . . 1311
8.1322.2.3 numLen . . . 1311
8.1322.2.4 numPlan . . . 1311
8.1322.2.5 numType . . . 1311
8.1322.2.6 PI . . . 1311
8.1322.2.7 SI . . . 1311
8.1322.2.8 SvcClass . . . 1311
8.1322.2.9 SvcStatus . . . 1311
8.1323 voice_callFWInfo Struct Reference . . . 1311
8.1323.1 Detailed Description . . . 1312
8.1323.2 Field Documentation . . . 1312
8.1323.2.1 noReplyTimer . . . 1312
8.1323.2.2 number . . . 1312
8.1323.2.3 numLen . . . 1312
8.1323.2.4 SvcClass . . . 1312
8.1323.2.5 SvcStatus . . . 1312
8.1324 voice_callInfo Struct Reference . . . 1312
8.1324.1 Detailed Description . . . 1312
8.1324.2 Field Documentation . . . 1313
8.1324.2.1 callID . . . 1313
8.1324.2.2 callState . . . 1314
8.1324.2.3 callType . . . 1314
8.1324.2.4 direction . . . 1314
8.1324.2.5 mode . . . 1314
8.1325 voice_ccSUPSType Struct Reference . . . 1314

8.1325.1 Detailed Description . . . 1314
8.1325.2 Field Documentation . . . 1314
8.1325.2.1 reason . . . 1314
8.1325.2.2 svcType . . . 1314
8.1326 voice_CLIPResp Struct Reference . . . 1314
8.1326.1 Detailed Description . . . 1315
8.1326.2 Field Documentation . . . 1315
8.1326.2.1 ActiveStatus . . . 1315
8.1326.2.2 ProvisionStatus . . . 1315
8.1327 voice_CLIRResp Struct Reference . . . 1315
8.1327.1 Detailed Description . . . 1315
8.1327.2 Field Documentation . . . 1316
8.1327.2.1 ActiveStatus . . . 1316
8.1327.2.2 ProvisionStatus . . . 1316
8.1328 voice_CNAPResp Struct Reference . . . 1316
8.1328.1 Detailed Description . . . 1316
8.1328.2 Field Documentation . . . 1316
8.1328.2.1 ActiveStatus . . . 1316
8.1328.2.2 ProvisionStatus . . . 1316
8.1329 voice_COLPResp Struct Reference . . . 1317
8.1329.1 Detailed Description . . . 1317
8.1329.2 Field Documentation . . . 1317
8.1329.2.1 ActiveStatus . . . 1317
8.1329.2.2 ProvisionStatus . . . 1317
8.1330 voice_COLRResp Struct Reference . . . 1317
8.1330.1 Detailed Description . . . 1317
8.1330.2 Field Documentation . . . 1318
8.1330.2.1 ActiveStatus . . . 1318
8.1330.2.2 ProvisionStatus . . . 1318
8.1331 voice_connectNumInfo Struct Reference . . . 1318
8.1331.1 Detailed Description . . . 1318
8.1331.2 Field Documentation . . . 1319
8.1331.2.1 callerID . . . 1319
8.1331.2.2 callerIDLen . . . 1319
8.1331.2.3 numPlan . . . 1319
8.1331.2.4 numPresInd . . . 1320
8.1331.2.5 numType . . . 1320
8.1331.2.6 screeningInd . . . 1320
8.1332 voice_CUGInfo Struct Reference . . . 1320
8.1332.1 Detailed Description . . . 1320

8.1332.2 Field Documentation . . . 1320
8.1332.2.1 CUGIndex . . . 1320
8.1332.2.2 SuppOA . . . 1320
8.1332.2.3 SuppPrefCUG . . . 1320
8.1333 voice_curAMRConfig Struct Reference . . . 1320
8.1333.1 Detailed Description . . . 1320
8.1333.2 Field Documentation . . . 1321
8.1333.2.1 gsmAmrStat . . . 1321
8.1333.2.2 wcdmaAmrStat . . . 1321
8.1334 voice_diagInfo Struct Reference . . . 1321
8.1334.1 Detailed Description . . . 1321
8.1334.2 Field Documentation . . . 1321
8.1334.2.1 diagInfoLen . . . 1321
8.1334.2.2 diagnosticInfo . . . 1321
8.1335 voice_DTMFLengths Struct Reference . . . 1322
8.1335.1 Detailed Description . . . 1322
8.1335.2 Field Documentation . . . 1322
8.1335.2.1 DTMFInterdigitInterval . . . 1322
8.1335.2.2 DTMFPulseWidth . . . 1322
8.1336 voice_getAllCallInformation Struct Reference . . . 1322
8.1336.1 Detailed Description . . . 1322
8.1336.2 Field Documentation . . . 1323
8.1336.2.1 ALS . . . 1323
8.1336.2.2 CallInfo . . . 1323
8.1336.2.3 isEmpty . . . 1323
8.1337 voice_getAllCallRmtPtyName Struct Reference . . . 1323
8.1337.1 Detailed Description . . . 1323
8.1337.2 Field Documentation . . . 1323
8.1337.2.1 callID . . . 1323
8.1337.2.2 RemotePartyName . . . 1323
8.1338 voice_getAllCallRmtPtyNum Struct Reference . . . 1323
8.1338.1 Detailed Description . . . 1324
8.1338.2 Field Documentation . . . 1324
8.1338.2.1 callID . . . 1324
8.1338.2.2 RemotePartyNum . . . 1324
8.1339 voice_getCallFWExtInfo Struct Reference . . . 1324
8.1339.1 Detailed Description . . . 1324
8.1339.2 Field Documentation . . . 1324
8.1339.2.1 CallFWExtInfo . . . 1324
8.1339.2.2 numInstances . . . 1324

8.1340 voice_getCallFWInfo Struct Reference . . . 1325
8.1340.1 Detailed Description . . . 1325
8.1340.2 Field Documentation . . . 1325
8.1340.2.1 CallFWInfo . . . 1325
8.1340.2.2 numInstances . . . 1325
8.1341 voice_peerNumberInfo Struct Reference . . . 1325
8.1341.1 Detailed Description . . . 1325
8.1341.2 Field Documentation . . . 1327
8.1341.2.1 callID . . . 1327
8.1341.2.2 number . . . 1327
8.1341.2.3 numLen . . . 1327
8.1341.2.4 numPI . . . 1327
8.1341.2.5 numPlan . . . 1327
8.1341.2.6 numSI . . . 1327
8.1341.2.7 numType . . . 1327
8.1342 voice_prefVoiceSO Struct Reference . . . 1327
8.1342.1 Detailed Description . . . 1327
8.1342.2 Field Documentation . . . 1329
8.1342.2.1 evrcCapability . . . 1329
8.1342.2.2 homeOrigVoiceSO . . . 1329
8.1342.2.3 homePageVoiceSO . . . 1329
8.1342.2.4 namID . . . 1329
8.1342.2.5 roamOrigVoiceSO . . . 1329
8.1343 voice_remotePartyName Struct Reference . . . 1329
8.1343.1 Detailed Description . . . 1329
8.1343.2 Field Documentation . . . 1329
8.1343.2.1 callerName . . . 1330
8.1343.2.2 codingScheme . . . 1330
8.1343.2.3 nameLen . . . 1330
8.1343.2.4 namePI . . . 1330
8.1344 voice_remotePartyNum Struct Reference . . . 1330
8.1344.1 Detailed Description . . . 1330
8.1344.2 Field Documentation . . . 1330
8.1344.2.1 numLen . . . 1330
8.1344.2.2 presentationInd . . . 1330
8.1344.2.3 remPartyNumber . . . 1330
8.1345 voice_roamTimer Struct Reference . . . 1330
8.1345.1 Detailed Description . . . 1331
8.1345.2 Field Documentation . . . 1331
8.1345.2.1 namID . . . 1331

8.1345.2.2 roamTimerValue . . . 1331
8.1346 voice_USSInfo Struct Reference . . . 1331
8.1346.1 Detailed Description . . . 1331
8.1346.2 Field Documentation . . . 1331
8.1346.2.1 ussData . . . 1332
8.1346.2.2 ussDCS . . . 1332
8.1346.2.3 ussLen . . . 1332
8.1347 voice_UUSInfo Struct Reference . . . 1332
8.1347.1 Detailed Description . . . 1332
8.1347.2 Field Documentation . . . 1333
8.1347.2.1 UUSData . . . 1333
8.1347.2.2 UUSDatalen . . . 1333
8.1347.2.3 UUSDcs . . . 1333
8.1347.2.4 UUSType . . . 1333
8.1348 voiceALSSelectLineInfo Struct Reference . . . 1333
8.1348.1 Detailed Description . . . 1333
8.1348.2 Field Documentation . . . 1333
8.1348.2.1 lineValue . . . 1333
8.1349 voiceALSSetLineSwitchInfo Struct Reference . . . 1333
8.1349.1 Detailed Description . . . 1333
8.1349.2 Field Documentation . . . 1333
8.1349.2.1 switchOption . . . 1334
8.1350 voiceAnswerCall Struct Reference . . . 1334
8.1350.1 Detailed Description . . . 1334
8.1350.2 Field Documentation . . . 1334
8.1350.2.1 pCallId . . . 1334
8.1351 voiceBindSubscriptionInfo Struct Reference . . . 1334
8.1351.1 Detailed Description . . . 1334
8.1351.2 Field Documentation . . . 1334
8.1351.2.1 subsType . . . 1334
8.1352 voiceBurstDTMFInfo Struct Reference . . . 1334
8.1352.1 Detailed Description . . . 1335
8.1352.2 Field Documentation . . . 1335
8.1352.2.1 BurstDTMFInfo . . . 1335
8.1352.2.2 pBurstDTMFLengths . . . 1335
8.1353 voiceCallInfoReq Struct Reference . . . 1335
8.1353.1 Detailed Description . . . 1335
8.1353.2 Field Documentation . . . 1335
8.1353.2.1 callID . . . 1335
8.1354 voiceCallInfoResp Struct Reference . . . 1335

8.1354.1 Detailed Description . . . 1336
8.1354.2 Field Documentation . . . 1338
8.1354.2.1 pAlertingPattern . . . 1338
8.1354.2.2 pAlertType . . . 1338
8.1354.2.3 pAlphaIDInfo . . . 1338
8.1354.2.4 pCallInfo . . . 1338
8.1354.2.5 pConnectNumInfo . . . 1338
8.1354.2.6 pDiagInfo . . . 1338
8.1354.2.7 pOTASPStatus . . . 1338
8.1354.2.8 pRemotePartyName . . . 1338
8.1354.2.9 pRemotePartyNum . . . 1338
8.1354.2.10 pSrvOpt . . . 1338
8.1354.2.11 pUUSInfo . . . 1338
8.1354.2.12 pVoicePrivacy . . . 1338
8.1355 VoiceCallRequestParams Struct Reference . . . 1338
8.1355.1 Detailed Description . . . 1339
8.1355.2 Field Documentation . . . 1340
8.1355.2.1 callNumber . . . 1340
8.1355.2.2 pCallPartySubAdd . . . 1340
8.1355.2.3 pCallType . . . 1340
8.1355.2.4 pCLIRType . . . 1340
8.1355.2.5 pCUGInfo . . . 1340
8.1355.2.6 pEmergencyCategory . . . 1340
8.1355.2.7 pSvcType . . . 1340
8.1355.2.8 pUUSInFo . . . 1340
8.1356 VoiceCallResponseParams Struct Reference . . . 1340
8.1356.1 Detailed Description . . . 1340
8.1356.2 Field Documentation . . . 1341
8.1356.2.1 pAlphaIDInfo . . . 1341
8.1356.2.2 pCallID . . . 1341
8.1356.2.3 pCCResultType . . . 1341
8.1356.2.4 pCCSUPSType . . . 1341
8.1357 VoiceContDTMFinfo Struct Reference . . . 1341
8.1357.1 Detailed Description . . . 1341
8.1357.2 Field Documentation . . . 1341
8.1357.2.1 DTMFdigit . . . 1341
8.1357.2.2 pCallID . . . 1342
8.1358 VoiceDTMFEventInfo Struct Reference . . . 1342
8.1358.1 Detailed Description . . . 1342
8.1358.2 Field Documentation . . . 1342

8.1358.2.1 DTMFInformation . . . 1342
8.1358.2.2 pOffLength . . . 1342
8.1358.2.3 pOnLength . . . 1342
8.1359 VoiceFlashInfo Struct Reference . . . 1342
8.1359.1 Detailed Description . . . 1343
8.1359.2 Field Documentation . . . 1343
8.1359.2.1 pCallID . . . 1343
8.1359.2.2 pFlashPayLd . . . 1343
8.1359.2.3 pFlashType . . . 1343
8.1360 VoiceGetAllCallInfo Struct Reference . . . 1343
8.1360.1 Detailed Description . . . 1344
8.1360.2 Field Documentation . . . 1345
8.1360.2.1 pArrAlertingPattern . . . 1345
8.1360.2.2 pArrAlertingType . . . 1345
8.1360.2.3 pArrAlphaID . . . 1345
8.1360.2.4 pArrCalledPartyNum . . . 1345
8.1360.2.5 pArrCallEndReason . . . 1345
8.1360.2.6 pArrCallInfo . . . 1345
8.1360.2.7 pArrConnectPartyNum . . . 1345
8.1360.2.8 pArrDiagInfo . . . 1345
8.1360.2.9 pArrRedirPartyNum . . . 1345
8.1360.2.10 pArrRemotePartyName . . . 1345
8.1360.2.11 pArrRemotePartyNum . . . 1345
8.1360.2.12 pArrSvcOption . . . 1346
8.1360.2.13 pArrUUSInfo . . . 1346
8.1360.2.14 pOTASPStatus . . . 1346
8.1360.2.15 pVoicePrivacy . . . 1346
8.1361 VoiceGetCallBarringReq Struct Reference . . . 1346
8.1361.1 Detailed Description . . . 1346
8.1361.2 Field Documentation . . . 1346
8.1361.2.1 pSvcClass . . . 1346
8.1361.2.2 reason . . . 1346
8.1362 VoiceGetCallBarringResp Struct Reference . . . 1346
8.1362.1 Detailed Description . . . 1347
8.1362.2 Field Documentation . . . 1347
8.1362.2.1 pAlphaIDInfo . . . 1348
8.1362.2.2 pCallID . . . 1348
8.1362.2.3 pCCResType . . . 1348
8.1362.2.4 pCCSUPSType . . . 1348
8.1362.2.5 pFailCause . . . 1348

8.1362.2.6 pSvcClass . . . 1348
8.1363 voiceGetCallFWReq Struct Reference . . . 1348
8.1363.1 Detailed Description . . . 1348
8.1363.2 Field Documentation . . . 1348
8.1363.2.1 pSvcClass . . . 1348
8.1363.2.2 Reason . . . 1348
8.1364 voiceGetCallFWResp Struct Reference . . . 1348
8.1364.1 Detailed Description . . . 1349
8.1364.2 Field Documentation . . . 1350
8.1364.2.1 pAlphaIDInfo . . . 1350
8.1364.2.2 pCallID . . . 1350
8.1364.2.3 pCCResType . . . 1350
8.1364.2.4 pCCSUPSType . . . 1350
8.1364.2.5 pFailCause . . . 1350
8.1364.2.6 pGetCallFWExtInfo . . . 1350
8.1364.2.7 pGetCallFWInfo . . . 1350
8.1365 voiceGetCallWaitInfo Struct Reference . . . 1350
8.1365.1 Detailed Description . . . 1350
8.1365.2 Field Documentation . . . 1351
8.1365.2.1 pAlphaIDInfo . . . 1351
8.1365.2.2 pCallID . . . 1351
8.1365.2.3 pCCResType . . . 1351
8.1365.2.4 pCCSUPSType . . . 1351
8.1365.2.5 pFailCause . . . 1351
8.1365.2.6 pSvcClass . . . 1351
8.1366 voiceGetCLIPResp Struct Reference . . . 1351
8.1366.1 Detailed Description . . . 1351
8.1366.2 Field Documentation . . . 1352
8.1366.2.1 pAlphaIDInfo . . . 1352
8.1366.2.2 pCallID . . . 1352
8.1366.2.3 pCCResType . . . 1352
8.1366.2.4 pCCSUPSType . . . 1352
8.1366.2.5 pCLIPResp . . . 1352
8.1366.2.6 pFailCause . . . 1352
8.1367 voiceGetCLIRResp Struct Reference . . . 1352
8.1367.1 Detailed Description . . . 1353
8.1367.2 Field Documentation . . . 1353
8.1367.2.1 pAlphaIDInfo . . . 1353
8.1367.2.2 pCallID . . . 1354
8.1367.2.3 pCCResType . . . 1354

8.1367.2.4 pCCSUPSType . . . 1354
8.1367.2.5 pCLIRResp . . . 1354
8.1367.2.6 pFailCause . . . 1354
8.1368 VoiceGetCNAPResp Struct Reference . . . 1354
8.1368.1 Detailed Description . . . 1354
8.1368.2 Field Documentation . . . 1355
8.1368.2.1 pAlphaIDInfo . . . 1355
8.1368.2.2 pCallID . . . 1355
8.1368.2.3 pCCResType . . . 1355
8.1368.2.4 pCCSUPSType . . . 1355
8.1368.2.5 pCNAPResp . . . 1355
8.1368.2.6 pFailCause . . . 1355
8.1369 VoiceGetCOLPResp Struct Reference . . . 1355
8.1369.1 Detailed Description . . . 1355
8.1369.2 Field Documentation . . . 1356
8.1369.2.1 pAlphaIDInfo . . . 1356
8.1369.2.2 pCallID . . . 1356
8.1369.2.3 pCCResType . . . 1356
8.1369.2.4 pCCSUPSType . . . 1356
8.1369.2.5 pCOLPResp . . . 1356
8.1369.2.6 pFailCause . . . 1356
8.1370 VoiceGetCOLRResp Struct Reference . . . 1356
8.1370.1 Detailed Description . . . 1356
8.1370.2 Field Documentation . . . 1357
8.1370.2.1 pAlphaIDInfo . . . 1357
8.1370.2.2 pCallID . . . 1357
8.1370.2.3 pCCResType . . . 1357
8.1370.2.4 pCCSUPSType . . . 1357
8.1370.2.5 pCOLRResp . . . 1357
8.1370.2.6 pFailCause . . . 1357
8.1371 VoiceGetConfigReq Struct Reference . . . 1357
8.1371.1 Detailed Description . . . 1358
8.1371.2 Field Documentation . . . 1359
8.1371.2.1 pAirTimer . . . 1359
8.1371.2.2 pAMRStatus . . . 1359
8.1371.2.3 pAutoAnswer . . . 1359
8.1371.2.4 pNamID . . . 1359
8.1371.2.5 pPrefVoicePrivacy . . . 1359
8.1371.2.6 pPrefVoiceSO . . . 1359
8.1371.2.7 pRoamTimer . . . 1359

8.1371.2.8 pTTYMode . . . 1359
8.1371.2.9 pVoiceDomainPref . . . 1359
8.1372 voiceGetConfigResp Struct Reference . . . 1359
8.1372.1 Detailed Description . . . 1360
8.1372.2 Field Documentation . . . 1361
8.1372.2.1 pAirTimerCnt . . . 1361
8.1372.2.2 pAutoAnswerStat . . . 1361
8.1372.2.3 pCurAMRConfig . . . 1361
8.1372.2.4 pCurPrefVoiceSO . . . 1361
8.1372.2.5 pCurrTTYMode . . . 1361
8.1372.2.6 pCurVoiceDomainPref . . . 1361
8.1372.2.7 pCurVoicePrivacyPref . . . 1361
8.1372.2.8 pRoamTimerCnt . . . 1361
8.1373 voiceIndicationRegisterInfo Struct Reference . . . 1361
8.1373.1 Detailed Description . . . 1361
8.1373.2 Field Documentation . . . 1362
8.1373.2.1 pRegDTMFEvents . . . 1362
8.1373.2.2 pRegVoicePrivacyEvents . . . 1362
8.1373.2.3 pSuppsNotifEvents . . . 1362
8.1374 voiceInfoRec Struct Reference . . . 1362
8.1374.1 Detailed Description . . . 1362
8.1374.2 Field Documentation . . . 1364
8.1374.2.1 callID . . . 1364
8.1374.2.2 pCalledPartyInfo . . . 1364
8.1374.2.3 pCallerIDInfo . . . 1364
8.1374.2.4 pCallerNameInfo . . . 1364
8.1374.2.5 pCallingPartyInfo . . . 1364
8.1374.2.6 pCallWaitInd . . . 1364
8.1374.2.7 pCLIRCause . . . 1364
8.1374.2.8 pConnectNumInfo . . . 1364
8.1374.2.9 pDispInfo . . . 1364
8.1374.2.10 pExtDispInfo . . . 1364
8.1374.2.11 pExtDispRecInfo . . . 1364
8.1374.2.12 pLineCtrlInfo . . . 1364
8.1374.2.13 pNSSAudioCtrl . . . 1364
8.1374.2.14 pNSSRelease . . . 1364
8.1374.2.15 pRedirNumInfo . . . 1364
8.1374.2.16 pSignalInfo . . . 1364
8.1375 voiceManageCallsReq Struct Reference . . . 1364
8.1375.1 Detailed Description . . . 1364

8.1375.2 Field Documentation . . . 1365
8.1375.2.1 pCallID . . . 1365
8.1375.2.2 SUPSType . . . 1365
8.1376 VoiceManageCallsResp Struct Reference . . . 1365
8.1376.1 Detailed Description . . . 1365
8.1376.2 Field Documentation . . . 1366
8.1376.2.1 pFailCause . . . 1366
8.1377 VoiceOrigUSSDNoWaitInfo Struct Reference . . . 1366
8.1377.1 Detailed Description . . . 1366
8.1377.2 Field Documentation . . . 1366
8.1377.2.1 USSInformation . . . 1366
8.1378 VoiceOTASPStatusInfo Struct Reference . . . 1366
8.1378.1 Detailed Description . . . 1366
8.1378.2 Field Documentation . . . 1367
8.1378.2.1 callID . . . 1367
8.1378.2.2 OTASPStatus . . . 1367
8.1379 VoicePrivacyInfo Struct Reference . . . 1367
8.1379.1 Detailed Description . . . 1367
8.1379.2 Field Documentation . . . 1368
8.1379.2.1 callID . . . 1368
8.1379.2.2 voicePrivacy . . . 1368
8.1380 VoiceSetAllCallStatusCbkInfo Struct Reference . . . 1368
8.1380.1 Detailed Description . . . 1368
8.1380.2 Field Documentation . . . 1370
8.1380.2.1 arrCallInfomation . . . 1370
8.1380.2.2 pArrAlertingPattern . . . 1370
8.1380.2.3 pArrAlertingType . . . 1370
8.1380.2.4 pArrAlphaID . . . 1370
8.1380.2.5 pArrCalledPartyNum . . . 1370
8.1380.2.6 pArrCallEndReason . . . 1370
8.1380.2.7 pArrConnectPartyNum . . . 1370
8.1380.2.8 pArrDiagInfo . . . 1370
8.1380.2.9 pArrRedirPartyNum . . . 1370
8.1380.2.10 pArrRemotePartyName . . . 1370
8.1380.2.11 pArrRemotePartyNum . . . 1370
8.1380.2.12 pArrSvcOption . . . 1370
8.1381 VoiceSetCallBarringPwdInfo Struct Reference . . . 1370
8.1381.1 Detailed Description . . . 1370
8.1381.2 Field Documentation . . . 1371
8.1381.2.1 newPasswd . . . 1371

8.1381.2.2 newPasswdAgain . . . 1371
8.1381.2.3 oldPasswd . . . 1371
8.1381.2.4 Reason . . . 1371
8.1382 VoiceSetCallBarringPwdResp Struct Reference . . . 1371
8.1382.1 Detailed Description . . . 1372
8.1382.2 Field Documentation . . . 1372
8.1382.2.1 pAlphaIDInfo . . . 1372
8.1382.2.2 pCallID . . . 1372
8.1382.2.3 pCCResType . . . 1372
8.1382.2.4 pCCSUPSType . . . 1372
8.1382.2.5 pFailCause . . . 1372
8.1383 VoiceSetConfigReq Struct Reference . . . 1372
8.1383.1 Detailed Description . . . 1373
8.1383.2 Field Documentation . . . 1373
8.1383.2.1 pAirTimerConfig . . . 1374
8.1383.2.2 pAutoAnswer . . . 1374
8.1383.2.3 pPrefVoiceDomain . . . 1374
8.1383.2.4 pPrefVoiceSO . . . 1374
8.1383.2.5 pRoamTimerConfig . . . 1374
8.1383.2.6 pTTYMode . . . 1374
8.1384 VoiceSetConfigResp Struct Reference . . . 1374
8.1384.1 Detailed Description . . . 1374
8.1384.2 Field Documentation . . . 1375
8.1384.2.1 pAirTimerStatus . . . 1375
8.1384.2.2 pAutoAnsStatus . . . 1375
8.1384.2.3 pPrefVoiceSOStatus . . . 1375
8.1384.2.4 pRoamTimerStatus . . . 1375
8.1384.2.5 pTTYConfigStatus . . . 1375
8.1384.2.6 pVoiceDomainPrefStatus . . . 1375
8.1385 VoiceSetPrefPrivacy Struct Reference . . . 1375
8.1385.1 Detailed Description . . . 1375
8.1385.2 Field Documentation . . . 1376
8.1385.2.1 privacyPref . . . 1376
8.1386 VoiceSetSUPSServiceReq Struct Reference . . . 1376
8.1386.1 Detailed Description . . . 1376
8.1386.2 Field Documentation . . . 1378
8.1386.2.1 pCallBarringPasswd . . . 1378
8.1386.2.2 pCallForwardingNumber . . . 1378
8.1386.2.3 pCallFwdTypeAndPlan . . . 1378
8.1386.2.4 pServiceClass . . . 1378

8.1386.2.5 pTimerVal . . . 1378
8.1386.2.6 reason . . . 1378
8.1386.2.7 voiceSvc . . . 1378
8.1387 VoiceSetSUPSServiceResp Struct Reference . . . 1378
8.1387.1 Detailed Description . . . 1378
8.1387.2 Field Documentation . . . 1379
8.1387.2.1 pAlphaIDInfo . . . 1379
8.1387.2.2 pCallID . . . 1379
8.1387.2.3 pCCResultType . . . 1379
8.1387.2.4 pCCSUPSType . . . 1379
8.1387.2.5 pFailCause . . . 1379
8.1388 VoiceStopContDTMFinfo Struct Reference . . . 1379
8.1388.1 Detailed Description . . . 1379
8.1388.2 Field Documentation . . . 1380
8.1388.2.1 callID . . . 1380
8.1389 VoiceSUPSInfo Struct Reference . . . 1380
8.1389.1 Detailed Description . . . 1380
8.1389.2 Field Documentation . . . 1382
8.1389.2.1 pAlphaIDInfo . . . 1382
8.1389.2.2 pCallBarPasswd . . . 1382
8.1389.2.3 pCallFwdInfo . . . 1382
8.1389.2.4 pCallFWNum . . . 1382
8.1389.2.5 pCallFWTimerVal . . . 1382
8.1389.2.6 pCallID . . . 1382
8.1389.2.7 pCLIPstatus . . . 1382
8.1389.2.8 pCLIRstatus . . . 1382
8.1389.2.9 pCNAPstatus . . . 1382
8.1389.2.10 pCOLPstatus . . . 1382
8.1389.2.11 pCOLRstatus . . . 1382
8.1389.2.12 pDataSrc . . . 1382
8.1389.2.13 pFailCause . . . 1382
8.1389.2.14 pNewPwdData . . . 1382
8.1389.2.15 pReason . . . 1382
8.1389.2.16 pSvcClass . . . 1382
8.1389.2.17 pUSSInfo . . . 1382
8.1389.2.18 pUPSInformation . . . 1382
8.1390 VoiceSUPSNotification Struct Reference . . . 1382
8.1390.1 Detailed Description . . . 1383
8.1390.2 Field Documentation . . . 1384
8.1390.2.1 callID . . . 1384

8.1390.2.2 notifType . . . 1384
8.1390.2.3 pCUGIndex . . . 1384
8.1390.2.4 pECTNum . . . 1384
8.1391 wcdmaCellInfo Struct Reference . . . 1384
8.1391.1 Detailed Description . . . 1384
8.1391.2 Field Documentation . . . 1384
8.1391.2.1 cpich_ecno . . . 1384
8.1391.2.2 cpich_rscp . . . 1384
8.1391.2.3 psc . . . 1385
8.1391.2.4 srxlev . . . 1385
8.1392 WCDMAECIOThresh Struct Reference . . . 1385
8.1392.1 Detailed Description . . . 1385
8.1392.2 Field Documentation . . . 1385
8.1392.2.1 pWCDMAECIOThreshList . . . 1385
8.1392.2.2 WCDMAECIOThreshListLen . . . 1385
8.1393 WCDMAInfoLTENeighborCell Struct Reference . . . 1385
8.1393.1 Detailed Description . . . 1385
8.1393.2 Field Documentation . . . 1386
8.1393.2.1 UMTSLTENbrCell . . . 1386
8.1393.2.2 umtsLTENbrCellLen . . . 1386
8.1393.2.3 wcdmaRRCState . . . 1386
8.1394 wcdmaLongMsgDecodingParams Struct Reference . . . 1386
8.1394.1 Detailed Description . . . 1386
8.1394.2 Field Documentation . . . 1387
8.1394.2.1 Date . . . 1387
8.1394.2.2 pIsUDHPresent . . . 1387
8.1394.2.3 pMessage . . . 1388
8.1394.2.4 pPartNum . . . 1388
8.1394.2.5 pReferenceNum . . . 1388
8.1394.2.6 pScAddr . . . 1388
8.1394.2.7 pScAddrLength . . . 1388
8.1394.2.8 pSenderAddr . . . 1388
8.1394.2.9 pSenderAddrLength . . . 1388
8.1394.2.10 pTextMsg . . . 1388
8.1394.2.11 pTextMsgLength . . . 1388
8.1394.2.12 pTotalNum . . . 1388
8.1394.2.13 Time . . . 1388
8.1395 wcdmaMsgDecodingParams Struct Reference . . . 1388
8.1395.1 Detailed Description . . . 1388
8.1395.2 Field Documentation . . . 1389

8.1395.2.1 Date . . . 1389
8.1395.2.2 pMessage . . . 1389
8.1395.2.3 pScAddr . . . 1389
8.1395.2.4 pScAddrLength . . . 1389
8.1395.2.5 pSenderAddr . . . 1389
8.1395.2.6 pSenderAddrLength . . . 1389
8.1395.2.7 pTextMsg . . . 1389
8.1395.2.8 pTextMsgLength . . . 1389
8.1395.2.9 Time . . . 1389
8.1396 wcdmaMsgEncodingParams Struct Reference . . . 1389
8.1396.1 Detailed Description . . . 1390
8.1396.2 Field Documentation . . . 1390
8.1396.2.1 alphabet . . . 1390
8.1396.2.2 messageSize . . . 1390
8.1396.2.3 pDestAddr . . . 1390
8.1396.2.4 pPDUMessage . . . 1390
8.1396.2.5 pTextMsg . . . 1390
8.1397 WCDMARSSIThresh Struct Reference . . . 1390
8.1397.1 Detailed Description . . . 1390
8.1397.2 Field Documentation . . . 1391
8.1397.2.1 pWCDMARSSIThreshList . . . 1391
8.1397.2.2 WCDMARSSIThreshListLen . . . 1391
8.1398 WCDMASysInfo Struct Reference . . . 1391
8.1398.1 Detailed Description . . . 1391
8.1398.2 Field Documentation . . . 1393
8.1398.2.1 cellId . . . 1394
8.1398.2.2 cellIdValid . . . 1394
8.1398.2.3 hsCallStatus . . . 1394
8.1398.2.4 hsCallStatusValid . . . 1394
8.1398.2.5 hsInd . . . 1394
8.1398.2.6 hsIndValid . . . 1394
8.1398.2.7 lac . . . 1394
8.1398.2.8 lacValid . . . 1394
8.1398.2.9 MCC . . . 1394
8.1398.2.10 MNC . . . 1394
8.1398.2.11 networkIdValid . . . 1394
8.1398.2.12 psc . . . 1394
8.1398.2.13 pscValid . . . 1394
8.1398.2.14 regRejectInfoValid . . . 1394
8.1398.2.15 rejCause . . . 1394

8.1398.2.16 rejectSrvDomain . . . 1394
8.1398.2.17 sysInfoWCDMA . . . 1394
8.1399 wcdmaUARFCN Struct Reference . . . 1394
8.1399.1 Detailed Description . . . 1394
8.1399.2 Field Documentation . . . 1395
8.1399.2.1 status . . . 1395
8.1399.2.2 uarfcn . . . 1395
8.1400 wds_currNetworkInfo Struct Reference . . . 1395
8.1400.1 Detailed Description . . . 1395
8.1400.2 Field Documentation . . . 1396
8.1400.2.1 NetworkType . . . 1396
8.1400.2.2 RATMask . . . 1396
8.1400.2.3 SOMask . . . 1396
8.1401 wds_DataULongLongTlv Struct Reference . . . 1396
8.1401.1 Detailed Description . . . 1396
8.1401.2 Field Documentation . . . 1396
8.1401.2.1 TlvPresent . . . 1397
8.1401.2.2 ullData . . . 1397
8.1402 wds_DataULongTlv Struct Reference . . . 1397
8.1402.1 Detailed Description . . . 1397
8.1402.2 Field Documentation . . . 1397
8.1402.2.1 TlvPresent . . . 1397
8.1402.2.2 ulData . . . 1397
8.1403 wds_DHCPLeaseOptTlv Struct Reference . . . 1397
8.1403.1 Detailed Description . . . 1397
8.1403.2 Field Documentation . . . 1397
8.1403.2.1 numOpt . . . 1397
8.1403.2.2 optList . . . 1397
8.1403.2.3 optListData . . . 1398
8.1403.2.4 TlvPresent . . . 1398
8.1404 wds_DHCPLeaseStateTlv Struct Reference . . . 1398
8.1404.1 Detailed Description . . . 1398
8.1404.2 Field Documentation . . . 1398
8.1404.2.1 leaseState . . . 1398
8.1404.2.2 TlvPresent . . . 1398
8.1405 wds_DHCPOpt Struct Reference . . . 1398
8.1405.1 Detailed Description . . . 1398
8.1405.2 Field Documentation . . . 1399
8.1405.2.1 optCode . . . 1399
8.1405.2.2 optValLen . . . 1399

8.1405.2.3 pOptVal . . . 1399
8.1406 wds_DHCPProfileIdTlv Struct Reference . . . 1399
8.1406.1 Detailed Description . . . 1399
8.1406.2 Field Documentation . . . 1399
8.1406.2.1 profileId . . . 1399
8.1406.2.2 profileType . . . 1399
8.1406.2.3 TlvPresent . . . 1399
8.1407 wds_DHCPv4HWConfig Struct Reference . . . 1399
8.1407.1 Detailed Description . . . 1400
8.1407.2 Field Documentation . . . 1400
8.1407.2.1 chaddr . . . 1400
8.1407.2.2 chaddrLen . . . 1400
8.1407.2.3 hwType . . . 1400
8.1408 wds_DHCPv4Option Struct Reference . . . 1400
8.1408.1 Detailed Description . . . 1400
8.1408.2 Field Documentation . . . 1400
8.1408.2.1 optCode . . . 1400
8.1408.2.2 optVal . . . 1401
8.1408.2.3 optValLen . . . 1401
8.1409 wds_DHCPv4OptionList Struct Reference . . . 1401
8.1409.1 Detailed Description . . . 1401
8.1409.2 Field Documentation . . . 1401
8.1409.2.1 numOpt . . . 1401
8.1409.2.2 pOptList . . . 1401
8.1410 wds_DHCPv4ProfileId Struct Reference . . . 1401
8.1410.1 Detailed Description . . . 1401
8.1410.2 Field Documentation . . . 1401
8.1410.2.1 profileId . . . 1402
8.1410.2.2 profileType . . . 1402
8.1411 wds_Domain Struct Reference . . . 1402
8.1411.1 Detailed Description . . . 1402
8.1411.2 Field Documentation . . . 1402
8.1411.2.1 domainLen . . . 1402
8.1411.2.2 domainName . . . 1402
8.1412 wds_DomainNameList Struct Reference . . . 1402
8.1412.1 Detailed Description . . . 1402
8.1412.2 Field Documentation . . . 1403
8.1412.2.1 domain . . . 1403
8.1412.2.2 numInstances . . . 1403
8.1413 wds_GPRSQoS Struct Reference . . . 1403

8.1413.1 Detailed Description . . . 1403
8.1413.2 Field Documentation . . . 1403
8.1413.2.1 delayClass . . . 1403
8.1413.2.2 meanThroughputClass . . . 1403
8.1413.2.3 peakThroughputClass . . . 1403
8.1413.2.4 precedenceClass . . . 1403
8.1413.2.5 reliabilityClass . . . 1403
8.1414 wds_IPv4AdTlv Struct Reference . . . 1404
8.1414.1 Detailed Description . . . 1404
8.1414.2 Field Documentation . . . 1404
8.1414.2.1 IPv4Addr . . . 1404
8.1414.2.2 TlvPresent . . . 1404
8.1415 wds_IPV6AddressInfo Struct Reference . . . 1404
8.1415.1 Detailed Description . . . 1404
8.1415.2 Field Documentation . . . 1404
8.1415.2.1 IPAddressV6 . . . 1404
8.1415.2.2 IPV6PrefixLen . . . 1404
8.1416 wds_IPV6GWAddressInfo Struct Reference . . . 1404
8.1416.1 Detailed Description . . . 1405
8.1416.2 Field Documentation . . . 1405
8.1416.2.1 gwAddressV6 . . . 1405
8.1416.2.2 gwV6PrefixLen . . . 1405
8.1417 wds_PCSCFFQDNAddress Struct Reference . . . 1405
8.1417.1 Detailed Description . . . 1405
8.1417.2 Field Documentation . . . 1405
8.1417.2.1 fqdnAddr . . . 1405
8.1417.2.2 fqdnLen . . . 1406
8.1418 wds_PCSCFFQDNAddressList Struct Reference . . . 1406
8.1418.1 Detailed Description . . . 1406
8.1418.2 Field Documentation . . . 1406
8.1418.2.1 numInstances . . . 1406
8.1418.2.2 pcsfFQDNAddress . . . 1406
8.1419 wds_PCSCFIPv4ServerAddressList Struct Reference . . . 1406
8.1419.1 Detailed Description . . . 1406
8.1419.2 Field Documentation . . . 1407
8.1419.2.1 numInstances . . . 1407
8.1419.2.2 pscsfIPv4Addr . . . 1407
8.1420 wds_ProfileIdentifier Struct Reference . . . 1407
8.1420.1 Detailed Description . . . 1407
8.1420.2 Field Documentation . . . 1407

8.1420.2.1 profileIndex . . . 1407
8.1420.2.2 profileType . . . 1407
8.1421 wds_profileInfo Union Reference . . . 1407
8.1421.1 Detailed Description . . . 1407
8.1421.2 Field Documentation . . . 1408
8.1421.2.1 SlqsProfile3GPP . . . 1408
8.1421.2.2 SlqsProfile3GPP2 . . . 1408
8.1422 wds_transferStatInd Struct Reference . . . 1408
8.1422.1 Detailed Description . . . 1408
8.1422.2 Field Documentation . . . 1408
8.1422.2.1 StatsMask . . . 1408
8.1422.2.2 StatsPeriod . . . 1408
8.1423 wds_TrStatInd Struct Reference . . . 1408
8.1423.1 Detailed Description . . . 1408
8.1423.2 Field Documentation . . . 1409
8.1423.2.1 statsMask . . . 1409
8.1423.2.2 statsPeriod . . . 1409
8.1424 wds_UMTSMinQoS Struct Reference . . . 1409
8.1424.1 Detailed Description . . . 1409
8.1424.2 Field Documentation . . . 1411
8.1424.2.1 deliveryErrSDU . . . 1411
8.1424.2.2 grntDownlinkBitrate . . . 1411
8.1424.2.3 grntUplinkBitrate . . . 1411
8.1424.2.4 maxDownlinkBitrate . . . 1411
8.1424.2.5 maxSDUSize . . . 1411
8.1424.2.6 maxUplinkBitrate . . . 1411
8.1424.2.7 qosDeliveryOrder . . . 1411
8.1424.2.8 resBerRatio . . . 1411
8.1424.2.9 sduErrorRatio . . . 1411
8.1424.2.10 trafficClass . . . 1411
8.1424.2.11 trafficPriority . . . 1411
8.1424.2.12 transferDelay . . . 1411
8.1425 WdsByteTotals Struct Reference . . . 1411
8.1425.1 Detailed Description . . . 1412
8.1425.2 Field Documentation . . . 1412
8.1425.2.1 ByteTotalsElmntsV4 . . . 1412
8.1425.2.2 ByteTotalsElmntsV6 . . . 1412
8.1425.2.3 pV4sessionId . . . 1412
8.1425.2.4 pV6sessionId . . . 1412
8.1426 WdsByteTotalsElmnts Struct Reference . . . 1412

8.1426.1 Detailed Description . . . 1412
8.1426.2 Field Documentation . . . 1413
8.1426.2.1 pRXTotalBytes . . . 1413
8.1426.2.2 pTXTotalBytes . . . 1413
8.1427 WdsClientLeaseChange Struct Reference . . . 1413
8.1427.1 Detailed Description . . . 1413
8.1427.2 Field Documentation . . . 1413
8.1427.2.1 pEnableNotification . . . 1413
8.1428 WdsConnectionRate Struct Reference . . . 1413
8.1428.1 Detailed Description . . . 1413
8.1428.2 Field Documentation . . . 1414
8.1428.2.1 ConnRateElmntsV4 . . . 1414
8.1428.2.2 ConnRateElmntsV6 . . . 1414
8.1428.2.3 pV4sessionId . . . 1414
8.1428.2.4 pV6sessionId . . . 1414
8.1429 WdsConnectionRateElmnts Struct Reference . . . 1414
8.1429.1 Detailed Description . . . 1414
8.1429.2 Field Documentation . . . 1415
8.1429.2.1 pCurrentChannelRXRate . . . 1415
8.1429.2.2 pCurrentChannelTXRate . . . 1415
8.1429.2.3 pMaxChannelRXRate . . . 1415
8.1429.2.4 pMaxChannelTXRate . . . 1415
8.1430 WdsDHCPv4ClientLeaseInd Struct Reference . . . 1415
8.1430.1 Detailed Description . . . 1415
8.1430.2 Field Documentation . . . 1415
8.1430.2.1 pIPv4Addr . . . 1415
8.1430.2.2 pLeaseState . . . 1415
8.1430.2.3 pOptList . . . 1416
8.1430.2.4 pProfileId . . . 1416
8.1431 WdsDHCPv4Config Struct Reference . . . 1416
8.1431.1 Detailed Description . . . 1416
8.1431.2 Field Documentation . . . 1416
8.1431.2.1 pHwConfig . . . 1416
8.1431.2.2 pProfileId . . . 1416
8.1431.2.3 pRequestOptionList . . . 1416
8.1432 WdsDHCPv4HWConfig Struct Reference . . . 1416
8.1432.1 Detailed Description . . . 1416
8.1432.2 Field Documentation . . . 1417
8.1432.2.1 chaddr . . . 1417
8.1432.2.2 chaddrLen . . . 1417

8.1432.2.3 hwType . . . 1417
8.1433 WdsDhcpv4HwConfig Struct Reference . . . 1417
8.1433.1 Detailed Description . . . 1417
8.1433.2 Field Documentation . . . 1417
8.1433.2.1 chaddr . . . 1417
8.1433.2.2 chaddrLen . . . 1417
8.1433.2.3 hwType . . . 1417
8.1434 WdsDhcpv4Option Struct Reference . . . 1417
8.1434.1 Detailed Description . . . 1418
8.1434.2 Field Documentation . . . 1418
8.1434.2.1 optCode . . . 1418
8.1434.2.2 optVal . . . 1418
8.1434.2.3 optValLen . . . 1418
8.1435 WdsDHCPv4Option Struct Reference . . . 1418
8.1435.1 Detailed Description . . . 1418
8.1435.2 Field Documentation . . . 1418
8.1435.2.1 optCode . . . 1419
8.1435.2.2 optVal . . . 1419
8.1435.2.3 optValLen . . . 1419
8.1436 WdsDhcpv4OptionList Struct Reference . . . 1419
8.1436.1 Detailed Description . . . 1419
8.1436.2 Field Documentation . . . 1419
8.1436.2.1 numOpt . . . 1419
8.1436.2.2 pOptList . . . 1419
8.1437 WdsDHCPv4OptionList Struct Reference . . . 1419
8.1437.1 Detailed Description . . . 1419
8.1437.2 Field Documentation . . . 1420
8.1437.2.1 numOpt . . . 1420
8.1437.2.2 pOptList . . . 1420
8.1438 WdsDHCPv4ProfileId Struct Reference . . . 1420
8.1438.1 Detailed Description . . . 1420
8.1438.2 Field Documentation . . . 1420
8.1438.2.1 profileId . . . 1420
8.1438.2.2 profileType . . . 1420
8.1439 WdsDhcpv4ProfileId Struct Reference . . . 1420
8.1439.1 Detailed Description . . . 1420
8.1439.2 Field Documentation . . . 1421
8.1439.2.1 profileId . . . 1421
8.1439.2.2 profileType . . . 1421
8.1440 WDSGetLoopbackData Struct Reference . . . 1421

8.1440.1 Detailed Description . . . 1421
8.1440.2 Field Documentation . . . 1421
8.1440.2.1 ByteLoopbackMode . . . 1421
8.1440.2.2 ByteLoopbackMultiplier . . . 1421
8.1441 WdsIpAddressInfoReq Struct Reference . . . 1421
8.1441.1 Detailed Description . . . 1422
8.1441.2 Field Documentation . . . 1422
8.1441.2.1 ip . . . 1422
8.1441.2.2 pv4sessionId . . . 1422
8.1441.2.3 pv6sessionId . . . 1422
8.1442 WdsPktStatisticsElmnts Struct Reference . . . 1422
8.1442.1 Detailed Description . . . 1422
8.1442.2 Field Documentation . . . 1423
8.1442.2.1 pRXDroppedCount . . . 1423
8.1442.2.2 pRXOkBytesCount . . . 1423
8.1442.2.3 pRXOKBytesLastCall . . . 1423
8.1442.2.4 pRXPacketErrors . . . 1423
8.1442.2.5 pRXPacketOverflows . . . 1423
8.1442.2.6 pRXPacketSuccesses . . . 1424
8.1442.2.7 pTXDroppedCount . . . 1424
8.1442.2.8 pTXOkBytesCount . . . 1424
8.1442.2.9 pTXOKBytesLastCall . . . 1424
8.1442.2.10 pTXPacketErrors . . . 1424
8.1442.2.11 pTXPacketOverflows . . . 1424
8.1442.2.12 pTXPacketSuccesses . . . 1424
8.1443 WdsPktStatisticsReq Struct Reference . . . 1424
8.1443.1 Detailed Description . . . 1424
8.1443.2 Field Documentation . . . 1424
8.1443.2.1 pStatMask . . . 1424
8.1444 WdsPktStatisticsResp Struct Reference . . . 1424
8.1444.1 Detailed Description . . . 1424
8.1444.2 Field Documentation . . . 1425
8.1444.2.1 PktStatElmntsV4 . . . 1425
8.1444.2.2 PktStatElmntsV6 . . . 1425
8.1444.2.3 pV4sessionId . . . 1425
8.1444.2.4 pV6sessionId . . . 1425
8.1445 WdsProfileParam Union Reference . . . 1425
8.1445.1 Detailed Description . . . 1425
8.1445.2 Field Documentation . . . 1425
8.1445.2.1 SlqsProfile3GPP . . . 1426

8.1445.2.2 SlqsProfile3GPP2 . . . 1426
8.1446 WdsRunTimeSettings Struct Reference . . . 1426
8.1446.1 Detailed Description . . . 1426
8.1446.2 Field Documentation . . . 1426
8.1446.2.1 rts . . . 1426
8.1446.2.2 v4sessionId . . . 1426
8.1446.2.3 v6sessionId . . . 1426
8.1447 WdsSetEventReportReq Struct Reference . . . 1426
8.1447.1 Detailed Description . . . 1427
8.1447.2 Field Documentation . . . 1428
8.1447.2.1 pCurrChannelRateInd . . . 1428
8.1447.2.2 pCurrDataBearerTechInd . . . 1428
8.1447.2.3 pCurrPrefDataSysInd . . . 1428
8.1447.2.4 pDataBearerTechInd . . . 1428
8.1447.2.5 pDataCallStatusChangeInd . . . 1428
8.1447.2.6 pDataSystemStatusChangeInd . . . 1428
8.1447.2.7 pDormancyStatusInd . . . 1428
8.1447.2.8 pEVDOPageMonPerChangeInd . . . 1428
8.1447.2.9 pMIPStatusInd . . . 1428
8.1447.2.10 pTransferStatInd . . . 1428
8.1448 WDSSetLoopbackData Struct Reference . . . 1428
8.1448.1 Detailed Description . . . 1428
8.1448.2 Field Documentation . . . 1429
8.1448.2.1 pLoopbackMode . . . 1429
8.1448.2.2 pLoopbackMultiplier . . . 1429
8.1449 WDSSWICurrentChannelRates Struct Reference . . . 1429
8.1449.1 Detailed Description . . . 1429
8.1449.2 Field Documentation . . . 1429
8.1449.2.1 current_channel_rx_rate . . . 1429
8.1449.2.2 current_channel_tx_rate . . . 1429
8.1449.2.3 max_channel_rx_rate . . . 1429
8.1449.2.4 max_channel_tx_rate . . . 1429

9 File Documentation **1431**

9.1 apdoxypages.c File Reference . . . 1431
9.1.1 Detailed Description . . . 1431
9.2 common.h File Reference . . . 1431
9.2.1 Macro Definition Documentation . . . 1433
9.2.1.1 DEAULT_LOC_TIMEOUT_IN_SEC . . . 1433
9.2.1.2 MINREQBKLEN . . . 1433

9.2.1.3 MSGID_AND_LEN . . . 1433
9.2.1.4 MSGID_DONT_CARE . . . 1433
9.2.1.5 SDK_VALIDATE_INPUT_PACK_PARAM . . . 1433
9.2.1.6 SDK_VALIDATE_INPUT_PACK_PARAM_AND_FILL_XID . . . 1433
9.2.1.7 SDU_HDR_LEN . . . 1433
9.2.1.8 UNUSEDPARAM . . . 1433
9.2.2 Typedef Documentation . . . 1433
9.2.2.1 logger . . . 1433
9.2.3 Enumeration Type Documentation . . . 1433
9.2.3.1 eLOG_LEVEL . . . 1433
9.2.3.2 eQMI_SVC . . . 1434
9.2.3.3 eTimeout . . . 1434
9.2.3.4 msgtype . . . 1435
9.2.4 Function Documentation . . . 1435
9.2.4.1 fill_pack_ctx . . . 1435
9.2.4.2 fill_sdu_hdr . . . 1435
9.2.4.3 get_version . . . 1435
9.2.4.4 helper_get_resp_ctx . . . 1435
9.2.4.5 helper_get_xid . . . 1435
9.2.4.6 helper_set_log_func . . . 1435
9.2.4.7 helper_set_log_lvl . . . 1435
9.2.4.8 liteqmi_GetVersion . . . 1435
9.2.4.9 liteqmi_helper_decode7bitAsciiEncString . . . 1436
9.2.4.10 liteqmi_log . . . 1436
9.2.4.11 unpack_result_code_only . . . 1436
9.2.5 Variable Documentation . . . 1436
9.2.5.1 glog . . . 1436
9.2.5.2 gloglvl . . . 1436
9.3 dms.h File Reference . . . 1436
9.3.1 Macro Definition Documentation . . . 1442
9.3.1.1 ACT_CODE_MAX_SIZE . . . 1442
9.3.1.2 CK_MAX_SIZE . . . 1442
9.3.1.3 DMS_IMGDETAILS_LEN . . . 1442
9.3.1.4 DMS_MAX_CUST_ID_LEN . . . 1442
9.3.1.5 DMS_MAX_CUST_VALUE_LEN . . . 1442
9.3.1.6 DMS_MAX_FWUPDATE_LOG_STR_SZ . . . 1442
9.3.1.7 DMS_MAX_FWUPDATE_REF_STR_SZ . . . 1442
9.3.1.8 DMS_PM_FACTORY . . . 1443
9.3.1.9 DMS_PM_LOW . . . 1443
9.3.1.10 DMS_PM_OFFLINE . . . 1443

9.3.1.11 DMS_PM_ONLINE . . . 1443
9.3.1.12 DMS_PM_PERSISTENT_LOW . . . 1443
9.3.1.13 DMS_PM_RESET . . . 1443
9.3.1.14 DMS_PM_SHUT_DOWN . . . 1443
9.3.1.15 DMS_SET_REPORT_DISABLE . . . 1443
9.3.1.16 DMS_SET_REPORT_ENABLE . . . 1443
9.3.1.17 DMS_SLQSFWINFO_APPVERSION_SZ . . . 1443
9.3.1.18 DMS_SLQSFWINFO_BOOTVERSION_SZ . . . 1443
9.3.1.19 DMS_SLQSFWINFO_CARRIER_SZ . . . 1443
9.3.1.20 DMS_SLQSFWINFO_CUR_CARR_NAME . . . 1443
9.3.1.21 DMS_SLQSFWINFO_CUR_CARR_REV . . . 1443
9.3.1.22 DMS_SLQSFWINFO_MODELID_SZ . . . 1443
9.3.1.23 DMS_SLQSFWINFO_PACKAGEID_SZ . . . 1443
9.3.1.24 DMS_SLQSFWINFO_PRIVERSION_SZ . . . 1443
9.3.1.25 DMS_SLQSFWINFO_SKU_SZ . . . 1443
9.3.1.26 DMS_SWI_SET_IND_DISABLE . . . 1443
9.3.1.27 DMS_SWI_SET_IND_ENABLE . . . 1443
9.3.1.28 DMS_UINT8_MAX_STRING_SZ . . . 1443
9.3.1.29 ERI_DATA_MAX_SIZE . . . 1443
9.3.1.30 MAX_BUILD_ID_LEN . . . 1443
9.3.1.31 MEID_MAX_SIZE . . . 1443
9.3.1.32 SLQS_MAX_DYING_GASP_CFG_SMS_CONTENT_LENGTH . . . 1443
9.3.1.33 SLQS_MAX_DYING_GASP_CFG_SMS_NUMBER_LENGTH . . . 1443
9.3.1.34 SPC_SIZE . . . 1443
9.3.1.35 UNIQUE_ID_LEN . . . 1443
9.3.2 Function Documentation . . . 1444
9.3.2.1 pack_dms_ActivateAutomatic . . . 1444
9.3.2.2 pack_dms_GetActivationState . . . 1444
9.3.2.3 pack_dms_GetBandCapability . . . 1444
9.3.2.4 pack_dms_GetCrashAction . . . 1445
9.3.2.5 pack_dms_GetCustFeature . . . 1445
9.3.2.6 pack_dms_GetCustFeaturesV2 . . . 1445
9.3.2.7 pack_dms_GetDeviceCap . . . 1446
9.3.2.8 pack_dms_GetDeviceCapabilities . . . 1446
9.3.2.9 pack_dms_GetDeviceHardwareRev . . . 1447
9.3.2.10 pack_dms_GetDeviceMfr . . . 1447
9.3.2.11 pack_dms_GetDeviceSerialNumbers . . . 1447
9.3.2.12 pack_dms_GetFirmwareInfo . . . 1448
9.3.2.13 pack_dms_GetFirmwareRevision . . . 1448
9.3.2.14 pack_dms_GetFirmwareRevisions . . . 1448

9.3.2.15 pack_dms_GetFSN . . . 1449
9.3.2.16 pack_dms_GetHardwareRevision . . . 1449
9.3.2.17 pack_dms_GetIMSI . . . 1449
9.3.2.18 pack_dms_GetManufacturer . . . 1450
9.3.2.19 pack_dms_GetModelID . . . 1450
9.3.2.20 pack_dms_GetNetworkTime . . . 1450
9.3.2.21 pack_dms_GetOfflineReason . . . 1451
9.3.2.22 pack_dms_GetPower . . . 1451
9.3.2.23 pack_dms_GetPRLVersion . . . 1451
9.3.2.24 pack_dms_GetSerialNumbers . . . 1452
9.3.2.25 pack_dms_GetUSBComp . . . 1452
9.3.2.26 pack_dms_GetVoiceNumber . . . 1453
9.3.2.27 pack_dms_ResetToFactoryDefaults . . . 1453
9.3.2.28 pack_dms_SetActivationStatusCallback . . . 1453
9.3.2.29 pack_dms_SetCrashAction . . . 1454
9.3.2.30 pack_dms_SetCustFeature . . . 1454
9.3.2.31 pack_dms_SetCustFeaturesV2 . . . 1454
9.3.2.32 pack_dms_SetEventReport . . . 1455
9.3.2.33 pack_dms_SetFirmwarePreference . . . 1455
9.3.2.34 pack_dms_SetPower . . . 1455
9.3.2.35 pack_dms_SetUSBComp . . . 1456
9.3.2.36 pack_dms_SLQSDmsSwiGetResetInfo . . . 1456
9.3.2.37 pack_dms_SLQSDmsSwiIndicationRegister . . . 1457
9.3.2.38 pack_dms_SLQSGetBandCapability . . . 1457
9.3.2.39 pack_dms_SLQSGetERIFile . . . 1457
9.3.2.40 pack_dms_SLQSSwiClearDyingGaspStatistics . . . 1458
9.3.2.41 pack_dms_SLQSSwiGetCrashInfo . . . 1458
9.3.2.42 pack_dms_SLQSSwiGetDyingGaspCfg . . . 1458
9.3.2.43 pack_dms_SLQSSwiGetDyingGaspStatistics . . . 1459
9.3.2.44 pack_dms_SLQSSwiGetFirmwareCurr . . . 1459
9.3.2.45 pack_dms_SLQSSwiGetFwUpdateStatus . . . 1459
9.3.2.46 pack_dms_SLQSSwiGetHostDevInfo . . . 1460
9.3.2.47 pack_dms_SLQSSwiGetOSInfo . . . 1460
9.3.2.48 pack_dms_SLQSSwiGetSerialNoExt . . . 1460
9.3.2.49 pack_dms_SLQSSwiSetDyingGaspCfg . . . 1461
9.3.2.50 pack_dms_SLQSSwiSetHostDevInfo . . . 1461
9.3.2.51 pack_dms_SLQSSwiSetOSInfo . . . 1461
9.3.2.52 pack_dms_SLQSUIMGetState . . . 1462
9.3.2.53 pack_dms_UIMChangePIN . . . 1462
9.3.2.54 pack_dms_UIMGetControlKeyStatus . . . 1463

9.3.2.55 pack_dms_UIMGetICCID . . . 1463
9.3.2.56 pack_dms_UIMGetPINStatus . . . 1463
9.3.2.57 pack_dms_UIMSetControlKeyProtection . . . 1464
9.3.2.58 pack_dms_UIMSetPINProtection . . . 1464
9.3.2.59 pack_dms_UIMUnblockControlKey . . . 1464
9.3.2.60 pack_dms_UIMUnblockPIN . . . 1465
9.3.2.61 pack_dms_UIMVerifyPIN . . . 1465
9.3.2.62 pack_dms_ValidateSPC . . . 1466
9.3.2.63 unpack_dms_ActivateAutomatic . . . 1466
9.3.2.64 unpack_dms_GetActivationState . . . 1466
9.3.2.65 unpack_dms_GetBandCapability . . . 1467
9.3.2.66 unpack_dms_GetCrashAction . . . 1467
9.3.2.67 unpack_dms_GetCustFeature . . . 1467
9.3.2.68 unpack_dms_GetCustFeaturesV2 . . . 1468
9.3.2.69 unpack_dms_GetDeviceCap . . . 1468
9.3.2.70 unpack_dms_GetDeviceCapabilities . . . 1468
9.3.2.71 unpack_dms_GetDeviceHardwareRev . . . 1469
9.3.2.72 unpack_dms_GetDeviceMfr . . . 1469
9.3.2.73 unpack_dms_GetDeviceSerialNumbers . . . 1469
9.3.2.74 unpack_dms_GetFirmwareInfo . . . 1470
9.3.2.75 unpack_dms_GetFirmwareRevision . . . 1470
9.3.2.76 unpack_dms_GetFirmwareRevisions . . . 1470
9.3.2.77 unpack_dms_GetFSN . . . 1471
9.3.2.78 unpack_dms_GetHardwareRevision . . . 1471
9.3.2.79 unpack_dms_GetIMSI . . . 1471
9.3.2.80 unpack_dms_GetManufacturer . . . 1472
9.3.2.81 unpack_dms_GetModelID . . . 1472
9.3.2.82 unpack_dms_GetNetworkTime . . . 1472
9.3.2.83 unpack_dms_GetOfflineReason . . . 1473
9.3.2.84 unpack_dms_GetPower . . . 1473
9.3.2.85 unpack_dms_GetPRLVersion . . . 1473
9.3.2.86 unpack_dms_GetSerialNumbers . . . 1474
9.3.2.87 unpack_dms_GetUSBComp . . . 1474
9.3.2.88 unpack_dms_GetVoiceNumber . . . 1474
9.3.2.89 unpack_dms_ResetToFactoryDefaults . . . 1475
9.3.2.90 unpack_dms_SetActivationStatusCallback . . . 1475
9.3.2.91 unpack_dms_SetCrashAction . . . 1475
9.3.2.92 unpack_dms_SetCustFeature . . . 1476
9.3.2.93 unpack_dms_SetCustFeaturesV2 . . . 1476
9.3.2.94 unpack_dms_SetEventReport . . . 1476

9.3.2.95 unpack_dms_SetEventReport_ind . . . 1477
9.3.2.96 unpack_dms_SetFirmwarePreference . . . 1477
9.3.2.97 unpack_dms_SetPower . . . 1477
9.3.2.98 unpack_dms_SetUSBComp . . . 1478
9.3.2.99 unpack_dms_SLQSDmsSwiGetResetInfo . . . 1478
9.3.2.100 unpack_dms_SLQSDmsSwiGetResetInfo_Ind . . . 1479
9.3.2.101 unpack_dms_SLQSDmsSwiIndicationRegister . . . 1479
9.3.2.102 unpack_dms_SLQSGetBandCapability . . . 1479
9.3.2.103 unpack_dms_SLQSGetERIFile . . . 1480
9.3.2.104 unpack_dms_SLQSSwiClearDyingGaspStatistics . . . 1480
9.3.2.105 unpack_dms_SLQSSwiGetCrashInfo . . . 1480
9.3.2.106 unpack_dms_SLQSSwiGetDyingGaspCfg . . . 1481
9.3.2.107 unpack_dms_SLQSSwiGetDyingGaspStatistics . . . 1481
9.3.2.108 unpack_dms_SLQSSwiGetFirmwareCurr . . . 1481
9.3.2.109 unpack_dms_SLQSSwiGetFwUpdateStatus . . . 1482
9.3.2.110 unpack_dms_SLQSSwiGetHostDevInfo . . . 1482
9.3.2.111 unpack_dms_SLQSSwiGetOSInfo . . . 1482
9.3.2.112 unpack_dms_SLQSSwiGetSerialNoExt . . . 1483
9.3.2.113 unpack_dms_SLQSSwiSetDyingGaspCfg . . . 1483
9.3.2.114 unpack_dms_SLQSSwiSetHostDevInfo . . . 1483
9.3.2.115 unpack_dms_SLQSSwiSetOSInfo . . . 1484
9.3.2.116 unpack_dms_SLQSUIMGetState . . . 1484
9.3.2.117 unpack_dms_UIMChangePIN . . . 1484
9.3.2.118 unpack_dms_UIMGetControlKeyStatus . . . 1485
9.3.2.119 unpack_dms_UIMGetICCID . . . 1485
9.3.2.120 unpack_dms_UIMGetPINStatus . . . 1485
9.3.2.121 unpack_dms_UIMSetControlKeyProtection . . . 1486
9.3.2.122 unpack_dms_UIMSetPINProtection . . . 1486
9.3.2.123 unpack_dms_UIMUnblockControlKey . . . 1487
9.3.2.124 unpack_dms_UIMUnblockPIN . . . 1487
9.3.2.125 unpack_dms_UIMVerifyPIN . . . 1487
9.3.2.126 unpack_dms_ValidateSPC . . . 1488
9.4 fms.h File Reference . . . 1488
9.4.1 Macro Definition Documentation . . . 1489
9.4.1.1 FMS_FW_PRI_BUILD_MATCH_LEN . . . 1489
9.4.1.2 FMS_GOBI_LISTENTRIES_MAX . . . 1489
9.4.1.3 FMS_GOBI_MBN_BUILD_ID_STR_LEN . . . 1489
9.4.1.4 FMS_GOBI_MBN_IMG_ID_STR_LEN . . . 1489
9.4.1.5 FMS_IMAGE_ID_BUILD_ID_LEN . . . 1489
9.4.1.6 FMS_IMAGE_ID_IMG_ID_LEN . . . 1489

9.4.1.7 FMS_IMAGE_ID_MAX_ENTRIES . . . 1489
9.4.1.8 FMS_IMAGE_ID_PRI_IMGTYPE . . . 1489
9.4.1.9 FMS_MAX_IMAGE_ID_ELEMENT . . . 1489
9.4.1.10 FMS_MAX_IMAGE_PREFERENCE_IMAGE_SIZE . . . 1489
9.4.2 Function Documentation . . . 1489
9.4.2.1 GetValidFwPriCombinations . . . 1489
9.4.2.2 pack_fms_GetImagesPreference . . . 1490
9.4.2.3 pack_fms_GetStoredImages . . . 1490
9.4.2.4 pack_fms_SetImagesPreference . . . 1490
9.4.2.5 unpack_fms_GetImagesPreference . . . 1491
9.4.2.6 unpack_fms_GetStoredImages . . . 1491
9.4.2.7 unpack_fms_SetImagesPreference . . . 1491
9.5 loc.h File Reference . . . 1491
9.5.1 Macro Definition Documentation . . . 1494
9.5.1.1 LOC_UINT8_MAX_STRING_SZ . . . 1494
9.5.1.2 LOCEVENTMASKBATCHFULLNOTIFICATION . . . 1494
9.5.1.3 LOCEVENTMASKENGINESTATE . . . 1494
9.5.1.4 LOCEVENTMASKFIXSESSIONSTATE . . . 1494
9.5.1.5 LOCEVENTMASKGEOFENCEBATCHBREACHNOTIFICATION . . . 1494
9.5.1.6 LOCEVENTMASKGEOFENCEBREACHNOTIFICATION . . . 1494
9.5.1.7 LOCEVENTMASKGEOFENCEGENALERT . . . 1494
9.5.1.8 LOCEVENTMASKGNSSMEASUREMENTREPORT . . . 1495
9.5.1.9 LOCEVENTMASKGNSSSVINFO . . . 1495
9.5.1.10 LOCEVENTMASKINJECTPOSITIONREQ . . . 1495
9.5.1.11 LOCEVENTMASKINJECTPREDICTEDORBITSREQ . . . 1495
9.5.1.12 LOCEVENTMASKINJECTTIMEREQ . . . 1495
9.5.1.13 LOCEVENTMASKINJECTWIFIAPDATAREQ . . . 1495
9.5.1.14 LOCEVENTMASKINVALIDVALUE . . . 1495
9.5.1.15 LOCEVENTMASKLIVEBATCHEDPOSITIONREPORT . . . 1495
9.5.1.16 LOCEVENTMASKLOCATIONSERVERCONNECTIONREQ . . . 1495
9.5.1.17 LOCEVENTMASKMOTIONDATACONTROL . . . 1495
9.5.1.18 LOCEVENTMASKNIGEOFENCENOTIFICATION . . . 1495
9.5.1.19 LOCEVENTMASKNINOTIFYVERIFYREQ . . . 1496
9.5.1.20 LOCEVENTMASKNMEA . . . 1496
9.5.1.21 LOCEVENTMASKPEDOMETERCONTROL . . . 1496
9.5.1.22 LOCEVENTMASKPOSITIONREPORT . . . 1496
9.5.1.23 LOCEVENTMASKSENSORSTREAMINGREADYSTATUS . . . 1496
9.5.1.24 LOCEVENTMASKSETSPISTREAMINGREPORT . . . 1496
9.5.1.25 LOCEVENTMASKTIMESYNCREQ . . . 1496
9.5.1.26 LOCEVENTMASKVEHICLEDATAREADYSTATUS . . . 1496

9.5.1.27 LOCEVENTMASKWIFIREQ . . . 1496
9.5.1.28 MAX_LOC_NMEA_STR_LEN . . . 1496
9.5.1.29 MAX_SENSOR_DATA_LEN . . . 1496
9.5.1.30 MAX_TEMP_DATA_LEN . . . 1496
9.5.2 Enumeration Type Documentation . . . 1496
9.5.2.1 anonymous enum . . . 1497
9.5.3 Function Documentation . . . 1497
9.5.3.1 pack_loc_DeleteAssistData . . . 1497
9.5.3.2 pack_loc_EventRegister . . . 1497
9.5.3.3 pack_loc_SetExtPowerState . . . 1497
9.5.3.4 pack_loc_SetOperationMode . . . 1498
9.5.3.5 pack_loc_SLQSLOCGetBestAvailPos . . . 1498
9.5.3.6 pack_loc_SLQSLOCGetServer . . . 1499
9.5.3.7 pack_loc_SLQSLOCInjectPosition . . . 1499
9.5.3.8 pack_loc_SLQSLOCInjectSensorData . . . 1499
9.5.3.9 pack_loc_SLQSLOCInjectUTCTime . . . 1500
9.5.3.10 pack_loc_SLQSLOCSetCradleMountConfig . . . 1500
9.5.3.11 pack_loc_Start . . . 1500
9.5.3.12 pack_loc_Stop . . . 1501
9.5.3.13 unpack_loc_BestAvailPos_Ind . . . 1501
9.5.3.14 unpack_loc_DeleteAssistData . . . 1502
9.5.3.15 unpack_loc_DeleteAssistData_Ind . . . 1502
9.5.3.16 unpack_loc_EngineState_Ind . . . 1502
9.5.3.17 unpack_loc_EventNMEA_Ind . . . 1503
9.5.3.18 unpack_loc_EventRegister . . . 1503
9.5.3.19 unpack_loc_GetServer_Ind . . . 1503
9.5.3.20 unpack_loc_GnssSvInfo_Ind . . . 1504
9.5.3.21 unpack_loc_PositionRpt_Ind . . . 1504
9.5.3.22 unpack_loc_SetExtPowerConfig_Ind . . . 1504
9.5.3.23 unpack_loc_SetExtPowerState . . . 1505
9.5.3.24 unpack_loc_SetOperationMode . . . 1505
9.5.3.25 unpack_loc_SetOperationMode_Ind . . . 1505
9.5.3.26 unpack_loc_SLQSLOCGetBestAvailPos . . . 1506
9.5.3.27 unpack_loc_SLQSLOCGetServer . . . 1506
9.5.3.28 unpack_loc_SLQSLOCInjectPosition . . . 1506
9.5.3.29 unpack_loc_SLQSLOCInjectSensorData . . . 1507
9.5.3.30 unpack_loc_SLQSLOCInjectUTCTime . . . 1507
9.5.3.31 unpack_loc_SLQSLOCSetCradleMountConfig . . . 1507
9.5.3.32 unpack_loc_Start . . . 1508
9.5.3.33 unpack_loc_Stop . . . 1508

9.6 nas.h File Reference . . . 1508
9.6.1 Macro Definition Documentation . . . 1516
9.6.1.1 NAS_IMSI_M_S1_LENGTH . . . 1516
9.6.1.2 NAS_IMSI_M_S2_LENGTH . . . 1516
9.6.1.3 NAS_MAX_DESCRIPTION_LENGTH . . . 1516
9.6.1.4 NAS_MAX_NUM_NETWORKS . . . 1516
9.6.1.5 NAS_MAX_SCC_RX_INFO_INSTANCES . . . 1516
9.6.1.6 NAS_NAM_NAME_LENGTH . . . 1516
9.6.1.7 NAS_OTA_MESSAGE_MAX_BUF_SIZE . . . 1516
9.6.1.8 NAS_PLMN_LENGTH . . . 1516
9.6.1.9 NAS_SERVING_SYSTEM_INFO_MAX_RADIO_INTERFACE_LIST . . . 1516
9.6.2 Enumeration Type Documentation . . . 1517
9.6.2.1 LITEQMI_NAS_LTE_CPHY_CA_BW_NRB . . . 1517
9.6.2.2 LITEQMI_NAS_LTE_CPHY_SCELL_STATE . . . 1517
9.6.2.3 NAS_LTE_CPHY_CA_BW_NRB_LITE . . . 1517
9.6.2.4 NAS_LTE_CPHY_SCELL_STATE_LITE . . . 1517
9.6.3 Function Documentation . . . 1517
9.6.3.1 pack_nas_GetACCOLC . . . 1517
9.6.3.2 pack_nas_GetANAAAAuthenticationStatus . . . 1518
9.6.3.3 pack_nas_GetCDMANetworkParameters . . . 1518
9.6.3.4 pack_nas_GetHomeNetwork . . . 1518
9.6.3.5 pack_nas_GetHomeNetwork3GPP2 . . . 1519
9.6.3.6 pack_nas_GetNetworkPreference . . . 1519
9.6.3.7 pack_nas_GetRFInfo . . . 1519
9.6.3.8 pack_nas_GetServingNetwork . . . 1520
9.6.3.9 pack_nas_GetServingNetworkCapabilities . . . 1520
9.6.3.10 pack_nas_GetSignalStrengths . . . 1520
9.6.3.11 pack_nas_InitiateDomainAttach . . . 1521
9.6.3.12 pack_nas_PerformNetworkScan . . . 1521
9.6.3.13 pack_nas_SetACCOLC . . . 1521
9.6.3.14 pack_nas_SetCDMANetworkParameters . . . 1522
9.6.3.15 pack_nas_SetLURejectCallback . . . 1522
9.6.3.16 pack_nas_SetNetworkPreference . . . 1522
9.6.3.17 pack_nas_SetRFInfoCallback . . . 1523
9.6.3.18 pack_nas_SLQSConfigSigInfo . . . 1523
9.6.3.19 pack_nas_SLQSGetErrorRate . . . 1523
9.6.3.20 pack_nas_SlqsGetLTECphyCAInfo . . . 1524
9.6.3.21 pack_nas_SLQSGetNetworkTime . . . 1524
9.6.3.22 pack_nas_SLQSGetOperatorNameData . . . 1524
9.6.3.23 pack_nas_SLQSGetPLMNName . . . 1524

9.6.3.24 pack_nas_SLQSGetServingSystem . . . 1525
9.6.3.25 pack_nas_SLQSGetSignalStrength . . . 1525
9.6.3.26 pack_nas_SLQSGetSysInfo . . . 1526
9.6.3.27 pack_nas_SLQSGetSysSelectionPref . . . 1526
9.6.3.28 pack_nas_SLQSInitiateNetworkRegistration . . . 1526
9.6.3.29 pack_nas_SLQSNasConfigSigInfo2 . . . 1527
9.6.3.30 pack_nas_SLQSNasGet3GPP2Subscription . . . 1527
9.6.3.31 pack_nas_SLQSNasGetCellLocationInfo . . . 1527
9.6.3.32 pack_nas_SLQSNasGetHDRColorCode . . . 1528
9.6.3.33 pack_nas_SLQSNasGetSigInfo . . . 1528
9.6.3.34 pack_nas_SLQSNasGetTxRxInfo . . . 1528
9.6.3.35 pack_nas_SLQSNasIndicationRegisterExt . . . 1529
9.6.3.36 pack_nas_SLQSNASSwiGetChannelLock . . . 1529
9.6.3.37 pack_nas_SLQSNasSwiIndicationRegister . . . 1530
9.6.3.38 pack_nas_SLQSNasSwiModemStatus . . . 1530
9.6.3.39 pack_nas_SLQSNASSwiSetChannelLock . . . 1530
9.6.3.40 pack_nas_SLQSSetBandPreference . . . 1531
9.6.3.41 pack_nas_SLQSSetSignalStrengthsCallback . . . 1531
9.6.3.42 pack_nas_SLQSSetSysSelectionPref . . . 1531
9.6.3.43 pack_nas_SLQSSwiGetHDRPersonality . . . 1532
9.6.3.44 pack_nas_SLQSSwiGetHDRProtSubtype . . . 1532
9.6.3.45 pack_nas_SLQSSwiGetHRPDStats . . . 1532
9.6.3.46 pack_nas_SLQSSwiGetLteCQI . . . 1533
9.6.3.47 pack_nas_SLQSSwiGetLteSccRxInfo . . . 1533
9.6.3.48 pack_nas_SLQSSwiNetworkDebug . . . 1533
9.6.3.49 pack_nas_SLQSSwiPSDetach . . . 1534
9.6.3.50 unpack_nas_GetACCOLC . . . 1534
9.6.3.51 unpack_nas_GetANAAAAuthenticationStatus . . . 1534
9.6.3.52 unpack_nas_GetCDMANetworkParameters . . . 1535
9.6.3.53 unpack_nas_GetHomeNetwork . . . 1535
9.6.3.54 unpack_nas_GetHomeNetwork3GPP2 . . . 1536
9.6.3.55 unpack_nas_GetNetworkPreference . . . 1536
9.6.3.56 unpack_nas_GetRFInfo . . . 1536
9.6.3.57 unpack_nas_GetServingNetwork . . . 1537
9.6.3.58 unpack_nas_GetServingNetworkCapabilities . . . 1537
9.6.3.59 unpack_nas_GetSignalStrengths . . . 1537
9.6.3.60 unpack_nas_InitiateDomainAttach . . . 1538
9.6.3.61 unpack_nas_PerformNetworkScan . . . 1538
9.6.3.62 unpack_nas_SetACCOLC . . . 1538
9.6.3.63 unpack_nas_SetCDMANetworkParameters . . . 1539

9.6.3.64 unpack_nas_SetDataCapabilitiesCallback_ind . . . 1539
9.6.3.65 unpack_nas_SetEventReportInd . . . 1539
9.6.3.66 unpack_nas_SetLURejectCallback . . . 1540
9.6.3.67 unpack_nas_SetNasLTECphyCaIndCallback_ind . . . 1540
9.6.3.68 unpack_nas_SetNetworkPreference . . . 1540
9.6.3.69 unpack_nas_SetRFInfoCallback . . . 1541
9.6.3.70 unpack_nas_SetRoamingIndicatorCallback_ind . . . 1541
9.6.3.71 unpack_nas_SetServingSystemCallback_ind . . . 1541
9.6.3.72 unpack_nas_SLQSConfigSigInfo . . . 1542
9.6.3.73 unpack_nas_SLQSGetErrorRate . . . 1542
9.6.3.74 unpack_nas_SlqsGetLTECphyCAInfo . . . 1542
9.6.3.75 unpack_nas_SLQSGetNetworkTime . . . 1543
9.6.3.76 unpack_nas_SLQSGetOperatorNameData . . . 1543
9.6.3.77 unpack_nas_SLQSGetPLMNName . . . 1544
9.6.3.78 unpack_nas_SLQSGetServingSystem . . . 1544
9.6.3.79 unpack_nas_SLQSGetSignalStrength . . . 1544
9.6.3.80 unpack_nas_SLQSGetSysInfo . . . 1545
9.6.3.81 unpack_nas_SLQSGetSysSelectionPref . . . 1545
9.6.3.82 unpack_nas_SLQSInitiateNetworkRegistration . . . 1545
9.6.3.83 unpack_nas_SLQSNasConfigSigInfo2 . . . 1546
9.6.3.84 unpack_nas_SLQSNasGet3GPP2Subscription . . . 1546
9.6.3.85 unpack_nas_SLQSNasGetCellLocationInfo . . . 1546
9.6.3.86 unpack_nas_SLQSNasGetHDRColorCode . . . 1547
9.6.3.87 unpack_nas_SLQSNasGetSigInfo . . . 1547
9.6.3.88 unpack_nas_SLQSNasGetTxRxInfo . . . 1548
9.6.3.89 unpack_nas_SLQSNasIndicationRegisterExt . . . 1548
9.6.3.90 unpack_nas_SLQSNasNetworkTimeCallBack_ind . . . 1548
9.6.3.91 unpack_nas_SLQSNasSigInfoCallback_ind . . . 1549
9.6.3.92 unpack_nas_SLQSNASSwiGetChannelLock . . . 1549
9.6.3.93 unpack_nas_SLQSNasSwiIndicationRegister . . . 1549
9.6.3.94 unpack_nas_SLQSNasSwiModemStatus . . . 1550
9.6.3.95 unpack_nas_SLQSNasSwiOTAMessageCallback_ind . . . 1550
9.6.3.96 unpack_nas_SLQSNASSwiSetChannelLock . . . 1550
9.6.3.97 unpack_nas_SLQSNasSysInfoCallback_ind . . . 1551
9.6.3.98 unpack_nas_SLQSNasTimerCallback_ind . . . 1551
9.6.3.99 unpack_nas_SLQSSetBandPreference . . . 1551
9.6.3.100 unpack_nas_SLQSSetSignalStrengthsCallback . . . 1552
9.6.3.101 unpack_nas_SLQSSetSysSelectionPref . . . 1552
9.6.3.102 unpack_nas_SLQSSetSysSelectionPrefCallBack_ind . . . 1552
9.6.3.103 unpack_nas_SLQSSwiGetHDRPersonality . . . 1553

9.6.3.104 unpack_nas_SLQSSwiGetHDRProtSubtype . . . 1553
9.6.3.105 unpack_nas_SLQSSwiGetHRPDStats . . . 1553
9.6.3.106 unpack_nas_SLQSSwiGetLteCQI . . . 1554
9.6.3.107 unpack_nas_SLQSSwiGetLteSccRxInfo . . . 1554
9.6.3.108 unpack_nas_SLQSSwiNetworkDebug . . . 1554
9.6.3.109 unpack_nas_SLQSSwiPSDetach . . . 1555
9.6.3.110 unpack_valid_nas_GetCDMANetworkParameters . . . 1555
9.6.3.111 unpack_valid_nas_SLQSGetServingSystem . . . 1555
9.6.3.112 unpack_valid_nas_SLQSGetSignalStrength . . . 1556
9.6.3.113 unpack_valid_nas_SLQSNasGetSigInfo . . . 1557
9.7 pds.h File Reference . . . 1557
9.7.1 Function Documentation . . . 1559
9.7.1.1 pack_pds_ForceXTRADownload . . . 1559
9.7.1.2 pack_pds_GetPDSDefaults . . . 1560
9.7.1.3 pack_pds_GetPDSState . . . 1560
9.7.1.4 pack_pds_GetPortAutomaticTracking . . . 1560
9.7.1.5 pack_pds_GetServiceAutomaticTracking . . . 1561
9.7.1.6 pack_pds_GetXTRAAutomaticDownload . . . 1561
9.7.1.7 pack_pds_GetXTRANetwork . . . 1561
9.7.1.8 pack_pds_GetXTRAValidity . . . 1562
9.7.1.9 pack_pds_PDSInjectTimeReference . . . 1562
9.7.1.10 pack_pds_ResetPDSData . . . 1562
9.7.1.11 pack_pds_SetEventReportCallback . . . 1563
9.7.1.12 pack_pds_SetPDSDefaults . . . 1563
9.7.1.13 pack_pds_SetPDSState . . . 1563
9.7.1.14 pack_pds_SetPortAutomaticTracking . . . 1564
9.7.1.15 pack_pds_SetServiceAutomaticTracking . . . 1564
9.7.1.16 pack_pds_SetXTRAAutomaticDownload . . . 1564
9.7.1.17 pack_pds_SetXTRANetwork . . . 1565
9.7.1.18 pack_pds_SLQSGetAGPSConfig . . . 1565
9.7.1.19 pack_pds_SLQSGetGPSStateInfo . . . 1566
9.7.1.20 pack_pds_SLQSPDSDeterminePosition . . . 1566
9.7.1.21 pack_pds_SLQSPDSInjectAbsoluteTimeReference . . . 1566
9.7.1.22 pack_pds_SLQSPDSInjectPositionData . . . 1567
9.7.1.23 pack_pds_SLQSSetAGPSConfig . . . 1567
9.7.1.24 pack_pds_SLQSSetPositionMethodState . . . 1567
9.7.1.25 pack_pds_StartPDSTrackingSessionExt . . . 1568
9.7.1.26 pack_pds_StopPDSTrackingSession . . . 1568
9.7.1.27 unpack_pds_ForceXTRADownload . . . 1568
9.7.1.28 unpack_pds_GetPDSDefaults . . . 1569

9.7.1.29 unpack_pds_GetPDSState . . . 1569
9.7.1.30 unpack_pds_GetPortAutomaticTracking . . . 1569
9.7.1.31 unpack_pds_GetServiceAutomaticTracking . . . 1570
9.7.1.32 unpack_pds_GetXTRAAutomaticDownload . . . 1570
9.7.1.33 unpack_pds_GetXTRANetwork . . . 1570
9.7.1.34 unpack_pds_GetXTRAValidity . . . 1571
9.7.1.35 unpack_pds_PDSInjectTimeReference . . . 1571
9.7.1.36 unpack_pds_ResetPDSData . . . 1571
9.7.1.37 unpack_pds_SetEventReport_Ind . . . 1572
9.7.1.38 unpack_pds_SetEventReportCallback . . . 1572
9.7.1.39 unpack_pds_SetPDSDefaults . . . 1572
9.7.1.40 unpack_pds_SetPDSState . . . 1573
9.7.1.41 unpack_pds_SetPdsState_Ind . . . 1573
9.7.1.42 unpack_pds_SetPortAutomaticTracking . . . 1573
9.7.1.43 unpack_pds_SetServiceAutomaticTracking . . . 1574
9.7.1.44 unpack_pds_SetXTRAAutomaticDownload . . . 1574
9.7.1.45 unpack_pds_SetXTRANetwork . . . 1574
9.7.1.46 unpack_pds_SLQSGetAGPSConfig . . . 1575
9.7.1.47 unpack_pds_SLQSGetGPSStateInfo . . . 1575
9.7.1.48 unpack_pds_SLQSPDSDeterminePosition . . . 1575
9.7.1.49 unpack_pds_SLQSPDSInjectAbsoluteTimeReference . . . 1576
9.7.1.50 unpack_pds_SLQSPDSInjectPositionData . . . 1576
9.7.1.51 unpack_pds_SLQSSetAGPSConfig . . . 1576
9.7.1.52 unpack_pds_SLQSSetPositionMethodState . . . 1577
9.7.1.53 unpack_pds_StartPDSTrackingSessionExt . . . 1577
9.7.1.54 unpack_pds_StopPDSTrackingSession . . . 1577
9.8 qaCbkCatEventReportInd.h File Reference . . . 1578
9.8.1 Macro Definition Documentation . . . 1579
9.8.1.1 QMI_CAN_COMMON_EVENT_TLV_NUMBER . . . 1579
9.8.1.2 QMI_MAX_CAT_EVENT_DATA_LENGTH . . . 1579
9.8.2 Enumeration Type Documentation . . . 1579
9.8.2.1 eQMI_CAT_EVENT_REPORT_IND_TLV . . . 1579
9.8.2.2 eQMI_CAT_EVENT_REPORT_IND_TLV_LENGTH . . . 1579
9.8.3 Function Documentation . . . 1579
9.8.3.1 UpkQmiCbkCatEventReportInd . . . 1579
9.9 qaCbkSwiOmaDmEventReportInd.h File Reference . . . 1579
9.9.1 Macro Definition Documentation . . . 1580
9.9.1.1 QMI_SWIOMA_DM_CONFIG . . . 1580
9.9.1.2 QMI_SWIOMA_DM_FOTA . . . 1580
9.9.1.3 QMI_SWIOMA_DM_NOT . . . 1580

9.9.2 Enumeration Type Documentation . . . 1580
9.9.2.1 eQMI_SWIOMA_DM_EVENT_REPORT_IND . . . 1580
9.9.3 Function Documentation . . . 1580
9.9.3.1 UpkQmiCbkSwiOmaDmEventReportInd . . . 1580
9.9.3.2 UpkQmiCbkSwiOmaDmEventReportIndExt . . . 1580
9.10 qaGobiApiAudio.h File Reference . . . 1580
9.10.1 Detailed Description . . . 1581
9.10.2 Function Documentation . . . 1581
9.10.2.1 SLQSGetAudioPathConfig . . . 1581
9.10.2.2 SLQSGetAudioProfile . . . 1582
9.10.2.3 SLQSGetAudioVolTLBConfig . . . 1582
9.10.2.4 SLQSSetAudioPathConfig . . . 1583
9.10.2.5 SLQSSetAudioProfile . . . 1583
9.10.2.6 SLQSSetAudioVolTLBConfig . . . 1583
9.11 qaGobiApiCat.h File Reference . . . 1584
9.11.1 Detailed Description . . . 1584
9.11.2 Function Documentation . . . 1584
9.11.2.1 CATSendEnvelopeCommand . . . 1584
9.11.2.2 CATSendTerminalResponse . . . 1585
9.12 qaGobiApiCbk.h File Reference . . . 1585
9.12.1 Detailed Description . . . 1594
9.12.2 Macro Definition Documentation . . . 1594
9.12.2.1 CBK_DISABLE_EVENT . . . 1594
9.12.2.2 CBK_ENABLE_EVENT . . . 1594
9.12.2.3 CBK_NOCHANGE . . . 1594
9.12.2.4 DEREGISTER_EVENT . . . 1594
9.12.2.5 DEREGISTER_SRV . . . 1594
9.12.2.6 DHCP_MAX_NUM_OPTIONS . . . 1594
9.12.2.7 DHCP_OPTION_DATA_BUF_SIZE . . . 1594
9.12.2.8 EVENT_MASK_CARD . . . 1594
9.12.2.9 EVENT_MASK_DEREGISTER_ALL . . . 1594
9.12.2.10 EVENT_MASK_PHY_SLOT_STATUS . . . 1594
9.12.2.11 FIRST_INSTANCE . . . 1594
9.12.2.12 INVALID_INSTACNE . . . 1594
9.12.2.13 IPV4 . . . 1594
9.12.2.14 IPV4V6 . . . 1594
9.12.2.15 IPV6 . . . 1594
9.12.2.16 LOC_EVENT_MASK_ENG_STATE . . . 1594
9.12.2.17 LOC_EVENT_MASK_GNSS_SV_INFO . . . 1594
9.12.2.18 LOC_EVENT_MASK_INJECT_TIME . . . 1594

9.12.2.19 LOC_EVENT_MASK_NMEA . . . 1594
9.12.2.20 LOC_EVENT_MASK_SENSOR_STREAM . . . 1594
9.12.2.21 LOC_EVENT_MASK_TIME_SYNC . . . 1594
9.12.2.22 LOC_EVENT_POSITION_REPORT . . . 1594
9.12.2.23 MAX_MITIGATION_DEV_ID_LEN . . . 1594
9.12.2.24 MAX_NMEA_STR_LEN . . . 1594
9.12.2.25 MAX_NO_OF_APPLICATIONS . . . 1594
9.12.2.26 MAX_NO_OF_CALLS . . . 1595
9.12.2.27 MAX_NO_OF_FILES . . . 1595
9.12.2.28 MAX_NO_OF_SLOTS . . . 1595
9.12.2.29 MAX_NO_OF_UUSINFO . . . 1595
9.12.2.30 MAX_PATH_LENGTH . . . 1595
9.12.2.31 MAX_RADIO_INTERFACE_LIST . . . 1595
9.12.2.32 MAX_URL_ADDR_STR_LEN . . . 1595
9.12.2.33 MAXUSSDLENGTH . . . 1595
9.12.2.34 NAS_SRV . . . 1595
9.12.2.35 NUM_OF_SET . . . 1595
9.12.2.36 PDS_SRV . . . 1595
9.12.2.37 QMI_ETWS_MAX_PAYLOAD_LENGTH . . . 1595
9.12.2.38 QMI_MAX_VOICE_NUMBER_LENGTH . . . 1595
9.12.2.39 QMI_WMS_MAX_PAYLOAD_LENGTH . . . 1595
9.12.2.40 REGISTER_EVENT . . . 1595
9.12.2.41 REGISTER_SRV . . . 1595
9.12.2.42 SECOND_INSTANCE . . . 1595
9.12.2.43 SIGSTRENGTH_THRESHOLD_ARR_SZ . . . 1595
9.12.2.44 THIRD_INSTANCE . . . 1595
9.12.2.45 USSD_DCS_8BIT . . . 1595
9.12.2.46 USSD_DCS_ASCII . . . 1595
9.12.2.47 USSD_DCS_UCS2 . . . 1595
9.12.2.48 VOICE_SRV . . . 1595
9.12.2.49 WDS_SRV . . . 1595
9.12.3 Typedef Documentation . . . 1595
9.12.3.1 accelAcceptReady . . . 1595
9.12.3.2 accelTempAcceptReady . . . 1596
9.12.3.3 eDevState . . . 1596
9.12.3.4 eSMSEventType . . . 1596
9.12.3.5 gpsTime . . . 1597
9.12.3.6 gyroAcceptReady . . . 1597
9.12.3.7 gyroTempAcceptReady . . . 1597
9.12.3.8 LteNasReleaseInfo . . . 1598

9.12.3.9 modemTempNotification . . . 1598
9.12.3.10 packetSrvStatus . . . 1599
9.12.3.11 precisionDilution . . . 1600
9.12.3.12 ResetInfoNotification . . . 1600
9.12.3.13 sensorDataUsage . . . 1601
9.12.3.14 sessionInformation . . . 1601
9.12.3.15 sessionInformationExt . . . 1602
9.12.3.16 SMSAsyncRawSend . . . 1602
9.12.3.17 SMSCAddressInfo . . . 1603
9.12.3.18 SMSEtwsMessageInfo . . . 1603
9.12.3.19 SMSEtwsPlmnInfo . . . 1603
9.12.3.20 SMSEventInfo . . . 1603
9.12.3.21 SMSMessageModeInfo . . . 1604
9.12.3.22 SMSMTMessageInfo . . . 1604
9.12.3.23 SMSOnIMSInfo . . . 1605
9.12.3.24 SMSTransferRouteMTMessageInfo . . . 1605
9.12.3.25 svUsedforFix . . . 1605
9.12.3.26 SwiAvmsEventInd . . . 1606
9.12.3.27 SwiOTAMsg . . . 1606
9.12.3.28 tFNActivationStatus . . . 1606
9.12.3.29 tFNAllCallStatus . . . 1607
9.12.3.30 tFNASwiLTECphyCaInfo . . . 1607
9.12.3.31 tFNASwiOTAMsg . . . 1607
9.12.3.32 tFNAsyncRawSend . . . 1607
9.12.3.33 tFNBandPreference . . . 1608
9.12.3.34 tFNBestAvailPos . . . 1610
9.12.3.35 tFNCATEvent . . . 1610
9.12.3.36 tFNCbkUimSlotStatusChangeInd . . . 1610
9.12.3.37 tFNDataCapabilities . . . 1611
9.12.3.38 tFNDataSysStatus . . . 1611
9.12.3.39 tFNDelAssistData . . . 1611
9.12.3.40 tFNDeviceStateChange . . . 1611
9.12.3.41 tFNDHCPv4ClientLeaseStatus . . . 1612
9.12.3.42 tFNDTMFEvent . . . 1612
9.12.3.43 tFNDUNCallInfo . . . 1612
9.12.3.44 tFNEventNmea . . . 1612
9.12.3.45 tFNEventPosition . . . 1613
9.12.3.46 tFNFwDldCompletion . . . 1613
9.12.3.47 tFNGetServer . . . 1613
9.12.3.48 tFNGnssSvInfo . . . 1613

9.12.3.49 tFNHDRPersonaity . . . 1614
9.12.3.50 tFNImsaPdpStatus . . . 1614
9.12.3.51 tFNImsaRatStatus . . . 1614
9.12.3.52 tFNImsaRegStatus . . . 1614
9.12.3.53 tFNImsaSvcStatus . . . 1614
9.12.3.54 tFNImsRegMgrConfig . . . 1615
9.12.3.55 tFNImsSIPConfig . . . 1615
9.12.3.56 tFNImsSMSConfig . . . 1615
9.12.3.57 tFNImsUserConfig . . . 1615
9.12.3.58 tFNImsVoIPConfig . . . 1615
9.12.3.59 tFNInfoRec . . . 1616
9.12.3.60 tFNInjectPosition . . . 1616
9.12.3.61 tFNInjectSensorData . . . 1616
9.12.3.62 tFNInjectTimeStatus . . . 1616
9.12.3.63 tFNInjectUTCTime . . . 1616
9.12.3.64 tFNLUReject . . . 1617
9.12.3.65 tFNMemoryFull . . . 1619
9.12.3.66 tFNMessageWaiting . . . 1619
9.12.3.67 tFNMitiLvlRpt . . . 1619
9.12.3.68 tFNMobileIPStatus . . . 1619
9.12.3.69 tFNModemTempInfo . . . 1619
9.12.3.70 tFNNasTimer . . . 1620
9.12.3.71 tFNNet . . . 1620
9.12.3.72 tFNNetworkReject . . . 1620
9.12.3.73 tFNNetworkTime . . . 1620
9.12.3.74 tFNNewGPS . . . 1621
9.12.3.75 tFNNewNMEA . . . 1621
9.12.3.76 tFNNewRMTransferStatistics . . . 1622
9.12.3.77 tFNNewSMS . . . 1622
9.12.3.78 tFNOMADMState . . . 1622
9.12.3.79 tFNOpMode . . . 1623
9.12.3.80 tFNOTASPStatus . . . 1624
9.12.3.81 tFNPacketSrvState . . . 1624
9.12.3.82 tFNPDSState . . . 1624
9.12.3.83 tFNPower . . . 1625
9.12.3.84 tFNPrivacyChange . . . 1625
9.12.3.85 tFNQosNWStatus . . . 1625
9.12.3.86 tFNQosPriEvent . . . 1626
9.12.3.87 tFNQosStatus . . . 1626
9.12.3.88 tFNRankIndicator . . . 1627

9.12.3.89 tFNResetInfo . . . 1627
9.12.3.90 tFNRFInfo . . . 1627
9.12.3.91 tFNRoamingIndicator . . . 1628
9.12.3.92 tFNSDKTerminated . . . 1628
9.12.3.93 tFNSensorStreaming . . . 1628
9.12.3.94 tFNServingSystem . . . 1629
9.12.3.95 tFNSetCradleMount . . . 1629
9.12.3.96 tFNSetEngineState . . . 1629
9.12.3.97 tFNSetEventTimeSync . . . 1629
9.12.3.98 tFNSetExtPowerConfig . . . 1629
9.12.3.99 tFNSigInfo . . . 1630
9.12.3.100 tFNSignalStrength . . . 1630
9.12.3.101 tFNSLQSOMADMAlert . . . 1630
9.12.3.102 tFNSLQSQOSEvent . . . 1630
9.12.3.103 tFNSLQSSessionState . . . 1631
9.12.3.104 tFNSLQSSignalStrengths . . . 1631
9.12.3.105 tFNSLQSWDSEvent . . . 1631
9.12.3.106 tFNSMSEvents . . . 1631
9.12.3.107 tFNSUPSInfo . . . 1631
9.12.3.108 tFNSUPSNotification . . . 1632
9.12.3.109 tFNSwiAvmEventReport . . . 1632
9.12.3.110 tFNSysInfo . . . 1632
9.12.3.111 tFNSysSelectionPref . . . 1632
9.12.3.112 tFNtransLayerInfo . . . 1632
9.12.3.113 tFNtransNWRegInfo . . . 1633
9.12.3.114 tFNUIMRefresh . . . 1633
9.12.3.115 tFNUIMStatusChangeInfo . . . 1633
9.12.3.116 tFNUSSDNotification . . . 1633
9.12.3.117 tFNUSSDNoWaitIndication . . . 1634
9.12.3.118 tFNUSSDRelease . . . 1634
9.12.3.119 transLayerNotification . . . 1634
9.12.3.120 transNWRegInfoNotification . . . 1634
9.12.4 Enumeration Type Documentation . . . 1635
9.12.4.1 device_state_enum . . . 1635
9.12.4.2 eQaQMIService . . . 1635
9.12.4.3 SMSEventType . . . 1635
9.12.4.4 TYPE_OF_SERVICE_DOMAIN . . . 1635
9.12.5 Function Documentation . . . 1636
9.12.5.1 iSetCATEventCallback . . . 1636
9.12.5.2 iSetSignalStrengthCallback . . . 1636

9.12.5.3 iSLQSSetDUNCallInfoCallback . . . 1636
9.12.5.4 iSLQSSetSignalStrengthsCallback . . . 1636
9.12.5.5 iSLQSSetWdsFirstInstEventCallback . . . 1636
9.12.5.6 iSLQSSetWdsSecondInstEventCallback . . . 1636
9.12.5.7 iSLQSSetWdsThirdInstEventCallback . . . 1636
9.12.5.8 iSLQSSetWdsXferStatsFirstInstCallback . . . 1636
9.12.5.9 iSLQSSetWdsXferStatsSecondInstCallback . . . 1636
9.12.5.10 SetActivationStatusCallback . . . 1636
9.12.5.11 SetCATEventCallback . . . 1636
9.12.5.12 SetDataCapabilitiesCallback . . . 1637
9.12.5.13 SetDeviceStateChangeCbk . . . 1638
9.12.5.14 SetFwDldCompletionCbk . . . 1638
9.12.5.15 SetGPSCallback . . . 1639
9.12.5.16 SetLocBestAvailPosCallback . . . 1639
9.12.5.17 SetLocCradleMountCallback . . . 1639
9.12.5.18 SetLocDeleteAssistDataCallback . . . 1640
9.12.5.19 SetLocEngineStateCallback . . . 1640
9.12.5.20 SetLocEventMaskNMEACallback . . . 1640
9.12.5.21 SetLocEventPositionCallback . . . 1640
9.12.5.22 SetLocEventTimeSyncCallback . . . 1641
9.12.5.23 SetLocGetServerCallback . . . 1641
9.12.5.24 SetLocGnssSvInfoCallback . . . 1641
9.12.5.25 SetLocInjectSensorDataCallback . . . 1642
9.12.5.26 SetLocInjectTimeCallback . . . 1642
9.12.5.27 SetLocOpModeCallback . . . 1642
9.12.5.28 SetLocSensorStreamingCallback . . . 1642
9.12.5.29 SetLocSetExtPowerConfigCallback . . . 1643
9.12.5.30 SetLURejectCallback . . . 1643
9.12.5.31 SetMobileIPStatusCallback . . . 1643
9.12.5.32 SetNasLTECphyCaIndCallback . . . 1644
9.12.5.33 SetNetChangeCbk . . . 1644
9.12.5.34 SetNewSMSCallback . . . 1645
9.12.5.35 SetNMEACallback . . . 1645
9.12.5.36 SetOMADMStateCallback . . . 1646
9.12.5.37 SetPDSStateCallback . . . 1646
9.12.5.38 SetPowerCallback . . . 1646
9.12.5.39 SetRankIndicatorCallback . . . 1647
9.12.5.40 SetRFInfoCallback . . . 1647
9.12.5.41 SetRMTransferStatisticsCallback . . . 1647
9.12.5.42 SetRoamingIndicatorCallback . . . 1648

9.12.5.43 SetSignalStrengthCallback . . . 1648
9.12.5.44 SetSLQSOMADMAlertCallback . . . 1649
9.12.5.45 SetSLQSOMADMAlertCallbackExt . . . 1649
9.12.5.46 SetUimSlotStatusChangeCallback . . . 1649
9.12.5.47 SetUSSDNotificationCallback . . . 1650
9.12.5.48 SetUSSDNoWaitIndicationCallback . . . 1650
9.12.5.49 SetUSSDReleaseCallback . . . 1650
9.12.5.50 SLQSNasNetworkRejectCallback . . . 1651
9.12.5.51 SLQSNasNetworkTimeCallBack . . . 1651
9.12.5.52 SLQSNasSigInfo2CallBack . . . 1651
9.12.5.53 SLQSNasSigInfoCallBack . . . 1652
9.12.5.54 SLQSNasSwiOTAMessageCallback . . . 1652
9.12.5.55 SLQSNasSysInfoCallBack . . . 1653
9.12.5.56 SLQSNasTimerCallback . . . 1653
9.12.5.57 SLQSSetBandPreferenceCbk . . . 1654
9.12.5.58 SLQSSetDataSystemStatusCallback . . . 1654
9.12.5.59 SLQSSetDHCPv4ClientLeaseStatusCallback . . . 1654
9.12.5.60 SLQSSetDUNCallInfoCallback . . . 1655
9.12.5.61 SLQSSetIMSAPdpStatusCallback . . . 1655
9.12.5.62 SLQSSetIMSARatStatusCallback . . . 1656
9.12.5.63 SLQSSetIMSARegStatusCallback . . . 1656
9.12.5.64 SLQSSetIMSASvcStatusCallback . . . 1656
9.12.5.65 SLQSSetIMSSMSConfigCallback . . . 1657
9.12.5.66 SLQSSetIMSUserConfigCallback . . . 1657
9.12.5.67 SLQSSetIMSVoIPConfigCallback . . . 1658
9.12.5.68 SLQSSetLocInjectPositionCallback . . . 1658
9.12.5.69 SLQSSetLocInjectUTCTimeCallback . . . 1658
9.12.5.70 SLQSSetModemTempCallback . . . 1658
9.12.5.71 SLQSSetPacketSrvStatusCallback . . . 1659
9.12.5.72 SLQSSetQosEventCallback . . . 1659
9.12.5.73 SLQSSetQosNWStatusCallback . . . 1660
9.12.5.74 SLQSSetQosPriEventCallback . . . 1660
9.12.5.75 SLQSSetQosStatusCallback . . . 1661
9.12.5.76 SLQSSetRegMgrConfigCallback . . . 1661
9.12.5.77 SLQSSetSDKTerminatedCallback . . . 1661
9.12.5.78 SLQSSetServingSystemCallback . . . 1662
9.12.5.79 SLQSSetSessionStateCallback . . . 1662
9.12.5.80 SLQSSetSignalStrengthsCallback . . . 1663
9.12.5.81 SLQSSetSIPConfigCallback . . . 1663
9.12.5.82 SLQSSetSMSEventCallback . . . 1664

9.12.5.83 SLQSSetSwiGetResetInfoCallback . . . 1664
9.12.5.84 SLQSSetSwiHDRPersCallback . . . 1664
9.12.5.85 SLQSSetSysSelectionPrefCallBack . . . 1665
9.12.5.86 SLQSSetTransLayerInfoCallback . . . 1665
9.12.5.87 SLQSSetTransNWRegInfoCallback . . . 1666
9.12.5.88 SLQSSetWdsEventCallback . . . 1666
9.12.5.89 SLQSSetWdsTransferStatisticCallback . . . 1667
9.12.5.90 SLQSSwiAVMSEventReportCallback . . . 1667
9.12.5.91 SLQSTmdMitigationLvlRptCallback . . . 1667
9.12.5.92 SLQSUIMSetRefreshCallBack . . . 1668
9.12.5.93 SLQSUIMSetStatusChangeCallBack . . . 1668
9.12.5.94 SLQSVoiceInfoRecCallback . . . 1668
9.12.5.95 SLQSVoiceSetAllCallStatusCallBack . . . 1669
9.12.5.96 SLQSVoiceSetDTMFEventCallBack . . . 1669
9.12.5.97 SLQSVoiceSetOTASPStatusCallBack . . . 1670
9.12.5.98 SLQSVoiceSetPrivacyChangeCallBack . . . 1670
9.12.5.99 SLQSVoiceSetSUPSCallBack . . . 1671
9.12.5.100 SLQSVoiceSetSUPSNotificationCallback . . . 1671
9.12.5.101 SLQSWmsAsyncRawSendCallBack . . . 1672
9.12.5.102 SLQSWmsMemoryFullCallBack . . . 1672
9.12.5.103 SLQSWmsMessageWaitingCallBack . . . 1672
9.13 qaGobiApiDcs.h File Reference . . . 1673
9.13.1 Detailed Description . . . 1673
9.13.2 Macro Definition Documentation . . . 1674
9.13.2.1 LEN . . . 1674
9.13.2.2 PORTNAM_LEN . . . 1674
9.13.3 Function Documentation . . . 1674
9.13.3.1 GetQmapIsSupported . . . 1674
9.13.3.2 QCWWAN2kConnect . . . 1674
9.13.3.3 QCWWAN2kEnumerateDevices . . . 1675
9.13.3.4 QCWWAN2kGetConnectedDeviceID . . . 1675
9.13.3.5 QCWWANConnect . . . 1676
9.13.3.6 QCWWANDisconnect . . . 1676
9.13.3.7 QCWWANEnumerateDevices . . . 1677
9.13.3.8 SetSDKImagePath . . . 1677
9.13.3.9 SLQSGetDeviceMode . . . 1677
9.13.3.10 SLQSGetNetStatistic . . . 1678
9.13.3.11 SLQSGetUsbPortNames . . . 1678
9.13.3.12 SLQSKillSDKProcess . . . 1679
9.13.3.13 SLQSSetLoggingMask . . . 1679

9.13.3.14 SLQSStart . . . 1679
9.13.3.15 SLQSStart_AVAgent . . . 1680
9.13.3.16 SLQSStartSrv . . . 1680
9.14 qaGobiApiDms.h File Reference . . . 1681
9.14.1 Detailed Description . . . 1683
9.14.2 Macro Definition Documentation . . . 1683
9.14.2.1 MAX_CUST_ID_LEN . . . 1683
9.14.2.2 MAX_CUST_VALUE_LEN . . . 1683
9.14.2.3 MAX_DYING_GASP_CFG_SMS_CONTENT_LENGTH . . . 1683
9.14.2.4 MAX_DYING_GASP_CFG_SMS_NUMBER_LENGTH . . . 1683
9.14.2.5 MAX_FSN_LENGTH . . . 1683
9.14.3 Typedef Documentation . . . 1683
9.14.3.1 custFeaturesInfo . . . 1683
9.14.3.2 custFeaturesSetting . . . 1685
9.14.3.3 dmsCurrentPRLInfo . . . 1687
9.14.3.4 ERIFileparams . . . 1687
9.14.3.5 serialNumbersInfo . . . 1687
9.14.3.6 SLQSSwiGetHostDevInfoParams . . . 1688
9.14.3.7 SLQSSwiGetOSInfoParams . . . 1688
9.14.3.8 SLQSSwiGetSerialNoExtParams . . . 1689
9.14.3.9 SLQSSwiSetHostDevInfoParams . . . 1689
9.14.3.10 SLQSSwiSetOSInfoParams . . . 1690
9.14.4 Function Documentation . . . 1690
9.14.4.1 ActivateAutomatic . . . 1690
9.14.4.2 GetActivationState . . . 1691
9.14.4.3 GetDeviceCapabilities . . . 1691
9.14.4.4 GetFirmwareRevision . . . 1692
9.14.4.5 GetFirmwareRevisions . . . 1693
9.14.4.6 GetHardwareRevision . . . 1693
9.14.4.7 GetManufacturer . . . 1694
9.14.4.8 GetModelID . . . 1694
9.14.4.9 GetNetworkTime . . . 1695
9.14.4.10 GetOfflineReason . . . 1695
9.14.4.11 GetPower . . . 1696
9.14.4.12 GetPRLVersion . . . 1697
9.14.4.13 GetSerialNumbers . . . 1697
9.14.4.14 GetVoiceNumber . . . 1698
9.14.4.15 ResetToFactoryDefaults . . . 1698
9.14.4.16 SetPower . . . 1699
9.14.4.17 SLQSDmsSwiGetResetInfo . . . 1699

9.14.4.18 SLQSDmsSwiIndicationRegister . . . 1699
9.14.4.19 SLQSGetBandCapabilities . . . 1700
9.14.4.20 SLQSGetBandCapability . . . 1700
9.14.4.21 SLQSGetCurrentPRLInfo . . . 1702
9.14.4.22 SLQSGetCustFeatures . . . 1702
9.14.4.23 SLQSGetCustFeaturesV2 . . . 1703
9.14.4.24 SLQSGetERIFile . . . 1703
9.14.4.25 SLQSGetSerialNumbers . . . 1703
9.14.4.26 SLQSSetCustFeatures . . . 1704
9.14.4.27 SLQSSetCustFeaturesV2 . . . 1704
9.14.4.28 SLQSSwiClearDyingGaspStatistics . . . 1705
9.14.4.29 SLQSSwiGetCrashAction . . . 1705
9.14.4.30 SLQSSwiGetCrashInfo . . . 1705
9.14.4.31 SLQSSwiGetDyingGaspCfg . . . 1706
9.14.4.32 SLQSSwiGetDyingGaspStatistics . . . 1706
9.14.4.33 SLQSSwiGetFSN . . . 1706
9.14.4.34 SLQSSwiGetFwUpdateStatus . . . 1707
9.14.4.35 SLQSSwiGetHostDevInfo . . . 1707
9.14.4.36 SLQSSwiGetOSInfo . . . 1708
9.14.4.37 SLQSSwiGetSerialNoExt . . . 1708
9.14.4.38 SLQSSwiGetUSBComp . . . 1708
9.14.4.39 SLQSSwiSetCrashAction . . . 1709
9.14.4.40 SLQSSwiSetDyingGaspCfg . . . 1709
9.14.4.41 SLQSSwiSetHostDevInfo . . . 1710
9.14.4.42 SLQSSwiSetOSInfo . . . 1710
9.14.4.43 SLQSSwiSetUSBComp . . . 1710
9.14.4.44 ValidateSPC . . . 1711
9.15 qaGobiApiFms.h File Reference . . . 1711
9.15.1 Detailed Description . . . 1714
9.15.2 Macro Definition Documentation . . . 1715
9.15.2.1 BUILD_ID_LEN . . . 1715
9.15.2.2 BUILD_ID_MAX_LEN . . . 1715
9.15.2.3 DEVICE_OFFLINE . . . 1715
9.15.2.4 DEVICE_RESET . . . 1715
9.15.2.5 DEVICE_SHUTDOWN . . . 1715
9.15.2.6 FIRMWARE_UPDATE_FAIL . . . 1715
9.15.2.7 FIRMWARE_UPDATE_SUCCESS . . . 1715
9.15.2.8 FIRMWARE_UPGRADE_SUCCESS . . . 1715
9.15.2.9 G3K_FIRMWARE_DOWNLOAD . . . 1715
9.15.2.10 GOBI_LISTENTRIES_MAX . . . 1715

9.15.2.11 GOBI_MBN_BUILD_ID_STR_LEN . . . 1715
9.15.2.12 GOBI_MBN_IMG_ID_STR_LEN . . . 1715
9.15.2.13 GOBI_SET_IMG_PREF_RSPLEN . . . 1715
9.15.2.14 IMG_ID_LEN . . . 1715
9.15.2.15 IMGDETAILS_LEN . . . 1715
9.15.2.16 MAX_IMAGE_IDE_ELEMENTS . . . 1715
9.15.2.17 PRI_UPDATE_FAIL . . . 1715
9.15.2.18 SLQSFWINFO_APPVERSION_SZ . . . 1715
9.15.2.19 SLQSFWINFO_BOOTVERSION_SZ . . . 1715
9.15.2.20 SLQSFWINFO_CARRIER_SZ . . . 1715
9.15.2.21 SLQSFWINFO_CUR_CARR_NAME . . . 1715
9.15.2.22 SLQSFWINFO_CUR_CARR_REV . . . 1715
9.15.2.23 SLQSFWINFO_MODELID_SZ . . . 1715
9.15.2.24 SLQSFWINFO_PACKAGEID_SZ . . . 1715
9.15.2.25 SLQSFWINFO_PRIVERSION_SZ . . . 1715
9.15.2.26 SLQSFWINFO_SKU_SZ . . . 1715
9.15.2.27 SPKG_FIRMWARE_DOWNLOAD . . . 1715
9.15.2.28 UNIQUE_ID_LEN . . . 1716
9.15.3 Enumeration Type Documentation . . . 1716
9.15.3.1 eGobiDeviceSeries . . . 1716
9.15.3.2 eGobiImageCarrier . . . 1716
9.15.3.3 eGobiImageGPS . . . 1717
9.15.3.4 eGobiImageRegion . . . 1717
9.15.3.5 eGobiImageTech . . . 1718
9.15.4 Function Documentation . . . 1718
9.15.4.1 DeleteStoredImage . . . 1718
9.15.4.2 DownloadToSlot . . . 1718
9.15.4.3 eGetDeviceSeries . . . 1719
9.15.4.4 GetImagesPreference . . . 1719
9.15.4.5 GetImageStore . . . 1720
9.15.4.6 GetStoredImages . . . 1720
9.15.4.7 SetImagesPreference . . . 1721
9.15.4.8 SLQSDownloadFirmwareToSlot . . . 1721
9.15.4.9 SLQSGetBootVersionNumber . . . 1722
9.15.4.10 SLQSGetFirmwareInfo . . . 1723
9.15.4.11 SLQSGetImageInfo . . . 1723
9.15.4.12 SLQSGetImageInfo_9x15 . . . 1724
9.15.4.13 SLQSGetImageInfoMC77xx . . . 1724
9.15.4.14 SLQSGetImageInfoMC83xx . . . 1725
9.15.4.15 SLQSGetValidFwPriCombinations . . . 1726

9.15.4.16 SLQSIsSpkgFormatRequired . . . 1726
9.15.4.17 SLQSSetCrashStateCheckIgnore . . . 1726
9.15.4.18 SLQSSetSIMBasedImageSwitching . . . 1727
9.15.4.19 SLQSSetSpkgFormatRequired . . . 1727
9.15.4.20 SLQSSwiGetAllCarrierImages . . . 1727
9.15.4.21 SLQSSwiGetFirmwareCurr . . . 1728
9.15.4.22 SLQSUpgradeFirmware9x07 . . . 1728
9.15.4.23 SLQSUpgradeFirmware9x15 . . . 1729
9.15.4.24 upgrade_mc77xx_fw . . . 1730
9.15.4.25 UpgradeFirmware2k . . . 1730
9.16 qaGobiApiIms.h File Reference . . . 1730
9.16.1 Detailed Description . . . 1731
9.16.2 Function Documentation . . . 1731
9.16.2.1 SLQSGetIMSSMSConfig . . . 1731
9.16.2.2 SLQSGetIMSUserConfig . . . 1732
9.16.2.3 SLQSGetIMSVoIPConfig . . . 1732
9.16.2.4 SLQSGetRegMgrConfig . . . 1733
9.16.2.5 SLQSGetSIPConfig . . . 1733
9.16.2.6 SLQSImsConfigIndicationRegister . . . 1734
9.16.2.7 SLQSSetIMSSMSConfig . . . 1734
9.16.2.8 SLQSSetIMSUserConfig . . . 1735
9.16.2.9 SLQSSetIMSVoIPConfig . . . 1735
9.16.2.10 SLQSSetRegMgrConfig . . . 1736
9.16.2.11 SLQSSetSIPConfig . . . 1736
9.17 qaGobiApiImsa.h File Reference . . . 1737
9.17.1 Detailed Description . . . 1737
9.17.2 Function Documentation . . . 1737
9.17.2.1 SLQSGetIMSARegStatus . . . 1737
9.17.2.2 SLQSGetIMSAServiceStatus . . . 1738
9.17.2.3 SLQSGetIMSASupportedFields . . . 1738
9.17.2.4 SLQSGetIMSASupportedMsg . . . 1739
9.17.2.5 SLQSRegisterIMSAIndication . . . 1739
9.18 qaGobiApiLoc.h File Reference . . . 1740
9.18.1 Detailed Description . . . 1741
9.18.2 Macro Definition Documentation . . . 1741
9.18.2.1 MAX_SENSOR_DATA_LEN . . . 1741
9.18.2.2 MAX_TEMP_DATA_LEN . . . 1741
9.18.3 Function Documentation . . . 1741
9.18.3.1 SLQSLOCDelAssData . . . 1741
9.18.3.2 SLQSLOCEventRegister . . . 1741

9.18.3.3 SLQSLOCGetBestAvailPos . . . 1742
9.18.3.4 SLQSLOCGetServer . . . 1742
9.18.3.5 SLQSLOCInjectPosition . . . 1743
9.18.3.6 SLQSLOCInjectSensorData . . . 1743
9.18.3.7 SLQSLOCInjectUTCTime . . . 1744
9.18.3.8 SLQSLOCSetCradleMountConfig . . . 1744
9.18.3.9 SLQSLOCSetExtPowerState . . . 1744
9.18.3.10 SLQSLOCSetOpMode . . . 1745
9.18.3.11 SLQSLOCStart . . . 1745
9.18.3.12 SLQSLOCStop . . . 1746
9.18.3.13 SwiLocGetAutoStart . . . 1746
9.18.3.14 SwiLocSetAutoStart . . . 1747
9.19 qaGobiApiNas.h File Reference . . . 1747
9.19.1 Detailed Description . . . 1752
9.19.2 Macro Definition Documentation . . . 1752
9.19.2.1 IMSI_M_S1_LENGTH . . . 1752
9.19.2.2 IMSI_M_S2_LENGTH . . . 1752
9.19.2.3 MAX_DATA_SRV_CAPABILITIES . . . 1752
9.19.2.4 MAX_DESCRIPTION_LENGTH . . . 1752
9.19.2.5 MAX_PILOT_SETS . . . 1752
9.19.2.6 MAX_SERV_SYSTEM_RADIO_INTERFACES . . . 1752
9.19.2.7 NAM_NAME_LENGTH . . . 1752
9.19.2.8 NAS_MAX_SCC_RX_INFO_INSTANCES . . . 1752
9.19.2.9 NAS_SIG_INFO_MAX_TDSCDMA_THRESHOLDS_LIST_SIZE . . . 1752
9.19.2.10 NAS_SIG_INFO_MIN_dB_FLOAT_VALUE . . . 1752
9.19.2.11 NAS_SIG_INFO_MIN_dBm_FLOAT_VALUE . . . 1752
9.19.2.12 PLMN_LENGTH . . . 1752
9.19.2.13 SLQS_SS_INFO_LIST_MAX_ELEMENTS . . . 1753
9.19.2.14 SLQS_SYSTEM_ID_SIZE . . . 1753
9.19.2.15 UATISIZE . . . 1753
9.19.3 Typedef Documentation . . . 1753
9.19.3.1 SlqsNas3GppNetworkRAT . . . 1753
9.19.3.2 slqsNetworkScanInfo . . . 1753
9.19.3.3 sysSelectPrefInfo . . . 1754
9.19.3.4 sysSelectPrefParams . . . 1758
9.19.4 Enumeration Type Documentation . . . 1764
9.19.4.1 _NAMS_RADIO_IF_TECHNOLOGY_ . . . 1764
9.19.4.2 eSYS_SRV_DOMAIN . . . 1764
9.19.4.3 NAS_LTE_CPHY_CA_BW_NRB . . . 1765
9.19.4.4 NAS_LTE_CPHY_SCELL_STATE . . . 1765

9.19.5 Function Documentation . . . 1765
9.19.5.1 GetACCOLC . . . 1765
9.19.5.2 GetANAAAAuthenticationStatus . . . 1765
9.19.5.3 GetCDMANetworkParameters . . . 1766
9.19.5.4 GetHomeNetwork . . . 1768
9.19.5.5 GetHomeNetwork3GPP2 . . . 1768
9.19.5.6 GetNetworkPreference . . . 1770
9.19.5.7 GetRFInfo . . . 1771
9.19.5.8 GetServingNetwork . . . 1771
9.19.5.9 GetServingNetworkCapabilities . . . 1772
9.19.5.10 GetSignalStrengths . . . 1773
9.19.5.11 InitiateDomainAttach . . . 1774
9.19.5.12 InitiateNetworkRegistration . . . 1774
9.19.5.13 PerformNetworkScan . . . 1775
9.19.5.14 SetACCOLC . . . 1775
9.19.5.15 SetCDMANetworkParameters . . . 1776
9.19.5.16 SetNetworkPreference . . . 1777
9.19.5.17 SLQSConfigSigInfo . . . 1778
9.19.5.18 SLQSGetErrorRate . . . 1779
9.19.5.19 SLQSGetNetworkTime . . . 1779
9.19.5.20 SLQSGetOperatorNameData . . . 1779
9.19.5.21 SLQSGetPLMNName . . . 1780
9.19.5.22 SLQSGetServingSystem . . . 1780
9.19.5.23 SLQSGetSignalStrength . . . 1781
9.19.5.24 SLQSGetSysSelectionPref . . . 1781
9.19.5.25 SLQSInitiateNetworkRegistration . . . 1782
9.19.5.26 SLQSNasConfigSigInfo2 . . . 1782
9.19.5.27 SLQSNasGet3GPP2Subscription . . . 1783
9.19.5.28 SLQSNasGetCellLocationInfo . . . 1783
9.19.5.29 SLQSNasGetHDRColorCode . . . 1784
9.19.5.30 SLQSNASGetLTECPHYCaInfo . . . 1784
9.19.5.31 SLQSNasGetSigInfo . . . 1784
9.19.5.32 SLQSNasGetSysInfo . . . 1785
9.19.5.33 SLQSNasGetTxRxInfo . . . 1785
9.19.5.34 SLQSNasIndicationRegister . . . 1786
9.19.5.35 SLQSNasIndicationRegisterExt . . . 1787
9.19.5.36 SLQSNasIndicationRegisterLTECphyCa . . . 1787
9.19.5.37 SLQSNASSwiGetChannelLock . . . 1787
9.19.5.38 SLQSNasSwiIndicationRegister . . . 1788
9.19.5.39 SLQSNasSwiModemStatus . . . 1788

9.19.5.40 SLQSNASSwiSetChannelLock . . . 1789
9.19.5.41 SLQSPerformNetworkScan . . . 1789
9.19.5.42 SLQSSetBandPreference . . . 1790
9.19.5.43 SLQSSetSysSelectionPref . . . 1791
9.19.5.44 SLQSSwiGetHDRPersonality . . . 1791
9.19.5.45 SLQSSwiGetHDRProtSubtype . . . 1792
9.19.5.46 SLQSSwiGetHRPDStats . . . 1792
9.19.5.47 SLQSSwiGetLteCQI . . . 1792
9.19.5.48 SLQSSwiGetLteSccRxInfo . . . 1793
9.19.5.49 SLQSSwiNetworkDebug . . . 1793
9.19.5.50 SLQSSwiPSDetach . . . 1794
9.20 qaGobiApiOmadm.h File Reference . . . 1794
9.20.1 Detailed Description . . . 1794
9.20.2 Function Documentation . . . 1794
9.20.2.1 OMADMCancelSession . . . 1794
9.20.2.2 OMADMGetPendingNIA . . . 1795
9.20.2.3 OMADMGetSessionInfo . . . 1795
9.20.2.4 OMADMStartSession . . . 1797
9.21 qaGobiApiPds.h File Reference . . . 1797
9.21.1 Detailed Description . . . 1798
9.21.2 Macro Definition Documentation . . . 1798
9.21.2.1 DEFAULTBYTEVALUE . . . 1798
9.21.2.2 DEFAULTLONGVALUE . . . 1798
9.21.2.3 DEFAULTWORDVALUE . . . 1798
9.21.3 Enumeration Type Documentation . . . 1798
9.21.3.1 anonymous enum . . . 1798
9.21.4 Function Documentation . . . 1799
9.21.4.1 ForceXTRADownload . . . 1799
9.21.4.2 GetPDSDefaults . . . 1799
9.21.4.3 GetPDSState . . . 1800
9.21.4.4 GetPortAutomaticTracking . . . 1800
9.21.4.5 GetServiceAutomaticTracking . . . 1801
9.21.4.6 GetXTRAAutomaticDownload . . . 1801
9.21.4.7 GetXTRANetwork . . . 1802
9.21.4.8 GetXTRAValidity . . . 1802
9.21.4.9 PDSInjectTimeReference . . . 1803
9.21.4.10 ResetPDSData . . . 1803
9.21.4.11 SetPDSDefaults . . . 1804
9.21.4.12 SetPDSState . . . 1805
9.21.4.13 SetPortAutomaticTracking . . . 1805

9.21.4.14 SetServiceAutomaticTracking . . . 1806
9.21.4.15 SetXTRAAutomaticDownload . . . 1806
9.21.4.16 SetXTRANetwork . . . 1807
9.21.4.17 SLQSGetAGPSConfig . . . 1807
9.21.4.18 SLQSGetGPSStateInfo . . . 1808
9.21.4.19 SLQSPDSDeterminePosition . . . 1808
9.21.4.20 SLQSPDSInjectAbsoluteTimeReference . . . 1809
9.21.4.21 SLQSPDSInjectPositionData . . . 1809
9.21.4.22 SLQSSetAGPSConfig . . . 1810
9.21.4.23 SLQSSetPositionMethodState . . . 1810
9.21.4.24 StartPDSTrackingSessionExt . . . 1811
9.21.4.25 StopPDSTrackingSession . . . 1812
9.22 qaGobiApiQos.h File Reference . . . 1812
9.22.1 Detailed Description . . . 1813
9.22.2 Macro Definition Documentation . . . 1813
9.22.2.1 MAX_QOS_FILTER_TLV . . . 1813
9.22.2.2 MAX_QOS_SPEC_PER_APN . . . 1813
9.22.3 Function Documentation . . . 1813
9.22.3.1 SLQSQosGetFlowStatus . . . 1813
9.22.3.2 SLQSQosGetGranted . . . 1813
9.22.3.3 SLQSQosGetNetworkStatus . . . 1814
9.22.3.4 SLQSQosGetNWProf . . . 1814
9.22.3.5 SLQSQosModify . . . 1815
9.22.3.6 SLQSQosRel . . . 1815
9.22.3.7 SLQSQosReq . . . 1816
9.22.3.8 SLQSQosReset . . . 1816
9.22.3.9 SLQSQosResume . . . 1817
9.22.3.10 SLQSQosSuspend . . . 1817
9.22.3.11 SLQSQosSwiReadApnExtraParams . . . 1818
9.22.3.12 SLQSQosSwiReadDataStats . . . 1818
9.23 qaGobiApiRms.h File Reference . . . 1818
9.23.1 Detailed Description . . . 1819
9.23.2 Function Documentation . . . 1819
9.23.2.1 GetSMSWake . . . 1819
9.23.2.2 SetSMSWake . . . 1819
9.24 qaGobiApiSar.h File Reference . . . 1820
9.24.1 Detailed Description . . . 1820
9.24.2 Enumeration Type Documentation . . . 1820
9.24.2.1 eQMISARRFState . . . 1820
9.24.3 Function Documentation . . . 1821

9.24.3.1 SLQSGetRfSarState . . . 1821
9.24.3.2 SLQSSetRfSarState . . . 1822
9.25 qaGobiApiSms.h File Reference . . . 1822
9.25.1 Detailed Description . . . 1824
9.25.2 Macro Definition Documentation . . . 1824
9.25.2.1 ABSOLUTE_VALIDITY . . . 1824
9.25.2.2 CONFIG_LEN . . . 1824
9.25.2.3 MAX_SMS_ROUTES . . . 1824
9.25.2.4 NUM_OF_SET . . . 1824
9.25.2.5 TIME_DATE_BUF . . . 1824
9.25.2.6 TIME_STAMP_BUF . . . 1824
9.25.3 Typedef Documentation . . . 1825
9.25.3.1 getIndicationRegResp . . . 1825
9.25.3.2 getTransLayerInfoResp . . . 1825
9.25.3.3 getTransNWRegInfoResp . . . 1826
9.25.3.4 qaQmi3GPP2BroadcastCfgInfo . . . 1826
9.25.3.5 qaQmi3GPPBroadcastCfgInfo . . . 1827
9.25.3.6 setIndicationRegReq . . . 1827
9.25.3.7 transLayerInfo . . . 1828
9.25.4 Function Documentation . . . 1828
9.25.4.1 GetSMSCAddress . . . 1828
9.25.4.2 SaveSMS . . . 1829
9.25.4.3 SendSMS . . . 1830
9.25.4.4 SetSMSCAddress . . . 1830
9.25.4.5 SLQSCDMADecodeMTTextMsg . . . 1831
9.25.4.6 SLQSCDMADecodeMTTextMsgExt . . . 1831
9.25.4.7 SLQSCDMAEncodeMOTextMsg . . . 1832
9.25.4.8 SLQSDeleteSMS . . . 1832
9.25.4.9 SLQSGetIndicationRegister . . . 1833
9.25.4.10 SLQSGetMessageWaiting . . . 1834
9.25.4.11 SLQSGetSMS . . . 1834
9.25.4.12 SLQSGetSmsBroadcastConfig . . . 1835
9.25.4.13 SLQSGetSMSList . . . 1836
9.25.4.14 SLQSGetTransLayerInfo . . . 1836
9.25.4.15 SLQSGetTransNWRegInfo . . . 1837
9.25.4.16 SLQSModifySMSStatus . . . 1837
9.25.4.17 SLQSSendAsyncSMS . . . 1838
9.25.4.18 SLQSSendLongSMS . . . 1839
9.25.4.19 SLQSSendSMS . . . 1839
9.25.4.20 SLQSSetIndicationRegister . . . 1840

9.25.4.21 SLQSSetSmsBroadcastActivation . . . 1840
9.25.4.22 SLQSSetSmsBroadcastConfig . . . 1841
9.25.4.23 SLQSSetSmsStorage . . . 1841
9.25.4.24 SLQSSmsGetMaxStorageSize . . . 1842
9.25.4.25 SLQSSmsGetMessageProtocol . . . 1842
9.25.4.26 SLQSSmsSetRoutes . . . 1843
9.25.4.27 SLQSSwiGetSMSStorage . . . 1843
9.25.4.28 SLQSWCDMADecodeLongTextMsg . . . 1844
9.25.4.29 SLQSWCDMADecodeMTTextMsg . . . 1844
9.25.4.30 SLQSWCDMAEncodeMOTextMsg . . . 1845
9.26 qaGobiApiSwi.h File Reference . . . 1845
9.26.1 Detailed Description . . . 1845
9.26.2 Function Documentation . . . 1845
9.26.2.1 SLQSGetPidof . . . 1845
9.26.2.2 SLQSGetSdkVersion . . . 1846
9.26.2.3 SLQSSendRawQMI . . . 1846
9.27 qaGobiApiSwiAudio.h File Reference . . . 1846
9.27.1 Detailed Description . . . 1847
9.27.2 Macro Definition Documentation . . . 1847
9.27.2.1 MAX_LEN_IFACE_TABLE . . . 1847
9.27.3 Function Documentation . . . 1847
9.27.3.1 SLQSGetM2MAudioProfile . . . 1847
9.27.3.2 SLQSGetM2MAudioVolume . . . 1848
9.27.3.3 SLQSGetM2MAVMute . . . 1848
9.27.3.4 SLQSGetM2MSpkrGain . . . 1849
9.27.3.5 SLQSSetM2MAudioAVCFG . . . 1849
9.27.3.6 SLQSSetM2MAudioLPBK . . . 1849
9.27.3.7 SLQSSetM2MAudioNVDef . . . 1850
9.27.3.8 SLQSSetM2MAudioProfile . . . 1850
9.27.3.9 SLQSSetM2MAudioVolume . . . 1850
9.27.3.10 SLQSSetM2MAVMute . . . 1851
9.27.3.11 SLQSSetM2MSpkrGain . . . 1851
9.28 qaGobiApiSwiAvms.h File Reference . . . 1852
9.28.1 Detailed Description . . . 1854
9.28.2 Macro Definition Documentation . . . 1854
9.28.2.1 MAX_AVMS_RETRY_TIMER_NUMBER . . . 1854
9.28.2.2 MAX_GET_SETTINGS_AVMS_APN_STRING_LENGTH . . . 1854
9.28.2.3 MAX_GET_SETTINGS_AVMS_PWD_STRING_LENGTH . . . 1854
9.28.2.4 MAX_GET_SETTINGS_AVMS_UNAME_STRING_LENGTH . . . 1854
9.28.2.5 MAX_SWI_AVMS_SET_SETTING_APN_LENGTH . . . 1854

9.28.2.6 MAX_SWI_AVMS_SET_SETTING_CONNECTION_RETRY_TIMMERS 1854
9.28.2.7 MAX_SWI_AVMS_SET_SETTING_PWD_LENGTH 1854
9.28.2.8 MAX_SWI_AVMS_SET_SETTING_UNAME_LENGTH 1854
9.28.2.9 QMISWIAVMSSESSIONGETINFO_BINARYUPDATESESSIONINFO_DESC_LENGTH 1854
9.28.2.10 QMISWIAVMSSESSIONGETINFO_BINARYUPDATESESSIONINFO_NAME_LENGTH 1854
9.28.2.11 QMISWIAVMSSESSIONGETINFO_BINARYUPDATESESSIONINFO_VERSION_LENGTH 1854
9.28.2.12 QMISWIAVMSSESSIONGETINFO_CONFIG_ALERT_MSG_LENGTH 1854
9.28.3 Typedef Documentation 1854
9.28.3.1 QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp 1854
9.28.3.2 QmiSwiAvmsEventReportConfigResp 1855
9.28.3.3 QmiSwiAvmsEventReportConnectionRequestResp 1855
9.28.3.4 QmiSwiAvmsEventReportDataSessionStatusResp 1856
9.28.3.5 QmiSwiAvmsEventReportHTTPStatusResp 1856
9.28.3.6 QmiSwiAvmsEventReportNotificationResp 1857
9.28.3.7 QmiSwiAvmsEventReportPackageIDResp 1858
9.28.3.8 QmiSwiAvmsEventReportRegStatusResp 1858
9.28.3.9 QmiSwiAvmsEventReportSessionTypeResp 1859
9.28.3.10 QmiSwiAvmsEventReportWAMSParamChangeResp 1859
9.28.3.11 QmiSwiAvmsSetSettingsAPNInfo 1859
9.28.3.12 QmiSwiAvmsSetSettingsConnectionRetryTimers 1859
9.28.3.13 QmiSwiAvmsSetSettingsPeriodInfo 1859
9.28.3.14 SLQSAVMSSetSettingsReq 1859
9.28.3.15 SLQSAVMSSettings 1860
9.28.3.16 SLQSAVMSSettingsAPNInfo 1860
9.28.3.17 SLQSAVMSSettingsConnectionRetryTimers 1860
9.28.3.18 SLQSAVMSSettingsPeriodsInfo 1861
9.28.4 Function Documentation 1861
9.28.4.1 SLQSAVMSGetSessionInfo 1861
9.28.4.2 SLQSAVMSGetSettings 1861
9.28.4.3 SLQSAVMSSelection 1861
9.28.4.4 SLQSAVMSSetEventReport 1862
9.28.4.5 SLQSAVMSSetSettings 1862
9.28.4.6 SLQSAVMSSetSettingsIgnoreAutoReboot 1863
9.28.4.7 SLQSAVMSStartSession 1863
9.28.4.8 SLQSAVMSStopSession 1864
9.29 qaGobiApiSwiOmadms.h File Reference 1864
9.29.1 Detailed Description 1865
9.29.2 Typedef Documentation 1865

9.29.2.1 SLQSOMADMSessionInfo . . . 1865
9.29.2.2 SLQSOMADMSettings . . . 1867
9.29.2.3 SLQSOMADMSettingsReqParams . . . 1868
9.29.2.4 SLQSOMADMSettingsReqParams3 . . . 1869
9.29.3 Function Documentation . . . 1870
9.29.3.1 SLQSOMADMCancelSession . . . 1870
9.29.3.2 SLQSOMADMGetSessionInfo . . . 1870
9.29.3.3 SLQSOMADMGetSettings . . . 1871
9.29.3.4 SLQSOMADMGetSettings2 . . . 1871
9.29.3.5 SLQSOMADMSendSelection . . . 1872
9.29.3.6 SLQSOMADMSendSelection2 . . . 1872
9.29.3.7 SLQSOMADMSetSettings . . . 1873
9.29.3.8 SLQSOMADMSetSettings2 . . . 1874
9.29.3.9 SLQSOMADMSetSettings3 . . . 1874
9.29.3.10 SLQSOMADMStartSession . . . 1874
9.29.3.11 SLQSOMADMStartSession2 . . . 1875
9.30 qaGobiApiTableBandClasses.h File Reference . . . 1875
9.30.1 Detailed Description . . . 1875
9.30.2 Band Classes (Value - Description) . . . 1875
9.30.2.1 LTE Bands . . . 1877
9.31 qaGobiApiTableCallControlReturnReasons.h File Reference . . . 1878
9.31.1 Detailed Description . . . 1878
9.31.2 S1 . . . 1878
9.32 qaGobiApiTableCallEndReasons.h File Reference . . . 1879
9.32.1 Detailed Description . . . 1879
9.32.2 Call end reason codes (Code - Reason) . . . 1879
9.32.2.1 Technology-agnostic call end reasons . . . 1879
9.32.2.2 EVDO CDMA 1xEV-DO . . . 1880
9.32.2.3 WCDMA/GSM call end reasons . . . 1880
9.32.2.4 EVDO CDMA 1xEV-DO . . . 1882
9.32.2.5 call end reason type . . . 1883
9.32.2.6 Mobile IP call end reasons (Type=1) . . . 1883
9.32.2.7 Internal call end reasons (Type=2) . . . 1885
9.32.2.8 Call Manager defined call end reasons (Type=3) . . . 1887
9.32.2.9 3GPP specification defined call end reasons (Type=6) . . . 1892
9.32.2.10 PPP call end reasons (Type=7) . . . 1894
9.32.2.11 EHRPD call end reasons (Type=8) . . . 1894
9.32.2.12 IPV6 call end reasons (Type=9) . . . 1895
9.33 qaGobiApiTableCarrierCodes.h File Reference . . . 1895
9.33.1 Detailed Description . . . 1895

9.33.2 Carrier Codes (Number - Carrier) . . . 1896
9.34 qaGobiApiTableCodingScheme.h File Reference . . . 1897
9.34.1 Detailed Description . . . 1897
9.34.2 S1 . . . 1898
9.34.2.1 Use of bits 3..0\n\n . . . 1898
9.34.3 Coding Group Bits 7..4(0001) . . . 1898
9.34.3.1 use of bits 3..0 . . . 1898
9.34.4 Coding Group Bits 7..4(0010) . . . 1898
9.34.4.1 use of bits 3..0 . . . 1898
9.34.5 Coding Group Bits 7..4(0011) . . . 1899
9.34.5.1 use of bits 3..0 . . . 1899
9.34.6 Coding Group Bits 7..4(01xx) . . . 1899
9.34.6.1 use of bits 3..0 . . . 1899
9.34.7 Coding Group Bits 7..4(1001) . . . 1899
9.34.7.1 Reserved coding groups . . . 1899
9.34.8 Coding Group Bits 7..4(1010..1101) . . . 1900
9.34.8.1 Reserved coding groups . . . 1900
9.34.9 Coding Group Bits 7..4(1110) . . . 1900
9.34.9.1 Defined by the WAP Forum . . . 1900
9.34.10 Coding Group Bits 7..4 (1111) . . . 1900
9.34.10.1 Data coding / message handling . . . 1900
9.34.11 Macro Definition Documentation . . . 1900
9.34.11.1 __GOBI_API_CODING_SCHEME_H__ . . . 1900
9.35 qaGobiApiTableGpsCapabilityCodes.h File Reference . . . 1900
9.35.1 Detailed Description . . . 1900
9.35.2 GPS capability (Value - Capability) . . . 1900
9.36 qaGobiApiTablePowerModes.h File Reference . . . 1901
9.36.1 Detailed Description . . . 1901
9.36.2 Power Modes (Value - Description) . . . 1901
9.37 qaGobiApiTableRadioInterfaces.h File Reference . . . 1901
9.37.1 Detailed Description . . . 1901
9.37.2 Radio interface . . . 1901
9.37.2.1 Technology (Value - Radio Interface Technology) . . . 1901
9.38 qaGobiApiTableRegionCodes.h File Reference . . . 1902
9.38.1 Detailed Description . . . 1902
9.38.2 Region Codes (Code - Region) . . . 1902
9.39 qaGobiApiTableServiceOptions.h File Reference . . . 1902
9.39.1 Detailed Description . . . 1902
9.39.2 Service Option codes (Code - Reason) . . . 1903
9.39.2.1 Description . . . 1903

9.40 qaGobiApiTableSupServiceInfoClasses.h File Reference . . . 1905
9.40.1 Detailed Description . . . 1905
9.40.2 Supplementary Service Information Classes (Value - Service Class) . . . 1905
9.41 qaGobiApiTableSwiAudio.h File Reference . . . 1905
9.41.1 Detailed Description . . . 1905
9.41.2 ACDB Device (Device ID - description) . . . 1905
9.41.3 Physical Interface (Device ID - description - Interface parameters) . . . 1906
9.42 qaGobiApiTableSwiOMADMSessionStatus.h File Reference . . . 1906
9.42.1 Detailed Description . . . 1906
9.42.2 OMA DM Session Status (Session Status - Meaning - Usage) . . . 1906
9.43 qaGobiApiTableSwiOMADMUpdateCompleteStatus.h File Reference . . . 1907
9.43.1 Detailed Description . . . 1907
9.43.2 OMA DM Update Complete Status (Update Complete Status - Meaning - Usage) . . . 1907
9.44 qaGobiApiTableVoiceCallEndReasons.h File Reference . . . 1908
9.44.1 Detailed Description . . . 1908
9.44.2 Voice Call and supplementary services end reason codes (Code - Reason) . . . 1908
9.44.2.1 General . . . 1908
9.44.2.2 service Errors . . . 1910
9.44.2.3 control cause values . . . 1911
9.44.2.4 reject causes . . . 1913
9.44.2.5 reject causes . . . 1913
9.44.2.6 reject causes . . . 1913
9.44.2.7 stratum reject causes . . . 1914
9.44.2.8 reject causes . . . 1914
9.44.2.9 IP end reasons . . . 1914
9.45 qaGobiApiTmd.h File Reference . . . 1915
9.45.1 Detailed Description . . . 1915
9.45.2 Macro Definition Documentation . . . 1915
9.45.2.1 MAX_MITIGATION_DEV_ID_LEN . . . 1915
9.45.2.2 MAX_MITIGATION_DEV_LIST_LEN . . . 1915
9.45.3 Function Documentation . . . 1915
9.45.3.1 SLQSTmdDeRegNotMitigationLvl . . . 1915
9.45.3.2 SLQSTmdGetMitigationDevList . . . 1916
9.45.3.3 SLQSTmdGetMitigationLvl . . . 1916
9.45.3.4 SLQSTmdRegNotMitigationLvl . . . 1916
9.46 qaGobiApiUim.h File Reference . . . 1917
9.46.1 Detailed Description . . . 1919
9.46.2 Macro Definition Documentation . . . 1919
9.46.2.1 MAX_ACTIVE_PERS_FEATURES . . . 1919
9.46.2.2 MAX_CONTENT_LENGTH . . . 1919

9.46.2.3 MAX_DESCRIPTION_LENGTH . . . 1919
9.46.2.4 MAX_ICCID_LENGTH . . . 1919
9.46.2.5 MAX_NO_OF_APPLICATIONS . . . 1919
9.46.2.6 MAX_NO_OF_SLOTS . . . 1919
9.46.2.7 MAX_PATH_LENGTH . . . 1919
9.46.2.8 MAX_PUK_LENGTH . . . 1919
9.46.2.9 MAX_SLOTS_STATUS . . . 1919
9.46.3 Function Documentation . . . 1919
9.46.3.1 SLQSUIMAuthenticate . . . 1919
9.46.3.2 SLQSUIMChangePin . . . 1920
9.46.3.3 SLQSUIMDepersonalization . . . 1920
9.46.3.4 SLQSUIMEventRegister . . . 1921
9.46.3.5 SLQSUIMGetCardStatus . . . 1921
9.46.3.6 SLQSUIMGetConfiguration . . . 1922
9.46.3.7 SLQSUIMGetFileAttributes . . . 1922
9.46.3.8 SLQSUIMGetSlotsStatus . . . 1923
9.46.3.9 SLQSUIMPowerDown . . . 1923
9.46.3.10 SLQSUIMPowerUp . . . 1924
9.46.3.11 SLQSUIMReadTransparent . . . 1924
9.46.3.12 SLQSUIMRefreshComplete . . . 1925
9.46.3.13 SLQSUIMRefreshGetLastEvent . . . 1925
9.46.3.14 SLQSUIMRefreshOK . . . 1926
9.46.3.15 SLQSUIMRefreshRegister . . . 1926
9.46.3.16 SLQSUIMReset . . . 1927
9.46.3.17 SLQSUIMSetPinProtection . . . 1927
9.46.3.18 SLQSUIMSwitchSlot . . . 1928
9.46.3.19 SLQSUIMUnblockPin . . . 1928
9.46.3.20 SLQSUIMVerifyPin . . . 1929
9.47 qaGobiApiVoice.h File Reference . . . 1929
9.47.1 Detailed Description . . . 1932
9.47.2 Macro Definition Documentation . . . 1932
9.47.2.1 MAX_CALL_NO_LEN . . . 1932
9.47.2.2 MAX_DESCRIPTION_LENGTH . . . 1932
9.47.2.3 MAX_NO_OF_CALLS . . . 1932
9.47.2.4 MAXUSSDLENGTH . . . 1933
9.47.2.5 PASSWORD_LENGTH . . . 1933
9.47.3 Enumeration Type Documentation . . . 1933
9.47.3.1 serviceClassInformation . . . 1933
9.47.4 Function Documentation . . . 1933
9.47.4.1 AnswerUSSD . . . 1933

9.47.4.2 CancelUSSD . . . 1933
9.47.4.3 OriginateUSSD . . . 1934
9.47.4.4 SLQSOriginateUSSD . . . 1934
9.47.4.5 SLQSVoiceALSSelectLine . . . 1935
9.47.4.6 SLQSVoiceALSSetLineSwitching . . . 1935
9.47.4.7 SLQSVoiceAnswerCall . . . 1936
9.47.4.8 SLQSVoiceBindSubscription . . . 1936
9.47.4.9 SLQSVoiceBurstDTMF . . . 1937
9.47.4.10 SLQSVoiceDialCall . . . 1937
9.47.4.11 SLQSVoiceEndCall . . . 1938
9.47.4.12 SLQSVoiceGetAllCallInfo . . . 1938
9.47.4.13 SLQSVoiceGetCallBarring . . . 1939
9.47.4.14 SLQSVoiceGetCallForwardingStatus . . . 1939
9.47.4.15 SLQSVoiceGetCallInfo . . . 1940
9.47.4.16 SLQSVoiceGetCallWaiting . . . 1941
9.47.4.17 SLQSVoiceGetCLIP . . . 1941
9.47.4.18 SLQSVoiceGetCLIR . . . 1942
9.47.4.19 SLQSVoiceGetCNAP . . . 1942
9.47.4.20 SLQSVoiceGetCOLP . . . 1943
9.47.4.21 SLQSVoiceGetCOLR . . . 1943
9.47.4.22 SLQSVoiceGetConfig . . . 1944
9.47.4.23 SLQSVoiceIndicationRegister . . . 1944
9.47.4.24 SLQSVoiceManageCalls . . . 1945
9.47.4.25 SLQSVoiceOrigUSSDNoWait . . . 1945
9.47.4.26 SLQSVoiceSendFlash . . . 1946
9.47.4.27 SLQSVoiceSetCallBarringPassword . . . 1946
9.47.4.28 SLQSVoiceSetConfig . . . 1947
9.47.4.29 SLQSVoiceSetPreferredPrivacy . . . 1948
9.47.4.30 SLQSVoiceSetSUPSService . . . 1948
9.47.4.31 SLQSVoiceStartContDTMF . . . 1949
9.47.4.32 SLQSVoiceStopContDTMF . . . 1949
9.48 qaGobiApiWds.h File Reference . . . 1950
9.48.1 Detailed Description . . . 1953
9.48.2 Macro Definition Documentation . . . 1953
9.48.2.1 IPV6_ADDRESS_ARRAY_SIZE . . . 1953
9.48.3 Typedef Documentation . . . 1953
9.48.3.1 GetProfileSettingIn . . . 1953
9.48.3.2 GetProfileSettingOut . . . 1954
9.48.3.3 QmiProfileInfo . . . 1954
9.48.3.4 QmiWDSDataBearers . . . 1954

9.48.3.5 QmiWDSDataBearerTechnology . . . 1955
9.48.3.6 slqs3GPPConfigItem . . . 1956
9.48.4 Enumeration Type Documentation . . . 1957
9.48.4.1 qmiDataBearerMasks . . . 1957
9.48.5 Function Documentation . . . 1957
9.48.5.1 GetAutoconnect . . . 1957
9.48.5.2 GetByteTotals . . . 1957
9.48.5.3 GetConnectionRate . . . 1958
9.48.5.4 GetDataBearerTechnology . . . 1959
9.48.5.5 GetDefaultProfile . . . 1960
9.48.5.6 GetDefaultProfileLTE . . . 1961
9.48.5.7 GetDefaultProfileNum . . . 1963
9.48.5.8 GetDormancyState . . . 1963
9.48.5.9 GetIPAddressLTE . . . 1964
9.48.5.10 GetLastMobileIPError . . . 1964
9.48.5.11 GetMobileIP . . . 1965
9.48.5.12 GetMobileIPProfile . . . 1965
9.48.5.13 GetPacketStatistics . . . 1967
9.48.5.14 GetPacketStatus . . . 1967
9.48.5.15 GetSessionDuration . . . 1968
9.48.5.16 GetSessionState . . . 1969
9.48.5.17 iGetByteTotals . . . 1969
9.48.5.18 iGetConnectionRate . . . 1969
9.48.5.19 iGetPacketStatistics . . . 1969
9.48.5.20 iSLQSMISetIPFamilyPreference . . . 1969
9.48.5.21 RMSetTransferStatistics . . . 1969
9.48.5.22 SetActiveMobileIPProfile . . . 1970
9.48.5.23 SetAutoconnect . . . 1970
9.48.5.24 SetDefaultProfile . . . 1971
9.48.5.25 SetDefaultProfileLTE . . . 1972
9.48.5.26 SetDefaultProfileLTEV2 . . . 1973
9.48.5.27 SetDefaultProfileNum . . . 1975
9.48.5.28 SetMobileIP . . . 1975
9.48.5.29 SetMobileIPParameters . . . 1976
9.48.5.30 SetMobileIPProfile . . . 1977
9.48.5.31 SLQSAutoConnect . . . 1978
9.48.5.32 SLQSCreateProfile . . . 1979
9.48.5.33 SLQSDeleteProfile . . . 1979
9.48.5.34 SLQSGet3GPPConfigItem . . . 1980
9.48.5.35 SLQSGetByteTotals . . . 1980

9.48.5.36 SLQSGetConnectionRate . . . 1980
9.48.5.37 SLQSGetCurrDataSystemStat . . . 1981
9.48.5.38 SLQSGetCurrentChannelRate . . . 1981
9.48.5.39 SLQSGetDataBearerTechnology . . . 1982
9.48.5.40 SLQSGetDataBearerTechnologyExt . . . 1982
9.48.5.41 SLQSGetDUNCallInfo . . . 1983
9.48.5.42 SLQSGetPacketStatistics . . . 1983
9.48.5.43 SLQSGetProfile . . . 1984
9.48.5.44 SLQSGetProfileSettings . . . 1985
9.48.5.45 SLQSGetRuntimeSettings . . . 1986
9.48.5.46 SLQSGetSessionState . . . 1986
9.48.5.47 SLQSModifyProfile . . . 1987
9.48.5.48 SLQSResetPacketStatics . . . 1987
9.48.5.49 SLQSSet3GPPConfigItem . . . 1988
9.48.5.50 SLQSSetProfile . . . 1988
9.48.5.51 SLQSSGetDHCPv4ClientConfig . . . 1989
9.48.5.52 SLQSSGetLoopback . . . 1990
9.48.5.53 SLQSSSetDHCPv4ClientConfig . . . 1990
9.48.5.54 SLQSSSetLoopback . . . 1990
9.48.5.55 SLQSStartStopDataSession . . . 1991
9.48.5.56 SLQSWdsGoActive . . . 1991
9.48.5.57 SLQSWdsGoDormant . . . 1992
9.48.5.58 SLQSWdsSetEventReport . . . 1992
9.48.5.59 SLQSWdsSwiPDPRuntimeSettings . . . 1993
9.48.5.60 WDS_IsGobiDevice . . . 1993
9.49 qaNasGetRFBandInfo.h File Reference . . . 1993
9.49.1 Enumeration Type Documentation . . . 1993
9.49.1.1 eQMI_NAS_GET_RF_INFO_RESP . . . 1993
9.49.2 Function Documentation . . . 1994
9.49.2.1 PkQmiNasGetRFBandInfo . . . 1994
9.49.2.2 UpkQmiNasGetRFBandInfo . . . 1994
9.50 qaNasPerformNetworkScan.h File Reference . . . 1994
9.50.1 Macro Definition Documentation . . . 1994
9.50.1.1 FORBIDDEN_INDEX . . . 1994
9.50.1.2 INDEX_ZERO . . . 1994
9.50.1.3 MAX_DESCRIPTION_LENGTH . . . 1994
9.50.1.4 PREFERRED_INDEX . . . 1994
9.50.1.5 QMI_NAS_MAX_INSTANCES . . . 1994
9.50.1.6 QMI_NAS_NETSTATUS_MASK . . . 1994
9.50.1.7 ROAMING_INDEX . . . 1994

9.50.2 Enumeration Type Documentation . . . 1994
9.50.2.1 eQMI_NAS_PERFORM_NETWORK_SCAN_RESP . . . 1995
9.50.3 Function Documentation . . . 1995
9.50.3.1 PkQmiNasPerformNetworkScan . . . 1995
9.50.3.2 UpkQmiNasPerformNetworkScan . . . 1995
9.51 qmerrno.h File Reference . . . 1995
9.51.1 Enumeration Type Documentation . . . 1997
9.51.1.1 eQCWWANError . . . 1997
9.51.1.2 qm_wds_ds_profile_extended_err_codes . . . 2002
9.52 qos.h File Reference . . . 2002
9.52.1 Macro Definition Documentation . . . 2004
9.52.1.1 LITEQMI_MAX_QOS_FILTERS . . . 2004
9.52.1.2 LITEQMI_MAX_QOS_FLOW_PER_APN_STATS . . . 2004
9.52.1.3 LITEQMI_MAX_QOS_FLOWS . . . 2004
9.52.2 Function Documentation . . . 2004
9.52.2.1 pack_qos_SLQSQosGetNetworkStatus . . . 2004
9.52.2.2 pack_qos_SLQSQosSwiReadApnExtraParams . . . 2004
9.52.2.3 pack_qos_SLQSQosSwiReadDataStats . . . 2005
9.52.2.4 pack_qos_SLQSSetQosEventCallback . . . 2006
9.52.2.5 unpack_qos_SLQSQosGetNetworkStatus . . . 2006
9.52.2.6 unpack_qos_SLQSQosSwiReadApnExtraParams . . . 2007
9.52.2.7 unpack_qos_SLQSQosSwiReadDataStats . . . 2007
9.52.2.8 unpack_qos_SLQSSetQosEventCallback . . . 2008
9.52.2.9 unpack_qos_SLQSSetQosEventCallback_ind . . . 2008
9.52.2.10 unpack_qos_SLQSSetQosNWStatusCallback_ind . . . 2009
9.52.2.11 unpack_qos_SLQSSetQosPriEventCallback_ind . . . 2009
9.52.2.12 unpack_qos_SLQSSetQosStatusCallback_ind . . . 2010
9.53 rms.h File Reference . . . 2010
9.53.1 Function Documentation . . . 2011
9.53.1.1 pack_rms_GetSMSWake . . . 2011
9.53.1.2 pack_rms_SetSMSWake . . . 2011
9.53.1.3 unpack_rms_GetSMSWake . . . 2011
9.53.1.4 unpack_rms_SetSMSWake . . . 2012
9.54 sar.h File Reference . . . 2012
9.54.1 Function Documentation . . . 2012
9.54.1.1 pack_sar_SLQSGetRfSarState . . . 2012
9.54.1.2 pack_sar_SLQSSetRfSarState . . . 2013
9.54.1.3 unpack_sar_SLQSGetRfSarState . . . 2013
9.54.1.4 unpack_sar_SLQSSetRfSarState . . . 2014
9.55 sms.h File Reference . . . 2014

9.55.1 Macro Definition Documentation . . . 2017
9.55.1.1 MAX_CDMA_ENC_MO_TXT_MSG_SIZE . . . 2017
9.55.1.2 MAX_MS_TRANSFER_ROUTE_MSG . . . 2017
9.55.1.3 MAX_MSC_ADDRESS_SIZE . . . 2017
9.55.1.4 MAX_MSE_TWS_MSG . . . 2017
9.55.1.5 MAX_SMS_LIST_SIZE . . . 2017
9.55.1.6 MAX_SMS_MESSAGE_SIZE . . . 2017
9.55.1.7 SMS_CONFIG_LEN . . . 2017
9.55.1.8 SMS_MAX_SMS_ROUTES . . . 2017
9.55.1.9 SMS_NUM_OF_SET . . . 2017
9.55.1.10 SMSC_TYPE_LEN . . . 2017
9.55.2 Enumeration Type Documentation . . . 2017
9.55.2.1 eqmiCbkSetStatus . . . 2017
9.55.3 Function Documentation . . . 2018
9.55.3.1 pack_sms_GetSMSCAddress . . . 2018
9.55.3.2 pack_sms_SaveSMS . . . 2018
9.55.3.3 pack_sms_SendSMS . . . 2018
9.55.3.4 pack_sms_SetNewSMSCallback . . . 2019
9.55.3.5 pack_sms_SetSMSCAddress . . . 2019
9.55.3.6 pack_sms_SLQSDeleteSMS . . . 2019
9.55.3.7 pack_sms_SLQSGetIndicationRegister . . . 2020
9.55.3.8 pack_sms_SLQSGetMessageWaiting . . . 2020
9.55.3.9 pack_sms_SLQSGetSMS . . . 2020
9.55.3.10 pack_sms_SLQSGetSmsBroadcastConfig . . . 2021
9.55.3.11 pack_sms_SLQSGetSMSList . . . 2021
9.55.3.12 pack_sms_SLQSGetTransLayerInfo . . . 2022
9.55.3.13 pack_sms_SLQSGetTransNWRegInfo . . . 2022
9.55.3.14 pack_sms_SLQSModifySMSStatus . . . 2022
9.55.3.15 pack_sms_SLQSSendAsyncSMS . . . 2023
9.55.3.16 pack_sms_SLQSSetIndicationRegister . . . 2023
9.55.3.17 pack_sms_SLQSSetSmsBroadcastActivation . . . 2023
9.55.3.18 pack_sms_SLQSSetSmsBroadcastConfig . . . 2024
9.55.3.19 pack_sms_SLQSSetSmsStorage . . . 2024
9.55.3.20 pack_sms_SLQSSmsGetMaxStorageSize . . . 2024
9.55.3.21 pack_sms_SLQSSmsGetMessageProtocol . . . 2025
9.55.3.22 pack_sms_SLQSSmsSetRoutes . . . 2025
9.55.3.23 pack_sms_SLQSSwiGetSMSStorage . . . 2026
9.55.3.24 unpack_sms_GetSMSCAddress . . . 2026
9.55.3.25 unpack_sms_SaveSMS . . . 2026
9.55.3.26 unpack_sms_SendSMS . . . 2027

9.55.3.27 unpack_sms_SetNewSMSCallback . . . 2027
9.55.3.28 unpack_sms_SetNewSMSCallback_ind . . . 2027
9.55.3.29 unpack_sms_SetSMSCAddress . . . 2028
9.55.3.30 unpack_sms_SLQSDeleteSMS . . . 2028
9.55.3.31 unpack_sms_SLQSGetIndicationRegister . . . 2028
9.55.3.32 unpack_sms_SLQSGetMessageWaiting . . . 2029
9.55.3.33 unpack_sms_SLQSGetSMS . . . 2029
9.55.3.34 unpack_sms_SLQSGetSmsBroadcastConfig . . . 2029
9.55.3.35 unpack_sms_SLQSGetSMSList . . . 2030
9.55.3.36 unpack_sms_SLQSGetTransLayerInfo . . . 2030
9.55.3.37 unpack_sms_SLQSGetTransNWRegInfo . . . 2030
9.55.3.38 unpack_sms_SLQSModifySMSStatus . . . 2031
9.55.3.39 unpack_sms_SLQSSendAsyncSMS . . . 2031
9.55.3.40 unpack_sms_SLQSSetIndicationRegister . . . 2031
9.55.3.41 unpack_sms_SLQSSetSmsBroadcastActivation . . . 2032
9.55.3.42 unpack_sms_SLQSSetSmsBroadcastConfig . . . 2032
9.55.3.43 unpack_sms_SLQSSetSmsStorage . . . 2032
9.55.3.44 unpack_sms_SLQSSmsGetMaxStorageSize . . . 2033
9.55.3.45 unpack_sms_SLQSSmsGetMessageProtocol . . . 2033
9.55.3.46 unpack_sms_SLQSSmsSetRoutes . . . 2033
9.55.3.47 unpack_sms_SLQSSwiGetSMSStorage . . . 2034
9.55.3.48 unpack_sms_SLQSWmsMemoryFullCallBack_ind . . . 2034
9.56 SwiDataTypes.h File Reference . . . 2034
9.56.1 Detailed Description . . . 2035
9.56.2 Macro Definition Documentation . . . 2035
9.56.2.1 QMI_NO_LTE_FW_SUPPORT . . . 2035
9.56.2.2 QMI_TLV_PLACEHOLDER . . . 2035
9.56.2.3 SWI_API . . . 2035
9.56.2.4 UNUSEDPARAM . . . 2035
9.56.3 Typedef Documentation . . . 2035
9.56.3.1 BOOL . . . 2035
9.56.3.2 BYTE . . . 2035
9.56.3.3 CHAR . . . 2035
9.56.3.4 FLOAT . . . 2035
9.56.3.5 INT32 . . . 2035
9.56.3.6 INT8 . . . 2035
9.56.3.7 LPCSTR . . . 2035
9.56.3.8 SHORT . . . 2035
9.56.3.9 ULONG . . . 2035
9.56.3.10 ULONGLONG . . . 2035

9.56.3.11 USHORT . . . 2036
9.56.3.12 WORD . . . 2036
9.57 swiloc.h File Reference . . . 2036
9.57.1 Function Documentation . . . 2036
9.57.1.1 pack_swiloc_SwiLocGetAutoStart . . . 2036
9.57.1.2 pack_swiloc_SwiLocSetAutoStart . . . 2036
9.57.1.3 unpack_swiloc_SwiLocGetAutoStart . . . 2037
9.57.1.4 unpack_swiloc_SwiLocSetAutoStart . . . 2037
9.58 swioma.h File Reference . . . 2037
9.58.1 Macro Definition Documentation . . . 2038
9.58.1.1 LITEQMI_MAX_SWIOMA_STR_LEN . . . 2038
9.58.2 Function Documentation . . . 2038
9.58.2.1 pack_swioma_SLQSOMADMAlertCallback . . . 2038
9.58.2.2 pack_swioma_SLQSOMADMCancelSession . . . 2039
9.58.2.3 pack_swioma_SLQSOMADMGetSessionInfo . . . 2040
9.58.2.4 pack_swioma_SLQSOMADMGetSettings . . . 2040
9.58.2.5 pack_swioma_SLQSOMADMSendSelection . . . 2041
9.58.2.6 pack_swioma_SLQSOMADMSetSettings . . . 2042
9.58.2.7 pack_swioma_SLQSOMADMStartSession . . . 2042
9.58.2.8 unpack_swioma_SLQSOMADMAlertCallback . . . 2043
9.58.2.9 unpack_swioma_SLQSOMADMAlertCallback_ind . . . 2043
9.58.2.10 unpack_swioma_SLQSOMADMCancelSession . . . 2044
9.58.2.11 unpack_swioma_SLQSOMADMGetSessionInfo . . . 2044
9.58.2.12 unpack_swioma_SLQSOMADMGetSettings . . . 2045
9.58.2.13 unpack_swioma_SLQSOMADMSendSelection . . . 2045
9.58.2.14 unpack_swioma_SLQSOMADMSetSettings . . . 2045
9.58.2.15 unpack_swioma_SLQSOMADMStartSession . . . 2046
9.59 SWIWWANCMAPI.h File Reference . . . 2046
9.60 uim.h File Reference . . . 2046
9.60.1 Macro Definition Documentation . . . 2049
9.60.1.1 MAX_DESCRIPTION_LENGTH . . . 2049
9.60.1.2 MAX_ICCID_LENGTH . . . 2049
9.60.1.3 MAX_NO_OF_APPLICATIONS . . . 2049
9.60.1.4 MAX_NO_OF_SLOTS . . . 2049
9.60.1.5 MAX_SLOTS_STATUS . . . 2049
9.60.1.6 UIM_MAX_ACTIVE_PERS_FEATURES . . . 2049
9.60.1.7 UIM_MAX_CONTENT_LENGTH . . . 2049
9.60.1.8 UIM_MAX_DESCRIPTION_LENGTH . . . 2049
9.60.1.9 UIM_MAX_NO_OF_APPLICATIONS . . . 2049
9.60.1.10 UIM_MAX_NO_OF_SLOTS . . . 2049

9.60.1.11 UIM_UINT8_MAX_STRING_SZ . . . 2049
9.60.2 Function Documentation . . . 2050
9.60.2.1 pack_uim_ChangePin . . . 2050
9.60.2.2 pack_uim_GetCardStatus . . . 2050
9.60.2.3 pack_uim_ReadTransparent . . . 2050
9.60.2.4 pack_uim_SetPinProtection . . . 2051
9.60.2.5 pack_uim_SLQSUIMAuthenticate . . . 2051
9.60.2.6 pack_uim_SLQSUIMDepersonalization . . . 2051
9.60.2.7 pack_uim_SLQSUIMEventRegister . . . 2052
9.60.2.8 pack_uim_SLQSUIMGetConfiguration . . . 2052
9.60.2.9 pack_uim_SLQSUIMGetFileAttributes . . . 2053
9.60.2.10 pack_uim_SLQSUIMGetSlotsStatus . . . 2053
9.60.2.11 pack_uim_SLQSUIMPowerDown . . . 2053
9.60.2.12 pack_uim_SLQSUIMPowerUp . . . 2054
9.60.2.13 pack_uim_SLQSUIMRefreshComplete . . . 2054
9.60.2.14 pack_uim_SLQSUIMRefreshGetLastEvent . . . 2054
9.60.2.15 pack_uim_SLQSUIMRefreshOK . . . 2055
9.60.2.16 pack_uim_SLQSUIMRefreshRegister . . . 2055
9.60.2.17 pack_uim_SLQSUIMReset . . . 2055
9.60.2.18 pack_uim_SLQSUIMSwitchSlot . . . 2056
9.60.2.19 pack_uim_UnblockPin . . . 2056
9.60.2.20 pack_uim_VerifyPin . . . 2057
9.60.2.21 unpack_uim_ChangePin . . . 2057
9.60.2.22 unpack_uim_GetCardStatus . . . 2057
9.60.2.23 unpack_uim_ReadTransparent . . . 2058
9.60.2.24 unpack_uim_SetPinProtection . . . 2058
9.60.2.25 unpack_uim_SetUimSlotStatusChangeCallback_ind . . . 2058
9.60.2.26 unpack_uim_SLQSUIMAuthenticate . . . 2059
9.60.2.27 unpack_uim_SLQSUIMDepersonalization . . . 2059
9.60.2.28 unpack_uim_SLQSUIMEventRegister . . . 2059
9.60.2.29 unpack_uim_SLQSUIMGetConfiguration . . . 2060
9.60.2.30 unpack_uim_SLQSUIMGetFileAttributes . . . 2060
9.60.2.31 unpack_uim_SLQSUIMGetSlotsStatus . . . 2061
9.60.2.32 unpack_uim_SLQSUIMPowerDown . . . 2061
9.60.2.33 unpack_uim_SLQSUIMPowerUp . . . 2061
9.60.2.34 unpack_uim_SLQSUIMRefreshComplete . . . 2062
9.60.2.35 unpack_uim_SLQSUIMRefreshGetLastEvent . . . 2062
9.60.2.36 unpack_uim_SLQSUIMRefreshOK . . . 2062
9.60.2.37 unpack_uim_SLQSUIMRefreshRegister . . . 2063
9.60.2.38 unpack_uim_SLQSUIMReset . . . 2063

9.60.2.39 unpack_uim_SLQSUIMSetStatusChangeCallBack_ind . . . 2063
9.60.2.40 unpack_uim_SLQSUIMSwitchSlot . . . 2064
9.60.2.41 unpack_uim_UnblockPin . . . 2064
9.60.2.42 unpack_uim_VerifyPin . . . 2064
9.61 voice.h File Reference . . . 2065
9.61.1 Detailed Description . . . 2069
9.61.2 S1 . . . 2069
9.61.3 Macro Definition Documentation . . . 2069
9.61.3.1 BARRING_PASSWORD_LENGTH . . . 2069
9.61.3.2 MAX_VOICE_CALL_NO_LEN . . . 2069
9.61.3.3 MAX_VOICE_DESCRIPTION_LENGTH . . . 2069
9.61.3.4 MAXVOICEUSSDLENGTH . . . 2069
9.61.3.5 VOICE_MAX_NO_OF_CALLS . . . 2069
9.61.4 Enumeration Type Documentation . . . 2069
9.61.4.1 liteServiceClassInformation . . . 2069
9.61.5 Function Documentation . . . 2070
9.61.5.1 pack_voice_AnswerUSSD . . . 2070
9.61.5.2 pack_voice_CancelUSSD . . . 2070
9.61.5.3 pack_voice_OriginateUSSD . . . 2070
9.61.5.4 pack_voice_SLQSOriginateUSSD . . . 2071
9.61.5.5 pack_voice_SLQSVoiceALSSelectLine . . . 2071
9.61.5.6 pack_voice_SLQSVoiceALSSetLineSwitching . . . 2072
9.61.5.7 pack_voice_SLQSVoiceAnswerCall . . . 2072
9.61.5.8 pack_voice_SLQSVoiceBindSubscription . . . 2072
9.61.5.9 pack_voice_SLQSVoiceBurstDTMF . . . 2073
9.61.5.10 pack_voice_SLQSVoiceDialCall . . . 2073
9.61.5.11 pack_voice_SLQSVoiceEndCall . . . 2073
9.61.5.12 pack_voice_SLQSVoiceGetAllCallInfo . . . 2074
9.61.5.13 pack_voice_SLQSVoiceGetCallBarring . . . 2074
9.61.5.14 pack_voice_SLQSVoiceGetCallForwardingStatus . . . 2075
9.61.5.15 pack_voice_SLQSVoiceGetCallInfo . . . 2075
9.61.5.16 pack_voice_SLQSVoiceGetCallWaiting . . . 2075
9.61.5.17 pack_voice_SLQSVoiceGetCLIP . . . 2076
9.61.5.18 pack_voice_SLQSVoiceGetCLIR . . . 2076
9.61.5.19 pack_voice_SLQSVoiceGetCNAP . . . 2076
9.61.5.20 pack_voice_SLQSVoiceGetCOLP . . . 2077
9.61.5.21 pack_voice_SLQSVoiceGetCOLR . . . 2077
9.61.5.22 pack_voice_SLQSVoiceGetConfig . . . 2077
9.61.5.23 pack_voice_SLQSVoiceIndicationRegister . . . 2078
9.61.5.24 pack_voice_SLQSVoiceManageCalls . . . 2078

9.61.5.25 pack_voice_SLQSVoiceOrigUSSDNoWait . . . 2078
9.61.5.26 pack_voice_SLQSVoiceSendFlash . . . 2079
9.61.5.27 pack_voice_SLQSVoiceSetCallBarringPassword . . . 2079
9.61.5.28 pack_voice_SLQSVoiceSetConfig . . . 2080
9.61.5.29 pack_voice_SLQSVoiceSetPreferredPrivacy . . . 2080
9.61.5.30 pack_voice_SLQSVoiceSetSUPSService . . . 2080
9.61.5.31 pack_voice_SLQSVoiceStartContDTMF . . . 2081
9.61.5.32 pack_voice_SLQSVoiceStopContDTMF . . . 2081
9.61.5.33 unpack_voice_AnswerUSSD . . . 2081
9.61.5.34 unpack_voice_CancelUSSD . . . 2082
9.61.5.35 unpack_voice_OriginateUSSD . . . 2082
9.61.5.36 unpack_voice_SLQSOriginateUSSD . . . 2082
9.61.5.37 unpack_voice_SLQSVoiceALSSelectLine . . . 2083
9.61.5.38 unpack_voice_SLQSVoiceALSSetLineSwitching . . . 2083
9.61.5.39 unpack_voice_SLQSVoiceAnswerCall . . . 2083
9.61.5.40 unpack_voice_SLQSVoiceBindSubscription . . . 2084
9.61.5.41 unpack_voice_SLQSVoiceBurstDTMF . . . 2084
9.61.5.42 unpack_voice_SLQSVoiceDialCall . . . 2084
9.61.5.43 unpack_voice_SLQSVoiceEndCall . . . 2085
9.61.5.44 unpack_voice_SLQSVoiceGetAllCallInfo . . . 2085
9.61.5.45 unpack_voice_SLQSVoiceGetCallBarring . . . 2085
9.61.5.46 unpack_voice_SLQSVoiceGetCallForwardingStatus . . . 2086
9.61.5.47 unpack_voice_SLQSVoiceGetCallInfo . . . 2086
9.61.5.48 unpack_voice_SLQSVoiceGetCallWaiting . . . 2086
9.61.5.49 unpack_voice_SLQSVoiceGetCLIP . . . 2087
9.61.5.50 unpack_voice_SLQSVoiceGetCLIR . . . 2087
9.61.5.51 unpack_voice_SLQSVoiceGetCNAP . . . 2087
9.61.5.52 unpack_voice_SLQSVoiceGetCOLP . . . 2088
9.61.5.53 unpack_voice_SLQSVoiceGetCOLR . . . 2088
9.61.5.54 unpack_voice_SLQSVoiceGetConfig . . . 2088
9.61.5.55 unpack_voice_SLQSVoiceIndicationRegister . . . 2089
9.61.5.56 unpack_voice_SLQSVoiceManageCalls . . . 2089
9.61.5.57 unpack_voice_SLQSVoiceOrigUSSDNoWait . . . 2089
9.61.5.58 unpack_voice_SLQSVoiceSendFlash . . . 2090
9.61.5.59 unpack_voice_SLQSVoiceSetCallBarringPassword . . . 2090
9.61.5.60 unpack_voice_SLQSVoiceSetConfig . . . 2090
9.61.5.61 unpack_voice_SLQSVoiceSetPreferredPrivacy . . . 2091
9.61.5.62 unpack_voice_SLQSVoiceSetSUPSService . . . 2091
9.61.5.63 unpack_voice_SLQSVoiceStartContDTMF . . . 2091
9.61.5.64 unpack_voice_SLQSVoiceStopContDTMF . . . 2092

9.62 wds.h File Reference . . . 2092
9.62.1 Macro Definition Documentation . . . 2098
9.62.1.1 BYT_STAT_STAT_MASK . . . 2098
9.62.1.2 IPV6_ADDRESS_ARRAY_SIZE . . . 2098
9.62.1.3 MAX_WDS_3GPP_CONF_LTE_ATTACH_PROFILE_LIST_SIZE . . . 2098
9.62.1.4 PACK_WDS_IPV4 . . . 2098
9.62.1.5 PACK_WDS_IPV6 . . . 2098
9.62.1.6 WDS_DHCP_MAX_NUM_OPTIONS . . . 2098
9.62.1.7 WDS_DHCP_OPTION_DATA_BUF_SIZE . . . 2098
9.62.2 Typedef Documentation . . . 2098
9.62.2.1 UnpackQmiProfileInfo . . . 2098
9.62.3 Enumeration Type Documentation . . . 2098
9.62.3.1 liteQmiDataBearerMasks . . . 2098
9.62.4 Function Documentation . . . 2099
9.62.4.1 pack_wds_DHCPv4ClientLeaseChange . . . 2099
9.62.4.2 pack_wds_GetAutoconnect . . . 2099
9.62.4.3 pack_wds_GetByteTotals . . . 2099
9.62.4.4 pack_wds_GetConnectionRate . . . 2100
9.62.4.5 pack_wds_GetDataBearerTechnology . . . 2100
9.62.4.6 pack_wds_GetDefaultProfile . . . 2101
9.62.4.7 pack_wds_GetDefaultProfileNum . . . 2101
9.62.4.8 pack_wds_GetDormancyState . . . 2101
9.62.4.9 pack_wds_GetLastMobileIPError . . . 2102
9.62.4.10 pack_wds_GetMobileIP . . . 2102
9.62.4.11 pack_wds_GetMobileIPProfile . . . 2103
9.62.4.12 pack_wds_GetPacketStatistics . . . 2103
9.62.4.13 pack_wds_GetPacketStatus . . . 2104
9.62.4.14 pack_wds_GetSessionDuration . . . 2104
9.62.4.15 pack_wds_GetSessionState . . . 2104
9.62.4.16 pack_wds_RMSetTransferStatistics . . . 2105
9.62.4.17 pack_wds_SetAutoconnect . . . 2105
9.62.4.18 pack_wds_SetDefaultProfile . . . 2106
9.62.4.19 pack_wds_SetDefaultProfileNum . . . 2106
9.62.4.20 pack_wds_SetMobileIP . . . 2106
9.62.4.21 pack_wds_SetMobileIPParameters . . . 2107
9.62.4.22 pack_wds_SetMobileIPProfile . . . 2107
9.62.4.23 pack_wds_SetMuxID . . . 2108
9.62.4.24 pack_wds_SLQSCreateProfile . . . 2108
9.62.4.25 pack_wds_SLQSDeleteProfile . . . 2109
9.62.4.26 pack_wds_SLQSGet3GPPConfigItem . . . 2109

9.62.4.27 pack_wds_SLQSGetCurrDataSystemStat . . . 2109
9.62.4.28 pack_wds_SLQSGetCurrentChannelRate . . . 2110
9.62.4.29 pack_wds_SLQSGetDataBearerTechnology . . . 2110
9.62.4.30 pack_wds_SLQSGetDUNCallInfo . . . 2111
9.62.4.31 pack_wds_SLQSGetProfileSettings . . . 2111
9.62.4.32 pack_wds_SLQSGetRuntimeSettings . . . 2112
9.62.4.33 pack_wds_SLQSModifyProfile . . . 2112
9.62.4.34 pack_wds_SLQSResetPacketStatics . . . 2112
9.62.4.35 pack_wds_SLQSSet3GPPConfigItem . . . 2113
9.62.4.36 pack_wds_SLQSSetIPFamilyPreference . . . 2113
9.62.4.37 pack_wds_SLQSSetWdsEventCallback . . . 2114
9.62.4.38 pack_wds_SLQSSGetDHCPv4ClientConfig . . . 2114
9.62.4.39 pack_wds_SLQSSGetLoopback . . . 2114
9.62.4.40 pack_wds_SLQSSSetDHCPv4ClientConfig . . . 2115
9.62.4.41 pack_wds_SLQSSSetLoopback . . . 2115
9.62.4.42 pack_wds_SLQSStartDataSession . . . 2116
9.62.4.43 pack_wds_SLQSStopDataSession . . . 2116
9.62.4.44 pack_wds_SLQSWdsGoActive . . . 2116
9.62.4.45 pack_wds_SLQSWdsGoDormant . . . 2117
9.62.4.46 pack_wds_SLQSWdsSetEventReport . . . 2117
9.62.4.47 pack_wds_SLQSWdsSwiPDPRuntimeSettings . . . 2118
9.62.4.48 unpack_wds_DHCPv4ClientLease_ind . . . 2118
9.62.4.49 unpack_wds_DHCPv4ClientLeaseChange . . . 2118
9.62.4.50 unpack_wds_GetAutoconnect . . . 2119
9.62.4.51 unpack_wds_GetByteTotals . . . 2119
9.62.4.52 unpack_wds_GetConnectionRate . . . 2119
9.62.4.53 unpack_wds_GetDataBearerTechnology . . . 2120
9.62.4.54 unpack_wds_GetDefaultProfile . . . 2120
9.62.4.55 unpack_wds_GetDefaultProfileNum . . . 2120
9.62.4.56 unpack_wds_GetDormancyState . . . 2121
9.62.4.57 unpack_wds_GetLastMobileIPError . . . 2121
9.62.4.58 unpack_wds_GetMobileIP . . . 2121
9.62.4.59 unpack_wds_GetMobileIPProfile . . . 2122
9.62.4.60 unpack_wds_GetPacketStatistics . . . 2122
9.62.4.61 unpack_wds_GetPacketStatus . . . 2122
9.62.4.62 unpack_wds_GetSessionDuration . . . 2123
9.62.4.63 unpack_wds_GetSessionState . . . 2123
9.62.4.64 unpack_wds_RMSetTransferStatistics . . . 2123
9.62.4.65 unpack_wds_RMTransferStatistics_ind . . . 2124
9.62.4.66 unpack_wds_SetAutoconnect . . . 2124

9.62.4.67 unpack_wds_SetDefaultProfile . . . 2124
9.62.4.68 unpack_wds_SetDefaultProfileNum . . . 2125
9.62.4.69 unpack_wds_SetMobileIP . . . 2125
9.62.4.70 unpack_wds_SetMobileIPParameters . . . 2125
9.62.4.71 unpack_wds_SetMobileIPProfile . . . 2126
9.62.4.72 unpack_wds_SetMuxID . . . 2126
9.62.4.73 unpack_wds_SLQSCreateProfile . . . 2126
9.62.4.74 unpack_wds_SLQSDeleteProfile . . . 2127
9.62.4.75 unpack_wds_SLQSGet3GPPConfigItem . . . 2127
9.62.4.76 unpack_wds_SLQSGetCurrDataSystemStat . . . 2127
9.62.4.77 unpack_wds_SLQSGetCurrentChannelRate . . . 2128
9.62.4.78 unpack_wds_SLQSGetDataBearerTechnology . . . 2128
9.62.4.79 unpack_wds_SLQSGetDUNCallInfo . . . 2128
9.62.4.80 unpack_wds_SLQSGetProfileSettings . . . 2129
9.62.4.81 unpack_wds_SLQSGetRuntimeSettings . . . 2129
9.62.4.82 unpack_wds_SLQSModifyProfile . . . 2129
9.62.4.83 unpack_wds_SLQSResetPacketStatics . . . 2130
9.62.4.84 unpack_wds_SLQSSet3GPPConfigItem . . . 2130
9.62.4.85 unpack_wds_SLQSSetIPFamilyPreference . . . 2130
9.62.4.86 unpack_wds_SLQSSetPacketSrvStatusCallback . . . 2131
9.62.4.87 unpack_wds_SLQSSetWdsEventCallback . . . 2131
9.62.4.88 unpack_wds_SLQSSetWdsEventCallback_ind . . . 2131
9.62.4.89 unpack_wds_SLQSSGetDHCPv4ClientConfig . . . 2132
9.62.4.90 unpack_wds_SLQSSGetLoopback . . . 2132
9.62.4.91 unpack_wds_SLQSSSetDHCPv4ClientConfig . . . 2132
9.62.4.92 unpack_wds_SLQSSSetLoopback . . . 2133
9.62.4.93 unpack_wds_SLQSStartDataSession . . . 2133
9.62.4.94 unpack_wds_SLQSStopDataSession . . . 2133
9.62.4.95 unpack_wds_SLQSWdsGoActive . . . 2134
9.62.4.96 unpack_wds_SLQSWdsGoDormant . . . 2134
9.62.4.97 unpack_wds_SLQSWdsSetEventReport . . . 2134
9.62.4.98 unpack_wds_SLQSWdsSwiPDPRuntimeSettings . . . 2134

Index **2136**

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
cdmaMsgDecodingParamsExt . . . 167
cdmaMsgEncodingParams . . . 170
CDMARSSIThresh . . . 172
CDMASSInfo . . . 172
cdmaSSInfo . . . 173
CDMASysInfo . . . 174
CDMASysInfoExt . . . 177
CellDb . . . 178
cellParams . . . 178
changeUIMPIN . . . 179
channelRate . . . 180
ChannelRate . . . 181
CLIPResp . . . 182
CLIRResp . . . 182
ClkInfo . . . 183
CNAPResp . . . 184
COLPResp . . . 185
COLRResp . . . 186
CommInfo . . . 186
ConnectionStatus . . . 188
connectionStatus . . . 189
connectNumInfo . . . 190
CrashInfo . . . 191
crashInfoParams . . . 192
CrashInfoParams . . . 193
crashInformation . . . 193
CreateProfileIn . . . 194
CreateProfileOut . . . 195
CsgId . . . 196
CSGID . . . 196
CUGInfo . . . 197
curAMRConfig . . . 198
CurrDataSysStat . . . 199
currentCatEvent . . . 200
CurrentImgList . . . 200
currentPLMN . . . 201
CurrImageInfo . . . 202
CurrNetworkInfo . . . 203
currNetworkInfo . . . 205
custFeaturesInfo . . . 206
custFeaturesSetting . . . 209
custSettingInfo . . . 210
custSettingList . . . 211
dataBearers . . . 212
DataBearerTech . . . 212

DataBearerTechExt . . . 214
dataBearerTechnology . . . 215
dataRate . . . 216
dataSrvCapabilities . . . 217
DataStatusDetail . . . 218
DataULongLongTlv . . . 220
DataULongTlv . . . 220
DcsUsbPortNames . . . 220
delAssistDataStatus . . . 220
depersonalizationInformation . . . 221
detailSvcInfo . . . 222
DeviceConfigDetail . . . 224
DHCPOption . . . 225
DHCPOptionList . . . 225
diagInfo . . . 226
dirNum . . . 226
dms_ActivationStatusTlv . . . 227
dms_OperatingModeTlv . . . 228
dmsCurrentPRLInfo . . . 228
DMScustSettingInfo . . . 229
DMScustSettingList . . . 230
DMSgetCustomFeatureV2 . . . 230
DMSgetCustomInput . . . 231
dmsIndicationRegisterReq . . . 232
dmsSwiGetResetInfo . . . 232
Domain . . . 233
DomainNameList . . . 233
DRCParams . . . 234
DTMFInfo . . . 234
DTMFLengths . . . 235
DUNCallInfoInd . . . 236
dunchannelRate . . . 237
ecioListElement . . . 237
ECIOThresh . . . 238
ECTNum . . . 239
encryptedPIN1 . . . 239
eriDataparams . . . 240
ERIFileparams . . . 240
errorRateListElement . . . 241
eTWSPLMNInfoTlv . . . 242
extDispRecInfo . . . 242
FactorySequenceNumber . . . 243
fileAttributes . . . 243
fileInfo . . . 247
FirmwareUpdatStat . . . 247
FMSImageElement . . . 249
FMSImageIdElement . . . 250
FMSImageIDEntries . . . 251
FMSImageList . . . 251
FMSPrefImageList . . . 252
fwinfo_s . . . 252
GERANInfo . . . 253
geranInstInfo . . . 255
getAllCallInformation . . . 256
getAllCallRmtPtyName . . . 256
getAllCallRmtPtyNum . . . 257
GetAudioPathConfigReq . . . 257
GetAudioPathConfigResp . . . 258

GetAudioProfileReq . . . 260
GetAudioProfileResp . . . 260
GetAudioVolTLBConfigReq . . . 262
GetAudioVolTLBConfigResp . . . 262
getCallFWExtInfo . . . 263
getCallFWInfo . . . 263
getCustomFeatureV2 . . . 264
getCustomInput . . . 265
getDUNCallInfoReq . . . 265
getDUNCallInfoResp . . . 267
getDyingGaspCfg . . . 269
getDyingGaspStatistics . . . 270
GetErrRateResp . . . 271
GetHRPDStatsResp . . . 272
GetIMSSMSConfigParams . . . 272
GetIMSUserConfigParams . . . 273
GetIMSVoIPConfigResp . . . 274
GetInstIDResp . . . 276
GetM2MAudioProfileReq . . . 277
GetM2MAudioProfileResp . . . 277
GetM2MAudioVolumeReq . . . 278
GetM2MAudioVolumeResp . . . 279
GetM2MAVMuteReq . . . 279
GetM2MAVMuteResp . . . 280
GetM2MSpkrGainReq . . . 280
GetM2MSpkrGainResp . . . 281
getMsgWaitingInfo . . . 281
GetNetworkTimeResp . . . 282
GetRegMgrConfigParams . . . 282
GetSessionIDResp . . . 283
GetSIPConfigResp . . . 283
GnssData . . . 285
gnssSvInfoNotification . . . 286
GPRSQoS . . . 287
GPRSRequestedQoS . . . 288
GPSStateInfo . . . 289
gpsTime_s . . . 293
gsmCellInfo . . . 293
GSMRSSIThresh . . . 295
GSMSrvStatusInfo . . . 295
GSMSysInfo . . . 296
gyroAcceptReady_s . . . 299
gyroTempAcceptReady_s . . . 300
HDRECIOThresh . . . 300
HDRIOThresh . . . 301
HDRPersonalityInd . . . 301
HDRPersonalityResp . . . 302
HDRProtSubtypResp . . . 302
HDRRSSIThresh . . . 303
HDRSINRThresh . . . 304
HDRSINRThreshold . . . 304
HDRSSInfo . . . 305
hdrSSInfo . . . 306
HDRSysInfo . . . 307
homeSIDNID . . . 309
hotSwapStatus . . . 310
image_info_t . . . 310
ImageElement . . . 311

ImageIdElement . . . 312
ImageIDEntries . . . 313
ImageList . . . 314
IMSAIndRegisterInfo . . . 314
imsaPdpStatusInfo . . . 315
imsaRatStatusInfo . . . 316
IMSARegistrationStatus . . . 317
imsaRegStatusInfo . . . 317
IMSAServiceStatus . . . 318
IMSASupportedFieldsResp . . . 320
IMSASupportedMsgInfo . . . 321
imsaSvcStatusInfo . . . 321
imsCfgIndRegisterInfo . . . 322
imsRegMgrConfigInfo . . . 323
imsSIPConfigInfo . . . 324
imsSMSConfigInfo . . . 325
imsUserConfigInfo . . . 326
imsVoIPConfigInfo . . . 326
IndFieldsList . . . 329
infoInterFreq . . . 329
IOThresh . . . 330
IPv4Addr . . . 331
IPv4Info . . . 331
IPv6Addr . . . 332
IPV6AddressInfo . . . 333
ipv6AddressInfo . . . 333
IPV6GWAddressInfo . . . 334
IPv6Info . . . 334
IPv6TrafCls . . . 335
LibPackGPRSRequestedQoS . . . 335
LibpackProfile3GPP . . . 336
LibpackProfile3GPP2 . . . 343
LibPackprofile_3GPP . . . 348
LibPackprofile_3GPP2 . . . 354
LibPackQosClassID . . . 359
LibPackTFTIDParams . . . 360
LibPackUMTSQoS . . . 362
LibPackUMTSReqQoSSigInd . . . 365
lineCtrlInfo . . . 365
loc_BdsSV . . . 366
loc_BdsSVInfo . . . 367
loc_CellDb . . . 367
loc_ClkInfo . . . 368
loc_GnssData . . . 369
loc_gpsTime . . . 371
loc_IPv4Info . . . 371
loc_IPv6Info . . . 372
loc_LocApplicationInfo . . . 372
loc_precisionDilution . . . 373
loc_satelliteInfo . . . 374
loc_sensorDataUsage . . . 376
loc_SV . . . 377
loc_SVInfo . . . 378
loc_svUsedforFix . . . 378
loc_urlAddr . . . 379
LocApplicationInfo . . . 380
LocDelAssDataReq . . . 381
LOCEventRegisterReqResp . . . 382

LOCExtPowerStateReqResp . . . 384
LocGetServerReq . . . 384
LocInjectPositionReq . . . 385
LocInjectSensorDataReq . . . 389
LocSetCradleMountReq . . . 390
LOCStartReq . . . 391
LOCStopReq . . . 393
LteCQIParm . . . 393
lteEARFCN . . . 394
lteGsmCellInfo . . . 395
LTEInfo . . . 396
LTEInfoInterfreq . . . 398
LTEInfoIntrafreq . . . 398
LTEInfoNeighboringGSM . . . 401
LTEInfoNeighboringWCDMA . . . 401
LteNasReleaseInfo_s . . . 402
ltePCI . . . 403
lteRsrpinformation . . . 403
LTERSRPThresh . . . 404
LTERSRQThresh . . . 404
LTERSSIThresh . . . 405
LteSccRxInfoResp . . . 405
LTESigRptCfg . . . 406
LTESigRptConfig . . . 406
lteSnrinformation . . . 407
LTESNRThresh . . . 408
LTESNRThreshold . . . 408
LTESSInfo . . . 409
lteSSInfo . . . 410
LTESysInfo . . . 411
lteWcdmaCellInfo . . . 413
messageModeTlv . . . 414
messageWaitingInfoContent . . . 414
minBasedIMSI . . . 415
mitigationDevList . . . 416
MNRInfo . . . 417
ModifyProfileIn . . . 417
ModifyProfileOut . . . 418
msgWaitingInfo . . . 418
namName . . . 419
nas_acqOrderPref . . . 420
nas_ActPilotPNElement . . . 420
nas_AddCDMASysInfo . . . 421
nas_AddSysInfo . . . 421
nas_CallBarringSysInfo . . . 422
nas_callBarStatus . . . 423
nas_CDMAChannel . . . 423
nas_CDMAECIOThresh . . . 424
nas_CDMAInfo . . . 425
nas_CDMARSSIThresh . . . 426
nas_CDMASysInfo . . . 426
nas_CDMASysInfoExt . . . 430
nas_cellParams . . . 430
nas_CommInfo . . . 431
nas_CSGID . . . 433
nas_currentPLMN . . . 434
nas_dataSrvCapabilities . . . 435
nas_DataStatusDetail . . . 436

nas_detailSvcInfo . . . 438
nas_DeviceConfigDetail . . . 439
nas_dirNum . . . 440
nas_DRCParams . . . 440
nas_ecioListElement . . . 441
nas_ECIOThresh . . . 442
nas_errorRateListElement . . . 442
nas_GERANInfo . . . 443
nas_geranInstInfo . . . 445
nas_gsmCellInfo . . . 446
nas_GSMRSSIThresh . . . 447
nas_GSMSrvStatusInfo . . . 447
nas_GSMSysInfo . . . 448
nas_HDRECIOThresh . . . 451
nas_HDRIOThresh . . . 451
nas_HDRRSSIThresh . . . 452
nas_HDRSINRThresh . . . 453
nas_HDRSINRThreshold . . . 453
nas_HDRSysInfo . . . 454
nas_homeSIDNID . . . 456
nas_infoInterFreq . . . 456
nas_IOThresh . . . 458
nas_lteEARFCN . . . 458
nas_lteGsmCellInfo . . . 459
nas_LTEInfo . . . 460
nas_LTEInfoInterfreq . . . 463
nas_LTEInfoIntrafreq . . . 463
nas_LTEInfoNeighboringGSM . . . 465
nas_LTEInfoNeighboringWCDMA . . . 466
nas_ltePCI . . . 467
nas_lteRsrpinformation . . . 467
nas_LTERSRPThresh . . . 468
nas_LTERSRQThresh . . . 468
nas_LTERSSIThresh . . . 469
nas_LTESigRptCfg . . . 469
nas_LTESigRptConfig . . . 470
nas_lteSnrinformation . . . 471
nas_LTESNRThresh . . . 472
nas_LTESNRThreshold . . . 472
nas_LTESysInfo . . . 473
nas_lteWcdmaCellInfo . . . 475
nas_minBasedIMSI . . . 476
nas_MNRInfo . . . 477
nas_namName . . . 478
nas_netSelectionPref . . . 478
nas_NetworkStat1x . . . 479
nas_NetworkStatEVDO . . . 481
nas_nmrCellInfo . . . 483
nas_operatorNameString . . . 484
nas_OperatorPLMNData . . . 484
nas_operatorPLMNList . . . 485
nas_PhyCaAggPcellInfo . . . 486
nas_PhyCaAggScellDlBw . . . 487
nas_PhyCaAggScellIndex . . . 487
nas_PhyCaAggScellIndType . . . 488
nas_PhyCaAggScellInfo . . . 489
nas_PilotSetData . . . 491
nas_PilotSetParams . . . 492

nas_PLMNNetworkName . . . 492
nas_PLMNNetworkNameData . . . 493
nas_protocolSubtypeElement . . . 494
nas_qaQmi3Gpp2TimeZone . . . 496
nas_QmiNas3GppNetworkInfo . . . 497
nas_QmiNas3GppNetworkRAT . . . 498
nas_QmisNasPcsDigit . . . 499
nas_RejectReasonTlv . . . 499
nas_RFInfoTlv . . . 500
nas_roamIndList . . . 501
nas_RSRPThresh . . . 502
nas_rsrqInformation . . . 502
nas_RSRQThresh . . . 503
nas_RSSIThresh . . . 503
nas_rxInfo . . . 504
nas_RxSigInfo . . . 505
nas_rxSignalStrengthListElement . . . 506
nas_SccRxInfo . . . 507
nas_serviceProviderName . . . 508
nas_servSystem . . . 509
nas_sidNid . . . 510
nas_SignalStrengthTlv . . . 511
nas_SLQSSignalStrengthsIndReq . . . 511
nas_SLQSSignalStrengthsInformation . . . 513
nas_SLQSSignalStrengthsTlv . . . 514
nas_SrvStatusInfo . . . 515
nas_sysInfoCommon . . . 515
nas_TDSCDMAECIOThresh . . . 518
nas_TDSCDMARSCPThresh . . . 518
nas_TDSCDMARSSIThresh . . . 519
nas_TDSCDMASINRCONFThresh . . . 519
nas_TDSCDMASINRThresh . . . 520
nas_timeInfo . . . 520
nas_trueIMSI . . . 522
nas_txInfo . . . 523
nas_UMTSInfo . . . 523
nas_UMTSinstInfo . . . 525
nas_umtsLTENbrCell . . . 526
nas_UniversalTime . . . 527
nas_wcdmaCellInfo . . . 528
nas_WCDMAECIOThresh . . . 529
nas_WCDMAInfoLTENeighborCell . . . 529
nas_WCDMARSSIThresh . . . 530
nas_WCDMASysInfo . . . 531
nas_wcdmaUARFCN . . . 534
NASBandPreferenceTlv . . . 535
nasCellLocationInfoResp . . . 535
NASEmergencyModeTlv . . . 536
nasGet3GPP2SubscriptionInfoReq . . . 537
nasGet3GPP2SubscriptionInfoResp . . . 537
nasGetHDRColorCodeResp . . . 538
nasGetLTECphyCa . . . 539
NasGetLTECphyCaInfo . . . 539
nasGetLTECphyCaResp . . . 540
nasGetSigInfoResp . . . 540
nasGetSysInfoResp . . . 541
nasGetTxRxInfoReq . . . 544
nasGetTxRxInfoResp . . . 545

NASGWAcqOrderPrefTlv . . . 545
nasIndicationRegisterReq . . . 546
nasInitNetworkReg . . . 549
NASLTEBandPreferenceTlv . . . 550
NASLteNasReleaseInfoTlv . . . 550
NASModePreferenceTlv . . . 551
NASNetSelPreferenceTlv . . . 552
nasNetworkReject . . . 552
nasNetworkTime . . . 555
nasOperatorNameResp . . . 556
NASOTAMessageTlv . . . 557
NASPhyCaAggPcellInfo . . . 558
NASPhyCaAggScellDlBw . . . 559
NASPhyCaAggScellIndex . . . 559
NASPhyCaAggScellIndType . . . 560
NASPhyCaAggScellInfo . . . 560
nasPLMNNameReq . . . 561
nasPLMNNameResp . . . 562
NASPRLPreferenceTlv . . . 565
NASQmiCbkNasSwiOTAMessageInd . . . 565
NASQmiCbkNasSystemSelPrefInd . . . 566
NASRoamPreferenceTlv . . . 567
NASServDomainPrefTlv . . . 567
NASServingSystemInfo . . . 568
nasSigInfo . . . 569
nasSwiGetChannelLockResp . . . 570
NasSwiIndReg . . . 571
nasSwiSetChannelLockReq . . . 572
nasSysInfo . . . 573
NASTimeInfoTlv . . . 576
nasTimers . . . 576
netSelectionPref . . . 577
NetStats . . . 578
NetworkDebugResp . . . 579
NetworkStat1x . . . 580
NetworkStatEVDO . . . 582
newMTMessageTlv . . . 583
newPwdData . . . 584
nmrCellInfo . . . 584
NSSAudioCtrl . . . 586
NWProfile . . . 586
omaDmConfigTlv . . . 587
omaDmConfigTlvExt . . . 588
omaDmFotaTlv . . . 590
omaDmFotaTlvExt . . . 592
omaDmNotificationsTlv . . . 594
operatorNameString . . . 595
OperatorPLMNData . . . 596
operatorPLMNList . . . 596
pack_dms_ActivateAutomatic_t . . . 597
pack_dms_GetCustFeaturesV2_t . . . 597
pack_dms_ResetToFactoryDefaults_t . . . 598
pack_dms_SetActivationStatusCallback_t . . . 598
pack_dms_SetCrashAction_t . . . 599
pack_dms_SetCustFeature_t . . . 599
pack_dms_SetCustFeaturesV2_t . . . 601
pack_dms_SetEventReport_t . . . 602
pack_dms_SetPower_t . . . 603

pack_dms_SetUSBComp_t . . . 603
pack_dms_SLQSDmsSwiIndicationRegister_t . . . 604
pack_dms_SLQSSwiGetCrashInfo_t . . . 605
pack_dms_SLQSSwiSetDyingGaspCfg_t . . . 605
pack_dms_SLQSSwiSetHostDevInfo_t . . . 606
pack_dms_SLQSSwiSetOSInfo_t . . . 606
pack_dms_UIMChangePIN_t . . . 607
pack_dms_UIMGetControlKeyStatus_t . . . 608
pack_dms_UIMGetICCID_t . . . 608
pack_dms_UIMSetControlKeyProtection_t . . . 608
pack_dms_UIMSetPINProtection_t . . . 609
pack_dms_UIMUnblockControlKey_t . . . 610
pack_dms_UIMUnblockPIN_t . . . 610
pack_dms_UIMVerifyPIN_t . . . 611
pack_fms_GetImagesPreference_t . . . 612
pack_fms_GetStoredImages_t . . . 612
pack_fms_SetImagesPreference_t . . . 612
pack_loc_Delete_Assist_Data_t . . . 613
pack_loc_EventRegister_t . . . 614
pack_loc_SetExtPowerState_t . . . 616
pack_loc_SetOperationMode_t . . . 616
pack_loc_SLQSLOCGetBestAvailPos_t . . . 617
pack_loc_SLQSLOCGetServer_t . . . 618
pack_loc_SLQSLOCInjectPosition_t . . . 618
pack_loc_SLQSLOCInjectSensorData_t . . . 623
pack_loc_SLQSLOCInjectUTCTime_t . . . 625
pack_loc_SLQSLOCSetCradleMountConfig_t . . . 625
pack_loc_Start_t . . . 626
pack_loc_Stop_t . . . 628
pack_nas_InitiateDomainAttach_t . . . 628
pack_nas_SetACCOLC_t . . . 629
pack_nas_SetCDMANetworkParameters_t . . . 629
pack_nas_SetNetworkPreference_t . . . 631
pack_nas_SLQSConfigSigInfo_t . . . 632
pack_nas_SLQSGetPLMNName_t . . . 633
pack_nas_SLQSInitiateNetworkRegistration_t . . . 634
pack_nas_SLQSNasConfigSigInfo2_t . . . 635
pack_nas_SLQSNasGet3GPP2Subscription_t . . . 640
pack_nas_SLQSNasGetTxRxInfo_t . . . 640
pack_nas_SLQSNasIndicationRegisterExt_t . . . 640
pack_nas_SLQSNasSwiIndicationRegister_t . . . 643
pack_nas_SLQSNASSwiSetChannelLock_t . . . 645
pack_nas_SLQSSetBandPreference_t . . . 645
pack_nas_SLQSSetSignalStrengthsCallback_t . . . 647
pack_nas_SLQSSetSysSelectionPref_t . . . 647
pack_nas_SLQSSwiPSDetach_t . . . 653
pack_pds_PDSInjectTimeReference_t . . . 653
pack_pds_ResetPDSData_t . . . 654
pack_pds_SetEventReportCallback_t . . . 654
pack_pds_SetPDSDefaults_t . . . 655
pack_pds_SetPDSState_t . . . 656
pack_pds_SetPortAutomaticTracking_t . . . 656
pack_pds_SetServiceAutomaticTracking_t . . . 657
pack_pds_SetXTRAAutomaticDownload_t . . . 657
pack_pds_SetXTRANetwork_t . . . 658
pack_pds_SLQSGetAGPSConfig_t . . . 658
pack_pds_SLQSPDSInjectAbsoluteTimeReference_t . . . 659
pack_pds_SLQSPDSInjectPositionData_t . . . 659

pack_pds_SLQSSetAGPSConfig_t . . . 661
pack_pds_SLQSSetPositionMethodState_t . . . 662
pack_pds_StartPDSTrackingSessionExt_t . . . 663
pack_qmi_t . . . 664
pack_qos_SLQSQosSwiReadApnExtraParams_t . . . 665
pack_qos_SLQSQosSwiReadDataStats_t . . . 665
pack_qos_SLQSSetQosEventCallback_t . . . 666
pack_rms_SetSMSWake_t . . . 666
pack_sar_SLQSSetRfSarState_t . . . 667
pack_sms_SaveSMS_t . . . 667
pack_sms_SendSMS_t . . . 668
pack_sms_SetNewSMSCallback_t . . . 669
pack_sms_SetSMSCAddress_t . . . 669
pack_sms_SLQSDeleteSMS_t . . . 670
pack_sms_SLQSGetSMS_t . . . 671
pack_sms_SLQSGetSmsBroadcastConfig_t . . . 671
pack_sms_SLQSGetSMSList_t . . . 672
pack_sms_SLQSModifySMSStatus_t . . . 673
pack_sms_SLQSSendAsyncSMS_t . . . 673
pack_sms_SLQSSetIndicationRegister_t . . . 674
pack_sms_SLQSSetSmsBroadcastActivation_t . . . 674
pack_sms_SLQSSetSmsBroadcastConfig_t . . . 675
pack_sms_SLQSSetSmsStorage_t . . . 675
pack_sms_SLQSSmsGetMaxStorageSize_t . . . 676
pack_sms_SLQSSmsSetRoutes_t . . . 676
pack_swiloc_SwiLocSetAutoStart_t . . . 677
pack_swioma_SLQSOMADMCancelSession_t . . . 678
pack_swioma_SLQSOMADMGetSessionInfo_t . . . 679
pack_swioma_SLQSOMADMSendSelection_t . . . 679
pack_swioma_SLQSOMADMSetSettings_t . . . 680
pack_swioma_SLQSOMADMStartSession_t . . . 681
pack_uim_ChangePin_t . . . 681
pack_uim_ReadTransparent_t . . . 682
pack_uim_SetPinProtection_t . . . 684
pack_uim_SLQSUIMAuthenticate_t . . . 685
pack_uim_SLQSUIMDepersonalization_t . . . 685
pack_uim_SLQSUIMEventRegister_t . . . 686
pack_uim_SLQSUIMGetConfiguration_t . . . 686
pack_uim_SLQSUIMGetFileAttributes_t . . . 687
pack_uim_SLQSUIMPowerDown_t . . . 687
pack_uim_SLQSUIMPowerUp_t . . . 688
pack_uim_SLQSUIMRefreshComplete_t . . . 688
pack_uim_SLQSUIMRefreshGetLastEvent_t . . . 689
pack_uim_SLQSUIMRefreshOK_t . . . 689
pack_uim_SLQSUIMRefreshRegister_t . . . 690
pack_uim_SLQSUIMSwitchSlot_t . . . 690
pack_uim_UnblockPin_t . . . 691
pack_uim_VerifyPin_t . . . 692
pack_voice_AnswerUSSD_t . . . 693
pack_voice_OriginateUSSD_t . . . 694
pack_voice_SLQSOriginateUSSD_t . . . 694
pack_voice_SLQSVoiceALSSelectLine_t . . . 695
pack_voice_SLQSVoiceALSSetLineSwitching_t . . . 695
pack_voice_SLQSVoiceAnswerCall_t . . . 696
pack_voice_SLQSVoiceBindSubscription_t . . . 696
pack_voice_SLQSVoiceBurstDTMF_t . . . 696
pack_voice_SLQSVoiceDialCall_t . . . 697
pack_voice_SLQSVoiceEndCall_t . . . 699

pack_voice_SLQSVoiceGetCallBarring_t . . . 699
pack_voice_SLQSVoiceGetCallForwardingStatus_t . . . 700
pack_voice_SLQSVoiceGetCallInfo_t . . . 701
pack_voice_SLQSVoiceGetCallWaiting_t . . . 701
pack_voice_SLQSVoiceGetConfig_t . . . 702
pack_voice_SLQSVoiceIndicationRegister_t . . . 704
pack_voice_SLQSVoiceManageCalls_t . . . 704
pack_voice_SLQSVoiceOrigUSSDNoWait_t . . . 705
pack_voice_SLQSVoiceSendFlash_t . . . 706
pack_voice_SLQSVoiceSetCallBarringPassword_t . . . 706
pack_voice_SLQSVoiceSetConfig_t . . . 707
pack_voice_SLQSVoiceSetPreferredPrivacy_t . . . 709
pack_voice_SLQSVoiceSetSUPSService_t . . . 709
pack_voice_SLQSVoiceStartContDTMF_t . . . 711
pack_voice_SLQSVoiceStopContDTMF_t . . . 712
pack_wds_DHCPv4ClientLeaseChange_t . . . 712
pack_wds_GetDefaultProfile_t . . . 713
pack_wds_GetDefaultProfileNum_t . . . 713
pack_wds_GetDormancyState_t . . . 714
pack_wds_GetLastMobileIPError_t . . . 714
pack_wds_GetMobileIP_t . . . 714
pack_wds_GetMobileIPProfile_t . . . 714
pack_wds_GetPacketStatistics_t . . . 715
pack_wds_GetPacketStatus_t . . . 715
pack_wds_GetSessionDuration_t . . . 716
pack_wds_RMSetTransferStatistics_t . . . 716
pack_wds_SetAutoconnect_t . . . 717
pack_wds_SetDefaultProfile_t . . . 717
pack_wds_SetDefaultProfileNum_t . . . 719
pack_wds_SetMobileIP_t . . . 720
pack_wds_SetMobileIPParameters_t . . . 720
pack_wds_SetMobileIPProfile_t . . . 721
pack_wds_SLQSCreateProfile_t . . . 723
pack_wds_SLQSDeleteProfile_t . . . 724
pack_wds_SLQSGetCurrDataSystemStat_t . . . 724
pack_wds_SLQSGetDataBearerTechnology_t . . . 725
pack_wds_SLQSGetDUNCallInfo_t . . . 725
pack_wds_SLQSGetProfileSettings_t . . . 726
pack_wds_SLQSGetRuntimeSettings_t . . . 727
pack_wds_SLQSModifyProfile_t . . . 728
pack_wds_SLQSSet3GPPConfigItem_t . . . 729
pack_wds_SLQSSetIPFamilyPreference_t . . . 731
pack_wds_SLQSSetWdsEventCallback_t . . . 731
pack_wds_SLQSSGetDHCPv4ClientConfig_t . . . 732
pack_wds_SLQSSSetDHCPv4ClientConfig_t . . . 732
pack_wds_SLQSSSetLoopback_t . . . 733
pack_wds_SLQSStartDataSession_t . . . 733
pack_wds_SLQSStopDataSession_t . . . 735
pack_wds_SLQSWdsSetEventReport_t . . . 735
pack_wds_SLQSWdsSwiPDPRuntimeSettings_t . . . 737
PackCreateProfileOut . . . 737
packgetDyingGaspCfg . . . 738
packgetDyingGaspStatistics . . . 738
PCMparams . . . 739
PCSCFFQDNAddress . . . 740
PCSCFFQDNAddressList . . . 740
PCSCFIPv4ServerAddressList . . . 741
PDSPositionData . . . 741

PDSPosMethodStateReq . . . 743
peerNumberInfo . . . 744
personalizationStatus . . . 746
PhyCaAggPcellInfo . . . 747
PhyCaAggScellDlBw . . . 747
PhyCaAggScellIndex . . . 748
PhyCaAggScellIndType . . . 748
PhyCaAggScellInfo . . . 749
PilotSetData . . . 752
PilotSetParams . . . 753
pktErrRate . . . 753
PlmnID . . . 754
PLMNNetworkName . . . 754
PLMNNetworkNameData . . . 755
Port . . . 757
precisionDilution_s . . . 757
PrefImageList . . . 758
prefVoiceSO . . . 758
Profile3GPP . . . 760
Profile3GPP2 . . . 766
ProfileIdentifier . . . 771
protocolSubtypeElement . . . 772
PSDetachReq . . . 773
qaQmi3Gpp2TimeZone . . . 774
qaQmiInterfaceInfo . . . 775
qaQmiServingSystemParam . . . 775
QmiCbkCatEventStatusReportInd . . . 779
QmiCbkLocBestAvailPosInd . . . 779
QmiCbkLocCradleMountInd . . . 785
QmiCbkLocEngineStateInd . . . 786
QmiCbkLocEventTimeSyncInd . . . 786
QmiCbkLocGetServerInd . . . 787
QmiCbkLocInjectPositionInd . . . 788
QmiCbkLocInjectSensorDataInd . . . 789
QmiCbkLocInjectTimeInd . . . 790
QmiCbkLocInjectUTCTimeInd . . . 791
QmiCbkLocPositionReportInd . . . 792
QmiCbkLocSensorStreamingInd . . . 797
QmiCbkLocSetExtPowerConfigInd . . . 798
QmiCbkNasLTECphyCaInfo . . . 799
QmiCbkSwiOmaDmEventStatusReportInd . . . 800
QmiCbkSwiOmaDmEventStatusReportIndExt . . . 800
QmiCbkTmdMitiLvlRptInd . . . 800
QmiCbkWdsStatisticsIndState . . . 801
qmifwinfo_s . . . 802
QmiNas3GppNetworkInfo . . . 802
QmiNasGetRFBandInfoResp . . . 804
QmiNasPerformNetworkScanResp . . . 804
qmiSmsMessageList . . . 804
qmiWDSDataBearerTechnology . . . 805
QmiWdsIpAddressInfo . . . 806
qmiWdsRunTimeSettings . . . 807
qmuxIPTable . . . 811
QosClassID . . . 811
QosEventInfo . . . 812
QosFlowInfo . . . 814
QosFlowInfoState . . . 815
QosMap . . . 815

RankIndicatorInd . . . 816
readResult . . . 816
readTransparentInfo . . . 817
redirNumInfo . . . 817
registerRefresh . . . 819
remainingRetries . . . 820
remotePartyName . . . 820
remotePartyNum . . . 821
ReqFieldsList . . . 822
RespFieldsList . . . 823
RFBandInfoElements . . . 823
rmTrasnferStaticsReq . . . 824
roamIndList . . . 825
RoamingInfo . . . 826
roamTimer . . . 826
RSRPThresh . . . 827
rsrqInformation . . . 827
RSRQThresh . . . 828
RSSIThresh . . . 828
RXAGCList . . . 829
RXAVCList . . . 830
rxInfo . . . 831
RXPCMIIRFltr . . . 832
RxSigInfo . . . 833
rxSignalStrengthListElement . . . 834
sApnExtraParams . . . 835
satelliteInfo . . . 836
SccRxInfo . . . 839
sensorData . . . 839
sensorData_t . . . 841
sensorDataUsage_s . . . 843
serialNumbersInfo . . . 843
serviceProviderName . . . 845
ServingSystemInfo . . . 845
servSystem . . . 847
sessionInfo . . . 848
sessionInfoExt . . . 849
sessionInfoTlv . . . 849
sessionInfoTlvExt . . . 850
SetAudioPathConfigReq . . . 850
SetAudioProfileReq . . . 852
SetAudioVolTLBConfigReq . . . 853
SetAudioVolTLBConfigResp . . . 854
setCustomSettingV2 . . . 855
setDyingGaspCfg . . . 855
SetIMSSMSConfigReq . . . 856
SetIMSSMSConfigResp . . . 856
SetIMSUserConfigReq . . . 857
SetIMSUserConfigResp . . . 857
SetIMSVoIPConfigReq . . . 858
SetIMSVoIPConfigResp . . . 860
SetM2MAudioAVCFGReq . . . 861
SetM2MAudioLPBKReq . . . 862
SetM2MAudioProfileReq . . . 862
SetM2MAudioVolumeReq . . . 863
SetM2MAVMuteReq . . . 864
SetM2MSpkrGainReq . . . 865
setPINProtection . . . 865

SetRegMgrConfigReq . . . 866
SetRegMgrConfigResp . . . 867
setSignalStrengthInfo . . . 867
SetSIPConfigReq . . . 872
SetSIPConfigResp . . . 873
sGetDeviceSeriesResult . . . 874
sidNid . . . 874
sigInfo . . . 875
signalInfo . . . 876
SignalStrengthDataType . . . 877
slot_t . . . 877
slotInf . . . 878
slotInfo . . . 880
slots_t . . . 881
slqsautoconnect . . . 882
SLQSDeleteProfileParams . . . 882
slqsfwinfo_s . . . 883
SlqsNas3GppNetworkInfo . . . 884
SlqsNasPcsDigit . . . 886
slqssendasyncsmsparams_s . . . 886
slqssendsmsparams_s . . . 888
slqsSessionStateInfo . . . 890
slqsSignalStrengthInfo . . . 890
SLQSSignalStrengthsIndReq . . . 893
SLQSSignalStrengthsInformation . . . 895
slqsWdsEventInfo . . . 896
sms_BroadcastConfig . . . 898
sms_CDMABroadcastConfig . . . 898
sms_getIndicationReg . . . 899
sms_getMsgWaitingInfo . . . 900
sms_getTransLayerInfo . . . 900
sms_getTransNWRegInfo . . . 901
sms_maxStorageSizeReq . . . 902
sms_maxStorageSizeResp . . . 902
sms_messageWaitingInfoContent . . . 903
sms_msgProtocolResp . . . 904
sms_qaQmi3GPP2BroadcastCfgInfo . . . 904
sms_qaQmi3GPPBroadcastCfgInfo . . . 905
sms_routeEntry . . . 905
sms_sendAsyncsmsParams . . . 906
sms_setIndicationReg . . . 909
sms_setRoutesReq . . . 909
sms_transLayerInfo . . . 910
SMSAsyncRawSend_s . . . 911
SMSCAddress . . . 912
sMSCAddressInfo . . . 913
sMSCAddressTlv . . . 913
SMSEtwsMessage . . . 914
sMSEtwsMessageInfo . . . 915
sMSEtwsMessageTlv . . . 915
SMSEtwsPlmn . . . 916
sMSEtwsPlmnInfo . . . 916
SMSEventInfo_s . . . 917
smsMaxStorageSizeReq . . . 918
smsMaxStorageSizeResp . . . 919
SMSMemoryInfo . . . 919
SMSMessageMode . . . 920
sMSMessageModeInfo . . . 920

smsMsgprotocolResp . . . 920
SMSMTMessage . . . 921
sMSMTMessageInfo . . . 921
SMSOnIMS . . . 922
sMSOnIMSInfo . . . 922
sMSOnIMSTlv . . . 923
smsRouteEntry . . . 923
smsSetRoutesReq . . . 924
SMSTransferRouteMTMessage . . . 925
sMSTransferRouteMTMessageInfo . . . 926
sQosFlowStat . . . 927
sQosStat . . . 927
SrvStatusInfo . . . 929
ssdatasession_params . . . 929
SupportedMsgList . . . 932
SUPSInfo . . . 933
SV . . . 933
SVInfo . . . 934
svUsedforFix_s . . . 935
SWI_STRUCT_CarrierImage . . . 935
SwiLocGetAutoStartResp . . . 937
SwiLocSetAutoStartReq . . . 938
swiModemStatusResp . . . 940
SwiOTAMsg_s . . . 940
swiPDPRuntimeSettingsReq . . . 941
swiPDPRuntimeSettingsResp . . . 942
swiQosFilter . . . 944
swiQosFlow . . . 947
swiQosGranted . . . 950
swiQosIds . . . 950
swiQosModifyReq . . . 951
swiQosReq . . . 951
swiRMTrasnferStaticsReq . . . 952
sysInfoCommon . . . 953
t_gpsTime . . . 955
t_sensor . . . 955
t_Sv . . . 956
TDSCDMAECIOThresh . . . 956
TDSCDMARSCPThresh . . . 956
TDSCDMARSSIThresh . . . 957
TDSCDMASigInfoExt . . . 957
tdscdmaSigInfoExt . . . 958
TDSCDMASINRCONFThresh . . . 959
TDSCDMASINRThresh . . . 959
tempData_t . . . 960
tempratureData . . . 961
TFTIDParams . . . 962
timeInfo . . . 964
TmdDeRegNotMitigationLvlReq . . . 966
TmdGetMitigationDevListResp . . . 966
TmdGetMitigationLvlReq . . . 967
TmdGetMitigationLvlResp . . . 967
TmdMitigationLvlIndReq . . . 968
TmdRegNotMitigationLvlReq . . . 968
tokenBucket . . . 969
Tos . . . 969
transferRouteMessageTlv . . . 970
TransferStatInd . . . 971

TransferStatsDataType . . . 971
TrStatInd . . . 971
trueIMSI . . . 972
TXAGCList . . . 973
txInfo . . . 974
TXPCMIIRFltr . . . 975
uim_appStatus . . . 976
uim_authenticateResult . . . 979
uim_authenticationData . . . 980
uim_cardResult . . . 981
uim_cardStatus . . . 981
uim_changeUIMPIN . . . 982
uim_depersonalizationInformation . . . 983
uim_encryptedPIN1 . . . 984
uim_fileAttributes . . . 985
uim_fileInfo . . . 988
uim_hotSwapStatus . . . 989
uim_personalizationStatus . . . 989
uim_readResult . . . 990
uim_readTransparentInfo . . . 991
uim_refreshevent . . . 991
uim_registerRefresh . . . 993
uim_remainingRetries . . . 994
uim_sessionInformation . . . 995
uim_setPINProtection . . . 995
uim_slotInfo . . . 996
uim_UIMSessionInformation . . . 998
uim_unblockUIMPIN . . . 999
uim_verifyUIMPIN . . . 1000
UIMAuthenticateReq . . . 1000
UIMAuthenticateResp . . . 1001
UIMChangePinReq . . . 1002
UIMDepersonalizationReq . . . 1003
UIMDepersonalizationResp . . . 1003
UIMEventRegisterReqResp . . . 1003
UIMGetCardStatusResp . . . 1004
UIMGetConfigurationReq . . . 1004
UIMGetConfigurationResp . . . 1005
UIMGetFileAttributesReq . . . 1006
UIMGetFileAttributesResp . . . 1006
UIMGetSlotsStatusResp . . . 1007
UIMPinResp . . . 1008
UIMPowerDownReq . . . 1008
UIMPowerUpReq . . . 1009
UIMReadTransparentReq . . . 1009
UIMReadTransparentResp . . . 1010
UIMRefreshCompleteReq . . . 1011
UIMRefreshEvent . . . 1012
UIMRefreshGetLastEventReq . . . 1013
UIMRefreshGetLastEventResp . . . 1014
UIMRefreshOKReq . . . 1014
UIMRefreshRegisterReq . . . 1015
UIMSessionInformation . . . 1015
UIMSetPinProtectionReq . . . 1016
UIMSlotsStatus . . . 1017
UIMSlotStatus . . . 1017
UIMSlotStatusChangeInfo . . . 1019
UIMStatusChangeInfo . . . 1019

UIMSwitchSlotReq . . . 1019
UIMUnblockPinReq . . . 1020
UIMVerifyPinReq . . . 1021
UMTSInfo . . . 1022
UMTSinstInfo . . . 1024
umtsLTENbrCell . . . 1024
UMTSMinQoS . . . 1025
UMTSQoS . . . 1028
UMTSReqQoSSigInd . . . 1030
unblockUIMPIN . . . 1031
UniversalTime . . . 1032
unpack_dms_GetActivationState_t . . . 1033
unpack_dms_GetBandCapability_t . . . 1034
unpack_dms_GetCrashAction_t . . . 1036
unpack_dms_GetCustFeature_t . . . 1036
unpack_dms_GetCustFeaturesV2_t . . . 1039
unpack_dms_GetDeviceCap_t . . . 1039
unpack_dms_GetDeviceCapabilities_t . . . 1040
unpack_dms_GetDeviceHardwareRev_t . . . 1042
unpack_dms_GetDeviceMfr_t . . . 1042
unpack_dms_GetDeviceSerialNumbers_t . . . 1043
unpack_dms_GetFirmwareInfo_t . . . 1044
unpack_dms_GetFirmwareRevision_t . . . 1046
unpack_dms_GetFirmwareRevisions_t . . . 1047
unpack_dms_GetFSN_t . . . 1048
unpack_dms_GetHardwareRevision_t . . . 1049
unpack_dms_GetIMSI_t . . . 1049
unpack_dms_GetManufacturer_t . . . 1050
unpack_dms_GetModelID_t . . . 1050
unpack_dms_GetNetworkTime_t . . . 1051
unpack_dms_GetOfflineReason_t . . . 1051
unpack_dms_GetPower_t . . . 1052
unpack_dms_GetPRLVersion_t . . . 1054
unpack_dms_GetSerialNumbers_t . . . 1054
unpack_dms_GetUSBComp_t . . . 1055
unpack_dms_GetVoiceNumber_t . . . 1057
unpack_dms_ResetToFactoryDefaults_t . . . 1058
unpack_dms_SetActivationStatusCallback_t . . . 1059
unpack_dms_SetCrashAction_t . . . 1059
unpack_dms_SetCustFeature_t . . . 1059
unpack_dms_SetCustFeaturesV2_t . . . 1060
unpack_dms_SetEventReport_ind_t . . . 1060
unpack_dms_SetEventReport_t . . . 1061
unpack_dms_SetFirmwarePreference_t . . . 1061
unpack_dms_SetPower_t . . . 1062
unpack_dms_SetUSBComp_t . . . 1062
unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t . . . 1062
unpack_dms_SLQSDmsSwiGetResetInfo_t . . . 1063
unpack_dms_SLQSDmsSwiIndicationRegister_t . . . 1064
unpack_dms_SLQSGetBandCapability_t . . . 1065
unpack_dms_SLQSGetERIFile_t . . . 1068
unpack_dms_SLQSSwiClearDyingGaspStatistics_t . . . 1068
unpack_dms_SLQSSwiGetCrashInfo_t . . . 1069
unpack_dms_SLQSSwiGetDyingGaspCfg_t . . . 1069
unpack_dms_SLQSSwiGetDyingGaspStatistics_t . . . 1070
unpack_dms_SLQSSwiGetFirmwareCurr_t . . . 1070
unpack_dms_SLQSSwiGetFwUpdateStatus_t . . . 1072
unpack_dms_SLQSSwiGetHostDevInfo_t . . . 1073

unpack_dms_SLQSSwiGetOSInfo_t . . . 1074
unpack_dms_SLQSSwiGetSerialNoExt_t . . . 1075
unpack_dms_SLQSSwiSetDyingGaspCfg_t . . . 1075
unpack_dms_SLQSSwiSetHostDevInfo_t . . . 1076
unpack_dms_SLQSSwiSetOSInfo_t . . . 1076
unpack_dms_SLQSUIMGetState_t . . . 1076
unpack_dms_UIMGetControlKeyStatus_t . . . 1077
unpack_dms_UIMGetICCID_t . . . 1078
unpack_dms_UIMGetPINStatus_t . . . 1078
unpack_dms_UIMSetControlKeyProtection_t . . . 1080
unpack_dms_UIMSetPINProtection_t . . . 1080
unpack_dms_UIMUnblockControlKey_t . . . 1081
unpack_fms_GetImagesPreference_t . . . 1082
unpack_fms_GetStoredImages_t . . . 1082
unpack_fms_SetImagesPreference_t . . . 1083
unpack_loc_BestAvailPos_Ind_t . . . 1084
unpack_loc_Delete_Assist_Data_t . . . 1089
unpack_loc_DeleteAssistData_Ind_t . . . 1090
unpack_loc_EngineState_Ind_t . . . 1091
unpack_loc_EventNMEA_Ind_t . . . 1091
unpack_loc_EventRegister_t . . . 1092
unpack_loc_GetServer_Ind_t . . . 1092
unpack_loc_GnssSvInfo_Ind_t . . . 1094
unpack_loc_PositionRpt_Ind_t . . . 1094
unpack_loc_SetExtPowerConfig_Ind_t . . . 1101
unpack_loc_SetExtPowerState_t . . . 1102
unpack_loc_SetOperationMode_Ind_t . . . 1102
unpack_loc_SetOperationMode_t . . . 1103
unpack_loc_SLQSLOCGetBestAvailPos_t . . . 1103
unpack_loc_Start_t . . . 1104
unpack_loc_Stop_t . . . 1104
unpack_nas_GetCDMANetworkParameters_t . . . 1104
unpack_nas_GetHomeNetwork3GPP2_t . . . 1107
unpack_nas_GetHomeNetwork_t . . . 1109
unpack_nas_GetNetworkPreference_t . . . 1110
unpack_nas_GetRFInfo_t . . . 1111
unpack_nas_GetServingNetwork_t . . . 1112
unpack_nas_GetServingNetworkCapabilities_t . . . 1114
unpack_nas_GetSignalStrengths_t . . . 1115
unpack_nas_PerformNetworkScan_t . . . 1116
unpack_nas_SetDataCapabilitiesCallback_ind_t . . . 1117
unpack_nas_SetEventReportInd_t . . . 1117
unpack_nas_SetNasLTECphyCaIndCallback_ind_t . . . 1118
unpack_nas_SetNetworkPreference_t . . . 1119
unpack_nas_SetRoamingIndicatorCallback_ind_t . . . 1119
unpack_nas_SetServingSystemCallback_ind_t . . . 1120
unpack_nas_SLQSGetErrorRate_t . . . 1120
unpack_nas_SlqsGetLTECphyCAInfo_t . . . 1121
unpack_nas_SLQSGetNetworkTime_t . . . 1122
unpack_nas_SLQSGetOperatorNameData_t . . . 1122
unpack_nas_SLQSGetPLMNName_t . . . 1123
unpack_nas_SLQSGetServingSystem_t . . . 1126
unpack_nas_SLQSGetSignalStrength_t . . . 1130
unpack_nas_SLQSGetSysInfo_t . . . 1132
unpack_nas_SLQSGetSysSelectionPref_t . . . 1136
unpack_nas_SLQSNasGet3GPP2Subscription_t . . . 1140
unpack_nas_SLQSNasGetCellLocationInfo_t . . . 1141
unpack_nas_SLQSNasGetHDRColorCode_t . . . 1143

unpack_nas_SLQSNasGetSigInfo_t . . . 1143
unpack_nas_SLQSNasGetTxRxInfo_t . . . 1144
unpack_nas_SLQSNasNetworkTimeCallBack_ind_t . . . 1145
unpack_nas_SLQSNasSigInfoCallback_ind_t . . . 1146
unpack_nas_SLQSNASSwiGetChannelLock_t . . . 1147
unpack_nas_SLQSNasSwiModemStatus_t . . . 1148
unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t . . . 1148
unpack_nas_SLQSNasTimerCallback_ind_t . . . 1149
unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t . . . 1150
unpack_nas_SLQSSwiGetHDRPersonality_t . . . 1150
unpack_nas_SLQSSwiGetHDRProtSubtype_t . . . 1151
unpack_nas_SLQSSwiGetHRPDStats_t . . . 1152
unpack_nas_SLQSSwiGetLteCQI_t . . . 1152
unpack_nas_SLQSSwiGetLteSccRxInfo_t . . . 1153
unpack_nas_SLQSSwiNetworkDebug_t . . . 1154
unpack_nas_SLQSSysInfoCallback_ind_t . . . 1154
unpack_omaDmConfigTlv_t . . . 1157
unpack_omaDmFotaTlv_t . . . 1158
unpack_omaDmNotificationsTlv_t . . . 1160
unpack_pds_GetPDSDefaults_t . . . 1161
unpack_pds_GetPDSState_t . . . 1162
unpack_pds_GetPortAutomaticTracking_t . . . 1163
unpack_pds_GetServiceAutomaticTracking_t . . . 1163
unpack_pds_GetXTRAAutomaticDownload_t . . . 1164
unpack_pds_GetXTRANetwork_t . . . 1164
unpack_pds_GetXTRAValidity_t . . . 1165
unpack_pds_SetEventReport_Ind_t . . . 1165
unpack_pds_SetPdsState_Ind_t . . . 1167
unpack_pds_SLQSGetAGPSConfig_t . . . 1168
unpack_pds_SLQSGetGPSStateInfo_t . . . 1169
unpack_qmi_t . . . 1173
unpack_qos_dataRate_t . . . 1173
unpack_qos_IPv4Addr_t . . . 1174
unpack_qos_IPv6Addr_t . . . 1174
unpack_qos_IPv6TrafCls_t . . . 1175
unpack_qos_pktErrRate_t . . . 1175
unpack_qos_Port_t . . . 1176
unpack_qos_QosFlowInfo_t . . . 1176
unpack_qos_QosFlowInfoState_t . . . 1178
unpack_qos_SLQSQosGetNetworkStatus_t . . . 1178
unpack_qos_SLQSQosSwiReadApnExtraParams_t . . . 1179
unpack_qos_SLQSQosSwiReadDataStats_t . . . 1180
unpack_qos_SLQSSetQosEventCallback_ind_t . . . 1181
unpack_qos_SLQSSetQosNWStatusCallback_ind_t . . . 1182
unpack_qos_SLQSSetQosPriEventCallback_ind_t . . . 1182
unpack_qos_SLQSSetQosStatusCallback_ind_t . . . 1183
unpack_qos_swiQosFilter_t . . . 1184
unpack_qos_swiQosFlow_t . . . 1188
unpack_qos_tokenBucket_t . . . 1192
unpack_qos_Tos_t . . . 1192
unpack_QosFlowStat_t . . . 1193
unpack_rms_GetSMSWake_t . . . 1193
unpack_RMTransferStatistics_ind_t . . . 1194
unpack_sar_SLQSGetRfSarState_t . . . 1195
unpack_sms_GetSMSCAddress_t . . . 1196
unpack_sms_SaveSMS_t . . . 1196
unpack_sms_SendSMS_t . . . 1197
unpack_sms_SetNewSMSCallback_ind_t . . . 1197

unpack_sms_SetNewSMSCallback_t . . . 1198
unpack_sms_SLQSDeleteSMS_t . . . 1199
unpack_sms_SLQSGetIndicationRegister_t . . . 1199
unpack_sms_SLQSGetMessageWaiting_t . . . 1199
unpack_sms_SLQSGetSMS_t . . . 1200
unpack_sms_SLQSGetSmsBroadcastConfig_t . . . 1201
unpack_sms_SLQSGetSMSList_t . . . 1201
unpack_sms_SLQSGetTransLayerInfo_t . . . 1202
unpack_sms_SLQSGetTransNWRegInfo_t . . . 1202
unpack_sms_SLQSModifySMSStatus_t . . . 1202
unpack_sms_SLQSSmsGetMaxStorageSize_t . . . 1203
unpack_sms_SLQSSmsGetMessageProtocol_t . . . 1203
unpack_sms_SLQSSwiGetSMSStorage_t . . . 1204
unpack_sms_SLQSWmsMemoryFullCallBack_ind_t . . . 1204
unpack_swiloc_SwiLocGetAutoStart_t . . . 1205
unpack_swioma_SLQSOMADMAlertCallback_ind_t . . . 1206
unpack_swioma_SLQSOMADMGetSessionInfo_t . . . 1207
unpack_swioma_SLQSOMADMGetSettings_t . . . 1210
unpack_swioma_SLQSOMADMStartSession_t . . . 1211
unpack_uim_ChangePin_t . . . 1212
unpack_uim_GetCardStatus_t . . . 1213
unpack_uim_ReadTransparent_t . . . 1213
unpack_uim_SetPinProtection_t . . . 1214
unpack_uim_SetUimSlotStatusChangeCallback_ind_t . . . 1215
unpack_uim_SLQSUIMAuthenticate_t . . . 1216
unpack_uim_SLQSUIMDepersonalization_t . . . 1216
unpack_uim_SLQSUIMEventRegister_t . . . 1217
unpack_uim_SLQSUIMGetConfiguration_t . . . 1217
unpack_uim_SLQSUIMGetFileAttributes_t . . . 1218
unpack_uim_SLQSUIMGetSlotsStatus_t . . . 1219
unpack_uim_SLQSUIMRefreshGetLastEvent_t . . . 1219
unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t . . . 1220
unpack_uim_UnblockPin_t . . . 1220
unpack_uim_VerifyPin_t . . . 1221
unpack_voice_SLQSOriginateUSSD_t . . . 1222
unpack_voice_SLQSVoiceAnswerCall_t . . . 1225
unpack_voice_SLQSVoiceBurstDTMF_t . . . 1226
unpack_voice_SLQSVoiceDialCall_t . . . 1226
unpack_voice_SLQSVoiceEndCall_t . . . 1227
unpack_voice_SLQSVoiceGetAllCallInfo_t . . . 1228
unpack_voice_SLQSVoiceGetCallBarring_t . . . 1230
unpack_voice_SLQSVoiceGetCallForwardingStatus_t . . . 1232
unpack_voice_SLQSVoiceGetCallInfo_t . . . 1233
unpack_voice_SLQSVoiceGetCallWaiting_t . . . 1237
unpack_voice_SLQSVoiceGetCLIP_t . . . 1238
unpack_voice_SLQSVoiceGetCLIR_t . . . 1239
unpack_voice_SLQSVoiceGetCNAP_t . . . 1241
unpack_voice_SLQSVoiceGetCOLP_t . . . 1242
unpack_voice_SLQSVoiceGetCOLR_t . . . 1243
unpack_voice_SLQSVoiceGetConfig_t . . . 1245
unpack_voice_SLQSVoiceManageCalls_t . . . 1247
unpack_voice_SLQSVoiceSendFlash_t . . . 1247
unpack_voice_SLQSVoiceSetCallBarringPassword_t . . . 1248
unpack_voice_SLQSVoiceSetConfig_t . . . 1249
unpack_voice_SLQSVoiceSetSUPSService_t . . . 1251
unpack_voice_SLQSVoiceStartContDTMF_t . . . 1252
unpack_voice_SLQSVoiceStopContDTMF_t . . . 1252
unpack_wds_DHCPv4ClientLease_ind_t . . . 1253

unpack_wds_GetAutoconnect_t . . . 1253
unpack_wds_GetByteTotals_t . . . 1254
unpack_wds_GetConnectionRate_t . . . 1254
unpack_wds_GetDataBearerTechnology_t . . . 1255
unpack_wds_GetDefaultProfile_t . . . 1256
unpack_wds_GetDefaultProfileNum_t . . . 1258
unpack_wds_GetDormancyState_t . . . 1259
unpack_wds_GetLastMobileIPError_t . . . 1259
unpack_wds_GetMobileIP_t . . . 1260
unpack_wds_GetMobileIPProfile_t . . . 1260
unpack_wds_GetPacketStatistics_t . . . 1262
unpack_wds_GetPacketStatus_t . . . 1264
unpack_wds_GetSessionDuration_t . . . 1266
unpack_wds_GetSessionState_t . . . 1266
unpack_wds_RMSetTransferStatistics_t . . . 1267
unpack_wds_SetMobileIPProfile_t . . . 1267
unpack_wds_SLQSCreateProfile_t . . . 1267
unpack_wds_SLQSDeleteProfile_t . . . 1268
unpack_wds_SLQSGet3GPPConfigItem_t . . . 1268
unpack_wds_SLQSGetCurrDataSystemStat_t . . . 1270
unpack_wds_SLQSGetCurrentChannelRate_t . . . 1271
unpack_wds_SLQSGetDataBearerTechnology_t . . . 1272
unpack_wds_SLQSGetDUNCallInfo_t . . . 1272
unpack_wds_SLQSGetProfileSettings_t . . . 1275
unpack_wds_SLQSGetRuntimeSettings_t . . . 1276
unpack_wds_SLQSModifyProfile_t . . . 1279
unpack_wds_SLQSSetIPFamilyPreference_t . . . 1279
unpack_wds_SLQSSetPacketSrvStatusCallback_t . . . 1280
unpack_wds_SLQSSetWdsEventCallback_ind_t . . . 1282
unpack_wds_SLQSSGetDHCPv4ClientConfig_t . . . 1284
unpack_wds_SLQSSGetLoopback_t . . . 1284
unpack_wds_SLQSStartDataSession_t . . . 1285
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t . . . 1285
UnPackGetProfileSettingOut . . . 1288
unpackWdsProfileParam . . . 1288
urlAddr . . . 1289
USBCompConfig . . . 1289
USBCompParams . . . 1290
USSDNoWaitIndicationInfo . . . 1292
USSDRespFNetwork . . . 1293
USSInfo . . . 1293
USSResp . . . 1294
UUSInfo . . . 1295
verifyUIMPIN . . . 1296
voice_airTimer . . . 1296
voice_allCallsAlphaIDInfo . . . 1297
voice_allCallsDiagInfo . . . 1297
voice_allCallsUUSInfo . . . 1298
voice_alphaIDInfo . . . 1298
voice_arrAlertingPattern . . . 1299
voice_arrAlertingType . . . 1300
voice_arrAlphaID . . . 1301
voice_arrCalledPartyNum . . . 1301
voice_arrCallEndReason . . . 1302
voice_arrCallInfo . . . 1302
voice_arrConnectPartyNum . . . 1303
voice_arrDiagInfo . . . 1304
voice_arrRedirPartyNum . . . 1304

voice_arrRemotePartyName . . . 1305
voice_arrRemotePartyNum . . . 1305
voice_arrSvcOption . . . 1306
voice_arrUUSInfo . . . 1306
voice_burstDTMFInfo . . . 1307
voice_calledPartySubAdd . . . 1308
voice_callFwdTypeAndPlan . . . 1309
voice_callFWExtInfo . . . 1309
voice_callFWInfo . . . 1311
voice_callInfo . . . 1312
voice_ccSUPSType . . . 1314
voice_CLIPResp . . . 1314
voice_CLIRResp . . . 1315
voice_CNAPResp . . . 1316
voice_COLPResp . . . 1317
voice_COLRResp . . . 1317
voice_connectNumInfo . . . 1318
voice_CUGInfo . . . 1320
voice_curAMRConfig . . . 1320
voice_diagInfo . . . 1321
voice_DTMFLengths . . . 1322
voice_getAllCallInformation . . . 1322
voice_getAllCallRmtPtyName . . . 1323
voice_getAllCallRmtPtyNum . . . 1323
voice_getCallFWExtInfo . . . 1324
voice_getCallFWInfo . . . 1325
voice_peerNumberInfo . . . 1325
voice_prefVoiceSO . . . 1327
voice_remotePartyName . . . 1329
voice_remotePartyNum . . . 1330
voice_roamTimer . . . 1330
voice_USSInfo . . . 1331
voice_UUSInfo . . . 1332
voiceALSSelectLineInfo . . . 1333
voiceALSSetLineSwitchInfo . . . 1333
voiceAnswerCall . . . 1334
voiceBindSubscriptionInfo . . . 1334
voiceBurstDTMFInfo . . . 1334
voiceCallInfoReq . . . 1335
voiceCallInfoResp . . . 1335
voiceCallRequestParams . . . 1338
voiceCallResponseParams . . . 1340
voiceContDTMFinfo . . . 1341
voiceDTMFEventInfo . . . 1342
voiceFlashInfo . . . 1342
voiceGetAllCallInfo . . . 1343
voiceGetCallBarringReq . . . 1346
voiceGetCallBarringResp . . . 1346
voiceGetCallFWReq . . . 1348
voiceGetCallFWResp . . . 1348
voiceGetCallWaitInfo . . . 1350
voiceGetCLIPResp . . . 1351
voiceGetCLIRResp . . . 1352
voiceGetCNAPResp . . . 1354
voiceGetCOLPResp . . . 1355
voiceGetCOLRResp . . . 1356
voiceGetConfigReq . . . 1357
voiceGetConfigResp . . . 1359

voiceIndicationRegisterInfo . . . 1361
voiceInfoRec . . . 1362
voiceManageCallsReq . . . 1364
voiceManageCallsResp . . . 1365
voiceOrigUSSDNoWaitInfo . . . 1366
voiceOTASPStatusInfo . . . 1366
voicePrivacyInfo . . . 1367
voiceSetAllCallStatusCbkInfo . . . 1368
voiceSetCallBarringPwdInfo . . . 1370
voiceSetCallBarringPwdResp . . . 1371
voiceSetConfigReq . . . 1372
voiceSetConfigResp . . . 1374
voiceSetPrefPrivacy . . . 1375
voiceSetSUPSServiceReq . . . 1376
voiceSetSUPSServiceResp . . . 1378
voiceStopContDTMFinfo . . . 1379
voiceSUPSInfo . . . 1380
voiceSUPSNotification . . . 1382
wcdmaCellInfo . . . 1384
WCDMAECIOThresh . . . 1385
WCDMAInfoLTENeighborCell . . . 1385
wcdmaLongMsgDecodingParams . . . 1386
wcdmaMsgDecodingParams . . . 1388
wcdmaMsgEncodingParams . . . 1389
WCDMARSSIThresh . . . 1390
WCDMASysInfo . . . 1391
wcdmaUARFCN . . . 1394
wds_currNetworkInfo . . . 1395
wds_DataULongLongTlv . . . 1396
wds_DataULongTlv . . . 1397
wds_DHCPLeaseOptTlv . . . 1397
wds_DHCPLeaseStateTlv . . . 1398
wds_DHCPOpt . . . 1398
wds_DHCPProfileIdTlv . . . 1399
wds_DHCPv4HWConfig . . . 1399
wds_DHCPv4Option . . . 1400
wds_DHCPv4OptionList . . . 1401
wds_DHCPv4ProfileId . . . 1401
wds_Domain . . . 1402
wds_DomainNameList . . . 1402
wds_GPRSQoS . . . 1403
wds_IPv4AdTlv . . . 1404
wds_IPV6AddressInfo . . . 1404
wds_IPV6GWAddressInfo . . . 1404
wds_PCSCFFQDNAddress . . . 1405
wds_PCSCFFQDNAddressList . . . 1406
wds_PCSCFIPv4ServerAddressList . . . 1406
wds_ProfileIdentifier . . . 1407
wds_profileInfo . . . 1407
wds_transferStatInd . . . 1408
wds_TrStatInd . . . 1408
wds_UMTSMinQoS . . . 1409
WdsByteTotals . . . 1411
WdsByteTotalsElmnts . . . 1412
WdsClientLeaseChange . . . 1413
WdsConnectionRate . . . 1413
WdsConnectionRateElmnts . . . 1414
WdsDHCPv4ClientLeaseInd . . . 1415

WdsDHCPv4Config 1416
WdsDHCPv4HWConfig 1416
wdsDhcpv4HwConfig 1417
wdsDhcpv4Option 1417
WdsDHCPv4Option 1418
wdsDhcpv4OptionList 1419
WdsDHCPv4OptionList 1419
WdsDHCPv4ProfileId 1420
wdsDhcpv4ProfileId 1420
WDSGetLoopbackData 1421
WdsIpAddressInfoReq 1421
WdsPktStatisticsElmnts 1422
WdsPktStatisticsReq 1424
WdsPktStatisticsResp 1424
WdsProfileParam 1425
WdsRunTimeSettings 1426
wdsSetEventReportReq 1426
WDSSetLoopbackData 1428
WDSSWICurrentChannelRates 1429

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 . . . 1431
common.h . . . 1431
dms.h . . . 1436
fms.h . . . 1488
loc.h . . . 1491
nas.h . . . 1508
pds.h . . . 1557
qaCbkCatEventReportInd.h . . . 1578
qaCbkSwiOmaDmEventReportInd.h . . . 1579
qaGobiApiAudio.h
Audio Service API function prototypes . . . 1580
qaGobiApiCat.h
Card Application Toolkit API function headers . . . 1584
qaGobiApiCbk.h
Callback Service API function prototypes . . . 1585
qaGobiApiDcs.h
Device Connectivity Service API function prototypes . . . 1673
qaGobiApiDms.h
Device Management Service API function prototypes . . . 1681
qaGobiApiFms.h
Firmware Management Service API function prototypes . . . 1711
qaGobiApiIms.h
IMS Service API function prototypes . . . 1730
qaGobiApiImsa.h
IMSA Service API function prototypes . . . 1737
qaGobiApiLoc.h
Location API function prototypes . . . 1740
qaGobiApiNas.h
Network Access Service API function prototypes . . . 1747
qaGobiApiOmadm.h
Open Mobile Alliance Device Management Service API function prototypes . . . 1794
qaGobiApiPds.h
Position Determination Service API function prototypes . . . 1797
qaGobiApiQos.h
Quality of Service API function prototypes . . . 1812

qaGobiApiRms.h
Remote Management Service API function prototypes . . . 1818
qaGobiApiSar.h
Specific Absorption Rate API function prototypes . . . 1820
qaGobiApiSms.h
Short Message Service API function prototypes . . . 1822
qaGobiApiSwi.h
SWI API function prototypes . . . 1845
qaGobiApiSwiAudio.h
M2M Audio Service API function prototypes . . . 1846
qaGobiApiSwiAvms.h
SWI Open Mobile Alliance Device Management Service API function prototypes SWI AVMS QMI Service revision 1.14 . . . 1852
qaGobiApiSwiOmadms.h
SWI Open Mobile Alliance Device Management Service API function prototypes SWI OMA-DM QMI Service revision 1.6 . . . 1864
qaGobiApiTableBandClasses.h
Network Access Service API Band Classes table . . . 1875
qaGobiApiTableCallControlReturnReasons.h
Call Control Return Reasons table . . . 1878
qaGobiApiTableCallEndReasons.h
Wireless Data Service Call End Reasons . . . 1879
qaGobiApiTableCarrierCodes.h
Carrier Codes table . . . 1895
qaGobiApiTableCodingScheme.h
Data Coding Scheme . . . 1897
qaGobiApiTableGpsCapabilityCodes.h
Position Determination Service API GPS Capability Codes . . . 1900
qaGobiApiTablePowerModes.h
Device Management Service API Power Modes table . . . 1901
qaGobiApiTableRadioInterfaces.h
Network Access Service API Radio Interfaces table . . . 1901
qaGobiApiTableRegionCodes.h
Region Codes table . . . 1902
qaGobiApiTableServiceOptions.h
Voice Service Options . . . 1902
qaGobiApiTableSupServiceInfoClasses.h
Voice Supplementary Service Information Classes . . . 1905
qaGobiApiTableSwiAudio.h
Swi Audio related tables . . . 1905
qaGobiApiTableSwiOMADMSessionStatus.h
Session Status table . . . 1906
qaGobiApiTableSwiOMADMUpdateCompleteStatus.h
Update Complete Status table . . . 1907
qaGobiApiTableVoiceCallEndReasons.h
Voice Service Call and supplementary services end reasons . . . 1908
qaGobiApiTmd.h
Thermal Mitigation Device API function prototypes . . . 1915
qaGobiApiUim.h
Uim Service API function prototypes . . . 1917
qaGobiApiVoice.h
Voice Service API function prototypes . . . 1929
qaGobiApiWds.h
Wireless Data Service API function prototypes . . . 1950
qaNasGetRFBandInfo.h . . . 1993
qaNasPerformNetworkScan.h . . . 1994
qmerrno.h . . . 1995
qos.h . . . 2002

rms.h 2010
sar.h 2012
sms.h 2014
SwiDataTypes.h
SWI data types 2034
swiloc.h 2036
swioma.h 2037
SWIWWANCMAPI.h 2046
uim.h 2046
voice.h 2065
wds.h 2092

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

pRegTrans-LayerInfoEvt	- • Optional 1 BYTE parameter indicating registration status of transport layer information events • Values: – 0x00 - Disabled – 0x01 - Enabled – NULL - No change - specifying NULL indicates that the device will continue to use the existing setting (disable/enable) which has been previously set for the device
pRegTransNW-RegInfoEvt	- • Optional 1 BYTE parameter indicating registration status of transport network registration information events • Values: – 0x00 - Disabled – 0x01 - Enabled – NULL - No change - specifying NULL indicates that the device will continue to use the existing setting (disable/enable) which has been previously set for the device
pRegCallStat-InfoEvt	- • Optional 1 BYTE parameter indicating registration status of call status information events • Values: – 0x00 - Disabled – 0x01 - Enabled – NULL - No change - specifying NULL indicates that the device will continue to use the existing setting (disable/enable) which has been previously set for the device

8.25.2 Field Documentation

8.25.2.1 BYTE* _setIndicationRegReq::pRegCallStatInfoEvt

8.25.2.2 BYTE* _setIndicationRegReq::pRegTransLayerInfoEvt

8.25.2.3 BYTE* _setIndicationRegReq::pRegTransNWRegInfoEvt

8.26 _slqs3GPPConfigItem Struct Reference

Data Fields

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

8.26.1 Detailed Description

This structure contains the 3gpp Configuration Item information.

Parameters

pLTEAttach-Profile	• LTE Attach Profile – points to a single WORD Value indicating the attached LTE Profile – Optional parameter with possible values 1-16 (EM/MC73xx or earlier) – function SLQSGet3GPPConfigItem() returns a default value 255 if no LTE Attach Profile is configured • This setting is deprecated on MC/EM74xx
pProfileList	• Profile List – an array of 4 profile configurations – Each element points to a single WORD value indicating profile – Optional parameter with possible values * 1 - 16 (MC/EM73xx and before) * 1 - 24 (MC/EM74xx and onwards) – function SLQSGet3GPPConfigItem() returns a default value 255 if no 3gpp configuration is present
pDefaultPDN-Enabled	• Always Connect Default PDN – A single BYTE value indicating the status of Always connect default PDN * 0 - disabled * 1 - enabled – Optional parameter – function SLQSGet3GPPConfigItem() returns a default value 255 if no 3gpp configuration is present
p3gppRelease	• 3gpp release – A single BYTE value indicating the 3gpp release * 0 - Release 99 * 1 - Release 5 * 2 - Release 6 * 3 - Release 7 * 4 - Release 8 – In 9x30 and onwards * 5 - Release 9 * 6 - Release 10 * 7 - Release 11 – Optional parameter – function SLQSGet3GPPConfigItem() returns a default value 255 if no 3gpp configuration is present

pLTEAttach-ProfileList	• pointer to WORD array indicating LTE Attach Profile List – Optional parameter – possible values: 1-24 – This setting is only supported for MC/EM74xx onwards – The new equivalent option for "pLTEAttachProfile" on 74xx modems is "pLTEAttachProfile-List". Please provide attach profiles in order of decreasing priority in this list.
LTEAttach-ProfileListLen	• Number of element in pLTEAttachProfileList – valid range: 1-24 * This setting is only supported for MC/EM74xx onwards

8.26.2 Field Documentation

8.26.2.1 WORD _slqs3GPPConfigItem::LTEAttachProfileListLen

8.26.2.2 BYTE* _slqs3GPPConfigItem::p3gppRelease

8.26.2.3 BYTE* _slqs3GPPConfigItem::pDefaultPDNEnabled

8.26.2.4 WORD* _slqs3GPPConfigItem::pLTEAttachProfile

8.26.2.5 WORD* _slqs3GPPConfigItem::pLTEAttachProfileList

8.26.2.6 WORD* _slqs3GPPConfigItem::pProfileList

8.27 _SLQSAVMSSetSettingsReqPara Struct Reference

Data Fields

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

8.27.1 Detailed Description

Sets the AVMS settings requested.

Parameters

bAutoConnect[IN]	• Auto Connect to AirVantage server – 0x00 - FALSE – 0x01 - TRUE
bAutoReboot[IN]	• Automatic device reboot when the request is received from AirVantage server – 0x00 - FALSE – 0x01 - TRUE
bFwPromptdownload[IN]	• Start Firmware Download promptly/automatically – 0x00 - FALSE – 0x01 - TRUE
bFwPromptUpdate[IN]	• Start Firmware Download promptly/automatically – 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::pFwAutoSDM

8.27.2.6 BYTE* _SLQSAVMSSetSettingsReqPara::pFwPromptdownload

8.27.2.7 BYTE* _SLQSAVMSSetSettingsReqPara::pFwPromptUpdate

8.27.2.8 BYTE* _SLQSAVMSSetSettingsReqPara::pNotifStore

8.27.2.9 QmiSwiAvmsSetSettingsPeriodInfo* _SLQSAVMSSetSettingsReqPara::pPeroidInfo

8.27.2.10 ULONG* _SLQSAVMSSetSettingsReqPara::pPollingTimer

8.28 _SLQSAVMSSettings Struct Reference

Data Fields

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

8.28.1 Field Documentation

8.28.1.1 SLQSAVMSSettingsAPNInfo* _SLQSAVMSSettings::pAPNInfo

8.28.1.2 BYTE* _SLQSAVMSSettings::pAutoConnect

8.28.1.3 BYTE* _SLQSAVMSSettings::pAutoReboot

8.28.1.4 SLQSAVMSSettingsConnectionRetryTimers* _SLQSAVMSSettings::pConnectionRetryTimers

8.28.1.5 BYTE* _SLQSAVMSSettings::pFwAutoSDM

8.28.1.6 BYTE* _SLQSAVMSSettings::pFwPromptdownload

8.28.1.7 BYTE* _SLQSAVMSSettings::pFwPromptUpdate

8.28.1.8 BYTE* _SLQSAVMSSettings::pNotificationStore

8.28.1.9 ULONG* _SLQSAVMSSettings::pOMADMEnabled

8.28.1.10 SLQSAVMSSettingsPeriodsInfo* _SLQSAVMSSettings::pPeroidsInfo

8.28.1.11 ULONG* _SLQSAVMSSettings::pPollingTimer

8.29 _SLQSAVMSSettingsAPNInfo Struct Reference

Data Fields

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

8.29.1 Field Documentation

8.29.1.1 BYTE _SLQSAVMSSettingsAPNInfo::bAPNLength

8.29.1.2 BYTE _SLQSAVMSSettingsAPNInfo::bPWDLength

8.29.1.3 BYTE _SLQSAVMSSettingsAPNInfo::bUnameLength

8.29.1.4 CHAR* _SLQSAVMSSettingsAPNInfo::pAPN

8.29.1.5 CHAR* _SLQSAVMSSettingsAPNInfo::pPWD

8.29.1.6 CHAR* _SLQSAVMSSettingsAPNInfo::pUname

8.30 _SLQSAVMSSettingsConnectionRetryTimers Struct Reference

Data Fields

- WORD Timers [8]

8.30.1 Field Documentation

8.30.1.1 WORD _SLQSAVMSSettingsConnectionRetryTimers::Timers[8]

8.31 _SLQSAVMSSettingsPeriodsInfo Struct Reference

Data Fields

- ULONG min
- ULONG max

8.31.1 Field Documentation

8.31.1.1 ULONG _SLQSAVMSSettingsPeriodsInfo::max

8.31.1.2 ULONG _SLQSAVMSSettingsPeriodsInfo::min

8.32 _SlqsNas3GppNetworkRAT_ Struct Reference

Data Fields

- WORD MCC
- WORD MNC
- BYTE RAT

8.32.1 Detailed Description

Contain the 3GPP radio access technology information.

Parameters

MCC	• Mobile Country Code
MNC	• Mobile Network Code
RAT	• Radio Access Technology – 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 cdmaMsgDecodingParamsExt Struct Reference

Data Fields

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

8.110.1 Detailed Description

Structure contains parameters which need to be decoded from message

Parameters

messageLength[IN]	• Length of the message to be decoded in bytes
pMessage[IN]	• Message read off the device via GetSMS
pSenderAddrLength[IN/OUT]	• Upon input, indicates the maximum number of ASCII characters (including NULL termination) that the pSenderAddr buffer can accommodate. Note that a length of 14 is reasonable.Upon successful output, returns the length of originating address string (including the NULL termination)
pSenderAddr[OUT]	• Returns NULL-terminated ASCII String containing the originating address.International number will be prepended with a '+'character
pTextMsgLength[IN/OUT]	• Upon input, specifies the number of UCS2 characters the given text message buffer can accommodate.Upon successful output, returns the number of UCS2 characters returns in the given text messagebuffer(including NULL-terminator)
pTextMsg[OUT]	• Returns the text message as NULL-terminated UCS2 string
pPriority[OUT]	(optional parameter) • 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

pRelative-Validity[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
pUser-Acknowledgement-Req[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
pRead-Acknowledgement-Req[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[O-UT]	(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
pCallbkAddr-Length[OUT]	(optional parameter) • returns the length of Callback address string (including the NULL termination)
pCallbkAddr[OU-T]	(optional parameter) • returns NULL-terminated ASCII String containing callback address String containing the Call Back number with a 32 maximum characters.
pReference-Num[OUT]	• Reference number of the sms
pTotalNum[OUT]	• Total number of the concatenated message
pPartNum[OUT]	• Sequence number of the current message
pIsUDHPresent	• Is User Data Header Present in the PDU? If yes, it means it is a • concatenated SMS.

8.110.2 Field Documentation

8.110.2.1 BYTE cdmaMsgDecodingParamsExt::absoluteValidity[0x08]

8.110.2.2 BYTE cdmaMsgDecodingParamsExt::mcTimeStamp[0x08]

8.110.2.3 ULONG cdmaMsgDecodingParamsExt::messageLength

8.110.2.4 BYTE* cdmaMsgDecodingParamsExt::pAlertPriority

8.110.2.5 CHAR* cdmaMsgDecodingParamsExt::pCallbkAddr

8.110.2.6 BYTE* cdmaMsgDecodingParamsExt::pCallbkAddrLength

8.110.2.7 BYTE* cdmaMsgDecodingParamsExt::pDisplayMode

8.110.2.8 BOOL* cdmaMsgDecodingParamsExt::pIsUDHPresent

8.110.2.9 BYTE* cdmaMsgDecodingParamsExt::pLanguage

8.110.2.10 BYTE* cdmaMsgDecodingParamsExt::pMessage

8.110.2.11 ULONG* cdmaMsgDecodingParamsExt::pMessageID

8.110.2.12 BYTE* cdmaMsgDecodingParamsExt::pPartNum

8.110.2.13 BYTE* cdmaMsgDecodingParamsExt::pPriority

8.110.2.14 BYTE* cdmaMsgDecodingParamsExt::pPrivacy

8.110.2.15 BOOL* cdmaMsgDecodingParamsExt::pReadAcknowledgementReq

8.110.2.16 BYTE* cdmaMsgDecodingParamsExt::pReferenceNum

8.110.2.17 BYTE* cdmaMsgDecodingParamsExt::pRelativeValidity

8.110.2.18 CHAR* cdmaMsgDecodingParamsExt::pSenderAddr

8.110.2.19 BYTE* cdmaMsgDecodingParamsExt::pSenderAddrLength

8.110.2.20 WORD* cdmaMsgDecodingParamsExt::pTextMsg

8.110.2.21 BYTE* cdmaMsgDecodingParamsExt::pTextMsgLength

8.110.2.22 BYTE* cdmaMsgDecodingParamsExt::pTotalNum

8.110.2.23 BOOL* cdmaMsgDecodingParamsExt::pUserAcknowledgementReq

8.111 cdmaMsgEncodingParams Struct Reference

Data Fields

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

8.111.1 Detailed Description

Structure contains parameters for message to be encoded

Parameters

pMessageSize[IN/OUT]	• Upon input, specifies the total number of bytes that the given pMessage buffer can hold (a buffer of length 240 is recommended).Upon successful output, specifies the length of the constructed message placed in the pMessage buffer (in bytes)
pMessage[OUT]	- The constructed raw message
messageId[IN]	• The message reference number for this message. This value should be incremented for every message the host application sends
pDestAddr[IN]	• Gives NULL-terminated ASCII String containing a destination address.International number will be prepended with a '+' character
pCallbackAddr[IN]	• Gives NULL-terminated ASCII String containing a callback address.International number will be prepended with a '+' character
textMsgLength[IN]	• Number of UCS2 characters in the text message(excluding NULL)
pTextMsg[IN]	• Text message to be encoded
pPriority[IN](optional	parameter) • 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.

8.111.2 Field Documentation

8.111.2.1 BYTE cdmaMsgEncodingParams::messageId

8.111.2.2 CHAR* cdmaMsgEncodingParams::pCallbackAddr

8.111.2.3 CHAR* cdmaMsgEncodingParams::pDestAddr

8.111.2.4 BYTE* cdmaMsgEncodingParams::pEncodingAlphabet

8.111.2.5 BYTE* cdmaMsgEncodingParams::pMessage

8.111.2.6 BYTE* cdmaMsgEncodingParams::pMessageSize

8.111.2.7 BYTE* cdmaMsgEncodingParams::pPriority

8.111.2.8 BYTE* cdmaMsgEncodingParams::pRelValidity

8.111.2.9 WORD* cdmaMsgEncodingParams::pTextMsg

8.111.2.10 ULONG cdmaMsgEncodingParams::textMsgLength

8.112 CDMARSSIThresh Struct Reference

Data Fields

- BYTE CDMARSSIThreshListLen
- WORD * pCDMARSSIThreshList

8.112.1 Detailed Description

This structure contains CDMA RSSI threshold related parameters.

Parameters

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

8.112.2 Field Documentation

8.112.2.1 BYTE CDMARSSIThresh::CDMARSSIThreshListLen

8.112.2.2 WORD* CDMARSSIThresh::pCDMARSSIThreshList

8.113 CDMASSInfo Struct Reference

Data Fields

- INT8 rssi
- SHORT ecio

8.113.1 Detailed Description

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

Parameters

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

8.113.2 Field Documentation

8.113.2.1 SHORT CDMASSInfo::ecio

8.113.2.2 INT8 CDMASSInfo::rssi

8.114 cdmaSSInfo Struct Reference

Data Fields

- int8_t rssi
- int16_t ecio

8.114.1 Detailed Description

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

Parameters

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

8.114.2 Field Documentation

8.114.2.1 int16_t cdmaSSInfo::ecio

8.114.2.2 int8_t cdmaSSInfo::rssi

8.115 CDMASysInfo Struct Reference

Data Fields

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

8.115.1 Detailed Description

Structure for storing the CDMA System Information.

Parameters

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

8.115.2 Field Documentation

8.115.2.1 WORD CDMASysInfo::baseId

8.115.2.2 ULONG CDMASysInfo::baseLat

8.115.2.3 ULONG CDMASysInfo::baseLong

8.115.2.4 BYTE CDMASysInfo::bsInfoValid

8.115.2.5 BYTE CDMASysInfo::bsPRev

8.115.2.6 BYTE CDMASysInfo::bsPRevValid

8.115.2.7 BYTE CDMASysInfo::ccsSupported

8.115.2.8 BYTE CDMASysInfo::ccsSupportedValid

8.115.2.9 BYTE CDMASysInfo::cdmaSysIdValid

8.115.2.10 BYTE CDMASysInfo::isSysPrlMatch

8.115.2.11 BYTE CDMASysInfo::isSysPrlMatchValid

8.115.2.12 BYTE CDMASysInfo::MCC[3]

8.115.2.13 BYTE CDMASysInfo::MNC[3]

8.115.2.14 WORD CDMASysInfo::networkID

8.115.2.15 BYTE CDMASysInfo::networkIdValid

8.115.2.16 WORD CDMASysInfo::packetZone

8.115.2.17 BYTE CDMASysInfo::packetZoneValid

8.115.2.18 BYTE CDMASysInfo::pRevInUse

8.115.2.19 BYTE CDMASysInfo::pRevInUseValid

8.115.2.20 sysInfoCommon CDMASysInfo::sysInfoCDMA

8.115.2.21 WORD CDMASysInfo::systemID

8.116 CDMASysInfoExt Struct Reference

Data Fields

- WORD MCC
- BYTE imsi_11_12

8.116.1 Detailed Description

This structure contains CDMA system information extension

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

Parameters

MCC	• Mobile Country Code
imsi_11_12	• IMSI_11_12

8.116.2 Field Documentation

8.116.2.1 BYTE CDMASysInfoExt::imsi_11_12

8.116.2.2 WORD CDMASysInfoExt::MCC

8.117 CellDb Struct Reference

Data Fields

- ULONG mask

8.117.1 Detailed Description

This structure contains the cell database

Parameters

mask	• Mask for the cell database assistance data that is to be deleted • 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

8.117.2 Field Documentation

8.117.2.1 ULONG CellDb::mask

8.118 cellParams Struct Reference

Data Fields

- WORD pci
- SHORT rsrq

- SHORT rsrp
- SHORT rssi
- SHORT srxlev

8.118.1 Detailed Description

This structure contains information about the Cell parameters.

Parameters

pci	• Physical cell ID. • 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.

8.118.2 Field Documentation

8.118.2.1 WORD cellParams::pci

8.118.2.2 SHORT cellParams::rsrp

8.118.2.3 SHORT cellParams::rsrq

8.118.2.4 SHORT cellParams::rssi

8.118.2.5 SHORT cellParams::srxlev

8.119 changeUIMPIN Struct Reference

Data Fields

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

8.119.1 Detailed Description

This structure contains the information about the pin parameters that need to be verified.

Parameters

pinID	• Indicates the PIN ID to be changed. – 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.

8.119.2 Field Documentation

8.119.2.1 BYTE changeUIMPIN::oldPINLen

8.119.2.2 BYTE changeUIMPIN::oldPINVal[255]

8.119.2.3 BYTE changeUIMPIN::pinID

8.119.2.4 BYTE changeUIMPIN::pinLen

8.119.2.5 BYTE changeUIMPIN::pinVal[255]

8.120 channelRate Struct Reference

Data Fields

- ULONG CurrChanTxRate
- ULONG CurrChanRxRate

8.120.1 Detailed Description

This structure contains Channel Rate

Parameters

CurrChanTxRate	• Max channel Tx rate in bits per second
CurrChanRx-Rate	• Max channel Rx rate in bits per second

8.120.2 Field Documentation

8.120.2.1 ULONG channelRate::CurrChanRxRate

8.120.2.2 ULONG channelRate::CurrChanTxRate

8.121 ChannelRate Struct Reference

Data Fields

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

8.121.1 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

8.121.2 Field Documentation

8.121.2.1 ULONG ChannelRate::CurrChanRxRate

8.121.2.2 ULONG ChannelRate::CurrChanTxRate

8.121.2.3 ULONG ChannelRate::MaxChanRxRate

8.121.2.4 ULONG ChannelRate::MaxChanTxRate

8.122 CLIPResp Struct Reference

Data Fields

- BYTE ActiveStatus
- BYTE ProvisionStatus

8.122.1 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

8.122.2 Field Documentation

8.122.2.1 BYTE CLIPResp::ActiveStatus

8.122.2.2 BYTE CLIPResp::ProvisionStatus

8.123 CLIRResp Struct Reference

Data Fields

- BYTE ActiveStatus
- BYTE ProvisionStatus

8.123.1 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

8.123.2 Field Documentation

8.123.2.1 BYTE CLIRResp::ActiveStatus

8.123.2.2 BYTE CLIRResp::ProvisionStatus

8.124 ClkInfo Struct Reference

Data Fields

- ULONG mask

8.124.1 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

8.124.2 Field Documentation

8.124.2.1 ULONG ClkInfo::mask

8.125 CNAPResp Struct Reference

Data Fields

- BYTE ActiveStatus
- BYTE ProvisionStatus

8.125.1 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

8.125.2 Field Documentation

8.125.2.1 BYTE CNAPResp::ActiveStatus

8.125.2.2 BYTE CNAPResp::ProvisionStatus

8.126 COLPResp Struct Reference

Data Fields

- BYTE ActiveStatus
- BYTE ProvisionStatus

8.126.1 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

8.126.2 Field Documentation

8.126.2.1 BYTE COLPResp::ActiveStatus

8.126.2.2 BYTE COLPResp::ProvisionStatus

8.127 COLRResp Struct Reference

Data Fields

- BYTE ActiveStatus
- BYTE ProvisionStatus

8.127.1 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

8.127.2 Field Documentation

8.127.2.1 BYTE COLRResp::ActiveStatus

8.127.2.2 BYTE COLRResp::ProvisionStatus

8.128 CommInfo Struct Reference

Data Fields

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

8.128.1 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

8.128.2 Field Documentation

8.128.2.1 BYTE CommInfo::imsRegState

8.128.2.2 BYTE CommInfo::modemMode

8.128.2.3 BYTE CommInfo::psState

8.128.2.4 BYTE CommInfo::systemMode

8.128.2.5 BYTE CommInfo::temperature

8.129 ConnectionStatus Struct Reference

Data Fields

- BYTE MDMConnStatus
- ULONGLONG 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 ULONGLONG ConnectionStatus::MDMCallDuration

8.129.2.2 BYTE ConnectionStatus::MDMConnStatus

8.130 connectionStatus Struct Reference

Data Fields

- uint8_t MDMConnStatus
- uint64_t MDMCallDuration

8.130.1 Detailed Description

This structure contains modem connection status

Parameters

MDMConn-Status	• Current link status – 0x01 - DISCONNECTED – 0x02 - CONNECTED – 0xff - Invalid data.
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). – 0xffffffffffffffff - Invalid data.

8.130.2 Field Documentation

8.130.2.1 uint64_t connectionStatus::MDMCallDuration

8.130.2.2 uint8_t connectionStatus::MDMConnStatus

8.131 connectNumInfo Struct Reference

Data Fields

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

8.131.1 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.

8.131.2 Field Documentation

8.131.2.1 BYTE connectNumInfo::callerID[81]

8.131.2.2 BYTE connectNumInfo::callerIDLen

8.131.2.3 BYTE connectNumInfo::numPlan

8.131.2.4 BYTE connectNumInfo::numPresInd

8.131.2.5 BYTE connectNumInfo::numType

8.131.2.6 BYTE connectNumInfo::screeningInd

8.132 CrashInfo Struct Reference

Data Fields

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

8.132.1 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

8.132.2 Field Documentation

8.132.2.1 ULONG CrashInfo::crashData

8.132.2.2 ULONG CrashInfo::crashId

8.132.2.3 WORD CrashInfo::crashStrLen

8.132.2.4 WORD CrashInfo::gcDumpStrLen

8.132.2.5 WORD CrashInfo::numCrashes

8.132.2.6 CHAR* CrashInfo::pCrashString

8.132.2.7 CHAR* CrashInfo::pGCDumpString

8.133 crashInfoParams Struct Reference

Data Fields

- uint8_t crashStatus
- crashInformation crashInfo

8.133.1 Detailed Description

This structure contains crash information parameters

Parameters

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

8.133.2 Field Documentation

8.133.2.1 crashInformation crashInfoParams::crashInfo

8.133.2.2 uint8_t crashInfoParams::crashStatus

8.134 CrashInfoParams Struct Reference

Data Fields

- BYTE * pDevCrashStatus
- CrashInfo * pCrashInfo

8.134.1 Detailed Description

This structure is used to store Crash Information

Parameters

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

8.134.2 Field Documentation

8.134.2.1 CrashInfo* CrashInfoParams::pCrashInfo

8.134.2.2 BYTE* CrashInfoParams::pDevCrashStatus

8.135 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]

8.135.1 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

8.135.2 Field Documentation

8.135.2.1 uint32_t crashInformation::crashData

8.135.2.2 uint32_t crashInformation::crashId

8.135.2.3 char crashInformation::crashString[255]

8.135.2.4 uint16_t crashInformation::crashStrlen

8.135.2.5 char crashInformation::gcdumpString[1024]

8.135.2.6 uint16_t crashInformation::gcdumpStrlen

8.135.2.7 uint16_t crashInformation::numCrashes

8.136 CreateProfileIn Struct Reference

Data Fields

- BYTE * pProfileID
- BYTE * pProfileType
- QmiProfileInfo curProfile

8.136.1 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

8.136.2 Field Documentation

8.136.2.1 QmiProfileInfo CreateProfileIn::curProfile

8.136.2.2 BYTE* CreateProfileIn::pProfileID

8.136.2.3 BYTE* CreateProfileIn::pProfileType

8.137 CreateProfileOut Struct Reference

Data Fields

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

8.137.1 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

8.137.2 Field Documentation

8.137.2.1 USHORT* CreateProfileOut::pExtErrorCode

8.137.2.2 BYTE* CreateProfileOut::pProfileIndex

8.137.2.3 BYTE* CreateProfileOut::pProfileType

8.138 CsgId Struct Reference

Data Fields

- ULONG csgId
- BYTE TlvPresent

8.138.1 Detailed Description

This structure contains csg ID.

Parameters

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

8.138.2 Field Documentation

8.138.2.1 ULONG CsgId::csgId

8.138.2.2 BYTE CsgId::TlvPresent

8.139 CSGID Struct Reference

Data Fields

- WORD mcc
- WORD mnc
- BYTE mncPcsDigits

- ULONG id
- BYTE rat

8.139.1 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

8.139.2 Field Documentation

8.139.2.1 ULONG CSGID::id

8.139.2.2 WORD CSGID::mcc

8.139.2.3 WORD CSGID::mnc

8.139.2.4 BYTE CSGID::mncPcsDigits

8.139.2.5 BYTE CSGID::rat

8.140 CUGInfo Struct Reference

Data Fields

- WORD CUGIndex
- BYTE SuppPrefCUG
- BYTE SuppOA

8.140.1 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

8.140.2 Field Documentation

8.140.2.1 WORD CUGInfo::CUGIndex

8.140.2.2 BYTE CUGInfo::SuppOA

8.140.2.3 BYTE CUGInfo::SuppPrefCUG

8.141 curAMRConfig Struct Reference

Data Fields

- BYTE gsmAmrStat
- BYTE wcdmaAmrStat

8.141.1 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

8.141.2 Field Documentation

8.141.2.1 BYTE curAMRConfig::gsmAmrStat

8.141.2.2 BYTE curAMRConfig::wcdmaAmrStat

8.142 CurrDataSysStat Struct Reference

Data Fields

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

8.142.1 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

8.142.2 Field Documentation

8.142.2.1 CurrNetworkInfo* CurrDataSysStat::pCurrNetworkInfo

8.142.2.2 BYTE* CurrDataSysStat::pNetworkInfoLen

8.142.2.3 BYTE* CurrDataSysStat::pPrefNetwork

8.143 currentCatEvent Union Reference

Data Fields

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

8.143.1 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

8.143.2 Field Documentation

8.143.2.1 struct CatAlPhaIdentifierTlv currentCatEvent::CatAlphaIdtfr

8.143.2.2 struct CatEndProactiveSessionTlv currentCatEvent::CatEndPS

8.143.2.3 struct CatEventListTlv currentCatEvent::CatEventLst

8.143.2.4 struct CatEventIDDataTlv currentCatEvent::CatEvIDData

8.143.2.5 struct CatRefreshTlv currentCatEvent::CatRefresh

8.144 CurrentImgList Struct Reference

Data Fields

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

8.144.1 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

8.144.2 Field Documentation

8.144.2.1 CHAR CurrentImgList::carrier[16]

8.144.2.2 CHAR CurrentImgList::fwvers[16]

8.144.2.3 BYTE CurrentImgList::numEntries

8.144.2.4 CurrImageInfo* CurrentImgList::pCurrImgInfo

8.144.2.5 CHAR CurrentImgList::pkgver[16]

8.144.2.6 CHAR CurrentImgList::priver[16]

8.145 currentPLMN Struct Reference

Data Fields

- WORD MCC

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

8.145.1 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

8.145.2 Field Documentation

8.145.2.1 WORD currentPLMN::MCC

8.145.2.2 WORD currentPLMN::MNC

8.145.2.3 BYTE currentPLMN::netDescr[255]

8.145.2.4 BYTE currentPLMN::netDescrLength

8.146 CurrImageInfo Struct Reference

Data Fields

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

8.146.1 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

8.146.2 Field Documentation

8.146.2.1 BYTE CurrImageInfo::buildID[255]

8.146.2.2 BYTE CurrImageInfo::buildIDLen

8.146.2.3 BYTE CurrImageInfo::imageType

8.146.2.4 BYTE CurrImageInfo::uniqueID[16]

8.147 CurrNetworkInfo Struct Reference

Data Fields

- BYTE NetworkType
- ULONG RATMask
- ULONG 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 BYTE CurrNetworkInfo::NetworkType

8.147.2.2 ULONG CurrNetworkInfo::RATMask

8.147.2.3 ULONG CurrNetworkInfo::SOMask

8.148 currNetworkInfo Struct Reference

Data Fields

- uint8_t NetworkType
- uint32_t RATMask
- uint32_t SOMask

8.148.1 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

8.148.2 Field Documentation

8.148.2.1 uint8_t currNetworkInfo::NetworkType

8.148.2.2 uint32_t currNetworkInfo::RATMask

8.148.2.3 uint32_t currNetworkInfo::SOMask

8.149 custFeaturesInfo Struct Reference

Data Fields

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

8.149.1 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

8.149.2 Field Documentation

8.149.2.1 ULONG custFeaturesInfo::GpsEnable

8.149.2.2 BYTE* custFeaturesInfo::pDHCPRelayEnabled

8.149.2.3 BYTE* custFeaturesInfo::pDisableIMSI

8.149.2.4 BYTE* custFeaturesInfo::pGPSLPM

8.149.2.5 BYTE* custFeaturesInfo::pGPSSel

8.149.2.6 WORD* custFeaturesInfo::pIPFamSupport

8.149.2.7 BYTE* custFeaturesInfo::pIsVoiceEnabled

8.149.2.8 BYTE* custFeaturesInfo::pRMAutoConnect

8.149.2.9 BYTE* custFeaturesInfo::pSMSSupport

8.150 custFeaturesSetting Struct Reference

Data Fields

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

8.150.1 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

8.150.2 Field Documentation

8.150.2.1 BYTE* custFeaturesSetting::pDHCPRelayEnabled

8.150.2.2 ULONG* custFeaturesSetting::pGPSEnable

8.150.2.3 BYTE* custFeaturesSetting::pGPSLPM

8.150.2.4 BYTE* custFeaturesSetting::pGPSSel

8.150.2.5 BYTE* custFeaturesSetting::pIsVoiceEnabled

8.151 custSettingInfo Struct Reference

Data Fields

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

8.151.1 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

8.151.2 Field Documentation

8.151.2.1 WORD custSettingInfo::cust_attr

8.151.2.2 CHAR custSettingInfo::cust_id[64+1]

8.151.2.3 BYTE custSettingInfo::cust_value[8+1]

8.151.2.4 WORD custSettingInfo::id_length

8.151.2.5 WORD custSettingInfo::value_length

8.152 custSettingList Struct Reference

Data Fields

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

8.152.1 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

8.152.2 Field Documentation

8.152.2.1 custSettingInfo custSettingList::custSetting[256]

8.152.2.2 BYTE custSettingList::list_type

8.152.2.3 WORD custSettingList::num_instances

8.153 dataBearers Struct Reference

Data Fields

- BYTE dataBearerMask
- QmiWDSDataBearerTechnology * pCurDataBearerTechnology
- QmiWDSDataBearerTechnology * pLastCallDataBearerTechnology

8.153.1 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

8.153.2 Field Documentation

8.153.2.1 BYTE dataBearers::dataBearerMask

8.153.2.2 QmiWDSDataBearerTechnology* dataBearers::pCurDataBearerTechnology

8.153.2.3 QmiWDSDataBearerTechnology* dataBearers::pLastCallDataBearerTechnology

8.154 DataBearerTech Struct Reference

Data Fields

- ULONG techType
- ULONG ratValue
- ULONGLONG soMask

8.154.1 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

8.154.2 Field Documentation

8.154.2.1 ULONG DataBearerTech::ratValue

8.154.2.2 ULONGLONG DataBearerTech::soMask

8.154.2.3 ULONG DataBearerTech::techType

8.155 DataBearerTechExt Struct Reference

Data Fields

- DataBearerTech * pBearerTech
- DataBearerTech * pLastBearerTech

8.155.1 Detailed Description

Data Bearer Technology Ext

Parameters

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

8.155.2 Field Documentation

8.155.2.1 DataBearerTech* DataBearerTechExt::pBearerTech

8.155.2.2 DataBearerTech* DataBearerTechExt::pLastBearerTech

8.156 dataBearerTechnology Struct Reference

Data Fields

- BYTE currentNetwork
- ULONG ratMask
- ULONG soMask

8.156.1 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

8.156.2 Field Documentation

8.156.2.1 BYTE dataBearerTechnology::currentNetwork

8.156.2.2 ULONG dataBearerTechnology::ratMask

8.156.2.3 ULONG dataBearerTechnology::soMask

8.157 dataRate Struct Reference

Data Fields

- ULONG dataRateMax
- ULONG guaranteedRate

8.157.1 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)

8.157.2 Field Documentation

8.157.2.1 ULONG dataRate::dataRateMax

8.157.2.2 ULONG dataRate::guaranteedRate

8.158 dataSrvCapabilities Struct Reference

Data Fields

- BYTE dataCapabilitiesLen
- BYTE dataCapabilities [0x20]

8.158.1 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+

8.158.2 Field Documentation

8.158.2.1 BYTE dataSrvCapabilities::dataCapabilities[0x20]

8.158.2.2 BYTE dataSrvCapabilities::dataCapabilitiesLen

8.159 DataStatusDetail Struct Reference

Data Fields

- ULONG IPAddress
- BYTE LastErrCode

8.159.1 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

8.159.2 Field Documentation

8.159.2.1 ULONG DataStatusDetail::IPAddress

8.159.2.2 BYTE DataStatusDetail::LastErrCode

8.160 DataULongLongTlv Struct Reference

Data Fields

- BYTE TlvPresent
- ULONGLONG ullData

8.160.1 Field Documentation

8.160.1.1 BYTE DataULongLongTlv::TlvPresent

8.160.1.2 ULONGLONG DataULongLongTlv::ullData

8.161 DataULongTlv Struct Reference

Data Fields

- BYTE TlvPresent
- ULONG ulData

8.161.1 Field Documentation

8.161.1.1 BYTE DataULongTlv::TlvPresent

8.161.1.2 ULONG DataULongTlv::ulData

8.162 DcsUsbPortNames Struct Reference

Data Fields

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

8.162.1 Field Documentation

8.162.1.1 CHAR DcsUsbPortNames::AtCmdPort[32]

8.162.1.2 CHAR DcsUsbPortNames::DmPort[32]

8.162.1.3 CHAR DcsUsbPortNames::NmeaPort[32]

8.163 delAssistDataStatus Struct Reference

Data Fields

- ULONG status

8.163.1 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

8.163.2 Field Documentation

8.163.2.1 ULONG delAssistDataStatus::status

8.164 depersonalizationInformation Struct Reference

Data Fields

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

8.164.1 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.

8.164.2 Field Documentation

8.164.2.1 BYTE depersonalizationInformation::ckLen

8.164.2.2 BYTE depersonalizationInformation::ckVal[255]

8.164.2.3 BYTE depersonalizationInformation::feature

8.164.2.4 BYTE depersonalizationInformation::operation

8.165 detailSvcInfo Struct Reference

Data Fields

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

8.165.1 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

8.165.2 Field Documentation

8.165.2.1 BYTE detailSvcInfo::hdrHybrid

8.165.2.2 BYTE detailSvcInfo::hdrSrvStatus

8.165.2.3 BYTE detailSvcInfo::isSysForbidden

8.165.2.4 BYTE detailSvcInfo::srvCapability

8.165.2.5 BYTE detailSvcInfo::srvStatus

8.166 DeviceConfigDetail Struct Reference

Data Fields

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

8.166.1 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

8.166.2 Field Documentation

8.166.2.1 BYTE DeviceConfigDetail::Chipset

8.166.2.2 BYTE DeviceConfigDetail::HWVersion

8.166.2.3 BYTE DeviceConfigDetail::QLIC

8.166.2.4 BYTE DeviceConfigDetail::Technology

8.167 DHCPOption Struct Reference

Data Fields

- BYTE optCode
- BYTE optValLen
- BYTE * pOptVal

8.167.1 Detailed Description

This structure contains DHCPv4 lease option values

Parameters

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

8.167.2 Field Documentation

8.167.2.1 BYTE DHCPOption::optCode

8.167.2.2 BYTE DHCPOption::optValLen

8.167.2.3 BYTE* DHCPOption::pOptVal

8.168 DHCPOptionList Struct Reference

Data Fields

- BYTE numOpt
- DHCPOption * pOptions

8.168.1 Detailed Description

This structure contains DHCPv4 lease option list

Parameters

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

8.168.2 Field Documentation

8.168.2.1 BYTE DHCPOptionList::numOpt

8.168.2.2 DHCPOption* DHCPOptionList::pOptions

8.169 diagInfo Struct Reference

Data Fields

- BYTE diagInfoLen
- BYTE diagnosticInfo [255]

8.169.1 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.

8.169.2 Field Documentation

8.169.2.1 BYTE diagInfo::diagInfoLen

8.169.2.2 BYTE diagInfo::diagnosticInfo[255]

8.170 dirNum Struct Reference

Data Fields

- BYTE dirNumLen
- BYTE dirNum [255]

8.170.1 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.

8.170.2 Field Documentation

8.170.2.1 BYTE dirNum::dirNum[255]

8.170.2.2 BYTE dirNum::dirNumLen

8.171 dms_ActivationStatusTlv Struct Reference

Data Fields

- uint16_t TlvPresent
- uint32_t activationStatus

8.171.1 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

8.171.2 Field Documentation

8.171.2.1 uint32_t dms_ActivationStatusTlv::activationStatus

8.171.2.2 uint16_t dms_ActivationStatusTlv::TlvPresent

8.172 dms_OperatingModeTlv Struct Reference

Data Fields

- uint16_t TlvPresent
- uint32_t operatingMode

8.172.1 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

8.172.2 Field Documentation

8.172.2.1 uint32_t dms_OperatingModeTlv::operatingMode

8.172.2.2 uint16_t dms_OperatingModeTlv::TlvPresent

8.173 dmsCurrentPRLInfo Struct Reference

Data Fields

- WORD * pPRLVersion
- BYTE * pPRLPreference

8.173.1 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

8.173.2 Field Documentation

8.173.2.1 BYTE* dmsCurrentPRLInfo::pPRLPreference

8.173.2.2 WORD* dmsCurrentPRLInfo::pPRLVersion

8.174 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

8.174.1 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

8.174.2 Field Documentation

8.174.2.1 uint16_t DMScustSettingInfo::cust_attr

8.174.2.2 uint8_t DMScustSettingInfo::cust_id[64+1]

8.174.2.3 uint8_t DMScustSettingInfo::cust_value[8+1]

8.174.2.4 uint16_t DMScustSettingInfo::id_length

8.174.2.5 uint16_t DMScustSettingInfo::value_length

8.175 DMScustSettingList Struct Reference

Data Fields

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

8.175.1 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

8.175.2 Field Documentation

8.175.2.1 DMScustSettingInfo DMScustSettingList::custSetting[255+1]

8.175.2.2 uint8_t DMScustSettingList::list_type

8.175.2.3 uint16_t DMScustSettingList::num_instances

8.176 DMSgetCustomFeatureV2 Struct Reference

Data Fields

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

8.176.1 Detailed Description

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

Parameters

pGetCustomInput[IN]	• See DMSgetCustomInput for more information
pCustSettingInfo[OUT]	• See DMScustSettingInfo for more information
pCustSettingList[OUT]	• See DMScustSettingList for more information

8.176.2 Field Documentation

8.176.2.1 DMScustSettingInfo∗ DMSgetCustomFeatureV2::pCustSettingInfo

8.176.2.2 DMScustSettingList∗ DMSgetCustomFeatureV2::pCustSettingList

8.176.2.3 DMSgetCustomInput∗ DMSgetCustomFeatureV2::pGetCustomInput

8.177 DMSgetCustomInput Struct Reference

Data Fields

- uint8_t cust_id [64+1]
- uint8_t list_type

8.177.1 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

8.177.2 Field Documentation

8.177.2.1 uint8_t DMSgetCustomInput::cust_id[64+1]

8.177.2.2 uint8_t DMSgetCustomInput::list_type

8.178 dmsIndicationRegisterReq Struct Reference

Data Fields

- BYTE * pSwiGetResetInd

8.178.1 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.

8.178.2 Field Documentation

8.178.2.1 BYTE* dmsIndicationRegisterReq::pSwiGetResetInd

8.179 dmsSwiGetResetInfo Struct Reference

Data Fields

- BYTE type
- BYTE source

8.179.1 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

8.179.2 Field Documentation

8.179.2.1 BYTE dmsSwiGetResetInfo::source

8.179.2.2 BYTE dmsSwiGetResetInfo::type

8.180 Domain Struct Reference

Data Fields

- WORD domainLen
- CHAR domainName [256]

8.180.1 Detailed Description

This structure contains the DomainName Information

Parameters

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

8.180.2 Field Documentation

8.180.2.1 WORD Domain::domainLen

8.180.2.2 CHAR Domain::domainName[256]

8.181 DomainNameList Struct Reference

Data Fields

- BYTE numInstances
- struct Domain domain [10]

8.181.1 Detailed Description

This structure contains the DomainNameList Information

Parameters

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

8.181.2 Field Documentation

8.181.2.1 struct Domain DomainNameList::domain[10]

8.181.2.2 BYTE DomainNameList::numInstances

8.182 DRCParams Struct Reference

Data Fields

- BYTE DRCValue
- BYTE DRCCover

8.182.1 Detailed Description

This structure contains Data Rate Channel parameters

Parameters

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

8.182.2 Field Documentation

8.182.2.1 BYTE DRCParams::DRCCover

8.182.2.2 BYTE DRCParams::DRCValue

8.183 DTMFInfo Struct Reference

Data Fields

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

8.183.1 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

8.183.2 Field Documentation

8.183.2.1 BYTE DTMFInfo::callID

8.183.2.2 BYTE DTMFInfo::digitBuff[255]

8.183.2.3 BYTE DTMFInfo::digitCnt

8.183.2.4 BYTE DTMFInfo::DTMFEvent

8.184 DTMFLengths Struct Reference

Data Fields

- BYTE DTMFPulseWidth
- BYTE DTMFInterdigitInterval

8.184.1 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

8.184.2 Field Documentation

8.184.2.1 BYTE DTMFLengths::DTMFInterdigitInterval

8.184.2.2 BYTE DTMFLengths::DTMFPulseWidth

8.185 DUNCallInfoInd Struct Reference

Data Fields

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

8.185.1 Field Documentation

8.185.1.1 WORD DUNCallInfoInd::CallEndReason

8.185.1.2 channelRate DUNCallInfoInd::ChannelRate

8.185.1.3 BYTE DUNCallInfoInd::DataBearerTech

8.185.1.4 BYTE DUNCallInfoInd::DormancyStatus

8.185.1.5 BYTE DUNCallInfoInd::MdmConnStatus

8.185.1.6 ULONGLONG DUNCallInfoInd::RXOKBytesCount

8.185.1.7 ULONGLONG DUNCallInfoInd::TXOKBytesCount

8.186 dunchannelRate Struct Reference

Data Fields

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

8.186.1 Detailed Description

This structure contains Channel Rate

Parameters

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

8.186.2 Field Documentation

8.186.2.1 uint32_t dunchannelRate::CurrChanRxRate

8.186.2.2 uint32_t dunchannelRate::CurrChanTxRate

8.186.2.3 uint32_t dunchannelRate::MaxChanRxRate

8.186.2.4 uint32_t dunchannelRate::MaxChanTxRate

8.187 ecioListElement Struct Reference

Data Fields

- SHORT ecio
- BYTE radioIf

8.187.1 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

8.187.2 Field Documentation

8.187.2.1 SHORT ecioListElement::ecio

8.187.2.2 BYTE ecioListElement::radioIf

8.188 ECIOThresh Struct Reference

Data Fields

- BYTE ECIOThresListLen
- SHORT * pECIOThresList

8.188.1 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

8.188.2 Field Documentation

8.188.2.1 BYTE ECIOThresh::ECIOThresListLen

8.188.2.2 SHORT* ECIOThresh::pECIOThresList

8.189 ECTNum Struct Reference

Data Fields

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

8.189.1 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

8.189.2 Field Documentation

8.189.2.1 BYTE ECTNum::ECTCallState

8.189.2.2 BYTE ECTNum::number[81]

8.189.2.3 BYTE ECTNum::presentationInd

8.190 encryptedPIN1 Struct Reference

Data Fields

- BYTE pin1Len
- BYTE pin1Val [255]

8.190.1 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.

8.190.2 Field Documentation

8.190.2.1 BYTE encryptedPIN1::pin1Len

8.190.2.2 BYTE encryptedPIN1::pin1Val[255]

8.191 eriDataparams Struct Reference

Data Fields

- uint16_t eriDataLen
- uint8_t eriData [1024]

8.191.1 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)

8.191.2 Field Documentation

8.191.2.1 uint8_t eriDataparams::eriData[1024]

8.191.2.2 uint16_t eriDataparams::eriDataLen

8.192 ERIFileparams Struct Reference

Data Fields

- WORD * pFileSize
- BYTE * pFile

8.192.1 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)

8.192.2 Field Documentation

8.192.2.1 BYTE* ERIFileparams::pFile

8.192.2.2 WORD* ERIFileparams::pFileSize

8.193 errorRateListElement Struct Reference

Data Fields

- USHORT errorRate
- BYTE radioIf

8.193.1 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

8.193.2 Field Documentation

8.193.2.1 USHORT errorRateListElement::errorRate

8.193.2.2 BYTE errorRateListElement::radioIf

8.194 eTWSPLMNInfoTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- sMSEtwsPlmnInfo ETWSPLMNInfo

8.194.1 Detailed Description

This structure contains ETWS PLMN TLV information

Parameters

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

8.194.2 Field Documentation

8.194.2.1 sMSEtwsPlmnInfo eTWSPLMNInfoTlv::ETWSPLMNInfo

8.194.2.2 uint8_t eTWSPLMNInfoTlv::TlvPresent

8.195 extDispRecInfo Struct Reference

Data Fields

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

8.195.1 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.

8.195.2 Field Documentation

8.195.2.1 BYTE extDispRecInfo::dispType

8.195.2.2 BYTE extDispRecInfo::extDispInfo[255]

8.195.2.3 BYTE extDispRecInfo::extDispInfoLen

8.196 FactorySequenceNumber Struct Reference

Data Fields

- BYTE FSNumber [255]

8.196.1 Detailed Description

This structure used to store Factory Sequence Number parameter

Parameters

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

8.196.2 Field Documentation

8.196.2.1 BYTE FactorySequenceNumber::FSNumber[255]

8.197 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]

8.197.1 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.

8.197.2 Field Documentation

8.197.2.1 WORD fileAttributes::fileID

8.197.2.2 WORD fileAttributes::fileSize

8.197.2.3 BYTE fileAttributes::fileType

8.197.2.4 WORD fileAttributes::rawLen

8.197.2.5 BYTE fileAttributes::rawValue[255]

8.197.2.6 WORD fileAttributes::recordCount

8.197.2.7 WORD fileAttributes::recordSize

8.197.2.8 BYTE fileAttributes::secActivate

8.197.2.9 WORD fileAttributes::secActivateMask

8.197.2.10 BYTE fileAttributes::secDeactivate

8.197.2.11 WORD fileAttributes::secDeactivateMask

8.197.2.12 BYTE fileAttributes::secIncrease

8.197.2.13 WORD fileAttributes::secIncreaseMask

8.197.2.14 BYTE fileAttributes::secRead

8.197.2.15 WORD fileAttributes::secReadMask

8.197.2.16 BYTE fileAttributes::secWrite

8.197.2.17 WORD fileAttributes::secWriteMask

8.198 fileInfo Struct Reference

Data Fields

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

8.198.1 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).

8.198.2 Field Documentation

8.198.2.1 WORD fileInfo::fileID

8.198.2.2 WORD fileInfo::path[255]

8.198.2.3 BYTE fileInfo::pathLen

8.199 FirmwareUpdatStat Struct Reference

Data Fields

- ULONG ResCode
- BYTE * pImgType

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

8.199.1 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.

8.199.2 Field Documentation

8.199.2.1 BYTE* FirmwareUpdatStat::pImgType

8.199.2.2 BYTE* FirmwareUpdatStat::pLogString

8.199.2.3 BYTE* FirmwareUpdatStat::pLogStringLen

8.199.2.4 ULONG* FirmwareUpdatStat::pRefData

8.199.2.5 BYTE* FirmwareUpdatStat::pRefString

8.199.2.6 BYTE* FirmwareUpdatStat::pRefStringLen

8.199.2.7 ULONG FirmwareUpdatStat::ResCode

8.200 FMSImageElement Struct Reference

Data Fields

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

8.200.1 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

8.200.2 Field Documentation

8.200.2.1 uint8_t FMSImageElement::buildId[100]

8.200.2.2 uint8_t FMSImageElement::buildIdLength

8.200.2.3 uint8_t FMSImageElement::imageId[16]

8.200.2.4 uint8_t FMSImageElement::imageType

8.201 FMSImageIdElement Struct Reference

Data Fields

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

8.201.1 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)

8.201.2 Field Documentation

8.201.2.1 uint8_t FMSImageIdElement::buildID[100]

8.201.2.2 uint8_t FMSImageIdElement::buildIDLength

8.201.2.3 uint8_t FMSImageIdElement::failureCount

8.201.2.4 uint8_t FMSImageIdElement::imageID[16]

8.201.2.5 uint8_t FMSImageIdElement::storageIndex

8.202 FMSImageIDEntries Struct Reference

Data Fields

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

8.202.1 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

8.202.2 Field Documentation

8.202.2.1 uint8_t FMSImageIDEntries::executingImage

8.202.2.2 FMSImageIdElement FMSImageIDEntries::imageIDElement[50]

8.202.2.3 uint8_t FMSImageIDEntries::imageIDSize

8.202.2.4 uint8_t FMSImageIDEntries::imageType

8.202.2.5 uint8_t FMSImageIDEntries::maxImages

8.203 FMSImageList Struct Reference

Data Fields

- uint8_t listSize
- FMSImageIDEntries imageIDEntries [2]

8.203.1 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)

8.203.2 Field Documentation

8.203.2.1 FMSImageIDEntries FMSImageList::imageIDEntries[2]

8.203.2.2 uint8_t FMSImageList::listSize

8.204 FMSPrefImageList Struct Reference

Data Fields

- uint8_t listSize
- FMSImageElement listEntries [2]

8.204.1 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

8.204.2 Field Documentation

8.204.2.1 FMSImageElement FMSPrefImageList::listEntries[2]

8.204.2.2 uint8_t FMSPrefImageList::listSize

8.205 fwinfo_s Struct Reference

Data Fields

- ULONG FirmwareID
- ULONG Technology
- ULONG Carrier

- ULONG Region
- ULONG GPSCapability

8.205.1 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)

8.205.2 Field Documentation

8.205.2.1 ULONG fwinfo_s::Carrier

8.205.2.2 ULONG fwinfo_s::FirmwareID

8.205.2.3 ULONG fwinfo_s::GPSCapability

8.205.2.4 ULONG fwinfo_s::Region

8.205.2.5 ULONG fwinfo_s::Technology

8.206 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]

8.206.1 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.

8.206.2 Field Documentation

8.206.2.1 WORD GERANInfo::arfcn

8.206.2.2 BYTE GERANInfo::bsic

8.206.2.3 ULONG GERANInfo::cellID

8.206.2.4 nmrCellInfo GERANInfo::insNmrCellInfo[255]

8.206.2.5 WORD GERANInfo::lac

8.206.2.6 BYTE GERANInfo::nmrInst

8.206.2.7 BYTE GERANInfo::plmn[3]

8.206.2.8 WORD GERANInfo::rxLev

8.206.2.9 ULONG GERANInfo::timingAdvance

8.207 geranInstInfo Struct Reference

Data Fields

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

8.207.1 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.

8.207.2 Field Documentation

8.207.2.1 WORD geranInstInfo::geranArfcn

8.207.2.2 BYTE geranInstInfo::geranBsicBcc

8.207.2.3 BYTE geranInstInfo::geranBsicNcc

8.207.2.4 SHORT geranInstInfo::geranRssi

8.208 getAllCallInformation Struct Reference

Data Fields

- callInfo CallInfo
- BYTE isEmpty
- BYTE ALS

8.208.1 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

8.208.2 Field Documentation

8.208.2.1 BYTE getAllCallInformation::ALS

8.208.2.2 callInfo getAllCallInformation::CallInfo

8.208.2.3 BYTE getAllCallInformation::isEmpty

8.209 getAllCallRmtPtyName Struct Reference

Data Fields

- BYTE callID
- remotePartyName RemotePartyName

8.209.1 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.

8.209.2 Field Documentation

8.209.2.1 BYTE getAllCallRmtPtyName::callID

8.209.2.2 remotePartyName getAllCallRmtPtyName::RemotePartyName

8.210 getAllCallRmtPtyNum Struct Reference

Data Fields

- BYTE callID
- remotePartyNum RemotePartyNum

8.210.1 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.

8.210.2 Field Documentation

8.210.2.1 BYTE getAllCallRmtPtyNum::callID

8.210.2.2 remotePartyNum getAllCallRmtPtyNum::RemotePartyNum

8.211 GetAudioPathConfigReq Struct Reference

Data Fields

- BYTE Profile
- BYTE Item

8.211.1 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

8.211.2 Field Documentation

8.211.2.1 BYTE GetAudioPathConfigReq::Item

8.211.2.2 BYTE GetAudioPathConfigReq::Profile

8.212 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

8.212.1 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

8.212.2 Field Documentation

8.212.2.1 WORD* GetAudioPathConfigResp::pCodecSTGain

8.212.2.2 WORD* GetAudioPathConfigResp::pDTMFTXGain

8.212.2.3 BYTE* GetAudioPathConfigResp::pECMode

8.212.2.4 BYTE* GetAudioPathConfigResp::pMICGainSelect

8.212.2.5 BYTE* GetAudioPathConfigResp::pNSEnable

8.212.2.6 RXAGCList* GetAudioPathConfigResp::pRXAGCList

8.212.2.7 BYTE* GetAudioPathConfigResp::pRXAVCAGCSwitch

8.212.2.8 RXAVCList* GetAudioPathConfigResp::pRXAVCList

8.212.2.9 RXPCMIIRFltr* GetAudioPathConfigResp::pRXPCMIIRFltr

8.212.2.10 TXAGCList* GetAudioPathConfigResp::pTXAGCList

8.212.2.11 BYTE* GetAudioPathConfigResp::pTXAVCSwitch

8.212.2.12 WORD* GetAudioPathConfigResp::pTXGain

8.212.2.13 TXPCMIIRFltr* GetAudioPathConfigResp::pTXPCMIIRFltr

8.213 GetAudioProfileReq Struct Reference

Data Fields

- BYTE Generator

8.213.1 Detailed Description

This structure contains the SLQSGetAudioProfile request parameters

Parameters

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

8.213.2 Field Documentation

8.213.2.1 BYTE GetAudioProfileReq::Generator

8.214 GetAudioProfileResp Struct Reference

Data Fields

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

8.214.1 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

8.214.2 Field Documentation

8.214.2.1 BYTE GetAudioProfileResp::EarMute

8.214.2.2 BYTE GetAudioProfileResp::MicMute

8.214.2.3 BYTE GetAudioProfileResp::Profile

8.214.2.4 BYTE GetAudioProfileResp::Volume

8.215 GetAudioVolTLBConfigReq Struct Reference

Data Fields

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

8.215.1 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

8.215.2 Field Documentation

8.215.2.1 BYTE GetAudioVolTLBConfigReq::Generator

8.215.2.2 BYTE GetAudioVolTLBConfigReq::Item

8.215.2.3 BYTE GetAudioVolTLBConfigReq::Profile

8.215.2.4 BYTE GetAudioVolTLBConfigReq::Volume

8.216 GetAudioVolTLBConfigResp Struct Reference

Data Fields

- WORD ResCode

8.216.1 Detailed Description

This structure contains the SLQSGetAudioVolTLBConfig response parameters.

Parameters

ResCode	• Result of requested item

8.216.2 Field Documentation

8.216.2.1 WORD GetAudioVolTLBConfigResp::ResCode

8.217 getCallFWExtInfo Struct Reference

Data Fields

- BYTE numInstances
- callFWExtInfo CallFWExtInfo [20]

8.217.1 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.

8.217.2 Field Documentation

8.217.2.1 callFWExtInfo getCallFWExtInfo::CallFWExtInfo[20]

8.217.2.2 BYTE getCallFWExtInfo::numInstances

8.218 getCallFWInfo Struct Reference

Data Fields

- BYTE numInstances
- callFWInfo CallFWInfo [20]

8.218.1 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.

8.218.2 Field Documentation

8.218.2.1 callFWInfo getCallFWInfo::CallFWInfo[20]

8.218.2.2 BYTE getCallFWInfo::numInstances

8.219 getCustomFeatureV2 Struct Reference

Data Fields

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

8.219.1 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

8.219.2 Field Documentation

8.219.2.1 custSettingInfo* getCustomFeatureV2::pCustSettingInfo

8.219.2.2 custSettingList* getCustomFeatureV2::pCustSettingList

8.219.2.3 getCustomInput* getCustomFeatureV2::pGetCustomInput

8.220 getCustomInput Struct Reference

Data Fields

- CHAR cust_id [64+1]
- BYTE list_type

8.220.1 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

8.220.2 Field Documentation

8.220.2.1 CHAR getCustomInput::cust_id[64+1]

8.220.2.2 BYTE getCustomInput::list_type

8.221 getDUNCallInfoReq Struct Reference

Data Fields

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

8.221.1 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

8.221.2 Field Documentation

8.221.2.1 ULONG getDUNCallInfoReq::Mask

8.221.2.2 BYTE* getDUNCallInfoReq::pReportChannelRate

8.221.2.3 BYTE* getDUNCallInfoReq::pReportConnStatus

8.221.2.4 BYTE* getDUNCallInfoReq::pReportDataBearerTech

8.221.2.5 BYTE* getDUNCallInfoReq::pReportDormStatus

8.221.2.6 TransferStatInd* getDUNCallInfoReq::pTransferStatInd

8.222 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

8.222.1 Detailed Description

This structure contains the DUN Call Info response parameters

Parameters

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

pDataBearer-Tech	• Current data bearer technology • Returned only if a data call is up – 0x01 - cdma2000 1X – 0x02 - cdma2000 HRPD (1xEV-DO) – 0x03 - GSM – 0x04 - UMTS – 0x05 - cdma200 HRPD (1xEV-DO RevA) – 0x06 - EDGE – 0x07 - HSDPA and WCDMA – 0x08 - WCDMA and HSUPA – 0x09 - HSDPA and HSUPA – 0x0A - LTE – 0x0B - cdma2000 EHRPD – 0x0C - HSDPA+ and WCDMA – 0x0D - HSDPA+ and HSUPA – 0x0E - DC_HSDPA+ and WCDMA – 0x0F - DC_HSDPA+ and HSUPA – 0x10 - HSDPA+ and 64QAM – 0x11 - HSDPA+, 64QAM and HSUPA – 0x12 - TDSCDMA – 0x13 - TDSCDMA and HSDPA – 0xFF - Unknown
pChannelRate	• See ChannelRate for more information
pLastCallTXOK-BytesCnt	• Number of bytes transmitted without error during the last data call (0 if no call was made). • Return only if not in a call and the previous call was made using DUN.
pLastCallRXOK-BytesCnt	• Number of bytes transmitted without error during the last data call (0 if no call was made). • Return only if not in a call and the previous call was made using DUN.
pMdmCall-DurationActive	• Duration that the call is active in milliseconds • If the modem connection status is connected, this represents the active duration of the current DUN call • If the modem connection status is disconnected, this represents the duration of the last DUN call since the device was powered up (0 if no call has been made or if the last call was not DUN)

pLastCallData-BearerTech	• Last Call Data Bearer Technology • Returned only if not in a call and when the previous call was made using DUN – 0x01 - cdma2000 1X – 0x02 - cdma2000 HRPD (1xEV-DO) – 0x03 - GSM – 0x04 - UMTS – 0x05 - cdma200 HRPD (1xEV-DO Rev A) – 0x06 - EDGE – 0x07 - HSDPA and WCDMA – 0x08 - WCDMA and HSUPA – 0x09 - HSDPA and HSUPA – 0x0A - LTE – 0x0B - cdma2000 EHRPD – 0x0C - HSDPA+ and WCDMA – 0x0D - HSDPA+ and HSUPA – 0x0E - DC_HSDPA+ and WCDMA – 0x0F - DC_HSDPA+ and HSUPA – 0x10 - HSDPA+ and 64QAM – 0x11 - HSDPA+, 64QAM and HSUPA – 0x12 - TDSCDMA – 0x13 - TDSCDMA and HSDPA – 0xFF - Unknown

8.222.2 Field Documentation

8.222.2.1 WORD* getDUNCallInfoResp::pCallEndReason

8.222.2.2 ChannelRate* getDUNCallInfoResp::pChannelRate

8.222.2.3 ConnectionStatus* getDUNCallInfoResp::pConnectionStatus

8.222.2.4 BYTE* getDUNCallInfoResp::pDataBearerTech

8.222.2.5 BYTE* getDUNCallInfoResp::pDormancyStatus

8.222.2.6 BYTE* getDUNCallInfoResp::pLastCallDataBearerTech

8.222.2.7 ULONGLONG* getDUNCallInfoResp::pLastCallRXOKBytesCnt

8.222.2.8 ULONGLONG* getDUNCallInfoResp::pLastCallTXOKBytesCnt

8.222.2.9 ULONGLONG* getDUNCallInfoResp::pMdmCallDurationActive

8.222.2.10 ULONGLONG* getDUNCallInfoResp::pRXOKBytesCount

8.222.2.11 ULONGLONG* getDUNCallInfoResp::pTXOKBytesCount

8.223 getDyingGaspCfg Struct Reference

Data Fields

- BYTE * pDestSMSNum
- BYTE * pDestSMSContent

8.223.1 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.

8.223.2 Field Documentation

8.223.2.1 BYTE* getDyingGaspCfg::pDestSMSContent

8.223.2.2 BYTE* getDyingGaspCfg::pDestSMSNum

8.224 getDyingGaspStatistics Struct Reference

Data Fields

- ULONG * pTimeStamp
- BYTE * pSMSAttemptedFlag

8.224.1 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.

8.224.2 Field Documentation

8.224.2.1 BYTE* getDyingGaspStatistics::pSMSAttemptedFlag

8.224.2.2 ULONG* getDyingGaspStatistics::pTimeStamp

8.225 GetErrRateResp Struct Reference

Data Fields

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

8.225.1 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.

8.225.2 Field Documentation

8.225.2.1 WORD* GetErrRateResp::pCDMAFrameErrRate

8.225.2.2 BYTE* GetErrRateResp::pGSMBER

8.225.2.3 WORD* GetErrRateResp::pHDRPackErrRate

8.225.2.4 BYTE* GetErrRateResp::pWCDMABER

8.226 GetHRPDStatsResp Struct Reference

Data Fields

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

8.226.1 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.

8.226.2 Field Documentation

8.226.2.1 DRCParams* GetHRPDStatsResp::pDRCParams

8.226.2.2 PilotSetData* GetHRPDStatsResp::pPilotSetData

8.226.2.3 BYTE* GetHRPDStatsResp::pUATI

8.227 GetIMSSMSConfigParams Struct Reference

Data Fields

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

8.227.1 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

8.227.2 Field Documentation

8.227.2.1 BYTE* GetIMSSMSConfigParams::pPhoneCtxtURI

8.227.2.2 BYTE* GetIMSSMSConfigParams::pPhoneCtxtURILen

8.227.2.3 BYTE* GetIMSSMSConfigParams::pSettingResp

8.227.2.4 BYTE* GetIMSSMSConfigParams::pSMSFormat

8.227.2.5 BYTE* GetIMSSMSConfigParams::pSMSOverIPNwInd

8.228 GetIMSUserConfigParams Struct Reference

Data Fields

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

8.228.1 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

8.228.2 Field Documentation

8.228.2.1 BYTE* GetIMSUserConfigParams::pIMSDomain

8.228.2.2 BYTE* GetIMSUserConfigParams::pIMSDomainLen

8.228.2.3 BYTE* GetIMSUserConfigParams::pSettingResp

8.229 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

8.229.1 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.

8.229.2 Field Documentation

8.229.2.1 BYTE* GetIMSVoIPConfigResp::pAmrMode

8.229.2.2 BYTE* GetIMSVoIPConfigResp::pAmrOctetAligned

8.229.2.3 BYTE* GetIMSVoIPConfigResp::pAmrWbEnable

8.229.2.4 WORD* GetIMSVoIPConfigResp::pAmrWBMode

8.229.2.5 BYTE* GetIMSVoIPConfigResp::pAmrWBOctetAligned

8.229.2.6 WORD* GetIMSVoIPConfigResp::pMinSessionExpiryTimer

8.229.2.7 WORD* GetIMSVoIPConfigResp::pRingBackTimer

8.229.2.8 WORD* GetIMSVoIPConfigResp::pRingingTimer

8.229.2.9 WORD* GetIMSVoIPConfigResp::pRTPRTCPInactTimer

8.229.2.10 BYTE* GetIMSVoIPConfigResp::pScrAmrEnable

8.229.2.11 BYTE* GetIMSVoIPConfigResp::pScrAmrWbEnable

8.229.2.12 WORD* GetIMSVoIPConfigResp::pSessionExpiryTimer

8.229.2.13 BYTE* GetIMSVoIPConfigResp::pSettingResp

8.230 GetInstIDResp Struct Reference

Data Fields

- BYTE * pInstanceID
- BYTE * pIPFamily

8.230.1 Field Documentation

8.230.1.1 BYTE* GetInstIDResp::pInstanceID

8.230.1.2 BYTE* GetInstIDResp::pIPFamily

8.231 GetM2MAudioProfileReq Struct Reference

Data Fields

- BYTE * pGenerator

8.231.1 Detailed Description

This structure contains the SLQSGetM2MAudioProfile request parameters.

Parameters

pGenerator[optional]	• Generator – 0 - Voice

8.231.2 Field Documentation

8.231.2.1 BYTE* GetM2MAudioProfileReq::pGenerator

8.232 GetM2MAudioProfileResp Struct Reference

Data Fields

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

8.232.1 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

8.232.2 Field Documentation

8.232.2.1 BYTE GetM2MAudioProfileResp::CwtMute

8.232.2.2 BYTE GetM2MAudioProfileResp::EarMute

8.232.2.3 BYTE GetM2MAudioProfileResp::Generator

8.232.2.4 BYTE GetM2MAudioProfileResp::MicMute

8.232.2.5 BYTE GetM2MAudioProfileResp::Profile

8.232.2.6 BYTE GetM2MAudioProfileResp::Volume

8.233 GetM2MAudioVolumeReq Struct Reference

Data Fields

- BYTE Profile
- BYTE Generator

8.233.1 Detailed Description

This structure contains the SLQSGetM2MAudioVolume request parameters.

Parameters

Profile	• Audio Profile – 0-5

Generator	• Generator – 0 - Voice

8.233.2 Field Documentation

8.233.2.1 BYTE GetM2MAudioVolumeReq::Generator

8.233.2.2 BYTE GetM2MAudioVolumeReq::Profile

8.234 GetM2MAudioVolumeResp Struct Reference

Data Fields

- BYTE Level

8.234.1 Detailed Description

This structure contains the SLQSGetM2MAudioVolume response parameters.

Parameters

Level	• The RX Volume Level – 0-5

8.234.2 Field Documentation

8.234.2.1 BYTE GetM2MAudioVolumeResp::Level

8.235 GetM2MAVMuteReq Struct Reference

Data Fields

- BYTE Profile

8.235.1 Detailed Description

This structure contains the SLQSGetM2MAVMute request parameters.

Parameters

Profile	• Audio Profile Number – 0-5

8.235.2 Field Documentation

8.235.2.1 BYTE GetM2MAVMuteReq::Profile

8.236 GetM2MAVMuteResp Struct Reference

Data Fields

- BYTE EarMute
- BYTE MicMute
- BYTE CwtMute

8.236.1 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

8.236.2 Field Documentation

8.236.2.1 BYTE GetM2MAVMuteResp::CwtMute

8.236.2.2 BYTE GetM2MAVMuteResp::EarMute

8.236.2.3 BYTE GetM2MAVMuteResp::MicMute

8.237 GetM2MSpkrGainReq Struct Reference

Data Fields

- BYTE Profile

8.237.1 Detailed Description

This structure contains the SLQSGetM2MSpkrGain request parameters.

Parameters

pProfile	• Audio Profile Number – 0-5

8.237.2 Field Documentation

8.237.2.1 BYTE GetM2MSpkrGainReq::Profile

8.238 GetM2MSpkrGainResp Struct Reference

Data Fields

- WORD Value

8.238.1 Detailed Description

This structure contains the SLQSGetM2MSpkrGain response parameters.

Parameters

Value	• RX speakerphone gain – 0x0 - 0x7fff

8.238.2 Field Documentation

8.238.2.1 WORD GetM2MSpkrGainResp::Value

8.239 getMsgWaitingInfo Struct Reference

Data Fields

- BYTE numInstances
- messageWaitingInfoContent msgWaitInfo [0xFF]

8.239.1 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.

8.239.2 Field Documentation

8.239.2.1 messageWaitingInfoContent getMsgWaitingInfo::msgWaitInfo[0xFF]

8.239.2.2 BYTE getMsgWaitingInfo::numInstances

8.240 GetNetworkTimeResp Struct Reference

Data Fields

- timeInfo * p3GPP2TimeInfo
- timeInfo * p3GPPTimeInfo

8.240.1 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

8.240.2 Field Documentation

8.240.2.1 timeInfo* GetNetworkTimeResp::p3GPP2TimeInfo

8.240.2.2 timeInfo* GetNetworkTimeResp::p3GPPTimeInfo

8.241 GetRegMgrConfigParams Struct Reference

Data Fields

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

8.241.1 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.

8.241.2 Field Documentation

8.241.2.1 BYTE* GetRegMgrConfigParams::pIMSTestMode

8.241.2.2 WORD* GetRegMgrConfigParams::pPCSCFPort

8.241.2.3 BYTE* GetRegMgrConfigParams::pPriCSCFPortName

8.241.2.4 BYTE* GetRegMgrConfigParams::pPriCSCFPortNameLen

8.241.2.5 BYTE* GetRegMgrConfigParams::pSettingResp

8.242 GetSessionIDResp Struct Reference

Data Fields

- ULONG * pSessionIDv4
- ULONG * pSessionIDv6

8.242.1 Field Documentation

8.242.1.1 ULONG* GetSessionIDResp::pSessionIDv4

8.242.1.2 ULONG* GetSessionIDResp::pSessionIDv6

8.243 GetSIPConfigResp Struct Reference

Data Fields

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

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

8.243.1 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

8.243.2 Field Documentation

8.243.2.1 BYTE* GetSIPConfigResp::pSettingResp

8.243.2.2 BYTE* GetSIPConfigResp::pSigCompEnabled

8.243.2.3 WORD* GetSIPConfigResp::pSIPLocalPort

8.243.2.4 ULONG* GetSIPConfigResp::pSubscribeTimer

8.243.2.5 ULONG* GetSIPConfigResp::pTimerSIPReg

8.243.2.6 ULONG* GetSIPConfigResp::pTimerT1

8.243.2.7 ULONG* GetSIPConfigResp::pTimerT2

8.243.2.8 ULONG* GetSIPConfigResp::pTimerTf

8.244 GnssData Struct Reference

Data Fields

- ULONGLONG mask

8.244.1 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

8.244.2 Field Documentation

8.244.2.1 ULONGLONG GnssData::mask

8.245 gnssSvInfoNotification Struct Reference

Data Fields

- BYTE bAltitudeAssumed
- satelliteInfo * pSatelliteInfo

8.245.1 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

8.245.2 Field Documentation

8.245.2.1 BYTE gnssSvInfoNotification::bAltitudeAssumed

8.245.2.2 satelliteInfo* gnssSvInfoNotification::pSatelliteInfo

8.246 GPRSQoS 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

- 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

8.246.2 Field Documentation

8.246.2.1 ULONG GPRSQoS::delayClass

8.246.2.2 ULONG GPRSQoS::meanThroughputClass

8.246.2.3 ULONG GPRSQoS::peakThroughputClass

8.246.2.4 ULONG GPRSQoS::precedenceClass

8.246.2.5 ULONG GPRSQoS::reliabilityClass

8.247 GPRSRequestedQoS Struct Reference

Data Fields

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

8.247.1 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

8.247.2 Field Documentation

8.247.2.1 ULONG GPRSRequestedQoS::delayClass

8.247.2.2 ULONG GPRSRequestedQoS::meanThroughputClass

8.247.2.3 ULONG GPRSRequestedQoS::peakThroughputClass

8.247.2.4 ULONG GPRSRequestedQoS::precedenceClass

8.247.2.5 ULONG GPRSRequestedQoS::reliabilityClass

8.248 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

8.248.1 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

8.248.2 Field Documentation

8.248.2.1 ULONG GPSStateInfo::Altitude

8.248.2.2 BYTE GPSStateInfo::EngineState

8.248.2.3 ULONG GPSStateInfo::glo_almanac_sv_msk

8.248.2.4 ULONG GPSStateInfo::glo_ephemeris_sv_msk

8.248.2.5 ULONG GPSStateInfo::glo_health_sv_msk

8.248.2.6 ULONG GPSStateInfo::glo_visible_sv_msk

8.248.2.7 ULONG GPSStateInfo::gps_almanac_sv_msk

8.248.2.8 ULONG GPSStateInfo::gps_ephemeris_sv_msk

8.248.2.9 ULONG GPSStateInfo::gps_health_sv_msk

8.248.2.10 ULONG GPSStateInfo::gps_visible_sv_msk

8.248.2.11 ULONG GPSStateInfo::HorizontalUncertainty

8.248.2.12 BYTE GPSStateInfo::Iono_valid

8.248.2.13 ULONGLONG GPSStateInfo::Latitude

8.248.2.14 ULONGLONG GPSStateInfo::Longitude

8.248.2.15 ULONG GPSStateInfo::sbas_almanac_sv_msk

8.248.2.16 ULONG GPSStateInfo::sbas_ephemeris_sv_msk

8.248.2.17 ULONG GPSStateInfo::sbas_health_sv_msk

8.248.2.18 ULONG GPSStateInfo::sbas_visible_sv_msk

8.248.2.19 ULONG GPSStateInfo::Time_uncert_ms

8.248.2.20 WORD GPSStateInfo::TimeStmp_gps_week

8.248.2.21 ULONG GPSStateInfo::TimeStmp_tow_ms

8.248.2.22 ULONG GPSStateInfo::ValidMask

8.248.2.23 ULONG GPSStateInfo::VerticalUncertainty

8.248.2.24 WORD GPSStateInfo::xtra_start_gps_minutes

8.248.2.25 WORD GPSStateInfo::xtra_start_gps_week

8.248.2.26 WORD GPSStateInfo::xtra_valid_duration_hours

8.249 gpsTime_s Struct Reference

Data Fields

- WORD gpsWeek
- ULONG gpsTimeOfWeekMs

8.249.1 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

8.249.2 Field Documentation

8.249.2.1 ULONG gpsTime_s::gpsTimeOfWeekMs

8.249.2.2 WORD gpsTime_s::gpsWeek

8.250 gsmCellInfo Struct Reference

Data Fields

- WORD arfcn
- BYTE band1900

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

8.250.1 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.

8.250.2 Field Documentation

8.250.2.1 WORD gsmCellInfo::arfcn

8.250.2.2 BYTE gsmCellInfo::band1900

8.250.2.3 BYTE gsmCellInfo::bsicId

8.250.2.4 BYTE gsmCellInfo::cellIdValid

8.250.2.5 SHORT gsmCellInfo::rssi

8.250.2.6 SHORT gsmCellInfo::srxlev

8.251 GSMRSSIThresh Struct Reference

Data Fields

- BYTE GSMRSSIThreshListLen
- WORD * pGSMRSSIThreshList

8.251.1 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)

8.251.2 Field Documentation

8.251.2.1 BYTE GSMRSSIThresh::GSMRSSIThreshListLen

8.251.2.2 WORD* GSMRSSIThresh::pGSMRSSIThreshList

8.252 GSMSrvStatusInfo Struct Reference

Data Fields

- BYTE srvStatus
- BYTE trueSrvStatus
- BYTE isPrefDataPath

8.252.1 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

8.252.2 Field Documentation

8.252.2.1 BYTE GSMSrvStatusInfo::isPrefDataPath

8.252.2.2 BYTE GSMSrvStatusInfo::srvStatus

8.252.2.3 BYTE GSMSrvStatusInfo::trueSrvStatus

8.253 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

8.253.1 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

8.253.2 Field Documentation

8.253.2.1 ULONG GSMSysInfo::cellId

8.253.2.2 BYTE GSMSysInfo::cellIdValid

8.253.2.3 BYTE GSMSysInfo::dtmSupp

8.253.2.4 BYTE GSMSysInfo::dtmSuppValid

8.253.2.5 BYTE GSMSysInfo::egprsSupp

8.253.2.6 BYTE GSMSysInfo::egprsSuppValid

8.253.2.7 WORD GSMSysInfo::lac

8.253.2.8 BYTE GSMSysInfo::lacValid

8.253.2.9 BYTE GSMSysInfo::MCC[3]

8.253.2.10 BYTE GSMSysInfo::MNC[3]

8.253.2.11 BYTE GSMSysInfo::networkIdValid

8.253.2.12 BYTE GSMSysInfo::regRejectInfoValid

8.253.2.13 BYTE GSMSysInfo::rejCause

8.253.2.14 BYTE GSMSysInfo::rejectSrvDomain

8.253.2.15 sysInfoCommon GSMSysInfo::sysInfoGSM

8.254 gyroAcceptReady_s Struct Reference

Data Fields

- BYTE injectEnable
- WORD samplesPerBatch
- WORD batchPerSec

8.254.1 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.

8.254.2 Field Documentation

8.254.2.1 WORD gyroAcceptReady_s::batchPerSec

8.254.2.2 BYTE gyroAcceptReady_s::injectEnable

8.254.2.3 WORD gyroAcceptReady_s::samplesPerBatch

8.255 gyroTempAcceptReady_s Struct Reference

Data Fields

- BYTE injectEnable
- WORD samplesPerBatch
- WORD batchPerSec

8.255.1 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.

8.255.2 Field Documentation

8.255.2.1 WORD gyroTempAcceptReady_s::batchPerSec

8.255.2.2 BYTE gyroTempAcceptReady_s::injectEnable

8.255.2.3 WORD gyroTempAcceptReady_s::samplesPerBatch

8.256 HDRECIOThresh Struct Reference

Data Fields

- BYTE HDRECIOThreshListLen
- WORD ∗ pHDRECIOThreshList

8.256.1 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).

8.256.2 Field Documentation

8.256.2.1 BYTE HDRECIOThresh::HDRECIOThreshListLen

8.256.2.2 WORD* HDRECIOThresh::pHDRECIOThreshList

8.257 HDRIOThresh Struct Reference

Data Fields

- BYTE HDRIOThreshListLen
- WORD * pHDRIOThreshList

8.257.1 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).

8.257.2 Field Documentation

8.257.2.1 BYTE HDRIOThresh::HDRIOThreshListLen

8.257.2.2 WORD* HDRIOThresh::pHDRIOThreshList

8.258 HDRPersonalityInd Struct Reference

Data Fields

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

8.258.1 Field Documentation

8.258.1.1 WORD* HDRPersonalityInd::pCurrentPersonality

8.258.1.2 BYTE* HDRPersonalityInd::pPersonalityListLength

8.258.1.3 protocolSubtypeElement* HDRPersonalityInd::pProtocolSubtypeElement

8.259 HDRPersonalityResp Struct Reference

Data Fields

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

8.259.1 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.

8.259.2 Field Documentation

8.259.2.1 WORD* HDRPersonalityResp::pCurrentPersonality

8.259.2.2 BYTE* HDRPersonalityResp::pPersonalityListLength

8.259.2.3 protocolSubtypeElement* HDRPersonalityResp::pProtocolSubtypeElement

8.260 HDRProtSubtypResp Struct Reference

Data Fields

- WORD * pCurrentPrsnlty
- BYTE * pPersonalityListLength

- protocolSubtypeElement * pProtoSubTypElmnt
- ULONGLONG * pAppSubType

8.260.1 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,

8.260.2 Field Documentation

8.260.2.1 ULONGLONG* HDRProtSubtypResp::pAppSubType

8.260.2.2 WORD* HDRProtSubtypResp::pCurrentPrsnlty

8.260.2.3 BYTE* HDRProtSubtypResp::pPersonalityListLength

8.260.2.4 protocolSubtypeElement* HDRProtSubtypResp::pProtoSubTypElmnt

8.261 HDRRSSIThresh Struct Reference

Data Fields

- BYTE HDRRSSIThreshListLen
- WORD * pHDRRSSIThreshList

8.261.1 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).

8.261.2 Field Documentation

8.261.2.1 BYTE HDRRSSIThresh::HDRRSSIThreshListLen

8.261.2.2 WORD* HDRRSSIThresh::pHDRRSSIThreshList

8.262 HDRSINRThresh Struct Reference

Data Fields

- BYTE HDRSINRThresListLen
- BYTE * pHDRSINRThresList

8.262.1 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

8.262.2 Field Documentation

8.262.2.1 BYTE HDRSINRThresh::HDRSINRThresListLen

8.262.2.2 BYTE* HDRSINRThresh::pHDRSINRThresList

8.263 HDRSINRThreshold Struct Reference

Data Fields

- BYTE HDRSINRThreshListLen
- WORD * pHDRSINRThreshList

8.263.1 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

8.263.2 Field Documentation

8.263.2.1 BYTE HDRSINRThreshold::HDRSINRThreshListLen

8.263.2.2 WORD* HDRSINRThreshold::pHDRSINRThreshList

8.264 HDRSSInfo Struct Reference

Data Fields

- INT8 rssi
- SHORT ecio
- BYTE sinr
- INT32 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 SHORT HDRSSInfo::ecio

8.264.2.2 INT32 HDRSSInfo::io

8.264.2.3 INT8 HDRSSInfo::rssi

8.264.2.4 BYTE HDRSSInfo::sinr

8.265 hdrSSInfo Struct Reference

Data Fields

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

8.265.1 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.

8.265.2 Field Documentation

8.265.2.1 int16_t hdrSSInfo::ecio

8.265.2.2 int32_t hdrSSInfo::io

8.265.2.3 int8_t hdrSSInfo::rssi

8.265.2.4 uint8_t hdrSSInfo::sinr

8.266 HDRSysInfo Struct Reference

Data Fields

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

8.266.1 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.

8.266.2 Field Documentation

8.266.2.1 BYTE HDRSysInfo::hdrActiveProt

8.266.2.2 BYTE HDRSysInfo::hdrActiveProtValid

8.266.2.3 BYTE HDRSysInfo::hdrPersonality

8.266.2.4 BYTE HDRSysInfo::hdrPersonalityValid

8.266.2.5 BYTE HDRSysInfo::is856SysId[16]

8.266.2.6 BYTE HDRSysInfo::is856SysIdValid

8.266.2.7 BYTE HDRSysInfo::isSysPrlMatch

8.266.2.8 BYTE HDRSysInfo::isSysPrlMatchValid

8.266.2.9 sysInfoCommon HDRSysInfo::sysInfoHDR

8.267 homeSIDNID Struct Reference

Data Fields

- BYTE numInstances
- sidNid SidNid [255]

8.267.1 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

8.267.2 Field Documentation

8.267.2.1 BYTE homeSIDNID::numInstances

8.267.2.2 sidNid homeSIDNID::SidNid[255]

8.268 hotSwapStatus Struct Reference

Data Fields

- BYTE hotSwapLength
- BYTE hotSwap [255]

8.268.1 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

8.268.2 Field Documentation

8.268.2.1 BYTE hotSwapStatus::hotSwap[255]

8.268.2.2 BYTE hotSwapStatus::hotSwapLength

8.269 image_info_t Struct Reference

Data Fields

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

8.269.1 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

8.269.2 Field Documentation

8.269.2.1 uint8_t image_info_t::buildID[255]

8.269.2.2 uint8_t image_info_t::buildIDLen

8.269.2.3 uint8_t image_info_t::imageType

8.269.2.4 uint8_t image_info_t::uniqueID[16]

8.270 ImageElement Struct Reference

Data Fields

- BYTE imageType
- 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 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)

8.270.2 Field Documentation

8.270.2.1 CHAR ImageElement::buildId[100]

8.270.2.2 BYTE ImageElement::buildIdLength

8.270.2.3 BYTE ImageElement::imageId[16]

8.270.2.4 BYTE ImageElement::imageType

8.271 ImageIdElement Struct Reference

Data Fields

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

8.271.1 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)

8.271.2 Field Documentation

8.271.2.1 CHAR ImageIdElement::buildID[100]

8.271.2.2 BYTE ImageIdElement::buildIDLength

8.271.2.3 BYTE ImageIdElement::failureCount

8.271.2.4 BYTE ImageIdElement::imageID[16]

8.271.2.5 BYTE ImageIdElement::storageIndex

8.272 ImageIDEntries Struct Reference

Data Fields

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

8.272.1 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)

8.272.2 Field Documentation

8.272.2.1 BYTE ImageIDEntries::executingImage

8.272.2.2 struct ImageIdElement ImageIDEntries::imageIDElement[50]

8.272.2.3 BYTE ImageIDEntries::imageIDSize

8.272.2.4 BYTE ImageIDEntries::imageType

8.272.2.5 BYTE ImageIDEntries::maxImages

8.273 ImageList Struct Reference

Data Fields

- BYTE listSize
- struct ImageIDEntries imageIDEntries [2]

8.273.1 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)

8.273.2 Field Documentation

8.273.2.1 struct ImageIDEntries ImageList::imageIDEntries[2]

8.273.2.2 BYTE ImageList::listSize

8.274 IMSAIndRegisterInfo Struct Reference

Data Fields

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

8.274.1 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.

8.274.2 Field Documentation

8.274.2.1 BYTE* IMSAIndRegisterInfo::pPdpStatusConfig

8.274.2.2 BYTE* IMSAIndRegisterInfo::pRatHandoverStatusConfig

8.274.2.3 BYTE* IMSAIndRegisterInfo::pRegStatusConfig

8.274.2.4 BYTE* IMSAIndRegisterInfo::pServiceStatusConfig

8.275 imsaPdpStatusInfo Struct Reference

Data Fields

- BYTE connetionState
- ULONG * pFailErrorCode

8.275.1 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.

8.275.2 Field Documentation

8.275.2.1 BYTE imsaPdpStatusInfo::connetionState

8.275.2.2 ULONG* imsaPdpStatusInfo::pFailErrorCode

8.276 imsaRatStatusInfo Struct Reference

Data Fields

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

8.276.1 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

8.276.2 Field Documentation

8.276.2.1 BYTE* imsaRatStatusInfo::pErrorCodeStr

8.276.2.2 ULONG* imsaRatStatusInfo::pRATStatus

8.276.2.3 ULONG* imsaRatStatusInfo::pSrcRAT

8.276.2.4 ULONG* imsaRatStatusInfo::pTgtRAT

8.277 IMSARegistrationStatus Struct Reference

Data Fields

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

8.277.1 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

8.277.2 Field Documentation

8.277.2.1 WORD* IMSARegistrationStatus::pImsRegErrCode

8.277.2.2 BYTE* IMSARegistrationStatus::pImsRegStatus

8.277.2.3 ULONG* IMSARegistrationStatus::pNewImsRegStatus

8.278 imsaRegStatusInfo Struct Reference

Data Fields

- BYTE * pbIMSRegistered

- WORD * pRegStatusErrorCode
- ULONG * pImsRegStatus

8.278.1 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

8.278.2 Field Documentation

8.278.2.1 BYTE* imsaRegStatusInfo::pbIMSRegistered

8.278.2.2 ULONG* imsaRegStatusInfo::pImsRegStatus

8.278.2.3 WORD* imsaRegStatusInfo::pRegStatusErrorCode

8.279 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

8.279.1 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

8.279.2 Field Documentation

8.279.2.1 ULONG* IMSAServiceStatus::pSmsServiceRat

8.279.2.2 ULONG* IMSAServiceStatus::pSmsServiceStatus

8.279.2.3 ULONG* IMSAServiceStatus::pUtServiceRat

8.279.2.4 ULONG* IMSAServiceStatus::pUtServiceStatus

8.279.2.5 ULONG* IMSAServiceStatus::pVoipServiceRat

8.279.2.6 ULONG* IMSAServiceStatus::pVoipServiceStatus

8.279.2.7 ULONG* IMSAServiceStatus::pVsServiceRat

8.279.2.8 ULONG* IMSAServiceStatus::pVsServiceStatus

8.279.2.9 ULONG* IMSAServiceStatus::pVtServiceRat

8.279.2.10 ULONG* IMSAServiceStatus::pVtServiceStatus

8.280 IMSASupportedFieldsResp Struct Reference

Data Fields

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

8.280.1 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

8.280.2 Field Documentation

8.280.2.1 struct IndFieldsList* IMSASupportedFieldsResp::pIndFieldsList

8.280.2.2 struct ReqFieldsList* IMSASupportedFieldsResp::pReqFieldsList

8.280.2.3 struct RespFieldsList* IMSASupportedFieldsResp::pRespFieldsList

8.281 IMSASupportedMsgInfo Struct Reference

Data Fields

- struct SupportedMsgList * pSupportedMsgList

8.281.1 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

8.281.2 Field Documentation

8.281.2.1 struct SupportedMsgList* IMSASupportedMsgInfo::pSupportedMsgList

8.282 imsaSvcStatusInfo Struct Reference

Data Fields

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

8.282.1 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

8.282.2 Field Documentation

8.282.2.1 ULONG∗ imsaSvcStatusInfo::pSMSSvcRAT

8.282.2.2 ULONG∗ imsaSvcStatusInfo::pSMSSvcStatus

8.282.2.3 ULONG∗ imsaSvcStatusInfo::pUTSvcRAT

8.282.2.4 ULONG∗ imsaSvcStatusInfo::pUTSvcStatus

8.282.2.5 ULONG∗ imsaSvcStatusInfo::pVOIPSvcRAT

8.282.2.6 ULONG∗ imsaSvcStatusInfo::pVOIPSvcStatus

8.282.2.7 ULONG∗ imsaSvcStatusInfo::pVTSvcRAT

8.282.2.8 ULONG∗ imsaSvcStatusInfo::pVTSvcStatus

8.283 imsCfgIndRegisterInfo Struct Reference

Data Fields

- BYTE ∗ pSIPConfigEvents
- BYTE ∗ pRegMgrConfigEvents
- BYTE ∗ pSMSConfigEvents
- BYTE ∗ pUserConfigEvents
- BYTE ∗ pVoIPConfigEvents

8.283.1 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.

8.283.2 Field Documentation

8.283.2.1 BYTE* imsCfgIndRegisterInfo::pRegMgrConfigEvents

8.283.2.2 BYTE* imsCfgIndRegisterInfo::pSIPConfigEvents

8.283.2.3 BYTE* imsCfgIndRegisterInfo::pSMSConfigEvents

8.283.2.4 BYTE* imsCfgIndRegisterInfo::pUserConfigEvents

8.283.2.5 BYTE* imsCfgIndRegisterInfo::pVoIPConfigEvents

8.284 imsRegMgrConfigInfo Struct Reference

Data Fields

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

8.284.1 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

8.284.2 Field Documentation

8.284.2.1 BYTE* imsRegMgrConfigInfo::pCSCFPortName

8.284.2.2 BYTE* imsRegMgrConfigInfo::pIMSTestMode

8.284.2.3 WORD* imsRegMgrConfigInfo::pPriCSCFPort

8.285 imsSIPConfigInfo Struct Reference

Data Fields

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

8.285.1 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

8.285.2 Field Documentation

8.285.2.1 BYTE* imsSIPConfigInfo::pSigCompEnabled

8.285.2.2 WORD* imsSIPConfigInfo::pSIPLocalPort

8.285.2.3 ULONG* imsSIPConfigInfo::pSubscribeTimer

8.285.2.4 ULONG* imsSIPConfigInfo::pTimerSIPReg

8.285.2.5 ULONG* imsSIPConfigInfo::pTimerT1

8.285.2.6 ULONG* imsSIPConfigInfo::pTimerT2

8.285.2.7 ULONG* imsSIPConfigInfo::pTimerTf

8.286 imsSMSConfigInfo Struct Reference

Data Fields

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

8.286.1 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

8.286.2 Field Documentation

8.286.2.1 BYTE* imsSMSConfigInfo::pPhoneCtxtURI

8.286.2.2 BYTE* imsSMSConfigInfo::pSMSFormat

8.286.2.3 BYTE* imsSMSConfigInfo::pSMSOverIPNwInd

8.287 imsUserConfigInfo Struct Reference

Data Fields

- BYTE * pIMSDomain

8.287.1 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

8.287.2 Field Documentation

8.287.2.1 BYTE* imsUserConfigInfo::pIMSDomain

8.288 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

8.288.1 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.

8.288.2 Field Documentation

8.288.2.1 BYTE* imsVoIPConfigInfo::pAmrMode

8.288.2.2 BYTE* imsVoIPConfigInfo::pAmrOctetAligned

8.288.2.3 BYTE* imsVoIPConfigInfo::pAmrWbEnable

8.288.2.4 WORD* imsVoIPConfigInfo::pAmrWBMode

8.288.2.5 BYTE* imsVoIPConfigInfo::pAmrWBOctetAligned

8.288.2.6 WORD* imsVoIPConfigInfo::pMinSessionExpiryTimer

8.288.2.7 WORD* imsVoIPConfigInfo::pRingBackTimer

8.288.2.8 WORD* imsVoIPConfigInfo::pRingingTimer

8.288.2.9 WORD* imsVoIPConfigInfo::pRTPRTCPInactTimer

8.288.2.10 BYTE* imsVoIPConfigInfo::pScrAmrEnable

8.288.2.11 BYTE* imsVoIPConfigInfo::pScrAmrWbEnable

8.288.2.12 WORD* imsVoIPConfigInfo::pSessionExpiryTimer

8.289 IndFieldsList Struct Reference

Data Fields

- BYTE indicationFieldsLen
- BYTE indicationFields [256]

8.289.1 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.

8.289.2 Field Documentation

8.289.2.1 BYTE IndFieldsList::indicationFields[256]

8.289.2.2 BYTE IndFieldsList::indicationFieldsLen

8.290 infoInterFreq Struct Reference

Data Fields

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

8.290.1 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.

8.290.2 Field Documentation

8.290.2.1 BYTE infoInterFreq::cell_resel_priority

8.290.2.2 cellParams infoInterFreq::cellInterFreqParams[255]

8.290.2.3 BYTE infoInterFreq::cells_len

8.290.2.4 WORD infoInterFreq::earfcn

8.290.2.5 BYTE infoInterFreq::threshXHigh

8.290.2.6 BYTE infoInterFreq::threshXLow

8.291 IOThresh Struct Reference

Data Fields

- BYTE IOThresListLen
- INT32 * pIOThresList

8.291.1 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

8.291.2 Field Documentation

8.291.2.1 BYTE IOThresh::IOThresListLen

8.291.2.2 INT32* IOThresh::pIOThresList

8.292 IPv4Addr Struct Reference

Data Fields

- ULONG addr
- ULONG subnetMask

8.292.1 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

8.292.2 Field Documentation

8.292.2.1 ULONG IPv4Addr::addr

8.292.2.2 ULONG IPv4Addr::subnetMask

8.293 IPv4Info Struct Reference

Data Fields

- ULONG address
- WORD port
- BYTE TlvPresent

8.293.1 Detailed Description

This structure contains LOC Server IPV4 info field.

Parameters

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

8.293.2 Field Documentation

8.293.2.1 ULONG IPv4Info::address

8.293.2.2 WORD IPv4Info::port

8.293.2.3 BYTE IPv4Info::TlvPresent

8.294 IPv6Addr Struct Reference

Data Fields

- BYTE addr [16]
- BYTE prefixLen

8.294.1 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.

8.294.2 Field Documentation

8.294.2.1 BYTE IPv6Addr::addr[16]

8.294.2.2 BYTE IPv6Addr::prefixLen

8.295 IPV6AddressInfo Struct Reference

Data Fields

- BYTE IPV6PrefixLen
- USHORT IPAddressV6 [8]

8.295.1 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.

8.295.2 Field Documentation

8.295.2.1 USHORT IPV6AddressInfo::IPAddressV6[8]

8.295.2.2 BYTE IPV6AddressInfo::IPV6PrefixLen

8.296 ipv6AddressInfo Struct Reference

Data Fields

- uint8_t IPV6PrefixLen
- uint16_t IPAddressV6 [8]

8.296.1 Detailed Description

This structure contains the IPv6 address information

Parameters

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

8.296.2 Field Documentation

8.296.2.1 uint16_t ipv6AddressInfo::IPAddressV6[8]

8.296.2.2 uint8_t ipv6AddressInfo::IPV6PrefixLen

8.297 IPV6GWAddressInfo Struct Reference

Data Fields

- BYTE gwV6PrefixLen
- USHORT gwAddressV6 [8]

8.297.1 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.

8.297.2 Field Documentation

8.297.2.1 USHORT IPV6GWAddressInfo::gwAddressV6[8]

8.297.2.2 BYTE IPV6GWAddressInfo::gwV6PrefixLen

8.298 IPv6Info Struct Reference

Data Fields

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

8.298.1 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

8.298.2 Field Documentation

8.298.2.1 WORD IPv6Info::address[8]

8.298.2.2 ULONG IPv6Info::port

8.298.2.3 BYTE IPv6Info::TlvPresent

8.299 IPv6TrafCls Struct Reference

Data Fields

- BYTE val
- BYTE mask

8.299.1 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

8.299.2 Field Documentation

8.299.2.1 BYTE IPv6TrafCls::mask

8.299.2.2 BYTE IPv6TrafCls::val

8.300 LibPackGPRSRequestedQoS Struct Reference

Data Fields

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

8.300.1 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

8.300.2 Field Documentation

8.300.2.1 uint32_t LibPackGPRSRequestedQoS::delayClass

8.300.2.2 uint32_t LibPackGPRSRequestedQoS::meanThroughputClass

8.300.2.3 uint32_t LibPackGPRSRequestedQoS::peakThroughputClass

8.300.2.4 uint32_t LibPackGPRSRequestedQoS::precedenceClass

8.300.2.5 uint32_t LibPackGPRSRequestedQoS::reliabilityClass

8.301 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

8.301.1 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 • NULL pointer - Invalid data.
pProfilename-Size;	• This parameter is an input parameter and should be initialised to the size of pProfileName field. Size of this parameter is 2 uint8_ts. • NULL pointer - Invalid data.
pPDPType	• Packet Data Protocol (PDP) type specifies the type of data payload exchanged over the air link when the packet data session is established with this profile – 0x00 - PDP-IP (IPv4) – 0x01 - PDP-PPP – 0x02 - PDP-IPV6 – 0x03 - PDP-IPV4V6 • NULL pointer - Invalid data.
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 • NULL pointer - Invalid data.

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 • NULL pointer - Invalid data.
pAPNName	• Access point name • NULL pointer - Invalid data.
pAPNnameSize;	• This parameter is an input parameter and should be initialised to the size of pAPNName field. Size of this parameter is 2 uint8_ts. • NULL pointer - Invalid data.
pPriDNSIPv4AddPref	• Primary DNS IPv4 Address Preference • NULL pointer - Invalid data.
pSecDNSIPv4AddPref	• Secondary DNS IPv4 Address Preference • NULL pointer - Invalid data.
pUMTSReqQoS	• UMTS Requested QoS • NULL pointer - Invalid data.
pUMTSMinQoS	• UMTS Minimum QoS • NULL pointer - Invalid data.
pGPRSRequestedQoS	• GPRS Requested QoS • NULL pointer - Invalid data.
pGPRSMinimumQoS	• GPRS Minimum QoS • NULL pointer - Invalid data.
pUsername	• User name • NULL pointer - Invalid data.
pUsernameSize;	• This parameter is an input parameter and should be initialised to the size of pUsername field. Size of this parameter is 2 uint8_ts. • NULL pointer - Invalid data.

pPassword	• Password • NULL pointer - Invalid data.
pPasswordSize;	• This parameter is an input parameter and should be initialised to the size of pPassword field. Size of this parameter is 2 uint8_ts. • NULL pointer - Invalid data.
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. • NULL pointer - Invalid data.
pIPv4AddrPref	• IPv4 Address Preference • NULL pointer - Invalid data.
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 • NULL pointer - Invalid data.
pPdpAccess-ConFlag	• PDP access control flag – 0 - PDP access control none – 1 - PDP access control reject – 2 - PDP access control permission • NULL pointer - Invalid data.
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 • NULL pointer - Invalid data.

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 • NULL pointer - Invalid data.
pTFTID1Params	• Traffic Flow Template • NULL pointer - Invalid data.
pTFTID2Params	• Traffic Flow Template • NULL pointer - Invalid data.
pPdpContext	• PDP context number • NULL pointer - Invalid data.
pSecondaryFlag	• PDP context secondary flag – 1 - (TRUE) implies this is secondary profile – 0 - (FALSE) implies this is not secondary profile • NULL pointer - Invalid data.
pPrimaryID	• PDP context primary ID • function SLQSGetProfileSettings() returns a default value 0xFF if this parameter is not returned by the device • NULL pointer - Invalid data.
pIPv6AddPref	• IPv6 address preference Preferred IPv6 address to be assigned to the TE; actual assigned address is negotiated with the network and may differ from this value; if not specified, the IPv6 address is obtained automatically from the network • NULL pointer - Invalid data.
pUMTSReqQoS-SigInd	• UMTS requested QoS with Signalling Indication flag • NULL pointer - Invalid data.
pUMTSMinQoS-SigInd	• UMTS minimum QoS with Signalling Indication flag • NULL pointer - Invalid data.
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 • NULL pointer - Invalid data.

pSecondaryDN-SIPv6addpref	• Secondary DNS IPv6 address preference • NULL pointer - Invalid data.
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 • NULL pointer - Invalid data.
pQosClassID	• 3GPP LTE QoS parameters • NULL pointer - Invalid data.
pAPNDisabled-Flag	• Optional 1 uint8_t Flag indicating if the APN is disabled/enabled • If set, the profile can not be used for making data calls • Any data call is failed locally • Values: – 0 - FALSE(default) – 1 - True • This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings(). • NULL pointer - Invalid data.
pPDNInactiv-Timeout	• Optional 4 uint8_ts indicating the duration of inactivity timer in seconds • If the PDP context/PDN connection is inactive for this duration i.e. No data Tx/Rx occurs, the PDP context/PDN connection is disconnected • Default value of zero indicates infinite value • This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings(). • NULL pointer - Invalid data.
pAPNClass	• Optional 1 uint8_t numeric identifier representing the APN in profile • Can be set and queried but is not used by the modem • This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings(). • NULL pointer - Invalid data.

8.301.2 Field Documentation

8.301.2.1 uint8_t* LibpackProfile3GPP::pAddrAllocPref

8.301.2.2 uint8_t* LibpackProfile3GPP::pAPNClass

8.301.2.3 uint8_t* LibpackProfile3GPP::pAPNDisabledFlag

8.301.2.4 uint8_t* LibpackProfile3GPP::pAPNName

8.301.2.5 uint16_t* LibpackProfile3GPP::pAPNnameSize

8.301.2.6 uint8_t* LibpackProfile3GPP::pAuthenticationPref

8.301.2.7 LibPackGPRSRequestedQoS* LibpackProfile3GPP::pGPRSMinimumQoS

8.301.2.8 LibPackGPRSRequestedQoS* LibpackProfile3GPP::pGPRSRequestedQos

8.301.2.9 uint8_t* LibpackProfile3GPP::pImCnFlag

8.301.2.10 uint32_t* LibpackProfile3GPP::pIPv4AddrPref

8.301.2.11 uint16_t* LibpackProfile3GPP::pIPv6AddPref

8.301.2.12 uint8_t* LibpackProfile3GPP::pPassword

8.301.2.13 uint16_t* LibpackProfile3GPP::pPasswordSize

8.301.2.14 uint8_t* LibpackProfile3GPP::pPcscfAddrUsingDhcp

8.301.2.15 uint8_t* LibpackProfile3GPP::pPcscfAddrUsingPCO

8.301.2.16 uint32_t* LibpackProfile3GPP::pPDNInactivTimeout

8.301.2.17 uint8_t* LibpackProfile3GPP::pPdpAccessConFlag

8.301.2.18 uint8_t* LibpackProfile3GPP::pPdpContext

8.301.2.19 uint8_t* LibpackProfile3GPP::pPdpDataCompType

8.301.2.20 uint8_t* LibpackProfile3GPP::pPdpHdrCompType

8.301.2.21 uint8_t* LibpackProfile3GPP::pPDPtype

8.301.2.22 uint32_t* LibpackProfile3GPP::pPriDNSIPv4AddPref

8.301.2.23 uint16_t* LibpackProfile3GPP::pPriDNSIPv6addpref

8.301.2.24 uint8_t* LibpackProfile3GPP::pPrimaryID

8.301.2.25 uint8_t* LibpackProfile3GPP::pProfilename

8.301.2.26 uint16_t* LibpackProfile3GPP::pProfilenameSize

8.301.2.27 LibPackQosClassID* LibpackProfile3GPP::pQosClassID

8.301.2.28 uint32_t* LibpackProfile3GPP::pSecDNSIPv4AddPref

8.301.2.29 uint16_t* LibpackProfile3GPP::pSecDNSIPv6addpref

8.301.2.30 uint8_t* LibpackProfile3GPP::pSecondaryFlag

8.301.2.31 LibPackTFTIDParams* LibpackProfile3GPP::pTFTID1Params

8.301.2.32 **LibPackTFTIDParams*** LibpackProfile3GPP::pTFTID2Params

8.301.2.33 **LibPackUMTSQoS*** LibpackProfile3GPP::pUMTSMinQoS

8.301.2.34 **LibPackUMTSReqQoSSigInd*** LibpackProfile3GPP::pUMTSMinQosSigInd

8.301.2.35 **LibPackUMTSQoS*** LibpackProfile3GPP::pUMTSReqQoS

8.301.2.36 **LibPackUMTSReqQoSSigInd*** LibpackProfile3GPP::pUMTSReqQoSSigInd

8.301.2.37 **uint8_t*** LibpackProfile3GPP::pUsername

8.301.2.38 **uint16_t*** LibpackProfile3GPP::pUsernameSize

8.302 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

8.302.1 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().
pPDNInactiv-Timeout3GPP2	• Optional 4 uint8_ts indicating the duration of inactivity timer in seconds • If the PDP context/PDN connection is inactive for this duration i.e. No data Tx/Rx occurs, the PDP context/PDN connection is disconnected • Default value of zero indicates infinite value • This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings().
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().

8.302.2 Field Documentation

8.302.2.1 uint8_t* LibpackProfile3GPP2::pAllowLinger

8.302.2.2 uint8_t* LibpackProfile3GPP2::pAPNClass3GPP2

8.302.2.3 uint8_t* LibpackProfile3GPP2::pAPNEnabled3GPP2

8.302.2.4 uint8_t* LibpackProfile3GPP2::pApnString

8.302.2.5 uint16_t* LibpackProfile3GPP2::pApnStringSize

8.302.2.6 uint8_t* LibpackProfile3GPP2::pAppPriority

8.302.2.7 uint32_t* LibpackProfile3GPP2::pAppType

8.302.2.8 uint8_t* LibpackProfile3GPP2::pAuthPassword

8.302.2.9 uint16_t* LibpackProfile3GPP2::pAuthPasswordSize

8.302.2.10 uint8_t* LibpackProfile3GPP2::pAuthProtocol

8.302.2.11 uint8_t* LibpackProfile3GPP2::pAuthRetryCount

8.302.2.12 uint16_t* LibpackProfile3GPP2::pAuthTimeout

8.302.2.13 uint8_t* LibpackProfile3GPP2::pDataMode

8.302.2.14 uint8_t* LibpackProfile3GPP2::pDataRate

8.302.2.15 uint16_t* LibpackProfile3GPP2::plpcpAckTimeout

8.302.2.16 uint8_t* LibpackProfile3GPP2::plpcpCreqRetryCount

8.302.2.17 uint8_t* LibpackProfile3GPP2::plsPcscfAddressNedded

8.302.2.18 uint16_t* LibpackProfile3GPP2::pLcpAckTimeout

8.302.2.19 uint8_t* LibpackProfile3GPP2::pLcpCreqRetryCount

8.302.2.20 uint8_t* LibpackProfile3GPP2::pNegoDnsSrvrPref

8.302.2.21 uint32_t* LibpackProfile3GPP2::pPDNInactivTimeout3GPP2

8.302.2.22 uint8_t* LibpackProfile3GPP2::pPdnType

8.302.2.23 uint32_t* LibpackProfile3GPP2::pPppSessCloseTimer1x

8.302.2.24 uint32_t* LibpackProfile3GPP2::pPppSessCloseTimerDO

8.302.2.25 uint32_t* LibpackProfile3GPP2::pPrimaryV4DnsAddress

8.302.2.26 uint16_t* LibpackProfile3GPP2::pPriV6DnsAddress

8.302.2.27 uint8_t* LibpackProfile3GPP2::pRATType

8.302.2.28 uint32_t* LibpackProfile3GPP2::pSecondaryV4DnsAddress

8.302.2.29 uint16_t* LibpackProfile3GPP2::pSecV6DnsAddress

8.302.2.30 uint8_t* LibpackProfile3GPP2::pUserId

8.302.2.31 uint16_t* LibpackProfile3GPP2::pUserIdSize

8.303 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

8.303.1 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

pPdpDataCompType	• PDP data compression type – 0 - PDP data compression is OFF – 1 - Manufacturer preferred compression – 2 - V.42BIS data compression – 3 - V.44 data compression
pAPNName	• Access point name
pAPNnameSize;	• This parameter is an input parameter and should be initialised to the size of pAPNName field. Size of this parameter is 2 bytes.
pPriDNSIPv4AddPref	• Primary DNS IPv4 Address Preference
pSecDNSIPv4AddPref	• Secondary DNS IPv4 Address Preference
pUMTSReqQoS	• UMTS Requested QoS
pUMTSMinQoS	• UMTS Minimum QoS
pGPRSRequestedQoS	• GPRS Minimum QoS
pUsername	• User name
pUsernameSize;	• This parameter is an input parameter and should be initialised to the size of pUsername field. Size of this parameter is 2 bytes.
pPassword	• Password
pPasswordSize;	• This parameter is an input parameter and should be initialised to the size of pPassword field. Size of this parameter is 2 bytes.

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

pPrimaryID	• PDP context primary ID • function SLQSGetProfileSettings() returns a default value 0xFF if this parameter is not returned by the device
pIPv6AddPref	• IPv6 address preference Preferred IPv6 address to be assigned to the TE; actual assigned address is negotiated with the network and may differ from this value; if not specified, the IPv6 address is obtained automatically from the network
pUMTSReqQoS-SigInd	• UMTS requested QoS with Signalling Indication flag
pUMTSMinQoS-SigInd	• UMTS minimum QoS with Signalling Indication flag
pPrimaryDNSI-Pv6addpref	• Primary DNS IPv6 address preference – The value may be used as a preference during negotiation with the network; if not specified, the wireless device will attempt to obtain the DNS address automatically from the network; the negotiated value is provided to the host via DHCP
pSecondaryDN-SIPv6addpref	• Secondary DNS IPv6 address preference
paddrAllocation-Pref	• DHCP/NAS preference – This enumerated value may be used to indicate the address allocation preference * 0 - NAS signaling is used for address allocation * 1 - DHCP is used for address allocation
pQosClassID	• 3GPP LTE QoS parameters
pAPNDisabled-Flag	• Optional 1 uint8_t Flag indicating if the APN is disabled/enabled • If set, the profile can not be used for making data calls • Any data call is failed locally • Values: – 0 - FALSE(default) – 1 - True • This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings().
pPDNInactiv-Timeout	• Optional 4 Bytes indicating the duration of inactivity timer in seconds • If the PDP context/PDN connection is inactive for this duration i.e. No data Tx/Rx occurs, the PDP context/PDN connection is disconnected • Default value of zero indicates infinite value • This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings().

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

8.303.2 Field Documentation

8.303.2.1 uint8_t* LibPackprofile_3GPP::pAddrAllocPref

8.303.2.2 uint8_t* LibPackprofile_3GPP::pAPNClass

8.303.2.3 uint8_t* LibPackprofile_3GPP::pAPNDisabledFlag

8.303.2.4 uint8_t* LibPackprofile_3GPP::pAPNName

8.303.2.5 uint16_t* LibPackprofile_3GPP::pAPNnameSize

8.303.2.6 uint8_t* LibPackprofile_3GPP::pAuthenticationPref

8.303.2.7 LibPackGPRSRequestedQoS* LibPackprofile_3GPP::pGPRSMinimumQoS

8.303.2.8 LibPackGPRSRequestedQoS* LibPackprofile_3GPP::pGPRSRequestedQos

8.303.2.9 uint8_t* LibPackprofile_3GPP::pImCnFlag

8.303.2.10 uint32_t* LibPackprofile_3GPP::pIPv4AddrPref

8.303.2.11 uint16_t* LibPackprofile_3GPP::pIPv6AddPref

8.303.2.12 uint8_t* LibPackprofile_3GPP::pPassword

8.303.2.13 uint16_t* LibPackprofile_3GPP::pPasswordSize

8.303.2.14 uint8_t* LibPackprofile_3GPP::pPcscfAddrUsingDhcp

8.303.2.15 uint8_t* LibPackprofile_3GPP::pPcscfAddrUsingPCO

8.303.2.16 uint32_t* LibPackprofile_3GPP::pPDNInactivTimeout

8.303.2.17 uint8_t* LibPackprofile_3GPP::pPdpAccessConFlag

8.303.2.18 uint8_t* LibPackprofile_3GPP::pPdpContext

8.303.2.19 uint8_t* LibPackprofile_3GPP::pPdpDataCompType

8.303.2.20 uint8_t* LibPackprofile_3GPP::pPdpHdrCompType

8.303.2.21 uint8_t* LibPackprofile_3GPP::pPDPtype

8.303.2.22 uint32_t* LibPackprofile_3GPP::pPriDNSIPv4AddPref

8.303.2.23 uint16_t* LibPackprofile_3GPP::pPriDNSIPv6addpref

8.303.2.24 uint8_t* LibPackprofile_3GPP::pPrimaryID

8.303.2.25 uint8_t* LibPackprofile_3GPP::pProfilename

8.303.2.26 uint16_t* LibPackprofile_3GPP::pProfilenameSize

8.303.2.27 LibPackQosClassID* LibPackprofile_3GPP::pQosClassID

8.303.2.28 uint32_t* LibPackprofile_3GPP::pSecDNSIPv4AddPref

8.303.2.29 uint16_t* LibPackprofile_3GPP::pSecDNSIPv6addpref

8.303.2.30 uint8_t* LibPackprofile_3GPP::pSecondaryFlag

8.303.2.31 LibPackTFTIDParams* LibPackprofile_3GPP::pTFTID1Params

8.303.2.32 LibPackTFTIDParams* LibPackprofile_3GPP::pTFTID2Params

8.303.2.33 LibPackUMTSQoS* LibPackprofile_3GPP::pUMTSMinQoS

8.303.2.34 LibPackUMTSReqQoSSigInd* LibPackprofile_3GPP::pUMTSMinQosSigInd

8.303.2.35 LibPackUMTSQoS* LibPackprofile_3GPP::pUMTSReqQoS

8.303.2.36 LibPackUMTSReqQoSSigInd* LibPackprofile_3GPP::pUMTSReqQoSSigInd

8.303.2.37 uint8_t* LibPackprofile_3GPP::pUsername

8.303.2.38 uint16_t* LibPackprofile_3GPP::pUsernameSize

8.304 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

8.304.1 Detailed Description

This structure contains the 3GPP2 profile parameters

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

Parameters

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

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

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

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

8.304.2 Field Documentation

8.304.2.1 uint8_t* LibPackprofile_3GPP2::pAllowLinger

8.304.2.2 uint8_t* LibPackprofile_3GPP2::pAPNClass3GPP2

8.304.2.3 uint8_t* LibPackprofile_3GPP2::pAPNEnabled3GPP2

8.304.2.4 uint8_t* LibPackprofile_3GPP2::pApnString

8.304.2.5 uint16_t* LibPackprofile_3GPP2::pApnStringSize

8.304.2.6 uint8_t* LibPackprofile_3GPP2::pAppPriority

8.304.2.7 uint32_t* LibPackprofile_3GPP2::pAppType

8.304.2.8 uint8_t* LibPackprofile_3GPP2::pAuthPassword

8.304.2.9 uint16_t* LibPackprofile_3GPP2::pAuthPassword_tSize

8.304.2.10 uint8_t* LibPackprofile_3GPP2::pAuthProtocol

8.304.2.11 uint8_t* LibPackprofile_3GPP2::pAuthRetryCount

8.304.2.12 uint16_t* LibPackprofile_3GPP2::pAuthTimeout

8.304.2.13 uint8_t* LibPackprofile_3GPP2::pDataMode

8.304.2.14 uint8_t* LibPackprofile_3GPP2::pDataRate

8.304.2.15 uint16_t* LibPackprofile_3GPP2::pIpcpAckTimeout

8.304.2.16 uint8_t* LibPackprofile_3GPP2::pIpcpCreqRetryCount

8.304.2.17 uint8_t* LibPackprofile_3GPP2::pIsPcscfAddressNedded

8.304.2.18 uint16_t* LibPackprofile_3GPP2::pLcpAckTimeout

8.304.2.19 uint8_t* LibPackprofile_3GPP2::pLcpCreqRetryCount

8.304.2.20 uint8_t* LibPackprofile_3GPP2::pNegoDnsSrvrPref

8.304.2.21 uint32_t* LibPackprofile_3GPP2::pPDNInactivTimeout3GPP2

8.304.2.22 uint8_t* LibPackprofile_3GPP2::pPdnType

8.304.2.23 uint32_t* LibPackprofile_3GPP2::pPppSessCloseTimer1x

8.304.2.24 uint32_t* LibPackprofile_3GPP2::pPppSessCloseTimerDO

8.304.2.25 uint32_t* LibPackprofile_3GPP2::pPrimaryV4DnsAddress

8.304.2.26 uint16_t* LibPackprofile_3GPP2::pPriV6DnsAddress

8.304.2.27 uint8_t* LibPackprofile_3GPP2::pRATType

8.304.2.28 uint32_t* LibPackprofile_3GPP2::pSecondaryV4DnsAddress

8.304.2.29 uint16_t* LibPackprofile_3GPP2::pSecV6DnsAddress

8.304.2.30 uint8_t* LibPackprofile_3GPP2::pUserId

8.304.2.31 uint16_t* LibPackprofile_3GPP2::pUserIdSize

8.305 LibPackQosClassID Struct Reference

Data Fields

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

8.305.1 Detailed Description

This structure contains 3GPP LTE QoS parameters

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

Parameters

QCI	• QOS specified using the QOS Class Identifier (QOS) values QCI value 0 - Requests the network to assign the appropriate QCI value QCI values 1-4 - Associated with guaranteed bit rates QCI values 5-9 - Associated with non-guaranteed bit rates
gDlBitRate	• Guaranteed DL bit rate
maxDlBitRate	• Maximum DL bit rate
gUlBitRate	• Guaranteed UL bit rate
maxUlBitRate	• Maximum UL bit rate

8.305.2 Field Documentation

8.305.2.1 uint8_t LibPackQosClassID::gDlBitRate

8.305.2.2 uint32_t LibPackQosClassID::gUlBitRate

8.305.2.3 uint32_t LibPackQosClassID::maxDlBitRate

8.305.2.4 uint32_t LibPackQosClassID::maxUlBitRate

8.305.2.5 uint8_t LibPackQosClassID::QCI

8.306 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

8.306.1 Detailed Description

This structure contains traffic flow template parameters

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

Parameters

filterId	• Filter identifier
eValid	• Evaluation precedence index
ipVersion	• IP version number – 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

8.306.2 Field Documentation

8.306.2.1 uint16_t LibPackTFTIDParams::destPortRangeEnd

8.306.2.2 uint32_t LibPackTFTIDParams::destPortRangeStart

8.306.2.3 uint8_t LibPackTFTIDParams::eValid

8.306.2.4 uint8_t LibPackTFTIDParams::filterId

8.306.2.5 uint32_t LibPackTFTIDParams::flowLabel

8.306.2.6 uint32_t LibPackTFTIDParams::IPSECSPI

8.306.2.7 uint8_t LibPackTFTIDParams::ipVersion

8.306.2.8 uint8_t LibPackTFTIDParams::nextHeader

8.306.2.9 uint16_t* LibPackTFTIDParams::pSourceIP

8.306.2.10 uint8_t LibPackTFTIDParams::sourceIPMask

8.306.2.11 uint16_t LibPackTFTIDParams::srcPortRangeEnd

8.306.2.12 uint16_t LibPackTFTIDParams::srcPortRangeStart

8.306.2.13 uint16_t LibPackTFTIDParams::tosMask

8.307 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

8.307.1 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 • 0xff - Invalid UMTS Quality Of Service Information.
maxUplinkBitrate	• Maximum uplink bit rate in bits/sec
maxDownlink-Bitrate	• Maximum downlink bit rate in bits/sec
grntUplinkBitrate	• Guaranteed uplink bit rate in bits/sec
grntDownlink-Bitrate	• Guranteed downlink bit rate in bits/sec
qosDelivery-Order	- Qos delivery order • 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 - $1*10^{\wedge}(-2)$ • 0x02 - $7*10^{\wedge}(-3)$ • 0x03 - $1*10^{\wedge}(-3)$ • 0x04 - $1*10^{\wedge}(-4)$ • 0x05 - $1*10^{\wedge}(-5)$ • 0x06 - $1*10^{\wedge}(-6)$ • 0x07 - $1*10^{\wedge}(-1)$

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.

Check trafficClass before use.

8.307.2 Field Documentation

8.307.2.1 uint8_t LibPackUMTSQoS::deliveryErrSDU

8.307.2.2 uint32_t LibPackUMTSQoS::grntDownlinkBitrate

8.307.2.3 uint32_t LibPackUMTSQoS::grntUplinkBitrate

8.307.2.4 uint32_t LibPackUMTSQoS::maxDownlinkBitrate

8.307.2.5 uint32_t LibPackUMTSQoS::maxSDUSize

8.307.2.6 uint32_t LibPackUMTSQoS::maxUplinkBitrate

8.307.2.7 uint8_t LibPackUMTSQoS::qosDeliveryOrder

8.307.2.8 uint8_t LibPackUMTSQoS::resBerRatio

8.307.2.9 uint8_t LibPackUMTSQoS::sduErrorRatio

8.307.2.10 uint8_t LibPackUMTSQoS::trafficClass

8.307.2.11 uint32_t LibPackUMTSQoS::trafficPriority

8.307.2.12 uint32_t LibPackUMTSQoS::transferDelay

8.308 LibPackUMTSReqQoSSigInd Struct Reference

Data Fields

- LibPackUMTSQoS UMTSReqQoS
- uint8_t SigInd

8.308.1 Detailed Description

This structure contains UMTS requested QoS with Signaling Indication flag

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

Parameters

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

8.308.2 Field Documentation

8.308.2.1 uint8_t LibPackUMTSReqQoSSigInd::SigInd

8.308.2.2 LibPackUMTSQoS LibPackUMTSReqQoSSigInd::UMTSReqQoS

8.309 lineCtrlInfo Struct Reference

Data Fields

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

8.309.1 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

8.309.2 Field Documentation

8.309.2.1 BYTE lineCtrlInfo::polarityIncluded

8.309.2.2 BYTE lineCtrlInfo::pwrDenialTime

8.309.2.3 BYTE lineCtrlInfo::revPolarity

8.309.2.4 BYTE lineCtrlInfo::toggleMode

8.310 loc_BdsSV Struct Reference

Data Fields

- uint16_t id
- uint8_t mask

8.310.1 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

8.310.2 Field Documentation

8.310.2.1 uint16_t loc_BdsSV::id

8.310.2.2 uint8_t loc_BdsSV::mask

8.311 loc_BdsSVInfo Struct Reference

Data Fields

- uint8_t len
- loc_BdsSV * pSV

8.311.1 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

8.311.2 Field Documentation

8.311.2.1 uint8_t loc_BdsSVInfo::len

8.311.2.2 loc_BdsSV* loc_BdsSVInfo::pSV

8.312 loc_CellDb Struct Reference

Data Fields

- uint32_t mask

8.312.1 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

8.312.2 Field Documentation

8.312.2.1 uint32_t loc_CellDb::mask

8.313 loc_ClkInfo Struct Reference

Data Fields

- uint32_t mask

8.313.1 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

8.313.2 Field Documentation

8.313.2.1 uint32_t loc_ClkInfo::mask

8.314 loc_GnssData Struct Reference

Data Fields

- uint64_t mask

8.314.1 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

8.314.2 Field Documentation

8.314.2.1 uint64_t loc_GnssData::mask

8.315 loc_gpsTime Struct Reference

Data Fields

- uint16_t gpsWeek
- uint32_t gpsTimeOfWeekMs

8.315.1 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

8.315.2 Field Documentation

8.315.2.1 uint32_t loc_gpsTime::gpsTimeOfWeekMs

8.315.2.2 uint16_t loc_gpsTime::gpsWeek

8.316 loc_IPv4Info Struct Reference

Data Fields

- uint32_t address
- uint16_t port
- uint8_t TlvPresent

8.316.1 Detailed Description

This structure contains LOC Server IPV4 info field.

Parameters

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

8.316.2 Field Documentation

8.316.2.1 uint32_t loc_IPv4Info::address

8.316.2.2 uint16_t loc_IPv4Info::port

8.316.2.3 uint8_t loc_IPv4Info::TlvPresent

8.317 loc_IPv6Info Struct Reference

Data Fields

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

8.317.1 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

8.317.2 Field Documentation

8.317.2.1 uint16_t loc_IPv6Info::address[8]

8.317.2.2 uint32_t loc_IPv6Info::port

8.317.2.3 uint8_t loc_IPv6Info::TlvPresent

8.318 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

8.318.1 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

8.318.2 Field Documentation

8.318.2.1 uint8_t loc_LocApplicationInfo::appNameLength

8.318.2.2 uint8_t loc_LocApplicationInfo::appProviderLength

8.318.2.3 uint8_t loc_LocApplicationInfo::appVersionLength

8.318.2.4 uint8_t loc_LocApplicationInfo::appVersionValid

8.318.2.5 uint8_t* loc_LocApplicationInfo::pAppName

8.318.2.6 uint8_t* loc_LocApplicationInfo::pAppProvider

8.318.2.7 uint8_t* loc_LocApplicationInfo::pAppVersion

8.319 loc_precisionDilution Struct Reference

Data Fields

- uint32_t PDOP

- uint32_t HDOP
- uint32_t VDOP

8.319.1 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)

8.319.2 Field Documentation

8.319.2.1 uint32_t loc_precisionDilution::HDOP

8.319.2.2 uint32_t loc_precisionDilution::PDOP

8.319.2.3 uint32_t loc_precisionDilution::VDOP

8.320 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

8.320.1 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

8.320.2 Field Documentation

8.320.2.1 float loc_satelliteInfo::azimuth

8.320.2.2 float loc_satelliteInfo::elevation

8.320.2.3 uint16_t loc_satelliteInfo::gnssSvId

8.320.2.4 uint8_t loc_satelliteInfo::healthStatus

8.320.2.5 float loc_satelliteInfo::snr

8.320.2.6 uint8_t loc_satelliteInfo::svInfoMask

8.320.2.7 uint8_t loc_satelliteInfo::svListLen

8.320.2.8 uint32_t loc_satelliteInfo::svStatus

8.320.2.9 uint32_t loc_satelliteInfo::system

8.320.2.10 uint32_t loc_satelliteInfo::validMask

8.321 loc_sensorDataUsage Struct Reference

Data Fields

- uint32_t usageMask
- uint32_t aidingIndicatorMask

8.321.1 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

8.321.2 Field Documentation

8.321.2.1 uint32_t loc_sensorDataUsage::aidingIndicatorMask

8.321.2.2 uint32_t loc_sensorDataUsage::usageMask

8.322 loc_SV Struct Reference

Data Fields

- uint16_t id
- uint32_t system
- uint8_t mask

8.322.1 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

8.322.2 Field Documentation

8.322.2.1 uint16_t loc_SV::id

8.322.2.2 uint8_t loc_SV::mask

8.322.2.3 uint32_t loc_SV::system

8.323 loc_SVInfo Struct Reference

Data Fields

- uint8_t len
- loc_SV * pSV

8.323.1 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

8.323.2 Field Documentation

8.323.2.1 uint8_t loc_SVInfo::len

8.323.2.2 loc_SV* loc_SVInfo::pSV

8.324 loc_svUsedforFix Struct Reference

Data Fields

- uint8_t gnssSvUsedList_len
- uint16_t gnssSvUsedList [255]

8.324.1 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

8.324.2 Field Documentation

8.324.2.1 uint16_t loc_svUsedforFix::gnssSvUsedList[255]

8.324.2.2 uint8_t loc_svUsedforFix::gnssSvUsedList_len

8.325 loc_urlAddr Struct Reference

Data Fields

- char address [256]
- uint TlvPresent

8.325.1 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

8.325.2 Field Documentation

8.325.2.1 char loc_urlAddr::address[256]

8.325.2.2 uint loc_urlAddr::TlvPresent

8.326 LocApplicationInfo Struct Reference

Data Fields

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

8.326.1 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

8.326.2 Field Documentation

8.326.2.1 BYTE LocApplicationInfo::appNameLength

8.326.2.2 BYTE LocApplicationInfo::appProviderLength

8.326.2.3 CHAR LocApplicationInfo::appVersionLength

8.326.2.4 BYTE LocApplicationInfo::appVersionValid

8.326.2.5 CHAR* LocApplicationInfo::pAppName

8.326.2.6 CHAR* LocApplicationInfo::pAppProvider

8.326.2.7 CHAR* LocApplicationInfo::pAppVersion

8.327 LocDelAssDataReq Struct Reference

Data Fields

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

8.327.1 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

8.327.2 Field Documentation

8.327.2.1 BdsSVInfo* LocDelAssDataReq::pBdsSVInfo

8.327.2.2 CellDb* LocDelAssDataReq::pCellDb

8.327.2.3 ClkInfo* LocDelAssDataReq::pClkInfo

8.327.2.4 GnssData* LocDelAssDataReq::pGnssData

8.327.2.5 SVInfo* LocDelAssDataReq::pSVInfo

8.328 LOCEventRegisterReqResp Struct Reference

Data Fields

- ULONGLONG eventRegister

8.328.1 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.

8.328.2 Field Documentation

8.328.2.1 ULONGLONG LOCEventRegisterReqResp::eventRegister

8.329 LOCExtPowerStateReqResp Struct Reference

Data Fields

- ULONG extPowerState

8.329.1 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

8.329.2 Field Documentation

8.329.2.1 ULONG LOCExtPowerStateReqResp::extPowerState

8.330 LocGetServerReq Struct Reference

Data Fields

- ULONG serverType
- BYTE * pServerAddrTypeMask

8.330.1 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

8.330.2 Field Documentation

8.330.2.1 BYTE* LocGetServerReq::pServerAddrTypeMask

8.330.2.2 ULONG LocGetServerReq::serverType

8.331 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

8.331.1 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.

8.331.2 Field Documentation

8.331.2.1 altitudeSrcInfo∗ LocInjectPositionReq::pAltitudeSrcInfo

8.331.2.2 ULONG∗ LocInjectPositionReq::pAltitudeWrtEllipsoid

8.331.2.3 ULONG∗ LocInjectPositionReq::pAltitudeWrtMeanSeaLevel

8.331.2.4 BYTE* LocInjectPositionReq::pHorConfidence

8.331.2.5 ULONG* LocInjectPositionReq::pHorReliability

8.331.2.6 ULONG* LocInjectPositionReq::pHorUncCircular

8.331.2.7 ULONGLONG* LocInjectPositionReq::pLatitude

8.331.2.8 ULONGLONG* LocInjectPositionReq::pLongitude

8.331.2.9 ULONG* LocInjectPositionReq::pPositionSrc

8.331.2.10 BYTE* LocInjectPositionReq::pRawHorConfidence

8.331.2.11 ULONG* LocInjectPositionReq::pRawHorUncCircular

8.331.2.12 ULONG* LocInjectPositionReq::pTimestampAge

8.331.2.13 ULONGLONG* LocInjectPositionReq::pTimestampUtc

8.331.2.14 BYTE* LocInjectPositionReq::pVertConfidence

8.331.2.15 ULONG* LocInjectPositionReq::pVertReliability

8.331.2.16 ULONG* LocInjectPositionReq::pVertUnc

8.332 LocInjectSensorDataReq Struct Reference

Data Fields

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

8.332.1 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

8.332.2 Field Documentation

8.332.2.1 sensorData* LocInjectSensorDataReq::pAcceleroData

8.332.2.2 tempratureData* LocInjectSensorDataReq::pAcceleroTempData

8.332.2.3 ULONG* LocInjectSensorDataReq::pAcceleroTimeSrc

8.332.2.4 sensorData* LocInjectSensorDataReq::pGyroData

8.332.2.5 tempratureData* LocInjectSensorDataReq::pGyroTempData

8.332.2.6 ULONG* LocInjectSensorDataReq::pGyroTimeSrc

8.332.2.7 ULONG* LocInjectSensorDataReq::pOpaqueIdentifier

8.333 LocSetCradleMountReq Struct Reference

Data Fields

- ULONG state
- BYTE * pConfidence

8.333.1 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

8.333.2 Field Documentation

8.333.2.1 BYTE* LocSetCradleMountReq::pConfidence

8.333.2.2 ULONG LocSetCradleMountReq::state

8.334 LOCStartReq Struct Reference

Data Fields

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

8.334.1 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

8.334.2 Field Documentation

8.334.2.1 struct LocApplicationInfo* LOCStartReq::pApplicationInfo

8.334.2.2 ULONG* LOCStartReq::pConfigAltitudeAssumed

8.334.2.3 ULONG* LOCStartReq::pHorizontalAccuracyLvl

8.334.2.4 ULONG* LOCStartReq::pIntermediateReportState

8.334.2.5 ULONG* LOCStartReq::pMinIntervalTime

8.334.2.6 ULONG* LOCStartReq::pRecurrenceType

8.334.2.7 BYTE LOCStartReq::SessionId

8.335 LOCStopReq Struct Reference

Data Fields

- BYTE sessionId

8.335.1 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

8.335.2 Field Documentation

8.335.2.1 BYTE LOCStopReq::sessionId

8.336 LteCQIParm Struct Reference

Data Fields

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

8.336.1 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

8.336.2 Field Documentation

8.336.2.1 BYTE LteCQIParm::CQIValueCW0

8.336.2.2 BYTE LteCQIParm::CQIValueCW1

8.336.2.3 BYTE LteCQIParm::ValidityCW0

8.336.2.4 BYTE LteCQIParm::ValidityCW1

8.337 lteEARFCN Struct Reference

Data Fields

- BYTE status
- ULONG earfcn0
- ULONG earfcn1

8.337.1 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.

8.337.2 Field Documentation

8.337.2.1 ULONG lteEARFCN::earfcn0

8.337.2.2 ULONG lteEARFCN::earfcn1

8.337.2.3 BYTE lteEARFCN::status

8.338 lteGsmCellInfo Struct Reference

Data Fields

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

8.338.1 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.

8.338.2 Field Documentation

8.338.2.1 BYTE lteGsmCellInfo::cellReselPriority

8.338.2.2 BYTE lteGsmCellInfo::cells_len

8.338.2.3 gsmCellInfo lteGsmCellInfo::GsmCellInfo[255]

8.338.2.4 BYTE lteGsmCellInfo::nccPermitted

8.338.2.5 BYTE lteGsmCellInfo::threshGsmHigh

8.338.2.6 BYTE lteGsmCellInfo::threshGsmLow

8.339 LTEInfo Struct Reference

Data Fields

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

8.339.1 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

8.339.2 Field Documentation

8.339.2.1 BYTE LTEInfo::band

8.339.2.2 BYTE LTEInfo::bandwidth

8.339.2.3 BYTE LTEInfo::emmConnState

8.339.2.4 BYTE LTEInfo::emmState

8.339.2.5 BYTE LTEInfo::emmSubState

8.339.2.6 WORD LTEInfo::RXChan

8.339.2.7 WORD LTEInfo::TXChan

8.340 LTEInfoInterfreq Struct Reference

Data Fields

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

8.340.1 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.

8.340.2 Field Documentation

8.340.2.1 BYTE LTEInfoInterfreq::freqsLen

8.340.2.2 infoInterFreq LTEInfoInterfreq::InfoInterfreq[255]

8.340.2.3 BYTE LTEInfoInterfreq::ueInIdle

8.341 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]

8.341.1 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.

8.341.2 Field Documentation

8.341.2.1 cellParams LTEInfoIntrafreq::CellParams[255]

8.341.2.2 BYTE LTEInfoIntrafreq::cellReselPriority

8.341.2.3 BYTE LTEInfoIntrafreq::cellsLen

8.341.2.4 WORD LTEInfoIntrafreq::earfcn

8.341.2.5 ULONG LTEInfoIntrafreq::globalCellId

8.341.2.6 BYTE LTEInfoIntrafreq::plmn[3]

8.341.2.7 WORD LTEInfoIntrafreq::servingCellId

8.341.2.8 BYTE LTEInfoIntrafreq::sIntraSearch

8.341.2.9 BYTE LTEInfoIntrafreq::sNonIntraSearch

8.341.2.10 WORD LTEInfoIntrafreq::tac

8.341.2.11 BYTE LTEInfoIntrafreq::threshServingLow

8.341.2.12 BYTE LTEInfoIntrafreq::ueInIdle

8.342 LTEInfoNeighboringGSM Struct Reference

Data Fields

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

8.342.1 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.

8.342.2 Field Documentation

8.342.2.1 BYTE LTEInfoNeighboringGSM::freqsLen

8.342.2.2 lteGsmCellInfo LTEInfoNeighboringGSM::LteGsmCellInfo[255]

8.342.2.3 BYTE LTEInfoNeighboringGSM::ueInIdle

8.343 LTEInfoNeighboringWCDMA Struct Reference

Data Fields

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

8.343.1 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.

8.343.2 Field Documentation

8.343.2.1 BYTE LTEInfoNeighboringWCDMA::freqsLen

8.343.2.2 lteWcdmaCellInfo LTEInfoNeighboringWCDMA::LTEWCDMACellInfo[255]

8.343.2.3 BYTE LTEInfoNeighboringWCDMA::ueInIdle

8.344 LteNasReleaseInfo_s Struct Reference

Data Fields

- BYTE nas_release
- BYTE nas_major
- BYTE nas_minor

8.344.1 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

8.344.2 Field Documentation

8.344.2.1 BYTE LteNasReleaseInfo_s::nas_major

8.344.2.2 BYTE LteNasReleaseInfo_s::nas_minor

8.344.2.3 BYTE LteNasReleaseInfo_s::nas_release

8.345 ltePCI Struct Reference

Data Fields

- BYTE status
- ULONG earfcn
- ULONG pci

8.345.1 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

8.345.2 Field Documentation

8.345.2.1 ULONG ltePCI::earfcn

8.345.2.2 ULONG ltePCI::pci

8.345.2.3 BYTE ltePCI::status

8.346 lteRsrpinformation Struct Reference

Data Fields

- SHORT rsrplevel

8.346.1 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).

8.346.2 Field Documentation

8.346.2.1 SHORT lteRsrpinformation::rsrplevel

8.347 LTERSRPThresh Struct Reference

Data Fields

- BYTE LTERSRPThreshListLen
- WORD * pLTERSRPThreshList

8.347.1 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).

8.347.2 Field Documentation

8.347.2.1 BYTE LTERSRPThresh::LTERSRPThreshListLen

8.347.2.2 WORD* LTERSRPThresh::pLTERSRPThreshList

8.348 LTERSRQThresh Struct Reference

Data Fields

- BYTE LTERSRQThreshListLen
- WORD * pLTERSRQThreshList

8.348.1 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)

8.348.2 Field Documentation

8.348.2.1 BYTE LTERSRQThresh::LTERSRQThreshListLen

8.348.2.2 WORD* LTERSRQThresh::pLTERSRQThreshList

8.349 LTERSSIThresh Struct Reference

Data Fields

- BYTE LTERSSIThreshListLen
- WORD * pLTERSSIThreshList

8.349.1 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)

8.349.2 Field Documentation

8.349.2.1 BYTE LTERSSIThresh::LTERSSIThreshListLen

8.349.2.2 WORD* LTERSSIThresh::pLTERSSIThreshList

8.350 LteSccRxInfoResp Struct Reference

Data Fields

- SccRxInfo * pSccRxInfo

8.350.1 Detailed Description

This structure contains information about the SLQSSwiGetLteSccRxInfo response parameters.

Parameters

pSccRxInfo	[Optional] • See SccRxInfo for more information

8.350.2 Field Documentation

8.350.2.1 SccRxInfo* LteSccRxInfoResp::pSccRxInfo

8.351 LTESigRptCfg Struct Reference

Data Fields

- BYTE rptRate
- BYTE avgPeriod

8.351.1 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

8.351.2 Field Documentation

8.351.2.1 BYTE LTESigRptCfg::avgPeriod

8.351.2.2 BYTE LTESigRptCfg::rptRate

8.352 LTESigRptConfig Struct Reference

Data Fields

- BYTE rptRate
- BYTE avgPeriod

8.352.1 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

8.352.2 Field Documentation

8.352.2.1 BYTE LTESigRptConfig::avgPeriod

8.352.2.2 BYTE LTESigRptConfig::rptRate

8.353 lteSnrinformation Struct Reference

Data Fields

- SHORT snrlevel

8.353.1 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.

8.353.2 Field Documentation

8.353.2.1 SHORT lteSnrinformation::snrlevel

8.354 LTESNRThresh Struct Reference

Data Fields

- BYTE LTESNRThresListLen
- SHORT * pLTESNRThresList

8.354.1 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

8.354.2 Field Documentation

8.354.2.1 BYTE LTESNRThresh::LTESNRThresListLen

8.354.2.2 SHORT* LTESNRThresh::pLTESNRThresList

8.355 LTESNRThreshold Struct Reference

Data Fields

- BYTE LTESNRThreshListLen
- WORD * pLTESNRThreshList

8.355.1 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).

8.355.2 Field Documentation

8.355.2.1 BYTE LTESNRThreshold::LTESNRThreshListLen

8.355.2.2 WORD* LTESNRThreshold::pLTESNRThreshList

8.356 LTESSInfo Struct Reference

Data Fields

- INT8 rssi
- INT8 rsrq
- SHORT rsrp
- SHORT 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 SHORT LTESSInfo::rsrp

8.356.2.2 INT8 LTESSInfo::rsrq

8.356.2.3 INT8 LTESSInfo::rssi

8.356.2.4 SHORT LTESSInfo::snr

8.357 lteSSInfo Struct Reference

Data Fields

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

8.357.1 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,

8.357.2 Field Documentation

8.357.2.1 int16_t lteSSInfo::rsrp

8.357.2.2 int8_t lteSSInfo::rsrq

8.357.2.3 int8_t lteSSInfo::rssi

8.357.2.4 int16_t lteSSInfo::snr

8.358 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

8.358.1 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

8.358.2 Field Documentation

8.358.2.1 ULONG LTESysInfo::cellId

8.358.2.2 BYTE LTESysInfo::cellIdValid

8.358.2.3 WORD LTESysInfo::lac

8.358.2.4 BYTE LTESysInfo::lacValid

8.358.2.5 BYTE LTESysInfo::MCC[3]

8.358.2.6 BYTE LTESysInfo::MNC[3]

8.358.2.7 BYTE LTESysInfo::networkIdValid

8.358.2.8 BYTE LTESysInfo::regRejectInfoValid

8.358.2.9 BYTE LTESysInfo::rejCause

8.358.2.10 BYTE LTESysInfo::rejectSrvDomain

8.358.2.11 sysInfoCommon LTESysInfo::sysInfoLTE

8.358.2.12 WORD LTESysInfo::tac

8.358.2.13 BYTE LTESysInfo::tacValid

8.359 lteWcdmaCellInfo Struct Reference

Data Fields

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

8.359.1 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.

8.359.2 Field Documentation

8.359.2.1 BYTE lteWcdmaCellInfo::cellReselPriority

8.359.2.2 BYTE lteWcdmaCellInfo::cellsLen

8.359.2.3 WORD lteWcdmaCellInfo::threshXhigh

8.359.2.4 WORD lteWcdmaCellInfo::threshXlow

8.359.2.5 WORD lteWcdmaCellInfo::uarfcn

8.359.2.6 wcdmaCellInfo lteWcdmaCellInfo::WCDMACellInfo[255]

8.360 messageModeTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- sMSMessageModeInfo MessageModeInfo

8.360.1 Detailed Description

This structure contains message mode TLV information.

Parameters

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

8.360.2 Field Documentation

8.360.2.1 sMSMessageModeInfo messageModeTlv::MessageModeInfo

8.360.2.2 uint8_t messageModeTlv::TlvPresent

8.361 messageWaitingInfoContent Struct Reference

Data Fields

- BYTE msgType
- BYTE activeInd
- BYTE msgCount

8.361.1 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

8.361.2 Field Documentation

8.361.2.1 BYTE messageWaitingInfoContent::activeInd

8.361.2.2 BYTE messageWaitingInfoContent::msgCount

8.361.2.3 BYTE messageWaitingInfoContent::msgType

8.362 minBasedIMSI Struct Reference

Data Fields

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

8.362.1 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

8.362.2 Field Documentation

8.362.2.1 WORD minBasedIMSI::imsiM1112

8.362.2.2 BYTE minBasedIMSI::imsiMS1[7]

8.362.2.3 BYTE minBasedIMSI::imsiMS2[3]

8.362.2.4 BYTE minBasedIMSI::mccM[3]

8.363 mitigationDevList Struct Reference

Data Fields

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

8.363.1 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.

8.363.2 Field Documentation

8.363.2.1 BYTE mitigationDevList::maxMitigationLevel

8.363.2.2 CHAR mitigationDevList::mitigationDevId[255]

8.363.2.3 BYTE mitigationDevList::mitigationDevIdLen

8.364 MNRInfo Struct Reference

Data Fields

- WORD mcc
- WORD mnc
- ULONG rat

8.364.1 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

8.364.2 Field Documentation

8.364.2.1 WORD MNRInfo::mcc

8.364.2.2 WORD MNRInfo::mnc

8.364.2.3 ULONG MNRInfo::rat

8.365 ModifyProfileIn Struct Reference

Data Fields

- BYTE * pProfileID
- BYTE * pProfileType
- QmiProfileInfo curProfile

8.365.1 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

8.365.2 Field Documentation

8.365.2.1 QmiProfileInfo ModifyProfileIn::curProfile

8.365.2.2 BYTE* ModifyProfileIn::pProfileID

8.365.2.3 BYTE* ModifyProfileIn::pProfileType

8.366 ModifyProfileOut Struct Reference

Data Fields

- USHORT * pExtErrorCode

8.366.1 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.

8.366.2 Field Documentation

8.366.2.1 USHORT* ModifyProfileOut::pExtErrorCode

8.367 msgWaitingInfo Struct Reference

Data Fields

- BYTE numInstances
- messageWaitingInfoContent msgWaitInfo [0xFF]

8.367.1 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.

8.367.2 Field Documentation

8.367.2.1 messageWaitingInfoContent msgWaitingInfo::msgWaitInfo[0xFF]

8.367.2.2 BYTE msgWaitingInfo::numInstances

8.368 namName Struct Reference

Data Fields

- BYTE namNameLen
- BYTE namName [12]

8.368.1 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.

8.368.2 Field Documentation

8.368.2.1 BYTE namName::namName[12]

8.368.2.2 BYTE namName::namNameLen

8.369 nas_acqOrderPref Struct Reference

Data Fields

- uint8_t acqOrdeLen
- uint8_t * pAcqOrder

8.369.1 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

8.369.2 Field Documentation

8.369.2.1 uint8_t nas_acqOrderPref::acqOrdeLen

8.369.2.2 uint8_t* nas_acqOrderPref::pAcqOrder

8.370 nas_ActPilotPNElement Struct Reference

Data Fields

- uint16_t ActSetPilotPN
- uint8_t ActSetPilotPNStrength

8.370.1 Detailed Description

This structure contains pilot PN Element.

Parameters

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

8.370.2 Field Documentation

8.370.2.1 uint16_t nas_ActPilotPNElement::ActSetPilotPN

8.370.2.2 uint8_t nas_ActPilotPNElement::ActSetPilotPNStrength

8.371 nas_AddCDMASysInfo Struct Reference

Data Fields

- uint16_t geoSysIdx
- uint16_t regPrd

8.371.1 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.

8.371.2 Field Documentation

8.371.2.1 uint16_t nas_AddCDMASysInfo::geoSysIdx

8.371.2.2 uint16_t nas_AddCDMASysInfo::regPrd

8.372 nas_AddSysInfo Struct Reference

Data Fields

- uint16_t geoSysIdx
- uint32_t cellBroadcastCap

8.372.1 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

8.372.2 Field Documentation

8.372.2.1 uint32_t nas_AddSysInfo::cellBroadcastCap

8.372.2.2 uint16_t nas_AddSysInfo::geoSysIdx

8.373 nas_CallBarringSysInfo Struct Reference

Data Fields

- uint32_t csBarStatus
- uint32_t psBarStatus

8.373.1 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

8.373.2 Field Documentation

8.373.2.1 uint32_t nas_CallBarringSysInfo::csBarStatus

8.373.2.2 uint32_t nas_CallBarringSysInfo::psBarStatus

8.374 nas_callBarStatus Struct Reference

Data Fields

- uint32_t csBarStatus
- uint32_t psBarStatus

8.374.1 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

8.374.2 Field Documentation

8.374.2.1 uint32_t nas_callBarStatus::csBarStatus

8.374.2.2 uint32_t nas_callBarStatus::psBarStatus

8.375 nas_CDMAChannel Struct Reference

Data Fields

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

8.375.1 Detailed Description

This structure contains CDMA channel.

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

8.375.2 Field Documentation

8.375.2.1 uint16_t nas_CDMAChannel::priChA

8.375.2.2 uint16_t nas_CDMAChannel::priChB

8.375.2.3 uint16_t nas_CDMAChannel::secChA

8.375.2.4 uint16_t nas_CDMAChannel::secChB

8.376 nas_CDMAECIOThresh Struct Reference

Data Fields

- uint8_t CDMAECIOThreshListLen
- int16_t * pCDMAECIOThreshList

8.376.1 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)

8.376.2 Field Documentation

8.376.2.1 uint8_t nas_CDMAECIOThresh::CDMAECIOThreshListLen

8.376.2.2 int16_t* nas_CDMAECIOThresh::pCDMAECIOThreshList

8.377 nas_CDMAInfo Struct Reference

Data Fields

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

8.377.1 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

8.377.2 Field Documentation

8.377.2.1 uint16_t nas_CDMAInfo::baseId

8.377.2.2 uint32_t nas_CDMAInfo::baseLat

8.377.2.3 uint32_t nas_CDMAInfo::baseLong

8.377.2.4 uint16_t nas_CDMAInfo::nid

8.377.2.5 uint16_t nas_CDMAInfo::refpn

8.377.2.6 uint16_t nas_CDMAInfo::sid

8.378 nas_CDMARSSIThresh Struct Reference

Data Fields

- uint8_t CDMARSSIThreshListLen
- int16_t * pCDMARSSIThreshList

8.378.1 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).

8.378.2 Field Documentation

8.378.2.1 uint8_t nas_CDMARSSIThresh::CDMARSSIThreshListLen

8.378.2.2 int16_t* nas_CDMARSSIThresh::pCDMARSSIThreshList

8.379 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]

8.379.1 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}.

8.379.2 Field Documentation

8.379.2.1 uint16_t nas_CDMASysInfo::baseId

8.379.2.2 uint32_t nas_CDMASysInfo::baseLat

8.379.2.3 uint32_t nas_CDMASysInfo::baseLong

8.379.2.4 uint8_t nas_CDMASysInfo::bsInfoValid

8.379.2.5 uint8_t nas_CDMASysInfo::bsPRev

8.379.2.6 uint8_t nas_CDMASysInfo::bsPRevValid

8.379.2.7 uint8_t nas_CDMASysInfo::ccsSupported

8.379.2.8 uint8_t nas_CDMASysInfo::ccsSupportedValid

8.379.2.9 uint8_t nas_CDMASysInfo::cdmaSysIdValid

8.379.2.10 uint8_t nas_CDMASysInfo::isSysPrlMatch

8.379.2.11 uint8_t nas_CDMASysInfo::isSysPrlMatchValid

8.379.2.12 uint8_t nas_CDMASysInfo::MCC[3]

8.379.2.13 uint8_t nas_CDMASysInfo::MNC[3]

8.379.2.14 uint16_t nas_CDMASysInfo::networkID

8.379.2.15 uint8_t nas_CDMASysInfo::networkIdValid

8.379.2.16 uint16_t nas_CDMASysInfo::packetZone

8.379.2.17 uint8_t nas_CDMASysInfo::packetZoneValid

8.379.2.18 uint8_t nas_CDMASysInfo::pRevInUse

8.379.2.19 uint8_t nas_CDMASysInfo::pRevInUseValid

8.379.2.20 nas_sysInfoCommon nas_CDMASysInfo::sysInfoCDMA

8.379.2.21 uint16_t nas_CDMASysInfo::systemID

8.380 nas_CDMASysInfoExt Struct Reference

Data Fields

- uint16_t MCC
- uint8_t imsi_11_12

8.380.1 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

8.380.2 Field Documentation

8.380.2.1 uint8_t nas_CDMASysInfoExt::imsi_11_12

8.380.2.2 uint16_t nas_CDMASysInfoExt::MCC

8.381 nas_cellParams Struct Reference

Data Fields

- uint16_t pci

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

8.381.1 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.

8.381.2 Field Documentation

8.381.2.1 uint16_t nas_cellParams::pci

8.381.2.2 int16_t nas_cellParams::rsrp

8.381.2.3 int16_t nas_cellParams::rsrq

8.381.2.4 int16_t nas_cellParams::rssi

8.381.2.5 int16_t nas_cellParams::srxlev

8.382 nas_CommInfo Struct Reference

Data Fields

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

- uint8_t psState

8.382.1 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

8.382.2 Field Documentation

8.382.2.1 uint8_t nas_CommInfo::imsRegState

8.382.2.2 uint8_t nas_CommInfo::modemMode

8.382.2.3 uint8_t nas_CommInfo::psState

8.382.2.4 uint8_t nas_CommInfo::systemMode

8.382.2.5 int8_t nas_CommInfo::temperature

8.383 nas_CSGID Struct Reference

Data Fields

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

8.383.1 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

8.383.2 Field Documentation

8.383.2.1 uint32_t nas_CSGID::id

8.383.2.2 uint16_t nas_CSGID::mcc

8.383.2.3 uint16_t nas_CSGID::mnc

8.383.2.4 uint8_t nas_CSGID::mncPcsDigits

8.383.2.5 uint8_t nas_CSGID::rat

8.384 nas_currentPLMN Struct Reference

Data Fields

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

8.384.1 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

8.384.2 Field Documentation

8.384.2.1 uint16_t nas_currentPLMN::MCC

8.384.2.2 uint16_t nas_currentPLMN::MNC

8.384.2.3 uint8_t nas_currentPLMN::netDescr[255]

8.384.2.4 uint8_t nas_currentPLMN::netDescrLength

8.385 nas_dataSrvCapabilities Struct Reference

Data Fields

- uint8_t dataCapabilitiesLen
- uint8_t dataCapabilities [32]

8.385.1 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+

8.385.2 Field Documentation

8.385.2.1 uint8_t nas_dataSrvCapabilities::dataCapabilities[32]

8.385.2.2 uint8_t nas_dataSrvCapabilities::dataCapabilitiesLen

8.386 nas_DataStatusDetail Struct Reference

Data Fields

- uint32_t IPAddress
- uint8_t LastErrCode

8.386.1 Detailed Description

This structure contains data status detail

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

8.386.2 Field Documentation

8.386.2.1 uint32_t nas_DataStatusDetail::IPAddress

8.386.2.2 uint8_t nas_DataStatusDetail::LastErrCode

8.387 nas_detailSvcInfo Struct Reference

Data Fields

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

8.387.1 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

8.387.2 Field Documentation

8.387.2.1 uint8_t nas_detailSvcInfo::hdrHybrid

8.387.2.2 uint8_t nas_detailSvcInfo::hdrSrvStatus

8.387.2.3 uint8_t nas_detailSvcInfo::isSysForbidden

8.387.2.4 uint8_t nas_detailSvcInfo::srvCapability

8.387.2.5 uint8_t nas_detailSvcInfo::srvStatus

8.388 nas_DeviceConfigDetail Struct Reference

Data Fields

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

8.388.1 Detailed Description

This structure contains device configure detail.

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

8.388.2 Field Documentation

8.388.2.1 uint8_t nas_DeviceConfigDetail::Chipset

8.388.2.2 uint8_t nas_DeviceConfigDetail::HWVersion

8.388.2.3 uint8_t nas_DeviceConfigDetail::QLIC

8.388.2.4 uint8_t nas_DeviceConfigDetail::Technology

8.389 nas_dirNum Struct Reference

Data Fields

- uint8_t dirNumLen
- uint8_t dirNum [255]

8.389.1 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.

8.389.2 Field Documentation

8.389.2.1 uint8_t nas_dirNum::dirNum[255]

8.389.2.2 uint8_t nas_dirNum::dirNumLen

8.390 nas_DRCParams Struct Reference

Data Fields

- uint8_t DRCValue
- uint8_t DRCCover

8.390.1 Detailed Description

This structure contains DRC information.

Parameters

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

8.390.2 Field Documentation

8.390.2.1 uint8_t nas_DRCParams::DRCCover

8.390.2.2 uint8_t nas_DRCParams::DRCValue

8.391 nas_ecioListElement Struct Reference

Data Fields

- int16_t ecio
- uint8_t radioIf

8.391.1 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

8.391.2 Field Documentation

8.391.2.1 int16_t nas_ecioListElement::ecio

8.391.2.2 uint8_t nas_ecioListElement::radioIf

8.392 nas_ECIOThresh Struct Reference

Data Fields

- uint8_t ECIOThresListLen
- int16_t * pECIOThresList

8.392.1 Detailed Description

This sturcture contain ECIO threshold list.

Parameters

ECIOThresList-Len	• Length of the ECIO threshold list parameter to follow
pECIOThresList	• Sequence of thresholds delimiting ECIO event reporting bands • 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

8.392.2 Field Documentation

8.392.2.1 uint8_t nas_ECIOThresh::ECIOThresListLen

8.392.2.2 int16_t* nas_ECIOThresh::pECIOThresList

8.393 nas_errorRateListElement Struct Reference

Data Fields

- uint16_t errorRate
- uint8_t radioIf

8.393.1 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

8.393.2 Field Documentation

8.393.2.1 uint16_t nas_errorRateListElement::errorRate

8.393.2.2 uint8_t nas_errorRateListElement::radioIf

8.394 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]

8.394.1 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.

8.394.2 Field Documentation

8.394.2.1 uint16_t nas_GERANInfo::arfcn

8.394.2.2 uint8_t nas_GERANInfo::bsic

8.394.2.3 uint32_t nas_GERANInfo::cellID

8.394.2.4 nas_nmrCellInfo nas_GERANInfo::insNmrCellInfo[255]

8.394.2.5 uint16_t nas_GERANInfo::lac

8.394.2.6 uint8_t nas_GERANInfo::nmrInst

8.394.2.7 uint8_t nas_GERANInfo::plmn[3]

8.394.2.8 uint16_t nas_GERANInfo::rxLev

8.394.2.9 uint32_t nas_GERANInfo::timingAdvance

8.395 nas_geranInstInfo Struct Reference

Data Fields

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

8.395.1 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.

8.395.2 Field Documentation

8.395.2.1 uint16_t nas_geranInstInfo::geranArfcn

8.395.2.2 uint8_t nas_geranInstInfo::geranBsicBcc

8.395.2.3 uint8_t nas_geranInstInfo::geranBsicNcc

8.395.2.4 int16_t nas_geranInstInfo::geranRssi

8.396 nas_gsmCellInfo Struct Reference

Data Fields

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

8.396.1 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.

8.396.2 Field Documentation

8.396.2.1 uint16_t nas_gsmCellInfo::arfcn

8.396.2.2 uint8_t nas_gsmCellInfo::band1900

8.396.2.3 uint8_t nas_gsmCellInfo::bsicId

8.396.2.4 uint8_t nas_gsmCellInfo::cellIdValid

8.396.2.5 int16_t nas_gsmCellInfo::rssi

8.396.2.6 int16_t nas_gsmCellInfo::srxlev

8.397 nas_GSMRSSIThresh Struct Reference

Data Fields

- uint8_t GSMRSSIThreshListLen
- int16_t * pGSMRSSIThreshList

8.397.1 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)

8.397.2 Field Documentation

8.397.2.1 uint8_t nas_GSMRSSIThresh::GSMRSSIThreshListLen

8.397.2.2 int16_t* nas_GSMRSSIThresh::pGSMRSSIThreshList

8.398 nas_GSMSrvStatusInfo Struct Reference

Data Fields

- uint8_t srvStatus
- uint8_t trueSrvStatus
- uint8_t isPrefDataPath

8.398.1 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

8.398.2 Field Documentation

8.398.2.1 uint8_t nas_GSMSrvStatusInfo::isPrefDataPath

8.398.2.2 uint8_t nas_GSMSrvStatusInfo::srvStatus

8.398.2.3 uint8_t nas_GSMSrvStatusInfo::trueSrvStatus

8.399 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

8.399.1 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 and 10.5.5.14] – 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

8.399.2 Field Documentation

8.399.2.1 uint32_t nas_GSMSysInfo::cellId

8.399.2.2 uint8_t nas_GSMSysInfo::cellIdValid

8.399.2.3 uint8_t nas_GSMSysInfo::dtmSupp

8.399.2.4 uint8_t nas_GSMSysInfo::dtmSuppValid

8.399.2.5 uint8_t nas_GSMSysInfo::egprsSupp

8.399.2.6 uint8_t nas_GSMSysInfo::egprsSuppValid

8.399.2.7 uint16_t nas_GSMSysInfo::lac

8.399.2.8 uint8_t nas_GSMSysInfo::lacValid

8.399.2.9 uint8_t nas_GSMSysInfo::MCC[3]

8.399.2.10 uint8_t nas_GSMSysInfo::MNC[3]

8.399.2.11 uint8_t nas_GSMSysInfo::networkIdValid

8.399.2.12 uint8_t nas_GSMSysInfo::regRejectInfoValid

8.399.2.13 uint8_t nas_GSMSysInfo::rejCause

8.399.2.14 uint8_t nas_GSMSysInfo::rejectSrvDomain

8.399.2.15 nas_sysInfoCommon nas_GSMSysInfo::sysInfoGSM

8.400 nas_HDRECIOThresh Struct Reference

Data Fields

- uint8_t HDRECIOThreshListLen
- int16_t * pHDRECIOThreshList

8.400.1 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).

8.400.2 Field Documentation

8.400.2.1 uint8_t nas_HDRECIOThresh::HDRECIOThreshListLen

8.400.2.2 int16_t* nas_HDRECIOThresh::pHDRECIOThreshList

8.401 nas_HDRIOThresh Struct Reference

Data Fields

- uint8_t HDRIOThreshListLen
- int16_t * pHDRIOThreshList

8.401.1 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).

8.401.2 Field Documentation

8.401.2.1 uint8_t nas_HDRIOThresh::HDRIOThreshListLen

8.401.2.2 int16_t* nas_HDRIOThresh::pHDRIOThreshList

8.402 nas_HDRRSSIThresh Struct Reference

Data Fields

- uint8_t HDRRSSIThreshListLen
- int16_t * pHDRRSSIThreshList

8.402.1 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).

8.402.2 Field Documentation

8.402.2.1 uint8_t nas_HDRRSSIThresh::HDRRSSIThreshListLen

8.402.2.2 int16_t* nas_HDRRSSIThresh::pHDRRSSIThreshList

8.403 nas_HDRSINRThresh Struct Reference

Data Fields

- uint8_t HDRSINRThresListLen
- uint8_t * pHDRSINRThresList

8.403.1 Detailed Description

This structure contains HDR SINR threshold list.

Parameters

HDRSINRThres-ListLen	• Length of the HDR SINR threshold list parameter to follow
pHDRSINR-ThresList	• Sequence of thresholds delimiting SINR event reporting bands • 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

8.403.2 Field Documentation

8.403.2.1 uint8_t nas_HDRSINRThresh::HDRSINRThresListLen

8.403.2.2 uint8_t* nas_HDRSINRThresh::pHDRSINRThresList

8.404 nas_HDRSINRThreshold Struct Reference

Data Fields

- uint8_t HDRSINRThreshListLen
- uint16_t * pHDRSINRThreshList

8.404.1 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

8.404.2 Field Documentation

8.404.2.1 uint8_t nas_HDRSINRThreshold::HDRSINRThreshListLen

8.404.2.2 uint16_t* nas_HDRSINRThreshold::pHDRSINRThreshList

8.405 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]

8.405.1 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[SLQ-S_SYSTEM_ID-_SIZE]	• IS-856 system ID. • Only applicable for HDR.

8.405.2 Field Documentation

8.405.2.1 uint8_t nas_HDRSysInfo::hdrActiveProt

8.405.2.2 uint8_t nas_HDRSysInfo::hdrActiveProtValid

8.405.2.3 uint8_t nas_HDRSysInfo::hdrPersonality

8.405.2.4 uint8_t nas_HDRSysInfo::hdrPersonalityValid

8.405.2.5 uint8_t nas_HDRSysInfo::is856SysId[16]

8.405.2.6 uint8_t nas_HDRSysInfo::is856SysIdValid

8.405.2.7 uint8_t nas_HDRSysInfo::isSysPrlMatch

8.405.2.8 uint8_t nas_HDRSysInfo::isSysPrlMatchValid

8.405.2.9 nas_sysInfoCommon nas_HDRSysInfo::sysInfoHDR

8.406 nas_homeSIDNID Struct Reference

Data Fields

- uint8_t numInstances
- nas_sidNid SidNid [255]

8.406.1 Detailed Description

This structure contains Home SID/NID

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

8.406.2 Field Documentation

8.406.2.1 uint8_t nas_homeSIDNID::numInstances

8.406.2.2 nas_sidNid nas_homeSIDNID::SidNid[255]

8.407 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]

8.407.1 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.

8.407.2 Field Documentation

8.407.2.1 uint8_t nas_infoInterFreq::cell_resel_priority

8.407.2.2 nas_cellParams nas_infoInterFreq::cellInterFreqParams[255]

8.407.2.3 uint8_t nas_infoInterFreq::cells_len

8.407.2.4 uint16_t nas_infoInterFreq::earfcn

8.407.2.5 uint8_t nas_infoInterFreq::threshXHigh

8.407.2.6 uint8_t nas_infoInterFreq::threshXLow

8.408 nas_IOThresh Struct Reference

Data Fields

- uint8_t IOThresListLen
- int32_t * pIOThresList

8.408.1 Detailed Description

This structure contains IO threshold list.

Parameters

IOThresListLen	• Length of the IO threshold list parameter to follow
pIOThresList	• Sequence of thresholds delimiting IO event reporting bands • 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

8.408.2 Field Documentation

8.408.2.1 uint8_t nas_IOThresh::IOThresListLen

8.408.2.2 int32_t* nas_IOThresh::pIOThresList

8.409 nas_lteEARFCN Struct Reference

Data Fields

- uint8_t status
- uint32_t earfcn0
- uint32_t earfcn1

8.409.1 Detailed Description

This structure contains LTE EARFCN infromation.

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.

8.409.2 Field Documentation

8.409.2.1 uint32_t nas_lteEARFCN::earfcn0

8.409.2.2 uint32_t nas_lteEARFCN::earfcn1

8.409.2.3 uint8_t nas_lteEARFCN::status

8.410 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]

8.410.1 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.

8.410.2 Field Documentation

8.410.2.1 uint8_t nas_lteGsmCellInfo::cellReselPriority

8.410.2.2 uint8_t nas_lteGsmCellInfo::cells_len

8.410.2.3 nas_gsmCellInfo nas_lteGsmCellInfo::GsmCellInfo[255]

8.410.2.4 uint8_t nas_lteGsmCellInfo::nccPermitted

8.410.2.5 uint8_t nas_lteGsmCellInfo::threshGsmHigh

8.410.2.6 uint8_t nas_lteGsmCellInfo::threshGsmLow

8.411 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

8.411.1 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

8.411.2 Field Documentation

8.411.2.1 uint8_t nas_LTEInfo::band

8.411.2.2 uint8_t nas_LTEInfo::bandwidth

8.411.2.3 uint8_t nas_LTEInfo::emmConnState

8.411.2.4 uint8_t nas_LTEInfo::emmState

8.411.2.5 uint8_t nas_LTEInfo::emmSubState

8.411.2.6 uint16_t nas_LTEInfo::RXChan

8.411.2.7 uint16_t nas_LTEInfo::TXChan

8.412 nas_LTEInfoInterfreq Struct Reference

Data Fields

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

8.412.1 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.

8.412.2 Field Documentation

8.412.2.1 uint8_t nas_LTEInfoInterfreq::freqsLen

8.412.2.2 nas_infoInterFreq nas_LTEInfoInterfreq::InfoInterfreq[255]

8.412.2.3 uint8_t nas_LTEInfoInterfreq::ueInIdle

8.413 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]

8.413.1 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.

8.413.2 Field Documentation

8.413.2.1 nas_cellParams nas_LTEInfoIntrafreq::CellParams[255]

8.413.2.2 uint8_t nas_LTEInfoIntrafreq::cellReselPriority

8.413.2.3 uint8_t nas_LTEInfoIntrafreq::cellsLen

8.413.2.4 uint16_t nas_LTEInfoIntrafreq::earfcn

8.413.2.5 uint32_t nas_LTEInfoIntrafreq::globalCellId

8.413.2.6 uint8_t nas_LTEInfoIntrafreq::plmn[3]

8.413.2.7 uint16_t nas_LTEInfoIntrafreq::servingCellId

8.413.2.8 uint8_t nas_LTEInfoIntrafreq::sIntraSearch

8.413.2.9 uint8_t nas_LTEInfoIntrafreq::sNonIntraSearch

8.413.2.10 uint16_t nas_LTEInfoIntrafreq::tac

8.413.2.11 uint8_t nas_LTEInfoIntrafreq::threshServingLow

8.413.2.12 uint8_t nas_LTEInfoIntrafreq::ueInIdle

8.414 nas_LTEInfoNeighboringGSM Struct Reference

Data Fields

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

8.414.1 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.

8.414.2 Field Documentation

8.414.2.1 uint8_t nas_LTEInfoNeighboringGSM::freqsLen

8.414.2.2 nas_lteGsmCellInfo nas_LTEInfoNeighboringGSM::LteGsmCellInfo[255]

8.414.2.3 uint8_t nas_LTEInfoNeighboringGSM::ueInIdle

8.415 nas_LTEInfoNeighboringWCDMA Struct Reference

Data Fields

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

8.415.1 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.

8.415.2 Field Documentation

8.415.2.1 uint8_t nas_LTEInfoNeighboringWCDMA::freqsLen

8.415.2.2 nas_lteWcdmaCellInfo nas_LTEInfoNeighboringWCDMA::LTEWCDMACellInfo[255]

8.415.2.3 uint8_t nas_LTEInfoNeighboringWCDMA::ueInIdle

8.416 nas_ltePCI Struct Reference

Data Fields

- uint8_t status
- uint32_t earfcn
- uint32_t pci

8.416.1 Detailed Description

This structure contains LTE PCI infromation.

Parameters

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

8.416.2 Field Documentation

8.416.2.1 uint32_t nas_ltePCI::earfcn

8.416.2.2 uint32_t nas_ltePCI::pci

8.416.2.3 uint8_t nas_ltePCI::status

8.417 nas_lteRsrpinformation Struct Reference

Data Fields

- int16_t rsrplevel

8.417.1 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).

8.417.2 Field Documentation

8.417.2.1 int16_t nas_lteRsrpinformation::rsrplevel

8.418 nas_LTERSRPThresh Struct Reference

Data Fields

- uint8_t LTERSRPThreshListLen
- int16_t * pLTERSRPThreshList

8.418.1 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).

8.418.2 Field Documentation

8.418.2.1 uint8_t nas_LTERSRPThresh::LTERSRPThreshListLen

8.418.2.2 int16_t* nas_LTERSRPThresh::pLTERSRPThreshList

8.419 nas_LTERSRQThresh Struct Reference

Data Fields

- uint8_t LTERSRQThreshListLen
- int16_t * pLTERSRQThreshList

8.419.1 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)

8.419.2 Field Documentation

8.419.2.1 uint8_t nas_LTERSRQThresh::LTERSRQThreshListLen

8.419.2.2 int16_t* nas_LTERSRQThresh::pLTERSRQThreshList

8.420 nas_LTERSSIThresh Struct Reference

Data Fields

- uint8_t LTERSSIThreshListLen
- int16_t * pLTERSSIThreshList

8.420.1 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)

8.420.2 Field Documentation

8.420.2.1 uint8_t nas_LTERSSIThresh::LTERSSIThreshListLen

8.420.2.2 int16_t* nas_LTERSSIThresh::pLTERSSIThreshList

8.421 nas_LTESigRptCfg Struct Reference

Data Fields

- uint8_t rptRate
- uint8_t avgPeriod

8.421.1 Detailed Description

This structure contains LTE signal report config.

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_LTESigRptCfg::avgPeriod

8.421.2.2 uint8_t nas_LTESigRptCfg::rptRate

8.422 nas_LTESigRptConfig Struct Reference

Data Fields

- uint8_t rptRate
- uint8_t avgPeriod

8.422.1 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

8.422.2 Field Documentation

8.422.2.1 uint8_t nas_LTESigRptConfig::avgPeriod

8.422.2.2 uint8_t nas_LTESigRptConfig::rptRate

8.423 nas_lteSnrinformation Struct Reference

Data Fields

- int16_t snrlevel

8.423.1 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.

8.423.2 Field Documentation

8.423.2.1 int16_t nas_lteSnrinformation::snrlevel

8.424 nas_LTESNRThresh Struct Reference

Data Fields

- uint8_t LTESNRThresListLen
- int16_t * pLTESNRThresList

8.424.1 Detailed Description

This structure contains LTE SNR threshold list.

Parameters

LTESNRThres-ListLen	• Length of the LTE SNR threshold list parameter to follow
pLTESNRThres-List	• Sequence of thresholds delimiting SNR event reporting bands • 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

8.424.2 Field Documentation

8.424.2.1 uint8_t nas_LTESNRThresh::LTESNRThresListLen

8.424.2.2 int16_t* nas_LTESNRThresh::pLTESNRThresList

8.425 nas_LTESNRThreshold Struct Reference

Data Fields

- uint8_t LTESNRThreshListLen
- int16_t * pLTESNRThreshList

8.425.1 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).

8.425.2 Field Documentation

8.425.2.1 uint8_t nas_LTESNRThreshold::LTESNRThreshListLen

8.425.2.2 int16_t* nas_LTESNRThreshold::pLTESNRThreshList

8.426 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

8.426.1 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]. and [3GPP TS 24.301, Section 9.9.3.9] – 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

8.426.2 Field Documentation

8.426.2.1 uint32_t nas_LTESysInfo::cellId

8.426.2.2 uint8_t nas_LTESysInfo::cellIdValid

8.426.2.3 uint16_t nas_LTESysInfo::lac

8.426.2.4 uint8_t nas_LTESysInfo::lacValid

8.426.2.5 uint8_t nas_LTESysInfo::MCC[3]

8.426.2.6 uint8_t nas_LTESysInfo::MNC[3]

8.426.2.7 uint8_t nas_LTESysInfo::networkIdValid

8.426.2.8 uint8_t nas_LTESysInfo::regRejectInfoValid

8.426.2.9 uint8_t nas_LTESysInfo::rejCause

8.426.2.10 uint8_t nas_LTESysInfo::rejectSrvDomain

8.426.2.11 nas_sysInfoCommon nas_LTESysInfo::sysInfoLTE

8.426.2.12 uint16_t nas_LTESysInfo::tac

8.426.2.13 uint8_t nas_LTESysInfo::tacValid

8.427 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]

8.427.1 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.

8.427.2 Field Documentation

8.427.2.1 uint8_t nas_lteWcdmaCellInfo::cellReselPriority

8.427.2.2 uint8_t nas_lteWcdmaCellInfo::cellsLen

8.427.2.3 uint16_t nas_lteWcdmaCellInfo::threshXhigh

8.427.2.4 uint16_t nas_lteWcdmaCellInfo::threshXlow

8.427.2.5 uint16_t nas_lteWcdmaCellInfo::uarfcn

8.427.2.6 nas_wcdmaCellInfo nas_lteWcdmaCellInfo::WCDMACellInfo[255]

8.428 nas_minBasedIMSI Struct Reference

Data Fields

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

- uint8_t imsiMS2 [3]

8.428.1 Detailed Description

This structure contains MIN-based IMSI.

Parameters

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

8.428.2 Field Documentation

8.428.2.1 uint16_t nas_minBasedIMSI::imsiM1112

8.428.2.2 uint8_t nas_minBasedIMSI::imsiMS1[7]

8.428.2.3 uint8_t nas_minBasedIMSI::imsiMS2[3]

8.428.2.4 uint8_t nas_minBasedIMSI::mccM[3]

8.429 nas_MNRInfo Struct Reference

Data Fields

- uint16_t mcc
- uint16_t mnc
- uint32_t rat

8.429.1 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

8.429.2 Field Documentation

8.429.2.1 uint16_t nas_MNRInfo::mcc

8.429.2.2 uint16_t nas_MNRInfo::mnc

8.429.2.3 uint32_t nas_MNRInfo::rat

8.430 nas_namName Struct Reference

Data Fields

- uint8_t namNameLen
- uint8_t namName [12]

8.430.1 Detailed Description

This structure contains NAM Name.

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.

8.430.2 Field Documentation

8.430.2.1 uint8_t nas_namName::namName[12]

8.430.2.2 uint8_t nas_namName::namNameLen

8.431 nas_netSelectionPref Struct Reference

Data Fields

- uint8_t netReg
- uint16_t mcc
- uint16_t mnc

8.431.1 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

8.431.2 Field Documentation

8.431.2.1 uint16_t nas_netSelectionPref::mcc

8.431.2.2 uint16_t nas_netSelectionPref::mnc

8.431.2.3 uint8_t nas_netSelectionPref::netReg

8.432 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

8.432.1 Detailed Description

This structure contains network stat 1x.

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

8.432.2 Field Documentation

8.432.2.1 uint8_t nas_NetworkStat1x::ActSetCnt

8.432.2.2 uint8_t nas_NetworkStat1x::NeighborSetCnt

8.432.2.3 nas_ActPilotPNElement* nas_NetworkStat1x::pActPilotPNElements

8.432.2.4 uint16_t* nas_NetworkStat1x::pNeighborSetPilotPN

8.432.2.5 uint16_t nas_NetworkStat1x::RX_EC_IO

8.432.2.6 uint32_t nas_NetworkStat1x::RX_PWR

8.432.2.7 uint16_t nas_NetworkStat1x::SO

8.432.2.8 uint8_t nas_NetworkStat1x::State

8.432.2.9 uint32_t nas_NetworkStat1x::TX_PWR

8.433 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

8.433.1 Detailed Description

This structure contains network stat EVDO.

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)

8.433.2 Field Documentation

8.433.2.1 uint8_t nas_NetworkStatEVDO::MACIndex

8.433.2.2 uint16_t nas_NetworkStatEVDO::PER

8.433.2.3 uint16_t nas_NetworkStatEVDO::PilotEnergy

8.433.2.4 uint16_t* nas_NetworkStatEVDO::pSectorID

8.433.2.5 uint16_t nas_NetworkStatEVDO::RX_PWR

8.433.2.6 uint8_t nas_NetworkStatEVDO::SectorIDLen

8.433.2.7 uint8_t nas_NetworkStatEVDO::SNR

8.433.2.8 uint8_t nas_NetworkStatEVDO::State

8.434 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

8.434.1 Detailed Description

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

Parameters

nmrCellID	• Cell ID. • 0xFFFFFFFF indicates cell ID information is not present.
nmrPlmn[NAS_PLMN_LENGTH]	• MCC/MNC information coded as octet 3, 4, and 5. • This field is ignored when nmrCellID is not present.
nmrLac	• Location area code. • This field is ignored when nmrCellID is not present. – 0xFFFF - Not Available
nmrArfcn	• Absolute RF channel number. – 0xFFFF - Not Available
nmrBsic	• Base station identity code. – 0xFF - Not Available

nmrRxLev	• Cell Rx measurement. • Values range between 0 and 63. • Mapped to a measured signal level: – Rxlev 0 is a signal strength less than -110 dBm – Rxlev 1 is -110 dBm to -109 dBm – Rxlev 2 is -109 dBm to -108 dBm – ... – Rxlev 62 is -49 dBm to -48 dBm – Rxlev 63 is greater than -48 dBm – 0xFFFF - Not Available

8.434.2 Field Documentation

8.434.2.1 uint16_t nas_nmrCellInfo::nmrArfcn

8.434.2.2 uint8_t nas_nmrCellInfo::nmrBsic

8.434.2.3 uint32_t nas_nmrCellInfo::nmrCellID

8.434.2.4 uint16_t nas_nmrCellInfo::nmrLac

8.434.2.5 uint8_t nas_nmrCellInfo::nmrPlmn[3]

8.434.2.6 uint16_t nas_nmrCellInfo::nmrRxLev

8.435 nas_operatorNameString Struct Reference

Data Fields

- uint8_t PLMNName [255]

8.435.1 Detailed Description

This structure contains PLMN name.

Parameters

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

8.435.2 Field Documentation

8.435.2.1 uint8_t nas_operatorNameString::PLMNName[255]

8.436 nas_OperatorPLMNData Struct Reference

Data Fields

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

8.436.1 Detailed Description

This structure contains Operator PLMN.

Parameters

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

8.436.2 Field Documentation

8.436.2.1 uint16_t nas_OperatorPLMNData::lac1

8.436.2.2 uint16_t nas_OperatorPLMNData::lac2

8.436.2.3 uint8_t nas_OperatorPLMNData::mcc[3]

8.436.2.4 uint8_t nas_OperatorPLMNData::mnc[3]

8.436.2.5 uint8_t nas_OperatorPLMNData::PLMNRecID

8.437 nas_operatorPLMNList Struct Reference

Data Fields

- uint16_t numInstance
- nas_OperatorPLMNData PLMNData [255]

8.437.1 Detailed Description

This structure contains Operator PLMN List.

Parameters

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

8.437.2 Field Documentation

8.437.2.1 uint16_t nas_operatorPLMNList::numInstance

8.437.2.2 nas_OperatorPLMNData nas_operatorPLMNList::PLMNData[255]

8.438 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

8.438.1 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.

8.438.2 Field Documentation

8.438.2.1 NAS_LTE_CPHY_CA_BW_NRB_LITE nas_PhyCaAggPcellInfo::dl_bw_value

8.438.2.2 uint16_t nas_PhyCaAggPcellInfo::freq

8.438.2.3 uint16_t nas_PhyCaAggPcellInfo::iLTEbandValue

8.438.2.4 uint16_t nas_PhyCaAggPcellInfo::pci

8.438.2.5 uint8_t nas_PhyCaAggPcellInfo::TlvPresent

8.439 nas_PhyCaAggScellDlBw Struct Reference

Data Fields

- NAS_LTE_CPHY_CA_BW_NRB_LITE dl_bw_value
- uint8_t TlvPresent

8.439.1 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.

8.439.2 Field Documentation

8.439.2.1 NAS_LTE_CPHY_CA_BW_NRB_LITE nas_PhyCaAggScellDlBw::dl_bw_value

8.439.2.2 uint8_t nas_PhyCaAggScellDlBw::TlvPresent

8.440 nas_PhyCaAggScellIndex Struct Reference

Data Fields

- uint8_t scell_idx
- uint8_t TlvPresent

8.440.1 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.

8.440.2 Field Documentation

8.440.2.1 uint8_t nas_PhyCaAggScellIndex::scell_idx

8.440.2.2 uint8_t nas_PhyCaAggScellIndex::TlvPresent

8.441 nas_PhyCaAggScellIndType Struct Reference

Data Fields

- uint16_t pci
- uint16_t freq
- 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 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.

8.441.2 Field Documentation

8.441.2.1 uint16_t nas_PhyCaAggScellIndType::freq

8.441.2.2 uint16_t nas_PhyCaAggScellIndType::pci

8.441.2.3 NAS_LTE_CPHY_SCELL_STATE_LITE nas_PhyCaAggScellIndType::scell_state

8.441.2.4 uint8_t nas_PhyCaAggScellIndType::TlvPresent

8.442 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

8.442.1 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.

8.442.2 Field Documentation

8.442.2.1 NAS_LTE_CPHY_CA_BW_NRB_LITE nas_PhyCaAggScellInfo::dl_bw_value

8.442.2.2 uint16_t nas_PhyCaAggScellInfo::freq

8.442.2.3 uint16_t nas_PhyCaAggScellInfo::iLTEbandValue

8.442.2.4 uint16_t nas_PhyCaAggScellInfo::pci

8.442.2.5 NAS_LTE_CPHY_SCELL_STATE_LITE nas_PhyCaAggScellInfo::scell_state

8.442.2.6 uint8_t nas_PhyCaAggScellInfo::TlvPresent

8.443 nas_PilotSetData Struct Reference

Data Fields

- uint8_t NumPilots
- nas_PilotSetParams * pPilotSetInfo

8.443.1 Detailed Description

This structure contains pilot sets data.

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.

8.443.2 Field Documentation

8.443.2.1 uint8_t nas_PilotSetData::NumPilots

8.443.2.2 nas_PilotSetParams* nas_PilotSetData::pPilotSetInfo

8.444 nas_PilotSetParams Struct Reference

Data Fields

- uint32_t PilotType
- uint16_t PilotPN
- uint16_t PilotStrength

8.444.1 Detailed Description

This structure contains pilot sets 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)

8.444.2 Field Documentation

8.444.2.1 uint16_t nas_PilotSetParams::PilotPN

8.444.2.2 uint16_t nas_PilotSetParams::PilotStrength

8.444.2.3 uint32_t nas_PilotSetParams::PilotType

8.445 nas_PLMNNetworkName Struct Reference

Data Fields

- uint8_t numInstance
- nas_PLMNNetworkNameData PLMNNetName [255]

8.445.1 Detailed Description

This structure contains NITZ Information list.

Parameters

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

8.445.2 Field Documentation

8.445.2.1 uint8_t nas_PLMNNetworkName::numInstance

8.445.2.2 nas_PLMNNetworkNameData nas_PLMNNetworkName::PLMNNetName[255]

8.446 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]

8.446.1 Detailed Description

This structure contains NITZ Information.

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.

8.446.2 Field Documentation

8.446.2.1 uint8_t nas_PLMNNetworkNameData::codingScheme

8.446.2.2 uint8_t nas_PLMNNetworkNameData::countryInitials

8.446.2.3 uint8_t nas_PLMNNetworkNameData::longName[255]

8.446.2.4 uint8_t nas_PLMNNetworkNameData::longNameLen

8.446.2.5 uint8_t nas_PLMNNetworkNameData::longNameSpareBits

8.446.2.6 uint8_t nas_PLMNNetworkNameData::shortName[255]

8.446.2.7 uint8_t nas_PLMNNetworkNameData::shortNameLen

8.446.2.8 uint8_t nas_PLMNNetworkNameData::shortNameSpareBits

8.447 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

8.447.1 Detailed Description

This structure contains protocol subtype element.

Parameters

PhysicalLayer	• Specifies Physical Layer Protocol subtype
ControlMac	• Specifies Control Channel MAC Protocol subtype
AccessMac	• Specifies Access Channel MAC Protocol subtype
ForwardMac	• Specifies Forward Traffic Channel MAC Protocol subtype
ReverseMac	• Specifies Reverse Traffic Channel MAC Protocol subtype
KeyExchange	• Specifies Key exchange Protocol subtype
AuthProt	• Specifies Authentication Protocol subtype
EncryptProt	• Specifies Encryption Protocol subtype
SecProt	• Specifies Security Protocol subtype
IdleState	• Specifies Idle state Protocol subtype
MultDisc	• Specifies Generic multimode capability discovery Protocol subtype
VirtStream	• Specifies Generic Virtual Stream Protocol subtype

8.447.2 Field Documentation

8.447.2.1 uint16_t nas_protocolSubtypeElement::AccessMac

8.447.2.2 uint16_t nas_protocolSubtypeElement::AuthProt

8.447.2.3 uint16_t nas_protocolSubtypeElement::ControlMac

8.447.2.4 uint16_t nas_protocolSubtypeElement::EncryptProt

8.447.2.5 uint16_t nas_protocolSubtypeElement::ForwardMac

8.447.2.6 uint16_t nas_protocolSubtypeElement::IdleState

8.447.2.7 uint16_t nas_protocolSubtypeElement::KeyExchange

8.447.2.8 uint16_t nas_protocolSubtypeElement::MultDisc

8.447.2.9 uint16_t nas_protocolSubtypeElement::PhysicalLayer

8.447.2.10 uint16_t nas_protocolSubtypeElement::ReverseMac

8.447.2.11 uint16_t nas_protocolSubtypeElement::SecProt

8.447.2.12 uint16_t nas_protocolSubtypeElement::VirtStream

8.448 nas_qaQmi3Gpp2TimeZone Struct Reference

Data Fields

- uint8_t leapSeconds
- uint8_t localTimeOffset
- uint8_t daylightSavings

8.448.1 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)

8.448.2 Field Documentation

8.448.2.1 uint8_t nas_qaQmi3Gpp2TimeZone::daylightSavings

8.448.2.2 uint8_t nas_qaQmi3Gpp2TimeZone::leapSeconds

8.448.2.3 uint8_t nas_qaQmi3Gpp2TimeZone::localTimeOffset

8.449 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]

8.449.1 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.

8.449.2 Field Documentation

8.449.2.1 char nas_QmiNas3GppNetworkInfo::Desription[255]

8.449.2.2 uint32_t nas_QmiNas3GppNetworkInfo::Forbidden

8.449.2.3 uint32_t nas_QmiNas3GppNetworkInfo::InUse

8.449.2.4 uint16_t nas_QmiNas3GppNetworkInfo::MCC

8.449.2.5 uint16_t nas_QmiNas3GppNetworkInfo::MNC

8.449.2.6 uint32_t nas_QmiNas3GppNetworkInfo::Preferred

8.449.2.7 uint32_t nas_QmiNas3GppNetworkInfo::Roaming

8.450 nas_QmiNas3GppNetworkRAT Struct Reference

Data Fields

- uint16_t MCC
- uint16_t MNC
- uint8_t RAT

8.450.1 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

8.450.2 Field Documentation

8.450.2.1 uint16_t nas_QmiNas3GppNetworkRAT::MCC

8.450.2.2 uint16_t nas_QmiNas3GppNetworkRAT::MNC

8.450.2.3 uint8_t nas_QmiNas3GppNetworkRAT::RAT

8.451 nas_QmisNasPcsDigit Struct Reference

Data Fields

- uint16_t MCC
- uint16_t MNC
- uint8_t includes_pcs_digit

8.451.1 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

8.451.2 Field Documentation

8.451.2.1 uint8_t nas_QmisNasPcsDigit::includes_pcs_digit

8.451.2.2 uint16_t nas_QmisNasPcsDigit::MCC

8.451.2.3 uint16_t nas_QmisNasPcsDigit::MNC

8.452 nas_RejectReasonTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint32_t serviceDomain
- uint32_t rejectCause

8.452.1 Detailed Description

This structure contains paramaters reject reason.

Parameters

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

8.452.2 Field Documentation

8.452.2.1 uint32_t nas_RejectReasonTlv::rejectCause

8.452.2.2 uint32_t nas_RejectReasonTlv::serviceDomain

8.452.2.3 uint8_t nas_RejectReasonTlv::TlvPresent

8.453 nas_RFInfoTlv Struct Reference

Data Fields

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

8.453.1 Detailed Description

This structure contains paramaters RF Band Information List.

Parameters

TlvPresent	indicating the presence of the TLV in the QMI ind
radioInterface-Size	Number of sets of the following elements • radioInterface • activeBandClass • activeChannel
radioInterface	radio interface technology of the signal being measured • 0x01 - cdma2000 1X • 0x02 - cdma2000 HRPD (1xEV-DO) • 0x03 - AMPS • 0x04 - GSM • 0x05 - UMTS • 0x08 - LTE • 0x09 - TD-SCDMA
activeBandClass	active band class • 00 to 39 - CDMA band classes • 40 to 79 - GSM band classes • 80 to 91 - WCDMA band classes • 120 to 160 - LTE band classes • 200 to 205 - TD-SCDMA band classes

activeChannel	active channel • Active channel. If the channel is not relevant to the technology, a value of 0 is returned.

8.453.2 Field Documentation

8.453.2.1 uint32_t nas_RFInfoTlv::activeBandClass[255]

8.453.2.2 uint32_t nas_RFInfoTlv::activeChannel[255]

8.453.2.3 uint32_t nas_RFInfoTlv::radioInterface[255]

8.453.2.4 uint8_t nas_RFInfoTlv::radioInterfaceSize

8.453.2.5 uint8_t nas_RFInfoTlv::TlvPresent

8.454 nas_roamIndList Struct Reference

Data Fields

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

8.454.1 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

8.454.2 Field Documentation

8.454.2.1 uint8_t nas_roamIndList::numInstances

8.454.2.2 uint8_t nas_roamIndList::radioInterface[32]

8.454.2.3 uint8_t nas_roamIndList::roamIndicator[32]

8.455 nas_RSRPThresh Struct Reference

Data Fields

- uint8_t RSRPThresListLen
- int16_t * pRSRPThresList

8.455.1 Detailed Description

This structure contains RSRP threshold list.

Parameters

RSRPThresList-Len	• Length of the RSRP threshold list parameter to follow
pRSRPThresList	• Sequence of thresholds delimiting current RSRP event reporting bands • 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

8.455.2 Field Documentation

8.455.2.1 int16_t* nas_RSRPThresh::pRSRPThresList

8.455.2.2 uint8_t nas_RSRPThresh::RSRPThresListLen

8.456 nas_rsrqInformation Struct Reference

Data Fields

- int8_t rsrq
- uint8_t radioIf

8.456.1 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

8.456.2 Field Documentation

8.456.2.1 uint8_t nas_rsrqInformation::radioIf

8.456.2.2 int8_t nas_rsrqInformation::rsrq

8.457 nas_RSRQThresh Struct Reference

Data Fields

- uint8_t RSRQThresListLen
- int8_t * pRSRQThresList

8.457.1 Detailed Description

This structure contains RSRQ threshold list.

Parameters

RSRQThresList-Len	• Length of the RSRQ threshold list parameter to follow
pRSRQThresList	• Sequence of thresholds delimiting current RSRQ event reporting bands • 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

8.457.2 Field Documentation

8.457.2.1 int8_t* nas_RSRQThresh::pRSRQThresList

8.457.2.2 uint8_t nas_RSRQThresh::RSRQThresListLen

8.458 nas_RSSIThresh Struct Reference

Data Fields

- uint8_t RSSIThresListLen
- int8_t * pRSSIThresList

8.458.1 Detailed Description

This sturcture contain RSSI threshold list.

Parameters

RSSIThresList-Len	• Length of the RSSI threshold list parameter to follow
pRSSIThresList	• RSSI in dBm(signed bytes) • 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.

8.458.2 Field Documentation

8.458.2.1 int8_t* nas_RSSIThresh::pRSSIThresList

8.458.2.2 uint8_t nas_RSSIThresh::RSSIThresListLen

8.459 nas_rxInfo Struct Reference

Data Fields

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

8.459.1 Detailed Description

This structure contains rx radio infomation.

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.

8.459.2 Field Documentation

8.459.2.1 int32_t nas_rxInfo::ecio

8.459.2.2 uint8_t nas_rxInfo::isRadioTuned

8.459.2.3 uint32_t nas_rxInfo::phase

8.459.2.4 int32_t nas_rxInfo::rscp

8.459.2.5 int32_t nas_rxInfo::rsrp

8.459.2.6 int32_t nas_rxInfo::rxPower

8.460 nas_RxSigInfo Struct Reference

Data Fields

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

8.460.1 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

8.460.2 Field Documentation

8.460.2.1 uint8_t nas_RxSigInfo::isRadioTuned

8.460.2.2 int32_t nas_RxSigInfo::rsrp

8.460.2.3 uint8_t nas_RxSigInfo::rxChainIndex

8.460.2.4 int32_t nas_RxSigInfo::rxPower

8.461 nas_rxSignalStrengthListElement Struct Reference

Data Fields

- int16_t rxSignalStrength
- uint8_t radioIf

8.461.1 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.

8.461.2 Field Documentation

8.461.2.1 uint8_t nas_rxSignalStrengthListElement::radioIf

8.461.2.2 int16_t nas_rxSignalStrengthListElement::rxSignalStrength

8.462 nas_SccRxInfo Struct Reference

Data Fields

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

8.462.1 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.

8.462.2 Field Documentation

8.462.2.1 uint8_t nas_SccRxInfo::numInstances

8.462.2.2 int32_t nas_SccRxInfo::rsrq

8.462.2.3 nas_RxSigInfo nas_SccRxInfo::sigInfo[255]

8.462.2.4 int16_t nas_SccRxInfo::snr

8.462.2.5 uint8_t nas_SccRxInfo::TlvPresent

8.463 nas_serviceProviderName Struct Reference

Data Fields

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

8.463.1 Detailed Description

This structure contains service provider name.

Parameters

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

8.463.2 Field Documentation

8.463.2.1 uint8_t nas_serviceProviderName::displayCondition

8.463.2.2 uint8_t nas_serviceProviderName::spn[255]

8.463.2.3 uint8_t nas_serviceProviderName::spnLength

8.464 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]

8.464.1 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

8.464.2 Field Documentation

8.464.2.1 uint8_t nas_servSystem::csAttachState

8.464.2.2 uint8_t nas_servSystem::numRadioInterfaces

8.464.2.3 uint8_t nas_servSystem::psAttachState

8.464.2.4 uint8_t nas_servSystem::radioInterface[32]

8.464.2.5 uint8_t nas_servSystem::regState

8.464.2.6 uint8_t nas_servSystem::selNetwork

8.465 nas_sidNid Struct Reference

Data Fields

- uint16_t nid
- uint16_t sid

8.465.1 Detailed Description

This structure contains the parameters for SidNid Information

Parameters

nid	• Network ID
sid	• System ID

8.465.2 Field Documentation

8.465.2.1 uint16_t nas_sidNid::nid

8.465.2.2 uint16_t nas_sidNid::sid

8.466 nas_SignalStrengthTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- int8_t signalStrength
- uint32_t radioInterface

8.466.1 Detailed Description

This structure contains paramaters Signal Strength.

Parameters

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

8.466.2 Field Documentation

8.466.2.1 uint32_t nas_SignalStrengthTlv::radioInterface

8.466.2.2 int8_t nas_SignalStrengthTlv::signalStrength

8.466.2.3 uint8_t nas_SignalStrengthTlv::TlvPresent

8.467 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

8.467.1 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.
sinrThresholdList	• 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.

8.467.2 Field Documentation

8.467.2.1 uint8_t nas_SLQSSignalStrengthsIndReq::ecioDelta

8.467.2.2 int16_t nas_SLQSSignalStrengthsIndReq::ecioThresholdList[10]

8.467.2.3 uint8_t nas_SLQSSignalStrengthsIndReq::ecioThresholdListLen

8.467.2.4 uint8_t nas_SLQSSignalStrengthsIndReq::ioDelta

8.467.2.5 uint8_t nas_SLQSSignalStrengthsIndReq::lteRsrpDelta

8.467.2.6 uint16_t nas_SLQSSignalStrengthsIndReq::lteSnrDelta

8.467.2.7 uint8_t nas_SLQSSignalStrengthsIndReq::rsrqDelta

8.467.2.8 uint8_t nas_SLQSSignalStrengthsIndReq::rxSignalStrengthDelta

8.467.2.9 uint8_t nas_SLQSSignalStrengthsIndReq::sinrDelta

8.467.2.10 uint8_t nas_SLQSSignalStrengthsIndReq::sinrThresholdList[5]

8.467.2.11 uint8_t nas_SLQSSignalStrengthsIndReq::sinrThresholdListLen

8.468 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

8.468.1 Detailed Description

This structure contains paramaters signal strengths information.

Parameters

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

8.468.2 Field Documentation

8.468.2.1 nas_ecioListElement nas_SLQSSignalStrengthsInformation::ecioInfo

8.468.2.2 nas_errorRateListElement nas_SLQSSignalStrengthsInformation::errorRateInfo

8.468.2.3 uint32_t nas_SLQSSignalStrengthsInformation::io

8.468.2.4 nas_lteRsrpinformation nas_SLQSSignalStrengthsInformation::lteRsrpinfo

8.468.2.5 nas_lteSnrinformation nas_SLQSSignalStrengthsInformation::lteSnrinfo

8.468.2.6 nas_rsrqInformation nas_SLQSSignalStrengthsInformation::rsrqInfo

8.468.2.7 nas_rxSignalStrengthListElement nas_SLQSSignalStrengthsInformation::rxSignalStrengthInfo

8.468.2.8 uint8_t nas_SLQSSignalStrengthsInformation::sinr

8.469 nas_SLQSSignalStrengthsTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- nas_SLQSSignalStrengthsInformation sSLQSSignalStrengthsInfo

8.469.1 Detailed Description

This structure contains paramaters SLQS Signal Strength.

Parameters

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

8.469.2 Field Documentation

8.469.2.1 nas_SLQSSignalStrengthsInformation nas_SLQSSignalStrengthsTlv::sSLQSSignalStrengthsInfo

8.469.2.2 uint8_t nas_SLQSSignalStrengthsTlv::TlvPresent

8.470 nas_SrvStatusInfo Struct Reference

Data Fields

- uint8_t srvStatus
- uint8_t isPrefDataPath

8.470.1 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

8.470.2 Field Documentation

8.470.2.1 uint8_t nas_SrvStatusInfo::isPrefDataPath

8.470.2.2 uint8_t nas_SrvStatusInfo::srvStatus

8.471 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

8.471.1 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

8.471.2 Field Documentation

8.471.2.1 uint8_t nas_sysInfoCommon::isSysForbidden

8.471.2.2 uint8_t nas_sysInfoCommon::isSysForbiddenValid

8.471.2.3 uint8_t nas_sysInfoCommon::roamStatus

8.471.2.4 uint8_t nas_sysInfoCommon::roamStatusValid

8.471.2.5 uint8_t nas_sysInfoCommon::srvCapability

8.471.2.6 uint8_t nas_sysInfoCommon::srvCapabilityValid

8.471.2.7 uint8_t nas_sysInfoCommon::srvDomain

8.471.2.8 uint8_t nas_sysInfoCommon::srvDomainValid

8.472 nas_TDSCDMAECIOThresh Struct Reference

Data Fields

- uint8_t TDSCDMAECIOThreshListLen
- float * pTDSCDMAECIOThreshList

8.472.1 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.

8.472.2 Field Documentation

8.472.2.1 float* nas_TDSCDMAECIOThresh::pTDSCDMAECIOThreshList

8.472.2.2 uint8_t nas_TDSCDMAECIOThresh::TDSCDMAECIOThreshListLen

8.473 nas_TDSCDMARSCPThresh Struct Reference

Data Fields

- uint8_t TDSCDMARSCPThreshListLen
- int16_t * pTDSCDMARSCPThreshList

8.473.1 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).

8.473.2 Field Documentation

8.473.2.1 int16_t* nas_TDSCDMARSCPThresh::pTDSCDMARSCPThreshList

8.473.2.2 uint8_t nas_TDSCDMARSCPThresh::TDSCDMARSCPThreshListLen

8.474 nas_TDSCDMARSSIThresh Struct Reference

Data Fields

- uint8_t TDSCDMARSSIThreshListLen
- float * pTDSCDMARSSIThreshList

8.474.1 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.

8.474.2 Field Documentation

8.474.2.1 float* nas_TDSCDMARSSIThresh::pTDSCDMARSSIThreshList

8.474.2.2 uint8_t nas_TDSCDMARSSIThresh::TDSCDMARSSIThreshListLen

8.475 nas_TDSCDMASINRCONFThresh Struct Reference

Data Fields

- uint8_t TDSCDMASINRCONFThreshListLen
- float * pTDSCDMASINRCONFThreshList

8.475.1 Detailed Description

This structure contains TDSCDMA SINR threshold list.

Parameters

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

8.475.2 Field Documentation

8.475.2.1 float* nas_TDSCDMASINRCONFThresh::pTDSCDMASINRCONFThreshList

8.475.2.2 uint8_t nas_TDSCDMASINRCONFThresh::TDSCDMASINRCONFThreshListLen

8.476 nas_TDSCDMASINRThresh Struct Reference

Data Fields

- uint8_t TDSCDMASINRThreshListLen
- float * pTDSCDMASINRThreshList

8.476.1 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

8.476.2 Field Documentation

8.476.2.1 float* nas_TDSCDMASINRThresh::pTDSCDMASINRThreshList

8.476.2.2 uint8_t nas_TDSCDMASINRThresh::TDSCDMASINRThreshListLen

8.477 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

8.477.1 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.

8.477.2 Field Documentation

8.477.2.1 uint8_t nas_timeInfo::day

8.477.2.2 uint8_t nas_timeInfo::dayLtSavingAdj

8.477.2.3 uint8_t nas_timeInfo::dayOfWeek

8.477.2.4 uint8_t nas_timeInfo::hour

8.477.2.5 uint8_t nas_timeInfo::minute

8.477.2.6 uint8_t nas_timeInfo::month

8.477.2.7 uint8_t nas_timeInfo::radioInterface

8.477.2.8 uint8_t nas_timeInfo::second

8.477.2.9 int8_t nas_timeInfo::timeZone

8.477.2.10 uint8_t nas_timeInfo::TlvPresent

8.477.2.11 uint16_t nas_timeInfo::year

8.478 nas_trueIMSI Struct Reference

Data Fields

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

8.478.1 Detailed Description

This structure contains True IMSI.

Parameters

mccT	• ASCII character representation of MCC_T
imsiT1112	• ASCII character representation of IMSI_T_11_12 value – 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

8.478.2 Field Documentation

8.478.2.1 uint16_t nas_truelMSI::imsiT1112

8.478.2.2 uint8_t nas_truelMSI::imsiTaddrNum

8.478.2.3 uint8_t nas_truelMSI::imsiTS1[7]

8.478.2.4 uint8_t nas_truelMSI::imsiTS2[3]

8.478.2.5 uint8_t nas_truelMSI::mccT[3]

8.479 nas_txInfo Struct Reference

Data Fields

- uint8_t isInTraffic
- int32_t txPower

8.479.1 Detailed Description

This structure contains tx radio infomation.

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.

8.479.2 Field Documentation

8.479.2.1 uint8_t nas_txInfo::isInTraffic

8.479.2.2 int32_t nas_txInfo::txPower

8.480 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]

8.480.1 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.

8.480.2 Field Documentation

8.480.2.1 uint16_t nas_UMTSInfo::cellID

8.480.2.2 int16_t nas_UMTSInfo::ecio

8.480.2.3 uint8_t nas_UMTSInfo::geranInst

8.480.2.4 nas_geranInstInfo nas_UMTSInfo::GeranInstInfo[255]

8.480.2.5 uint16_t nas_UMTSInfo::lac

8.480.2.6 uint8_t nas_UMTSInfo::plmn[3]

8.480.2.7 uint16_t nas_UMTSInfo::psc

8.480.2.8 int16_t nas_UMTSInfo::rscp

8.480.2.9 uint16_t nas_UMTSInfo::uarfcn

8.480.2.10 uint8_t nas_UMTSInfo::umtsInst

8.480.2.11 nas_UMTSinstInfo nas_UMTSInfo::UMTSInstInfo[255]

8.481 nas_UMTSinstInfo Struct Reference

Data Fields

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

8.481.1 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).

8.481.2 Field Documentation

8.481.2.1 int16_t nas_UMTSinstInfo::umtsEcio

8.481.2.2 uint16_t nas_UMTSinstInfo::umtsPsc

8.481.2.3 int16_t nas_UMTSinstInfo::umtsRscp

8.481.2.4 uint16_t nas_UMTSinstInfo::umtsUarfcn

8.482 nas_umtsLTENbrCell Struct Reference

Data Fields

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

8.482.1 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.

8.482.2 Field Documentation

8.482.2.1 uint8_t nas_umtsLTENbrCell::cellIsTDD

8.482.2.2 uint16_t nas_umtsLTENbrCell::earfcn

8.482.2.3 uint16_t nas_umtsLTENbrCell::pci

8.482.2.4 uint32_t nas_umtsLTENbrCell::rsrp

8.482.2.5 uint32_t nas_umtsLTENbrCell::rsrq

8.482.2.6 int16_t nas_umtsLTENbrCell::srxlev

8.483 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

8.483.1 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.

8.483.2 Field Documentation

8.483.2.1 uint8_t nas_UniversalTime::day

8.483.2.2 uint8_t nas_UniversalTime::dayOfWeek

8.483.2.3 uint8_t nas_UniversalTime::hour

8.483.2.4 uint8_t nas_UniversalTime::minute

8.483.2.5 uint8_t nas_UniversalTime::month

8.483.2.6 uint8_t nas_UniversalTime::second

8.483.2.7 uint16_t nas_UniversalTime::year

8.484 nas_wcdmaCellInfo Struct Reference

Data Fields

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

8.484.1 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.

8.484.2 Field Documentation

8.484.2.1 int16_t nas_wcdmaCellInfo::cpich_ecno

8.484.2.2 int16_t nas_wcdmaCellInfo::cpich_rscp

8.484.2.3 uint16_t nas_wcdmaCellInfo::psc

8.484.2.4 int16_t nas_wcdmaCellInfo::srxlev

8.485 nas_WCDMAECIOThresh Struct Reference

Data Fields

- uint8_t WCDMAECIOThreshListLen
- int16_t * pWCDMAECIOThreshList

8.485.1 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)

8.485.2 Field Documentation

8.485.2.1 int16_t* nas_WCDMAECIOThresh::pWCDMAECIOThreshList

8.485.2.2 uint8_t nas_WCDMAECIOThresh::WCDMAECIOThreshListLen

8.486 nas_WCDMAInfoLTENeighborCell Struct Reference

Data Fields

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

8.486.1 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.

8.486.2 Field Documentation

8.486.2.1 nas_umtsLTENbrCell nas_WCDMAInfoLTENeighborCell::UMTSLTENbrCell[255]

8.486.2.2 uint8_t nas_WCDMAInfoLTENeighborCell::umtsLTENbrCellLen

8.486.2.3 uint32_t nas_WCDMAInfoLTENeighborCell::wcdmaRRCState

8.487 nas_WCDMARSSIThresh Struct Reference

Data Fields

- uint8_t WCDMARSSIThreshListLen
- int16_t * pWCDMARSSIThreshList

8.487.1 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)

8.487.2 Field Documentation

8.487.2.1 int16_t* nas_WCDMARSSIThresh::pWCDMARSSIThreshList

8.487.2.2 uint8_t nas_WCDMARSSIThresh::WCDMARSSIThreshListLen

8.488 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

8.488.1 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

8.488.2 Field Documentation

8.488.2.1 uint32_t nas_WCDMASysInfo::cellId

8.488.2.2 uint8_t nas_WCDMASysInfo::cellIdValid

8.488.2.3 uint8_t nas_WCDMASysInfo::hsCallStatus

8.488.2.4 uint8_t nas_WCDMASysInfo::hsCallStatusValid

8.488.2.5 uint8_t nas_WCDMASysInfo::hsInd

8.488.2.6 uint8_t nas_WCDMASysInfo::hsIndValid

8.488.2.7 uint16_t nas_WCDMASysInfo::lac

8.488.2.8 uint8_t nas_WCDMASysInfo::lacValid

8.488.2.9 uint8_t nas_WCDMASysInfo::MCC[3]

8.488.2.10 uint8_t nas_WCDMASysInfo::MNC[3]

8.488.2.11 uint8_t nas_WCDMASysInfo::networkIdValid

8.488.2.12 uint16_t nas_WCDMASysInfo::psc

8.488.2.13 uint8_t nas_WCDMASysInfo::pscValid

8.488.2.14 uint8_t nas_WCDMASysInfo::regRejectInfoValid

8.488.2.15 uint8_t nas_WCDMASysInfo::rejCause

8.488.2.16 uint8_t nas_WCDMASysInfo::rejectSrvDomain

8.488.2.17 nas_sysInfoCommon nas_WCDMASysInfo::sysInfoWCDMA

8.489 nas_wcdmaUARFCN Struct Reference

Data Fields

- uint8_t status
- uint32_t uarfcn

8.489.1 Detailed Description

This structure contains WCDMA UARFCN infromation.

Parameters

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

8.489.2 Field Documentation

8.489.2.1 uint8_t nas_wcdmaUARFCN::status

8.489.2.2 uint32_t nas_wcdmaUARFCN::uarfcn

8.490 NASBandPreferenceTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint64_t band_pref

8.490.1 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

8.490.2 Field Documentation

8.490.2.1 uint64_t NASBandPreferenceTlv::band_pref

8.490.2.2 uint8_t NASBandPreferenceTlv::TlvPresent

8.491 nasCellLocationInfoResp Struct Reference

Data Fields

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

8.491.1 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.

8.491.2 Field Documentation

8.491.2.1 CDMAInfo∗ nasCellLocationInfoResp::pCDMAInfo

8.491.2.2 GERANInfo∗ nasCellLocationInfoResp::pGERANInfo

8.491.2.3 LTEInfoInterfreq∗ nasCellLocationInfoResp::pLTEInfoInterfreq

8.491.2.4 LTEInfoIntrafreq∗ nasCellLocationInfoResp::pLTEInfoIntrafreq

8.491.2.5 LTEInfoNeighboringGSM∗ nasCellLocationInfoResp::pLTEInfoNeighboringGSM

8.491.2.6 LTEInfoNeighboringWCDMA∗ nasCellLocationInfoResp::pLTEInfoNeighboringWCDMA

8.491.2.7 ULONG∗ nasCellLocationInfoResp::pUMTSCellID

8.491.2.8 UMTSInfo∗ nasCellLocationInfoResp::pUMTSInfo

8.491.2.9 WCDMAInfoLTENeighborCell∗ nasCellLocationInfoResp::pWCDMAInfoLTENeighborCell

8.492 NASEmergencyModeTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint8_t EmerMode

8.492.1 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

8.492.2 Field Documentation

8.492.2.1 uint8_t NASEmergencyModeTlv::EmerMode

8.492.2.2 uint8_t NASEmergencyModeTlv::TlvPresent

8.493 nasGet3GPP2SubscriptionInfoReq Struct Reference

Data Fields

- BYTE namID

8.493.1 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.

8.493.2 Field Documentation

8.493.2.1 BYTE nasGet3GPP2SubscriptionInfoReq::namID

8.494 nasGet3GPP2SubscriptionInfoResp Struct Reference

Data Fields

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

8.494.1 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

8.494.2 Field Documentation

8.494.2.1 CDMAChannel∗ nasGet3GPP2SubscriptionInfoResp::pCDMAChannel

8.494.2.2 dirNum∗ nasGet3GPP2SubscriptionInfoResp::pDirNum

8.494.2.3 homeSIDNID∗ nasGet3GPP2SubscriptionInfoResp::pHomeSIDNID

8.494.2.4 minBasedIMSI∗ nasGet3GPP2SubscriptionInfoResp::pMinBasedIMSI

8.494.2.5 namName∗ nasGet3GPP2SubscriptionInfoResp::pNAMNameInfo

8.494.2.6 trueIMSI∗ nasGet3GPP2SubscriptionInfoResp::pTrueIMSI

8.495 nasGetHDRColorCodeResp Struct Reference

Data Fields

- BYTE ∗ pColorCode

8.495.1 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

8.495.2 Field Documentation

8.495.2.1 BYTE* nasGetHDRColorCodeResp::pColorCode

8.496 nasGetLTECphyCa Struct Reference

Data Fields

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

8.496.1 Field Documentation

8.496.1.1 PhyCaAggPcellInfo nasGetLTECphyCa::sPhyCaAggPcellInfo

8.496.1.2 PhyCaAggScellDlBw nasGetLTECphyCa::sPhyCaAggScellDlBw

8.496.1.3 PhyCaAggScellIndex nasGetLTECphyCa::sPhyCaAggScellIndex

8.496.1.4 PhyCaAggScellIndType nasGetLTECphyCa::sPhyCaAggScellIndType

8.496.1.5 PhyCaAggScellInfo nasGetLTECphyCa::sPhyCaAggScellInfo

8.497 NasGetLTECphyCaInfo Struct Reference

Data Fields

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

8.497.1 Detailed Description

This structure contains the parameters for aggregation event information.

Parameters

PhyCaAggScell-IndType	• see NASPhyCaAggScellIndType
PhyCaAggScell-DlBw	• see NASPhyCaAggScellDlBw
PhyCaAggScell-Info	• see NASPhyCaAggScellInfo
PhyCaAggScell-IndType	• see NASPhyCaAggScellIndType

PhyCaAggPcellInfo	• see NASPhyCaAggPcellInfo
PhyCaAggScellIndex	• see NASPhyCaAggScellIndex

8.497.2 Field Documentation

8.497.2.1 NASPhyCaAggPcellInfo NasGetLTECphyCaInfo::PhyCaAggPcellInfo

8.497.2.2 NASPhyCaAggScellDlBw NasGetLTECphyCaInfo::PhyCaAggScellDlBw

8.497.2.3 NASPhyCaAggScellIndex NasGetLTECphyCaInfo::PhyCaAggScellIndex

8.497.2.4 NASPhyCaAggScellIndType NasGetLTECphyCaInfo::PhyCaAggScellIndType

8.497.2.5 NASPhyCaAggScellInfo NasGetLTECphyCaInfo::PhyCaAggScellInfo

8.498 nasGetLTECphyCaResp Struct Reference

Data Fields

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

8.498.1 Field Documentation

8.498.1.1 PhyCaAggPcellInfo* nasGetLTECphyCaResp::pPhyCaAggPcellInfo

8.498.1.2 PhyCaAggScellDlBw* nasGetLTECphyCaResp::pPhyCaAggScellDlBw

8.498.1.3 PhyCaAggScellIndex* nasGetLTECphyCaResp::pPhyCaAggScellIndex

8.498.1.4 PhyCaAggScellIndType* nasGetLTECphyCaResp::pPhyCaAggScellIndType

8.498.1.5 PhyCaAggScellInfo* nasGetLTECphyCaResp::pPhyCaAggScellInfo

8.499 nasGetSigInfoResp Struct Reference

Data Fields

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

8.499.1 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.

8.499.2 Field Documentation

8.499.2.1 CDMASSInfo* nasGetSigInfoResp::pCDMASSInfo

8.499.2.2 INT8* nasGetSigInfoResp::pGSMSSInfo

8.499.2.3 HDRSSInfo* nasGetSigInfoResp::pHDRSSInfo

8.499.2.4 LTESSInfo* nasGetSigInfoResp::pLTESSInfo

8.499.2.5 TDSCDMASigInfoExt* nasGetSigInfoResp::pTDSCDMASigInfoExt

8.499.2.6 INT8* nasGetSigInfoResp::pTDSCDMASigInfoRscp

8.499.2.7 CDMASSInfo* nasGetSigInfoResp::pWCDMASSInfo

8.500 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

8.500.1 Detailed Description

Structure for storing the SLQSNasGetSysInfo response parameters.

Parameters

pCDMASrvStatusInfo	• See SrvStatusInfo for more information.
pHDRSrvStatusInfo	• See SrvStatusInfo for more information.
pGSMSrvStatusInfo	• See GSMSrvStatusInfo for more information.
pWCDMASrvStatusInfo	• See GSMSrvStatusInfo for more information.
pLTESrvStatusInfo	• See GSMSrvStatusInfo for more information.
pCDMASysInfo	• See CDMASysInfo for more information.
pHDRSysInfo	• See HDRSysInfo for more information.
pGSMSysInfo	• See GSMSysInfo for more information.
pWCDMASysInfo	• See WCDMASysInfo for more information.
pLTESysInfo	• See LTESysInfo for more information.

pAddCDMASysInfo	• See AddCDMASysInfo for more information.
pAddHDRSysInfo	• System table index referencing the beginning of the geo in which the current serving system is present. • When the system index is not known, 0xFFFF is used.
pAddGSMSysInfo	• See AddSysInfo for more information.
pAddWCDMASysInfo	• See AddSysInfo for more information.
pAddLTESysInfo	• System table index referencing the beginning of the geo in which the current serving system is present. • When the system index is not known, 0xFFFF is used.
pGSMCallBarringSysInfo	• See CallBarringSysInfo for more information.
pWCDMACallBarringSysInfo	• See CallBarringSysInfo for more information.
pLTEVoiceSupportSysInfo	• Indicates voice support status on LTE. – 0x00 - Voice is not supported – 0x01 - Voice is supported
pGSMCipherDomainSysInfo	• Ciphering on the service domain. – 0x00 - No service – 0x01 - Circuit-switched only – 0x02 - Packet-switched only – 0x03 - Circuit-switched and packet-switched
pWCDMACipherDomainSysInfo	• Ciphering on the service domain. – 0x00 - No service – 0x01 - Circuit-switched only – 0x02 - Packet-switched only – 0x03 - Circuit-switched and packet-switched

8.500.2 Field Documentation

8.500.2.1 AddCDMASysInfo∗ nasGetSysInfoResp::pAddCDMASysInfo

8.500.2.2 AddSysInfo∗ nasGetSysInfoResp::pAddGSMSysInfo

8.500.2.3 WORD* nasGetSysInfoResp::pAddHDRSysInfo

8.500.2.4 WORD* nasGetSysInfoResp::pAddLTESysInfo

8.500.2.5 AddSysInfo* nasGetSysInfoResp::pAddWCDMASysInfo

8.500.2.6 SrvStatusInfo* nasGetSysInfoResp::pCDMASrvStatusInfo

8.500.2.7 CDMASysInfo* nasGetSysInfoResp::pCDMASysInfo

8.500.2.8 CallBarringSysInfo* nasGetSysInfoResp::pGSMCallBarringSysInfo

8.500.2.9 BYTE* nasGetSysInfoResp::pGSMCipherDomainSysInfo

8.500.2.10 GSMSrvStatusInfo* nasGetSysInfoResp::pGSMSrvStatusInfo

8.500.2.11 GSMSysInfo* nasGetSysInfoResp::pGSMSysInfo

8.500.2.12 SrvStatusInfo* nasGetSysInfoResp::pHDRSrvStatusInfo

8.500.2.13 HDRSysInfo* nasGetSysInfoResp::pHDRSysInfo

8.500.2.14 GSMSrvStatusInfo* nasGetSysInfoResp::pLTESrvStatusInfo

8.500.2.15 LTESysInfo* nasGetSysInfoResp::pLTESysInfo

8.500.2.16 BYTE* nasGetSysInfoResp::pLTEVoiceSupportSysInfo

8.500.2.17 CallBarringSysInfo* nasGetSysInfoResp::pWCDMACallBarringSysInfo

8.500.2.18 BYTE* nasGetSysInfoResp::pWCDMACipherDomainSysInfo

8.500.2.19 GSMSrvStatusInfo* nasGetSysInfoResp::pWCDMASrvStatusInfo

8.500.2.20 WCDMASysInfo* nasGetSysInfoResp::pWCDMASysInfo

8.501 nasGetTxRxInfoReq Struct Reference

Data Fields

- BYTE radio_if

8.501.1 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

8.501.2 Field Documentation

8.501.2.1 BYTE nasGetTxRxInfoReq::radio_if

8.502 nasGetTxRxInfoResp Struct Reference

Data Fields

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

8.502.1 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.

8.502.2 Field Documentation

8.502.2.1 rxInfo* nasGetTxRxInfoResp::pRXChain0Info

8.502.2.2 rxInfo* nasGetTxRxInfoResp::pRXChain1Info

8.502.2.3 txInfo* nasGetTxRxInfoResp::pTXInfo

8.503 NASGWAcqOrderPrefTlv Struct Reference

Data Fields

- uint8_t TlvPresent

- uint32_t GWAcqOrderPref

8.503.1 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

8.503.2 Field Documentation

8.503.2.1 uint32_t NASGWAcqOrderPrefTlv::GWAcqOrderPref

8.503.2.2 uint8_t NASGWAcqOrderPrefTlv::TlvPresent

8.504 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

8.504.1 Detailed Description

This structure contains the SLQSNasIndicationRegisterExt request parameters.

Parameters

pSystem-SelectionInd	[Optional] • System Selection Preference indication registration. The following callbacks would not be invoked if the indication is disabled. tFNRoamingIndicator tFNDataCapabilities and tFNServingSystem – 0x00 - Disable – 0x01 - Enable

pDDTMInd	[Optional] • DDTM (Data Dedicated Transmission Mode) indication registration. The following callbacks would not be invoked if the indication is disabled. tFNDDTM – 0x00 - Disable – 0x01 - Enable
pServing-SystemInd	[Optional] • Serving System indication registration. The following callbacks would not be invoked if the indication is disabled. tFNBandPreference – 0x00 - Disable – 0x01 - Enable
pDualStandBy-PrefInd	[Optional] • Dual Standby Preference indication registration.The following callbacks would not be invoked if the indication is disabled. tFNDualStandByPref – 0x00 - Disable – 0x01 - Enable
pSubscription-InfoInd	[Optional] • Subscription Information indication registration.The following callbacks would not be invoked if the indication is disabled. tFNSubscriptionInfo – 0x00 - Disable – 0x01 - Enable
pNetworkTime-Ind	[Optional] • Network Time indication registration.The following callbacks would not be invoked if the indication is disabled. tFNNetworkTime – 0x00 - Disable – 0x01 - Enable
pSysInfoInd	[Optional] • System Information indication registration.The following callbacks would not be invoked if the indication is disabled. tFNSysInfo – 0x00 - Disable – 0x01 - Enable
pSignalStrength-Ind	[Optional] • Signal Strength indication registration.The following callbacks would not be invoked if the indication is disabled. tFNSigInfo – 0x00 - Disable – 0x01 - Enable

pErrorRateInd	[Optional] • Error Rate indication registration.The following callbacks would not be invoked if the indication is disabled. tFNErrRate – 0x00 - Disable – 0x01 - Enable
pHDRNewUATI-AssInd	[Optional] • HDR New UATI Assigned indication registration.The following callbacks would not be invoked if the indication is disabled. tFNHDRUATIUpdate – 0x00 - Disable – 0x01 - Enable
pHDRSession-CloseInd	[Optional] • HDR Session Closed indication registration.The following callbacks would not be invoked if the indication is disabled. tFNHDRSessionClose – 0x00 - Disable – 0x01 - Enable
pManaged-RoamingInd	[Optional] • Managed Roaming indication registration.The following callbacks would not be invoked if the indication is disabled. tFNManagedRoaming – 0x00 - Disable – 0x01 - Enable
pNetworkReject-Ind	[Optional] • Network reject information indication registration.The following callbacks would not be invoked if the indication is disabled. – 0x00 - Disable – 0x01 - Enable
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
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.

8.504.2 Field Documentation

8.504.2.1 BYTE* nasIndicationRegisterReq::pDDTMInd

8.504.2.2 BYTE* nasIndicationRegisterReq::pDualStandByPrefInd

8.504.2.3 BYTE* nasIndicationRegisterReq::pErrorRateInd

8.504.2.4 BYTE* nasIndicationRegisterReq::pHDRNewUATIAssInd

8.504.2.5 BYTE* nasIndicationRegisterReq::pHDRSessionCloseInd

8.504.2.6 BYTE* nasIndicationRegisterReq::pLTECphyCa

8.504.2.7 BYTE* nasIndicationRegisterReq::pManagedRoamingInd

8.504.2.8 BYTE* nasIndicationRegisterReq::pNetworkRejectInd

8.504.2.9 BYTE* nasIndicationRegisterReq::pNetworkTimeInd

8.504.2.10 BYTE* nasIndicationRegisterReq::pServingSystemInd

8.504.2.11 BYTE* nasIndicationRegisterReq::pSignalStrengthInd

8.504.2.12 BYTE* nasIndicationRegisterReq::pSubscriptionInfoInd

8.504.2.13 BYTE* nasIndicationRegisterReq::pSuppressSysInfoInd

8.504.2.14 BYTE* nasIndicationRegisterReq::pSysInfoInd

8.504.2.15 BYTE* nasIndicationRegisterReq::pSystemSelectionInd

8.505 nasInitNetworkReg Struct Reference

Data Fields

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

8.505.1 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.

8.505.2 Field Documentation

8.505.2.1 ULONG* nasInitNetworkReg::pChangeDuration

8.505.2.2 BYTE* nasInitNetworkReg::pMncPcsDigitStatus

8.505.2.3 MNRInfo* nasInitNetworkReg::pMNRInfo

8.505.2.4 ULONG nasInitNetworkReg::regAction

8.506 NASLTEBandPreferenceTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint64_t LTEBandPref

8.506.1 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

8.506.2 Field Documentation

8.506.2.1 uint64_t NASLTEBandPreferenceTlv::LTEBandPref

8.506.2.2 uint8_t NASLTEBandPreferenceTlv::TlvPresent

8.507 NASLteNasReleaseInfoTlv Struct Reference

Data Fields

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

8.507.1 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

8.507.2 Field Documentation

8.507.2.1 uint8_t NASLteNasReleaseInfoTlv::nas_major

8.507.2.2 uint8_t NASLteNasReleaseInfoTlv::nas_minor

8.507.2.3 uint8_t NASLteNasReleaseInfoTlv::nas_release

8.507.2.4 uint8_t NASLteNasReleaseInfoTlv::TlvPresent

8.508 NASModePreferenceTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint16_t ModePref

8.508.1 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

8.508.2 Field Documentation

8.508.2.1 uint16_t NASModePreferenceTlv::ModePref

8.508.2.2 uint8_t NASModePreferenceTlv::TlvPresent

8.509 NASNetSelPreferenceTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint8_t NetSelPref

8.509.1 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

8.509.2 Field Documentation

8.509.2.1 uint8_t NASNetSelPreferenceTlv::NetSelPref

8.509.2.2 uint8_t NASNetSelPreferenceTlv::TlvPresent

8.510 nasNetworkReject Struct Reference

Data Fields

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

8.510.1 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.

8.510.2 Field Documentation

8.510.2.1 CsgId* nasNetworkReject::pCsgId

8.510.2.2 PlmnID* nasNetworkReject::pPlmnId

8.510.2.3 BYTE nasNetworkReject::radioIf

8.510.2.4 BYTE nasNetworkReject::rejectCause

8.510.2.5 TYPE_OF_SERVICE_DOMAIN nasNetworkReject::serviceDomain

8.511 nasNetworkTime Struct Reference

Data Fields

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

8.511.1 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

8.511.2 Field Documentation

8.511.2.1 BYTE* nasNetworkTime::pDayltSavAdj

8.511.2.2 BYTE* nasNetworkTime::pRadioInterface

8.511.2.3 BYTE* nasNetworkTime::pTimeZone

8.511.2.4 UniversalTime nasNetworkTime::universalTime

8.512 nasOperatorNameResp Struct Reference

Data Fields

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

8.512.1 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.

8.512.2 Field Documentation

8.512.2.1 PLMNNetworkNameData∗ nasOperatorNameResp::pNITZInformation

8.512.2.2 operatorNameString∗ nasOperatorNameResp::pOperatorNameString

8.512.2.3 operatorPLMNList∗ nasOperatorNameResp::pOperatorPLMNList

8.512.2.4 PLMNNetworkName∗ nasOperatorNameResp::pPLMNNetworkName

8.512.2.5 serviceProviderName∗ nasOperatorNameResp::pSrvcProviderName

8.513 NASOTAMessageTlv Struct Reference

Data Fields

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

8.513.1 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

8.513.2 Field Documentation

8.513.2.1 uint8_t NASOTAMessageTlv::data_buf[2048]

8.513.2.2 uint16_t NASOTAMessageTlv::data_len

8.513.2.3 uint32_t NASOTAMessageTlv::message_type

8.513.2.4 uint8_t NASOTAMessageTlv::TlvPresent

8.514 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

8.514.1 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.

8.514.2 Field Documentation

8.514.2.1 LITEQMI_NAS_LTE_CPHY_CA_BW_NRB NASPhyCaAggPcellInfo::dl_bw_value

8.514.2.2 uint32_t NASPhyCaAggPcellInfo::freq

8.514.2.3 uint32_t NASPhyCaAggPcellInfo::iLTEbandValue

8.514.2.4 uint32_t NASPhyCaAggPcellInfo::pci

8.514.2.5 uint8_t NASPhyCaAggPcellInfo::TlvPresent

8.515 NASPhyCaAggScellDlBw Struct Reference

Data Fields

- LITEQMI_NAS_LTE_CPHY_CA_BW_NRB dl_bw_value
- uint8_t TlvPresent

8.515.1 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.

8.515.2 Field Documentation

8.515.2.1 LITEQMI_NAS_LTE_CPHY_CA_BW_NRB NASPhyCaAggScellDlBw::dl_bw_value

8.515.2.2 uint8_t NASPhyCaAggScellDlBw::TlvPresent

8.516 NASPhyCaAggScellIndex Struct Reference

Data Fields

- uint8_t scell_idx
- uint8_t TlvPresent

8.516.1 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.

8.516.2 Field Documentation

8.516.2.1 uint8_t NASPhyCaAggScellIndex::scell_idx

8.516.2.2 uint8_t NASPhyCaAggScellIndex::TlvPresent

8.517 NASPhyCaAggScellIndType Struct Reference

Data Fields

- uint32_t pci
- uint32_t freq
- 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 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.

8.517.2 Field Documentation

8.517.2.1 uint32_t NASPhyCaAggScellIndType::freq

8.517.2.2 uint32_t NASPhyCaAggScellIndType::pci

8.517.2.3 LITEQMI_NAS_LTE_CPHY_SCELL_STATE NASPhyCaAggScellIndType::scell_state

8.517.2.4 uint8_t NASPhyCaAggScellIndType::TlvPresent

8.518 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

8.518.1 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.

8.518.2 Field Documentation

8.518.2.1 LITEQMI_NAS_LTE_CPHY_CA_BW_NRB NASPhyCaAggScellInfo::dl_bw_value

8.518.2.2 uint32_t NASPhyCaAggScellInfo::freq

8.518.2.3 uint32_t NASPhyCaAggScellInfo::iLTEbandValue

8.518.2.4 uint32_t NASPhyCaAggScellInfo::pci

8.518.2.5 LITEQMI_NAS_LTE_CPHY_SCELL_STATE NASPhyCaAggScellInfo::scell_state

8.518.2.6 uint8_t NASPhyCaAggScellInfo::TlvPresent

8.519 nasPLMNNameReq Struct Reference

Data Fields

- WORD mcc
- WORD mnc

- BYTE * pMncPcsStatus

8.519.1 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.

8.519.2 Field Documentation

8.519.2.1 WORD nasPLMNNameReq::mcc

8.519.2.2 WORD nasPLMNNameReq::mnc

8.519.2.3 BYTE* nasPLMNNameReq::pMncPcsStatus

8.520 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]

8.520.1 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

8.520.2 Field Documentation

8.520.2.1 BYTE nasPLMNNameResp::longName[255]

8.520.2.2 BYTE nasPLMNNameResp::longNameCI

8.520.2.3 BYTE nasPLMNNameResp::longNameEn

8.520.2.4 BYTE nasPLMNNameResp::longNameLen

8.520.2.5 BYTE nasPLMNNameResp::longNameSB

8.520.2.6 BYTE nasPLMNNameResp::shortName[255]

8.520.2.7 BYTE nasPLMNNameResp::shortNameCI

8.520.2.8 BYTE nasPLMNNameResp::shortNameEn

8.520.2.9 BYTE nasPLMNNameResp::shortNameLen

8.520.2.10 BYTE nasPLMNNameResp::shortNameSB

8.520.2.11 BYTE nasPLMNNameResp::spn[255]

8.520.2.12 BYTE nasPLMNNameResp::spnEncoding

8.520.2.13 BYTE nasPLMNNameResp::spnLength

8.521 NASPRLPreferenceTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint16_t PRLPref

8.521.1 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

8.521.2 Field Documentation

8.521.2.1 uint16_t NASPRLPreferenceTlv::PRLPref

8.521.2.2 uint8_t NASPRLPreferenceTlv::TlvPresent

8.522 NASQmiCbkNasSwiOTAMessageInd Struct Reference

Data Fields

- NASOTAMessageTlv otaMsgTlv
- NASLteNasReleaseInfoTlv nasRelInfoTlv
- NASTimeInfoTlv timeTlv

8.522.1 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

8.522.2 Field Documentation

8.522.2.1 NASLteNasReleaseInfoTlv NASQmiCbkNasSwiOTAMessageInd::nasRelInfoTlv

8.522.2.2 NASOTAMessageTlv NASQmiCbkNasSwiOTAMessageInd::otaMsgTlv

8.522.2.3 NASTimeInfoTlv NASQmiCbkNasSwiOTAMessageInd::timeTlv

8.523 NASQmiCbkNasSystemSelPrefInd Struct Reference

Data Fields

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

8.523.1 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

8.523.2 Field Documentation

8.523.2.1 NASBandPreferenceTlv NASQmiCbkNasSystemSelPrefInd::BPTlv

8.523.2.2 NASEmergencyModeTlv NASQmiCbkNasSystemSelPrefInd::EMTlv

8.523.2.3 NASGWAcqOrderPrefTlv NASQmiCbkNasSystemSelPrefInd::GWAOPTlv

8.523.2.4 NASLTEBandPreferenceTlv NASQmiCbkNasSystemSelPrefInd::LBPTlv

8.523.2.5 NASModePreferenceTlv NASQmiCbkNasSystemSelPrefInd::MPTlv

8.523.2.6 NASNetSelPreferenceTlv NASQmiCbkNasSystemSelPrefInd::NSPTlv

8.523.2.7 NASPRLPreferenceTlv NASQmiCbkNasSystemSelPrefInd::PRLPTlv

8.523.2.8 NASRoamPreferenceTlv NASQmiCbkNasSystemSelPrefInd::RPTlv

8.523.2.9 NASServDomainPrefTlv NASQmiCbkNasSystemSelPrefInd::SDPTlv

8.524 NASRoamPreferenceTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint16_t RoamPref

8.524.1 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

8.524.2 Field Documentation

8.524.2.1 uint16_t NASRoamPreferenceTlv::RoamPref

8.524.2.2 uint8_t NASRoamPreferenceTlv::TlvPresent

8.525 NASServDomainPrefTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint32_t SrvDomainPref

8.525.1 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

8.525.2 Field Documentation

8.525.2.1 uint32_t NASServDomainPrefTlv::SrvDomainPref

8.525.2.2 uint8_t NASServDomainPrefTlv::TlvPresent

8.526 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

8.526.1 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

8.526.2 Field Documentation

8.526.2.1 uint8_t NASServingSystemInfo::csAttachState

8.526.2.2 uint8_t NASServingSystemInfo::hdrPersonality

8.526.2.3 uint8_t NASServingSystemInfo::psAttachState

8.526.2.4 uint8_t NASServingSystemInfo::radioInterfaceList[255]

8.526.2.5 uint8_t NASServingSystemInfo::radioInterfaceNo

8.526.2.6 uint8_t NASServingSystemInfo::registrationState

8.526.2.7 uint8_t NASServingSystemInfo::selectedNetwork

8.527 nasSigInfo Struct Reference

Data Fields

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

8.527.1 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.

8.527.2 Field Documentation

8.527.2.1 CDMASSInfo* nasSigInfo::pCDMASigInfo

8.527.2.2 INT8* nasSigInfo::pGSMSigInfo

8.527.2.3 HDRSSInfo* nasSigInfo::pHDRSigInfo

8.527.2.4 LTESSInfo* nasSigInfo::pLTESigInfo

8.527.2.5 INT8* nasSigInfo::pRscp

8.527.2.6 TDSCDMASigInfoExt* nasSigInfo::pTDSCDMASigInfoExt

8.527.2.7 CDMASSInfo* nasSigInfo::pWCDMASigInfo

8.528 nasSwiGetChannelLockResp Struct Reference

Data Fields

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

8.528.1 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

8.528.2 Field Documentation

8.528.2.1 lteEARFCN∗ nasSwiGetChannelLockResp::pLteEARFCN

8.528.2.2 ltePCI∗ nasSwiGetChannelLockResp::pLtePCI

8.528.2.3 wcdmaUARFCN∗ nasSwiGetChannelLockResp::pWcdmaUARFCN

8.529 NasSwiIndReg Struct Reference

Data Fields

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

8.529.1 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

8.529.2 Field Documentation

8.529.2.1 BYTE NasSwiIndReg::gsmUmtsDl

8.529.2.2 BYTE NasSwiIndReg::gsmUmtsUl

8.529.2.3 BYTE NasSwiIndReg::lteEmmDl

8.529.2.4 BYTE NasSwiIndReg::lteEmmUl

8.529.2.5 BYTE NasSwiIndReg::lteEsmDl

8.529.2.6 BYTE NasSwiIndReg::lteEsmUl

8.529.2.7 BYTE* NasSwiIndReg::pRankIndicatorInd

8.529.2.8 BYTE* NasSwiIndReg::pTimer

8.530 nasSwiSetChannelLockReq Struct Reference

Data Fields

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

8.530.1 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

8.530.2 Field Documentation

8.530.2.1 lteEARFCN∗ nasSwiSetChannelLockReq::pLteEARFCN

8.530.2.2 ltePCI∗ nasSwiSetChannelLockReq::pLtePCI

8.530.2.3 wcdmaUARFCN∗ nasSwiSetChannelLockReq::pWcdmaUARFCN

8.531 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

8.531.1 Detailed Description

Structure for storing the nasSysInfo indication parameters.

Parameters

pCDMASrvStatusInfo	• See SrvStatusInfo for more information.
pHDRSrvStatusInfo	• See SrvStatusInfo for more information.
pGSMSrvStatusInfo	• See GSMSrvStatusInfo for more information.
pWCDMASrvStatusInfo	• See GSMSrvStatusInfo for more information.
pLTESrvStatusInfo	• See GSMSrvStatusInfo for more information.
pCDMASysInfo	• See CDMASysInfo for more information.
pHDRSysInfo	• See HDRSysInfo for more information.
pGSMSysInfo	• See GSMSysInfo for more information.
pWCDMASysInfo	• See WCDMASysInfo for more information.
pLTESysInfo	• See LTESysInfo for more information.
pAddCDMASysInfo	• See AddCDMASysInfo for more information.
pAddHDRSysInfo	• System table index referencing the beginning of the geo in which the current serving system is present. • When the system index is not known, 0xFFFF is used.
pAddGSMSysInfo	• See AddSysInfo for more information.
pAddWCDMASysInfo	• See AddSysInfo for more information.
pAddLTESysInfo	• System table index referencing the beginning of the geo in which the current serving system is present. • When the system index is not known, 0xFFFF is used.
pGSMCallBarringSysInfo	• See CallBarringSysInfo for more information.

pWCDMACallBarringSysInfo	• See CallBarringSysInfo for more information.
pLTEVoiceSupportSysInfo	• Indicates voice support status on LTE. – 0x00 - Voice is not supported – 0x01 - Voice is supported
pGSMCipherDomainSysInfo	• Ciphering on the service domain. – 0x00 - No service – 0x01 - Circuit-switched only – 0x02 - Packet-switched only – 0x03 - Circuit-switched and packet-switched
pWCDMACipherDomainSysInfo	• Ciphering on the service domain. – 0x00 - No service – 0x01 - Circuit-switched only – 0x02 - Packet-switched only – 0x03 - Circuit-switched and packet-switched
pSysInfoNoChange	• System Info No Change. • Flag used to notify clients that a request to select a network ended with no change in the PLMN. – 0x01 - No change in system information

8.531.2 Field Documentation

8.531.2.1 AddCDMASysInfo∗ nasSysInfo::pAddCDMASysInfo

8.531.2.2 AddSysInfo∗ nasSysInfo::pAddGSMSysInfo

8.531.2.3 WORD∗ nasSysInfo::pAddHDRSysInfo

8.531.2.4 WORD∗ nasSysInfo::pAddLTESysInfo

8.531.2.5 AddSysInfo∗ nasSysInfo::pAddWCDMASysInfo

8.531.2.6 SrvStatusInfo∗ nasSysInfo::pCDMASrvStatusInfo

8.531.2.7 CDMASysInfo∗ nasSysInfo::pCDMASysInfo

8.531.2.8 CallBarringSysInfo∗ nasSysInfo::pGSMCallBarringSysInfo

8.531.2.9 BYTE∗ nasSysInfo::pGSMCipherDomainSysInfo

8.531.2.10 GSMSrvStatusInfo∗ nasSysInfo::pGSMSrvStatusInfo

8.531.2.11 GSMSysInfo∗ nasSysInfo::pGSMSysInfo

8.531.2.12 SrvStatusInfo* nasSysInfo::pHDRSrvStatusInfo

8.531.2.13 HDRSysInfo* nasSysInfo::pHDRSysInfo

8.531.2.14 GSMSrvStatusInfo* nasSysInfo::pLTESrvStatusInfo

8.531.2.15 LTESysInfo* nasSysInfo::pLTESysInfo

8.531.2.16 BYTE* nasSysInfo::pLTEVoiceSupportSysInfo

8.531.2.17 BYTE* nasSysInfo::pSysInfoNoChange

8.531.2.18 CallBarringSysInfo* nasSysInfo::pWCDMACallBarringSysInfo

8.531.2.19 BYTE* nasSysInfo::pWCDMACipherDomainSysInfo

8.531.2.20 GSMSrvStatusInfo* nasSysInfo::pWCDMASrvStatusInfo

8.531.2.21 WCDMASysInfo* nasSysInfo::pWCDMASysInfo

8.532 NASTimeInfoTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint64_t time

8.532.1 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

8.532.2 Field Documentation

8.532.2.1 uint64_t NASTimeInfoTlv::time

8.532.2.2 uint8_t NASTimeInfoTlv::TlvPresent

8.533 nasTimers Struct Reference

Data Fields

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

8.533.1 Detailed Description

This structure contains Network Timer

Parameters

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

8.533.2 Field Documentation

8.533.2.1 char nasTimers::t3396_apn[101]

8.533.2.2 BYTE nasTimers::t3396_plmn_id[3]

8.533.2.3 ULONG nasTimers::t3396_val

8.534 netSelectionPref Struct Reference

Data Fields

- BYTE netReg
- WORD mcc
- WORD mnc

8.534.1 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

8.534.2 Field Documentation

8.534.2.1 WORD netSelectionPref::mcc

8.534.2.2 WORD netSelectionPref::mnc

8.534.2.3 BYTE netSelectionPref::netReg

8.535 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

8.535.1 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

8.535.2 Field Documentation

8.535.2.1 ULONGLONG NetStats::rx_bytes

8.535.2.2 ULONG NetStats::rx_errors

8.535.2.3 ULONG NetStats::rx_overflows

8.535.2.4 ULONG NetStats::rx_packets

8.535.2.5 ULONGLONG NetStats::tx_bytes

8.535.2.6 ULONG NetStats::tx_errors

8.535.2.7 ULONG NetStats::tx_overflows

8.535.2.8 ULONG NetStats::tx_packets

8.536 NetworkDebugResp Struct Reference

Data Fields

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

8.536.1 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.

8.536.2 Field Documentation

8.536.2.1 DataStatusDetail* NetworkDebugResp::pDataStatusDetail

8.536.2.2 DeviceConfigDetail* NetworkDebugResp::pDeviceConfigDetail

8.536.2.3 NetworkStat1x∗ NetworkDebugResp::pNetworkStat1x

8.536.2.4 NetworkStatEVDO∗ NetworkDebugResp::pNetworkStatEVDO

8.536.2.5 BYTE∗ NetworkDebugResp::pObjectVer

8.537 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

8.537.1 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

8.537.2 Field Documentation

8.537.2.1 BYTE NetworkStat1x::ActSetCnt

8.537.2.2 BYTE NetworkStat1x::NeighborSetCnt

8.537.2.3 ActPilotPNElement* NetworkStat1x::pActPilotPNElements

8.537.2.4 WORD* NetworkStat1x::pNeighborSetPilotPN

8.537.2.5 WORD NetworkStat1x::RX_EC_IO

8.537.2.6 ULONG NetworkStat1x::RX_PWR

8.537.2.7 WORD NetworkStat1x::SO

8.537.2.8 BYTE NetworkStat1x::State

8.537.2.9 ULONG NetworkStat1x::TX_PWR

8.538 NetworkStatEVDO Struct Reference

Data Fields

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

8.538.1 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)

8.538.2 Field Documentation

8.538.2.1 BYTE NetworkStatEVDO::MACIndex

8.538.2.2 WORD NetworkStatEVDO::PER

8.538.2.3 WORD NetworkStatEVDO::PilotEnergy

8.538.2.4 WORD* NetworkStatEVDO::pSectorID

8.538.2.5 WORD NetworkStatEVDO::RX_PWR

8.538.2.6 BYTE NetworkStatEVDO::SectorIDLen

8.538.2.7 BYTE NetworkStatEVDO::SNR

8.538.2.8 BYTE NetworkStatEVDO::State

8.539 newMTMessageTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- sMSMTMessageInfo MTMessageInfo

8.539.1 Detailed Description

This structure contains MT message TLV information.

Parameters

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

8.539.2 Field Documentation

8.539.2.1 sMSMTMessageInfo newMTMessageTlv::MTMessageInfo

8.539.2.2 uint8_t newMTMessageTlv::TlvPresent

8.540 newPwdData Struct Reference

Data Fields

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

8.540.1 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.

8.540.2 Field Documentation

8.540.2.1 BYTE newPwdData::newPwd[4]

8.540.2.2 BYTE newPwdData::newPwdAgain[4]

8.541 nmrCellInfo Struct Reference

Data Fields

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

8.541.1 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

8.541.2 Field Documentation

8.541.2.1 WORD nmrCellInfo::nmrArfcn

8.541.2.2 BYTE nmrCellInfo::nmrBsic

8.541.2.3 ULONG nmrCellInfo::nmrCellID

8.541.2.4 WORD nmrCellInfo::nmrLac

8.541.2.5 BYTE nmrCellInfo::nmrPlmn[3]

8.541.2.6 WORD nmrCellInfo::nmrRxLev

8.542 NSSAudioCtrl Struct Reference

Data Fields

- BYTE upLink
- BYTE downLink

8.542.1 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].

8.542.2 Field Documentation

8.542.2.1 BYTE NSSAudioCtrl::downLink

8.542.2.2 BYTE NSSAudioCtrl::upLink

8.543 NWProfile Struct Reference

Data Fields

- WORD tech
- BYTE * pProfSz
- WORD * pProfValues

8.543.1 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	

8.543.2 Field Documentation

8.543.2.1 BYTE* NWProfile::pProfSz

8.543.2.2 WORD* NWProfile::pProfValues

8.543.2.3 WORD NWProfile::tech

8.544 omaDmConfigTlv 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	• 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)

8.544.2 Field Documentation

8.544.2.1 BYTE omaDmConfigTlv::alertmsg[256]

8.544.2.2 USHORT omaDmConfigTlv::alertmsglength

8.544.2.3 BYTE omaDmConfigTlv::state

8.544.2.4 BYTE omaDmConfigTlv::userInputReq

8.544.2.5 USHORT omaDmConfigTlv::userInputTimeout

8.545 omaDmConfigTlvExt Struct Reference

Data Fields

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

8.545.1 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

8.545.2 Field Documentation

8.545.2.1 BYTE omaDmConfigTlvExt::alertmsg[256]

8.545.2.2 USHORT omaDmConfigTlvExt::alertmsglength

8.545.2.3 BYTE omaDmConfigTlvExt::state

8.545.2.4 BYTE omaDmConfigTlvExt::userInputReq

8.545.2.5 USHORT omaDmConfigTlvExt::userInputTimeout

8.546 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

8.546.1 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

8.546.2 Field Documentation

8.546.2.1 BYTE omaDmFotaTlv::description[256]

8.546.2.2 USHORT omaDmFotaTlv::descriptionlength

8.546.2.3 ULONG omaDmFotaTlv::fwdloadsize

8.546.2.4 ULONG omaDmFotaTlv::fwloadComplete

8.546.2.5 USHORT omaDmFotaTlv::namelength

8.546.2.6 BYTE omaDmFotaTlv::package_name[256]

8.546.2.7 BYTE omaDmFotaTlv::sessionType

8.546.2.8 BYTE omaDmFotaTlv::severity

8.546.2.9 BYTE omaDmFotaTlv::state

8.546.2.10 USHORT omaDmFotaTlv::updateCompleteStatus

8.546.2.11 BYTE omaDmFotaTlv::userInputReq

8.546.2.12 USHORT omaDmFotaTlv::userInputTimeout

8.546.2.13 BYTE omaDmFotaTlv::version[256]

8.546.2.14 USHORT omaDmFotaTlv::versionlength

8.547 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]

8.547.1 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)

8.547.2 Field Documentation

8.547.2.1 BYTE omaDmFotaTlvExt::description[256]

8.547.2.2 USHORT omaDmFotaTlvExt::descriptionlength

8.547.2.3 USHORT omaDmFotaTlvExt::fumoResultCode

8.547.2.4 USHORT omaDmFotaTlvExt::namelength

8.547.2.5 BYTE omaDmFotaTlvExt::package_name[256]

8.547.2.6 ULONG omaDmFotaTlvExt::packageSize

8.547.2.7 ULONG omaDmFotaTlvExt::receivedBytes

8.547.2.8 BYTE omaDmFotaTlvExt::reserved

8.547.2.9 BYTE omaDmFotaTlvExt::state

8.547.2.10 USHORT omaDmFotaTlvExt::userInputTimeout

8.547.2.11 BYTE omaDmFotaTlvExt::version[256]

8.547.2.12 USHORT omaDmFotaTlvExt::versionlength

8.548 omaDmNotificationsTlv Struct Reference

Data Fields

- BYTE notification
- USHORT sessionStatus

8.548.1 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.

8.548.2 Field Documentation

8.548.2.1 BYTE omaDmNotificationsTlv::notification

8.548.2.2 USHORT omaDmNotificationsTlv::sessionStatus

8.549 operatorNameString Struct Reference

Data Fields

- BYTE PLMNName [255]

8.549.1 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.

8.549.2 Field Documentation

8.549.2.1 BYTE operatorNameString::PLMNName[255]

8.550 OperatorPLMNData Struct Reference

Data Fields

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

8.550.1 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.

8.550.2 Field Documentation

8.550.2.1 WORD OperatorPLMNData::lac1

8.550.2.2 WORD OperatorPLMNData::lac2

8.550.2.3 BYTE OperatorPLMNData::mcc[3]

8.550.2.4 BYTE OperatorPLMNData::mnc[3]

8.550.2.5 BYTE OperatorPLMNData::PLMNRecID

8.551 operatorPLMNList Struct Reference

Data Fields

- WORD numInstance
- OperatorPLMNData PLMNData [255]

8.551.1 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).

8.551.2 Field Documentation

8.551.2.1 WORD operatorPLMNList::numInstance

8.551.2.2 OperatorPLMNData operatorPLMNList::PLMNData[255]

8.552 pack_dms_ActivateAutomatic_t Struct Reference

Data Fields

- uint8_t actCode [81]

8.552.1 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"

8.552.2 Field Documentation

8.552.2.1 uint8_t pack_dms_ActivateAutomatic_t::actCode[81]

8.553 pack_dms_GetCustFeaturesV2_t Struct Reference

Data Fields

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

8.553.1 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

8.553.2 Field Documentation

8.553.2.1 uint8_t pack_dms_GetCustFeaturesV2_t::cust_id[64+1]

8.553.2.2 uint8_t pack_dms_GetCustFeaturesV2_t::list_type

8.553.2.3 uint16_t pack_dms_GetCustFeaturesV2_t::Tlvresult

8.554 pack_dms_ResetToFactoryDefaults_t Struct Reference

Data Fields

- uint8_t spc [6]

8.554.1 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)

8.554.2 Field Documentation

8.554.2.1 uint8_t pack_dms_ResetToFactoryDefaults_t::spc[6]

8.555 pack_dms_SetActivationStatusCallback_t Struct Reference

Data Fields

- uint8_t activationState

8.555.1 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

8.555.2 Field Documentation

8.555.2.1 uint8_t pack_dms_SetActivationStatusCallback_t::activationState

8.556 pack_dms_SetCrashAction_t Struct Reference

Data Fields

- uint8_t crashAction

8.556.1 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

8.556.2 Field Documentation

8.556.2.1 uint8_t pack_dms_SetCrashAction_t::crashAction

8.557 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

8.557.1 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

8.557.2 Field Documentation

8.557.2.1 uint8_t pack_dms_SetCustFeature_t::DHCPRelayEnabled

8.557.2.2 uint8_t pack_dms_SetCustFeature_t::DisableIMSI

8.557.2.3 uint32_t pack_dms_SetCustFeature_t::GpsEnable

8.557.2.4 uint8_t pack_dms_SetCustFeature_t::GPSLPM

8.557.2.5 uint8_t pack_dms_SetCustFeature_t::GPSSel

8.557.2.6 uint16_t pack_dms_SetCustFeature_t::IPFamSupport

8.557.2.7 uint8_t pack_dms_SetCustFeature_t::IsVoiceEnabled

8.557.2.8 uint8_t pack_dms_SetCustFeature_t::RMAutoConnect

8.557.2.9 uint8_t pack_dms_SetCustFeature_t::SMSSupport

8.558 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

8.558.1 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

8.558.2 Field Documentation

8.558.2.1 uint8_t pack_dms_SetCustFeaturesV2_t::cust_id[64+1]

8.558.2.2 uint8_t pack_dms_SetCustFeaturesV2_t::cust_value[8+1]

8.558.2.3 uint16_t pack_dms_SetCustFeaturesV2_t::Tlvresult

8.558.2.4 uint16_t pack_dms_SetCustFeaturesV2_t::value_length

8.559 pack_dms_SetEventReport_t Struct Reference

Data Fields

- uint8_t mode

8.559.1 Detailed Description

This structure is used to store pack_dms_SetEventReport parameters.

Parameters

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

8.559.2 Field Documentation

8.559.2.1 uint8_t pack_dms_SetEventReport_t::mode

8.560 pack_dms_SetPower_t Struct Reference

Data Fields

- uint32_t mode
- uint16_t Tlvresult

8.560.1 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.

8.560.2 Field Documentation

8.560.2.1 uint32_t pack_dms_SetPower_t::mode

8.560.2.2 uint16_t pack_dms_SetPower_t::Tlvresult

8.561 pack_dms_SetUSBComp_t Struct Reference

Data Fields

- uint8_t USBComp
- uint16_t Tlvresult

8.561.1 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.

8.561.2 Field Documentation

8.561.2.1 uint16_t pack_dms_SetUSBComp_t::Tlvresult

8.561.2.2 uint8_t pack_dms_SetUSBComp_t::USBComp

8.562 pack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference

Data Fields

- uint8_t resetInfoInd

8.562.1 Detailed Description

This structure contains set registration state for different indication pack

Parameters

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

8.562.2 Field Documentation

8.562.2.1 uint8_t pack_dms_SLQSDmsSwiIndicationRegister_t::resetInfoInd

8.563 pack_dms_SLQSSwiGetCrashInfo_t Struct Reference

Data Fields

- uint8_t clear

8.563.1 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

8.563.2 Field Documentation

8.563.2.1 uint8_t pack_dms_SLQSSwiGetCrashInfo_t::clear

8.564 pack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference

Data Fields

- uint8_t * pDestSMSNum
- uint8_t * pDestSMSContent

8.564.1 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.

8.564.2 Field Documentation

8.564.2.1 uint8_t* pack_dms_SLQSSwiSetDyingGaspCfg_t::pDestSMSContent

8.564.2.2 uint8_t* pack_dms_SLQSSwiSetDyingGaspCfg_t::pDestSMSNum

8.565 pack_dms_SLQSSwiSetHostDevInfo_t Struct Reference

Data Fields

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

8.565.1 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

8.565.2 Field Documentation

8.565.2.1 char pack_dms_SLQSSwiSetHostDevInfo_t::manString[255]

8.565.2.2 char pack_dms_SLQSSwiSetHostDevInfo_t::modelString[255]

8.565.2.3 char pack_dms_SLQSSwiSetHostDevInfo_t::plasmaIDString[255]

8.565.2.4 char pack_dms_SLQSSwiSetHostDevInfo_t::swVerString[255]

8.566 pack_dms_SLQSSwiSetOSInfo_t Struct Reference

Data Fields

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

8.566.1 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.

8.566.2 Field Documentation

8.566.2.1 char pack_dms_SLQSSwiSetOSInfo_t::nameString[255]

8.566.2.2 char pack_dms_SLQSSwiSetOSInfo_t::versionString[255]

8.567 pack_dms_UIMChangePIN_t Struct Reference

Data Fields

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

8.567.1 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

8.567.2 Field Documentation

8.567.2.1 uint8_t pack_dms_UIMChangePIN_t::id

8.567.2.2 uint8_t pack_dms_UIMChangePIN_t::newValue[255]

8.567.2.3 uint8_t pack_dms_UIMChangePIN_t::oldValue[255]

8.568 pack_dms_UIMGetControlKeyStatus_t Struct Reference

Data Fields

- uint8_t facility

8.568.1 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)

8.568.2 Field Documentation

8.568.2.1 uint8_t pack_dms_UIMGetControlKeyStatus_t::facility

8.569 pack_dms_UIMGetICCID_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.569.1 Detailed Description

This structure contains UIM Get ICCID pack

Parameters

Tlvresult	• Pack result.

8.569.2 Field Documentation

8.569.2.1 uint16_t pack_dms_UIMGetICCID_t::Tlvresult

8.570 pack_dms_UIMSetControlKeyProtection_t Struct Reference

Data Fields

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

8.570.1 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)

8.570.2 Field Documentation

8.570.2.1 uint8_t pack_dms_UIMSetControlKeyProtection_t::facility

8.570.2.2 uint8_t pack_dms_UIMSetControlKeyProtection_t::facilityCk[8]

8.570.2.3 uint8_t pack_dms_UIMSetControlKeyProtection_t::facilityState

8.571 pack_dms_UIMSetPINProtection_t Struct Reference

Data Fields

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

8.571.1 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

8.571.2 Field Documentation

8.571.2.1 uint8_t pack_dms_UIMSetPINProtection_t::bEnable

8.571.2.2 uint8_t pack_dms_UIMSetPINProtection_t::id

8.571.2.3 uint8_t pack_dms_UIMSetPINProtection_t::value[255]

8.572 pack_dms_UIMUnblockControlKey_t Struct Reference

Data Fields

- uint8_t facility
- uint8_t facilityCk [8]

8.572.1 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)

8.572.2 Field Documentation

8.572.2.1 uint8_t pack_dms_UIMUnblockControlKey_t::facility

8.572.2.2 uint8_t pack_dms_UIMUnblockControlKey_t::facilityCk[8]

8.573 pack_dms_UIMUnblockPIN_t Struct Reference

Data Fields

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

8.573.1 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

8.573.2 Field Documentation

8.573.2.1 uint8_t pack_dms_UIMUnblockPIN_t::id

8.573.2.2 uint8_t pack_dms_UIMUnblockPIN_t::newPin[255]

8.573.2.3 uint8_t pack_dms_UIMUnblockPIN_t::pukValue[255]

8.574 pack_dms_UIMVerifyPIN_t Struct Reference

Data Fields

- uint8_t id
- uint8_t value [255]

8.574.1 Detailed Description

This structure contains PIN Value Information

Parameters

id[IN]	• PIN ID – 1 (PIN1 / CHV1) – 2 (PIN2 / CHV2)
value[IN]	• PIN value of the PIN to be enabled/disabled

8.574.2 Field Documentation

8.574.2.1 uint8_t pack_dms_UIMVerifyPIN_t::id

8.574.2.2 uint8_t pack_dms_UIMVerifyPIN_t::value[255]

8.575 pack_fms_GetImagesPreference_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.575.1 Detailed Description

This structure contains the Get Image Preference information pack

Parameters

Tlvresult	• Pack result

8.575.2 Field Documentation

8.575.2.1 uint16_t pack_fms_GetImagesPreference_t::Tlvresult

8.576 pack_fms_GetStoredImages_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.576.1 Detailed Description

This structure contains the Get Stored Images pack

Parameters

Tlvresult	• Pack result

8.576.2 Field Documentation

8.576.2.1 uint16_t pack_fms_GetStoredImages_t::Tlvresult

8.577 pack_fms_SetImagesPreference_t Struct Reference

Data Fields

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

8.577.1 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

8.577.2 Field Documentation

8.577.2.1 uint32_t pack_fms_SetImagesPreference_t::bForceDownload

8.577.2.2 uint32_t pack_fms_SetImagesPreference_t::imageListSize

8.577.2.3 uint8_t pack_fms_SetImagesPreference_t::modemindex

8.577.2.4 FMSPrefImageList* pack_fms_SetImagesPreference_t::pImageList

8.577.2.5 uint16_t pack_fms_SetImagesPreference_t::Tlvresult

8.578 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

8.578.1 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.

8.578.2 Field Documentation

8.578.2.1 loc_BdsSVInfo* pack_loc_Delete_Assist_Data_t::pBdsSVInfo

8.578.2.2 loc_CellDb* pack_loc_Delete_Assist_Data_t::pCellDb

8.578.2.3 loc_ClkInfo* pack_loc_Delete_Assist_Data_t::pClkInfo

8.578.2.4 loc_GnssData* pack_loc_Delete_Assist_Data_t::pGnssData

8.578.2.5 loc_SVInfo* pack_loc_Delete_Assist_Data_t::pSVInfo

8.578.2.6 uint16_t pack_loc_Delete_Assist_Data_t::Tlvresult

8.579 pack_loc_EventRegister_t Struct Reference

Data Fields

- uint64_t eventRegister
- uint16_t Tlvresult

8.579.1 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.

8.579.2 Field Documentation

8.579.2.1 uint64_t pack_loc_EventRegister_t::eventRegister

8.579.2.2 uint16_t pack_loc_EventRegister_t::TlvResult

8.580 pack_loc_SetExtPowerState_t Struct Reference

Data Fields

- uint32_t extPowerState
- uint16_t TlvResult

8.580.1 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.

8.580.2 Field Documentation

8.580.2.1 uint32_t pack_loc_SetExtPowerState_t::extPowerState

8.580.2.2 uint16_t pack_loc_SetExtPowerState_t::TlvResult

8.581 pack_loc_SetOperationMode_t Struct Reference

Data Fields

- uint32_t mode
- uint16_t TlvResult

8.581.1 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.

8.581.2 Field Documentation

8.581.2.1 uint32_t pack_loc_SetOperationMode_t::mode

8.581.2.2 uint16_t pack_loc_SetOperationMode_t::Tlvresult

8.582 pack_loc_SLQSLOCGetBestAvailPos_t Struct Reference

Data Fields

- uint32_t xid
- uint16_t Tlvresult

8.582.1 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.

8.582.2 Field Documentation

8.582.2.1 uint16_t pack_loc_SLQSLOCGetBestAvailPos_t::Tlvresult

8.582.2.2 uint32_t pack_loc_SLQSLOCGetBestAvailPos_t::xid

8.583 pack_loc_SLQSLOCGetServer_t Struct Reference

Data Fields

- uint32_t serverType
- int has_serverAddrTypeMask
- uint8_t serverAddrTypeMask

8.583.1 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

8.583.2 Field Documentation

8.583.2.1 int pack_loc_SLQSLOCGetServer_t::has_serverAddrTypeMask

8.583.2.2 uint8_t pack_loc_SLQSLOCGetServer_t::serverAddrTypeMask

8.583.2.3 uint32_t pack_loc_SLQSLOCGetServer_t::serverType

8.584 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

8.584.1 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.

8.584.2 Field Documentation

8.584.2.1 altSrcInfo_t pack_loc_SLQSLOCInjectPosition_t::altitudeSrcInfo

8.584.2.2 float pack_loc_SLQSLOCInjectPosition_t::altitudeWrtEllipsoid

8.584.2.3 float pack_loc_SLQSLOCInjectPosition_t::altitudeWrtMeanSeaLevel

8.584.2.4 int pack_loc_SLQSLOCInjectPosition_t::has_altitudeSrcInfo

8.584.2.5 int pack_loc_SLQSLOCInjectPosition_t::has_altitudeWrtEllipsoid

8.584.2.6 int pack_loc_SLQSLOCInjectPosition_t::has_altitudeWrtMeanSeaLevel

8.584.2.7 int pack_loc_SLQSLOCInjectPosition_t::has_horConfidence

8.584.2.8 int pack_loc_SLQSLOCInjectPosition_t::has_horReliability

8.584.2.9 int pack_loc_SLQSLOCInjectPosition_t::has_horUncCircular

8.584.2.10 int pack_loc_SLQSLOCInjectPosition_t::has_latitude

8.584.2.11 int pack_loc_SLQSLOCInjectPosition_t::has_longitude

8.584.2.12 int pack_loc_SLQSLOCInjectPosition_t::has_positionSrc

8.584.2.13 int pack_loc_SLQSLOCInjectPosition_t::has_rawHorConfidence

8.584.2.14 int pack_loc_SLQSLOCInjectPosition_t::has_rawHorUncCircular

8.584.2.15 int pack_loc_SLQSLOCInjectPosition_t::has_timestampAge

8.584.2.16 int pack_loc_SLQSLOCInjectPosition_t::has_timestampUtc

8.584.2.17 int pack_loc_SLQSLOCInjectPosition_t::has_vertConfidence

8.584.2.18 int pack_loc_SLQSLOCInjectPosition_t::has_vertRelicability

8.584.2.19 int pack_loc_SLQSLOCInjectPosition_t::has_vertUnc

8.584.2.20 uint8_t pack_loc_SLQSLOCInjectPosition_t::horConfidence

8.584.2.21 uint32_t pack_loc_SLQSLOCInjectPosition_t::horReliability

8.584.2.22 float pack_loc_SLQSLOCInjectPosition_t::horUncCircular

8.584.2.23 double pack_loc_SLQSLOCInjectPosition_t::latitude

8.584.2.24 double pack_loc_SLQSLOCInjectPosition_t::longitude

8.584.2.25 uint32_t pack_loc_SLQSLOCInjectPosition_t::positionSrc

8.584.2.26 uint8_t pack_loc_SLQSLOCInjectPosition_t::rawHorConfidence

8.584.2.27 float pack_loc_SLQSLOCInjectPosition_t::rawHorUncCircular

8.584.2.28 uint32_t pack_loc_SLQSLOCInjectPosition_t::timestampAge

8.584.2.29 uint64_t pack_loc_SLQSLOCInjectPosition_t::timestampUtc

8.584.2.30 uint8_t pack_loc_SLQSLOCInjectPosition_t::vertConfidence

8.584.2.31 uint32_t pack_loc_SLQSLOCInjectPosition_t::vertReliability

8.584.2.32 float pack_loc_SLQSLOCInjectPosition_t::vertUnc

8.585 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

8.585.1 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

8.585.2 Field Documentation

8.585.2.1 sensorData_t pack_loc_SLQSLOCInjectSensorData_t::acceleroData

8.585.2.2 uint32_t pack_loc_SLQSLOCInjectSensorData_t::acceleroTimeSrc

8.585.2.3 tempData_t pack_loc_SLQSLOCInjectSensorData_t::accelTemp

8.585.2.4 sensorData_t pack_loc_SLQSLOCInjectSensorData_t::gyroData

8.585.2.5 tempData_t pack_loc_SLQSLOCInjectSensorData_t::gyroTemp

8.585.2.6 uint32_t pack_loc_SLQSLOCInjectSensorData_t::gyroTimeSrc

8.585.2.7 int pack_loc_SLQSLOCInjectSensorData_t::has_acceleroTimeSrc

8.585.2.8 int pack_loc_SLQSLOCInjectSensorData_t::has_accelTemp

8.585.2.9 int pack_loc_SLQSLOCInjectSensorData_t::has_accleroData

8.585.2.10 int pack_loc_SLQSLOCInjectSensorData_t::has_gyroData

8.585.2.11 int pack_loc_SLQSLOCInjectSensorData_t::has_gyroTemp

8.585.2.12 int pack_loc_SLQSLOCInjectSensorData_t::has_gyroTimeSrc

8.585.2.13 int pack_loc_SLQSLOCInjectSensorData_t::has_opaqueId

8.585.2.14 uint32_t pack_loc_SLQSLOCInjectSensorData_t::opaqueId

8.586 pack_loc_SLQSLOCInjectUTCTime_t Struct Reference

Data Fields

- uint64_t timeMsec
- uint32_t timeUncMsec

8.586.1 Detailed Description

This structure contains inject UTC time parameter.

Parameters

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

8.586.2 Field Documentation

8.586.2.1 uint64_t pack_loc_SLQSLOCInjectUTCTime_t::timeMsec

8.586.2.2 uint32_t pack_loc_SLQSLOCInjectUTCTime_t::timeUncMsec

8.587 pack_loc_SLQSLOCSetCradleMountConfig_t Struct Reference

Data Fields

- uint32_t state
- int has_confidence
- uint8_t confidence

8.587.1 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

8.587.2 Field Documentation

8.587.2.1 uint8_t pack_loc_SLQSLOCSetCradleMountConfig_t::confidence

8.587.2.2 int pack_loc_SLQSLOCSetCradleMountConfig_t::has_confidence

8.587.2.3 uint32_t pack_loc_SLQSLOCSetCradleMountConfig_t::state

8.588 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

8.588.1 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.

8.588.2 Field Documentation

8.588.2.1 loc_LocApplicationInfo* pack_loc_Start_t::pApplicationInfo

8.588.2.2 uint32_t* pack_loc_Start_t::pConfigAltitudeAssumed

8.588.2.3 uint32_t* pack_loc_Start_t::pHorizontalAccuracyLvl

8.588.2.4 uint32_t* pack_loc_Start_t::pIntermediateReportState

8.588.2.5 uint32_t* pack_loc_Start_t::pMinIntervalTime

8.588.2.6 uint32_t* pack_loc_Start_t::pRecurrenceType

8.588.2.7 uint8_t pack_loc_Start_t::SessionId

8.588.2.8 uint16_t pack_loc_Start_t::Tlvresult

8.589 pack_loc_Stop_t Struct Reference

Data Fields

- uint8_t SessionId
- uint16_t Tlvresult

8.589.1 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.

8.589.2 Field Documentation

8.589.2.1 uint8_t pack_loc_Stop_t::SessionId

8.589.2.2 uint16_t pack_loc_Stop_t::Tlvresult

8.590 pack_nas_InitiateDomainAttach_t Struct Reference

Data Fields

- uint32_t action

8.590.1 Detailed Description

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

Parameters

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

8.590.2 Field Documentation

8.590.2.1 uint32_t pack_nas_InitiateDomainAttach_t::action

8.591 pack_nas_SetACCOLC_t Struct Reference

Data Fields

- int8_t spc [6]
- uint8_t accolc

8.591.1 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

8.591.2 Field Documentation

8.591.2.1 uint8_t pack_nas_SetACCOLC_t::accolc

8.591.2.2 int8_t pack_nas_SetACCOLC_t::spc[6]

8.592 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

8.592.1 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

8.592.2 Field Documentation

8.592.2.1 uint32_t* pack_nas_SetCDMANetworkParameters_t::pApplication

8.592.2.2 uint32_t* pack_nas_SetCDMANetworkParameters_t::pBroadcast

8.592.2.3 uint8_t* pack_nas_SetCDMANetworkParameters_t::pCustomSCP

8.592.2.4 uint8_t* pack_nas_SetCDMANetworkParameters_t::pForceRev0

8.592.2.5 uint32_t* pack_nas_SetCDMANetworkParameters_t::pProtocol

8.592.2.6 uint32_t* pack_nas_SetCDMANetworkParameters_t::pRoaming

8.592.2.7 char* pack_nas_SetCDMANetworkParameters_t::pSPC

8.593 pack_nas_SetNetworkPreference_t Struct Reference

Data Fields

- uint32_t TechnologyPref
- uint32_t Duration
- uint16_t Tlvresult

8.593.1 Detailed Description

This structure contains pack set network preference parameters.

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

8.593.2 Field Documentation

8.593.2.1 uint32_t pack_nas_SetNetworkPreference_t::Duration

8.593.2.2 uint32_t pack_nas_SetNetworkPreference_t::TechnologyPref

8.593.2.3 uint16_t pack_nas_SetNetworkPreference_t::Tlvresult

8.594 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

8.594.1 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

8.594.2 Field Documentation

8.594.2.1 nas_ECIOThresh∗ pack_nas_SLQSConfigSigInfo_t::pECIOThresh

8.594.2.2 nas_HDRSINRThresh∗ pack_nas_SLQSConfigSigInfo_t::pHDRSINRThresh

8.594.2.3 nas_IOThresh∗ pack_nas_SLQSConfigSigInfo_t::pIOThresh

8.594.2.4 nas_LTESigRptCfg∗ pack_nas_SLQSConfigSigInfo_t::pLTESigRptCfg

8.594.2.5 nas_LTESNRThresh∗ pack_nas_SLQSConfigSigInfo_t::pLTESNRThresh

8.594.2.6 nas_RSRPThresh∗ pack_nas_SLQSConfigSigInfo_t::pRSRPThresh

8.594.2.7 nas_RSRQThresh∗ pack_nas_SLQSConfigSigInfo_t::pRSRQThresh

8.594.2.8 nas_RSSIThresh∗ pack_nas_SLQSConfigSigInfo_t::pRSSIThresh

8.594.2.9 nas_TDSCDMASINRCONFThresh∗ pack_nas_SLQSConfigSigInfo_t::pTDSCDMASINRCONFThresh

8.595 pack_nas_SLQSGetPLMNName_t Struct Reference

Data Fields

- uint16_t mcc
- uint16_t mnc
- uint8_t ∗ pMncPcsStatus

8.595.1 Detailed Description

This structure contains pack get operator name for specificed network parameters.

Parameters

mcc	• A 16-bit integer representation of MCC. Range: 0 to 999
mnc	• A 16-bit integer representation of MNC. Range: 0 to 999

pMncPcsStatus	• MNC PCS Digit Include Status • 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.

8.595.2 Field Documentation

8.595.2.1 uint16_t pack_nas_SLQSGetPLMNName_t::mcc

8.595.2.2 uint16_t pack_nas_SLQSGetPLMNName_t::mnc

8.595.2.3 uint8_t* pack_nas_SLQSGetPLMNName_t::pMncPcsStatus

8.596 pack_nas_SLQSInitiateNetworkRegistration_t Struct Reference

Data Fields

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

8.596.1 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.

8.596.2 Field Documentation

8.596.2.1 uint32_t* pack_nas_SLQSInitiateNetworkRegistration_t::pChangeDuration

8.596.2.2 uint8_t* pack_nas_SLQSInitiateNetworkRegistration_t::pMncPcsDigitStatus

8.596.2.3 nas_MNRInfo* pack_nas_SLQSInitiateNetworkRegistration_t::pMNRInfo

8.596.2.4 uint32_t pack_nas_SLQSInitiateNetworkRegistration_t::regAction

8.597 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

8.597.1 Detailed Description

This structure contains pack set the signal strength reporting thresholds parameters.

Parameters

pCDMARSSI-Thresh	• CDMA RSSI threshold List
pCDMARSSI-Delta	• RSSI delta (in units of 0.1 dBm). • 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.
pTDSCDMARS-SIThresh	• TDSCDMA RSSI Threshold List
pTDSCDMARS-SIDelta	• RSSI delta (in dBm) used by TD-SCDMA.
pTDSCDMAECI-OThresh	• TDSCDMA ECIO Threshold List
pTDSCDMAECI-ODelta	• ECIO delta (in dB) used by TD-SCDMA
pTDSCDMASIN-RThresh	• TDSCDMA SINR Threshold List
pTDSCDMASIN-RDelta	• 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.

8.597.2 Field Documentation

8.597.2.1 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pCDMAECIODelta

8.597.2.2 nas_CDMAECIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pCDMAECIOThresh

8.597.2.3 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pCDMARSSIDelta

8.597.2.4 nas_CDMARSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pCDMARSSIThresh

8.597.2.5 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pGSMRSSIDelta

8.597.2.6 nas_GSMRSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pGSMRSSIThresh

8.597.2.7 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHDRECIODelta

8.597.2.8 **nas_HDRECIOThresh*** pack_nas_SLQSNasConfigSigInfo2_t::pHDRECIOThresh

8.597.2.9 **uint16_t*** pack_nas_SLQSNasConfigSigInfo2_t::pHDRIODelta

8.597.2.10 **nas_HDRIOThresh*** pack_nas_SLQSNasConfigSigInfo2_t::pHDRIOThresh

8.597.2.11 **uint16_t*** pack_nas_SLQSNasConfigSigInfo2_t::pHDRRSSIDelta

8.597.2.12 **nas_HDRRSSIThresh*** pack_nas_SLQSNasConfigSigInfo2_t::pHDRRSSIThresh

8.597.2.13 **uint16_t*** pack_nas_SLQSNasConfigSigInfo2_t::pHDRSINRDelta

8.597.2.14 **nas_HDRSINRThreshold*** pack_nas_SLQSNasConfigSigInfo2_t::pHDRSINRThresh

8.597.2.15 **uint16_t*** pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRPDelta

8.597.2.16 **nas_LTERSRPThresh*** pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRPThresh

8.597.2.17 **uint16_t*** pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRQDelta

8.597.2.18 **nas_LTERSRQThresh*** pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRQThresh

8.597.2.19 **uint16_t*** pack_nas_SLQSNasConfigSigInfo2_t::pLTERSSIDelta

8.597.2.20 **nas_LTERSSIThresh*** pack_nas_SLQSNasConfigSigInfo2_t::pLTERSSIThresh

8.597.2.21 **nas_LTESigRptConfig*** pack_nas_SLQSNasConfigSigInfo2_t::pLTESigRptConfig

8.597.2.22 **uint16_t*** pack_nas_SLQSNasConfigSigInfo2_t::pLTESNRDelta

8.597.2.23 **nas_LTESNRThreshold*** pack_nas_SLQSNasConfigSigInfo2_t::pLTESNRThresh

8.597.2.24 **float*** pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMAECIODelta

8.597.2.25 **nas_TDSCDMAECIOThresh*** pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMAECIOThresh

8.597.2.26 **uint16_t*** pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSCPDelta

8.597.2.27 **nas_TDSCDMARSCPThresh*** pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSCPThresh

8.597.2.28 **float*** pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSSIDelta

8.597.2.29 **nas_TDSCDMARSSIThresh*** pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSSIThresh

8.597.2.30 **float*** pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMASINRDelta

8.597.2.31 **nas_TDSCDMASINRThresh*** pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMASINRThresh

8.597.2.32 **uint16_t*** pack_nas_SLQSNasConfigSigInfo2_t::pWCDMAECIODelta

8.597.2.33 **nas_WCDMAECIOThresh*** pack_nas_SLQSNasConfigSigInfo2_t::pWCDMAECIOThresh

8.597.2.34 **uint16_t*** pack_nas_SLQSNasConfigSigInfo2_t::pWCDMARSSIDelta

8.597.2.35 **nas_WCDMARSSIThresh*** pack_nas_SLQSNasConfigSigInfo2_t::pWCDMARSSIThresh

8.598 pack_nas_SLQSNasGet3GPP2Subscription_t Struct Reference

Data Fields

- uint8_t namID

8.598.1 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.

8.598.2 Field Documentation

8.598.2.1 uint8_t pack_nas_SLQSNasGet3GPP2Subscription_t::namID

8.599 pack_nas_SLQSNasGetTxRxInfo_t Struct Reference

Data Fields

- uint8_t radio_if

8.599.1 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

8.599.2 Field Documentation

8.599.2.1 uint8_t pack_nas_SLQSNasGetTxRxInfo_t::radio_if

8.600 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

8.600.1 Detailed Description

This structure contains pack indication register parameters.

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
pSignalStrengthInd	[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
pHDRNewUATIAssInd	[Optional] • HDR New UATI Assigned indication registration.The callbacks would not be invoked if the indication is disabled. – 0x00 - Disable – 0x01 - Enable
pHDRSessionCloseInd	[Optional] • HDR Session Closed indication registration.The callbacks would not be invoked if the indication is disabled. – 0x00 - Disable – 0x01 - Enable
pManagedRoamingInd	[Optional] • Managed Roaming indication registration.The callbacks would not be invoked if the indication is disabled. – 0x00 - Disable
pNetworkRejectInd	[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- 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

8.600.2 Field Documentation

8.600.2.1 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pDDTMInd

8.600.2.2 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pDualStandByPrefInd

8.600.2.3 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pErrorRateInd

8.600.2.4 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pHDRNewUATIAssInd

8.600.2.5 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pHDRSessionCloseInd

8.600.2.6 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pLTECphyCa

8.600.2.7 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pManagedRoamingInd

8.600.2.8 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pNetworkRejectInd

8.600.2.9 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pNetworkTimeInd

8.600.2.10 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pServingSystemInd

8.600.2.11 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSignalStrengthInd

8.600.2.12 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSubscriptionInfoInd

8.600.2.13 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSuppressSysInfoInd

8.600.2.14 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSysInfoInd

8.600.2.15 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSystemSelectionInd

8.601 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

8.601.1 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

8.601.2 Field Documentation

8.601.2.1 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::gsmUmtsDl

8.601.2.2 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::gsmUmtsUl

8.601.2.3 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::lteEmmDl

8.601.2.4 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::lteEmmUl

8.601.2.5 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::lteEsmDl

8.601.2.6 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::lteEsmUl

8.601.2.7 uint8_t* pack_nas_SLQSNasSwiIndicationRegister_t::pRankIndicatorInd

8.601.2.8 uint8_t* pack_nas_SLQSNasSwiIndicationRegister_t::pTimer

8.602 pack_nas_SLQSNASSwiSetChannelLock_t Struct Reference

Data Fields

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

8.602.1 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

8.602.2 Field Documentation

8.602.2.1 nas_lteEARFCN* pack_nas_SLQSNASSwiSetChannelLock_t::pLteEARFCN

8.602.2.2 nas_ltePCI* pack_nas_SLQSNASSwiSetChannelLock_t::pLtePCI

8.602.2.3 nas_wcdmaUARFCN* pack_nas_SLQSNASSwiSetChannelLock_t::pWcdmaUARFCN

8.603 pack_nas_SLQSSetBandPreference_t Struct Reference

Data Fields

- uint64_t bandPref

8.603.1 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

8.603.2 Field Documentation

8.603.2.1 uint64_t pack_nas_SLQSSetBandPreference_t::bandPref

8.604 pack_nas_SLQSSetSignalStrengthsCallback_t Struct Reference

Data Fields

- uint8_t bEnable
- nas_SLQSSignalStrengthsIndReq * pSigIndReq

8.604.1 Detailed Description

This structure contains paramaters pack set strength thresholds callback.

Parameters

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

8.604.2 Field Documentation

8.604.2.1 uint8_t pack_nas_SLQSSetSignalStrengthsCallback_t::bEnable

8.604.2.2 nas_SLQSSignalStrengthsIndReq* pack_nas_SLQSSetSignalStrengthsCallback_t::pSigIndReq

8.605 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

8.605.1 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

8.605.2 Field Documentation

8.605.2.1 struct nas_acqOrderPref* pack_nas_SLQSSetSysSelectionPref_t::pAcqOrderPref

8.605.2.2 uint64_t* pack_nas_SLQSSetSysSelectionPref_t::pBandPref

8.605.2.3 uint8_t* pack_nas_SLQSSetSysSelectionPref_t::pChgDuration

8.605.2.4 struct nas_CSGID* pack_nas_SLQSSetSysSelectionPref_t::pCSGID

8.605.2.5 uint8_t* pack_nas_SLQSSetSysSelectionPref_t::pEmerMode

8.605.2.6 uint32_t* pack_nas_SLQSSetSysSelectionPref_t::pGWAcqOrderPref

8.605.2.7 uint64_t* pack_nas_SLQSSetSysSelectionPref_t::pLTEBandPref

8.605.2.8 uint8_t* pack_nas_SLQSSetSysSelectionPref_t::pMNCIncPCSDigStat

8.605.2.9 uint16_t* pack_nas_SLQSSetSysSelectionPref_t::pModePref

8.605.2.10 struct nas_netSelectionPref* pack_nas_SLQSSetSysSelectionPref_t::pNetSelPref

8.605.2.11 uint16_t* pack_nas_SLQSSetSysSelectionPref_t::pPRLPref

8.605.2.12 unsigned char* pack_nas_SLQSSetSysSelectionPref_t::pRAT

8.605.2.13 uint16_t* pack_nas_SLQSSetSysSelectionPref_t::pRoamPref

8.605.2.14 uint32_t* pack_nas_SLQSSetSysSelectionPref_t::pSrvDomainPref

8.605.2.15 uint32_t* pack_nas_SLQSSetSysSelectionPref_t::pSrvRegRestriction

8.605.2.16 uint64_t* pack_nas_SLQSSetSysSelectionPref_t::pTdscdmaBandPref

8.606 pack_nas_SLQSSwiPSDetach_t Struct Reference

Data Fields

- uint8_t * pDetachAction

8.606.1 Detailed Description

Structure for PS connection detach pack.

Parameters

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

8.606.2 Field Documentation

8.606.2.1 uint8_t* pack_nas_SLQSSwiPSDetach_t::pDetachAction

8.607 pack_pds_PDSInjectTimeReference_t Struct Reference

Data Fields

- uint64_t systemTime
- uint16_t systemDiscontinuities

8.607.1 Detailed Description

Structure contain the parameter for PDS Inject time.

Parameters

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

8.607.2 Field Documentation

8.607.2.1 uint16_t pack_pds_PDSInjectTimeReference_t::systemDiscontinuities

8.607.2.2 uint64_t pack_pds_PDSInjectTimeReference_t::systemTime

8.608 pack_pds_ResetPDSData_t Struct Reference

Data Fields

- uint32_t * pGPSDataMask
- uint32_t * pCellDataMask

8.608.1 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

8.608.2 Field Documentation

8.608.2.1 uint32_t* pack_pds_ResetPDSData_t::pCellDataMask

8.608.2.2 uint32_t* pack_pds_ResetPDSData_t::pGPSDataMask

8.609 pack_pds_SetEventReportCallback_t Struct Reference

Data Fields

- uint8_t posDataNmea
- uint8_t rptPosData

8.609.1 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

8.609.2 Field Documentation

8.609.2.1 uint8_t pack_pds_SetEventReportCallback_t::posDataNmea

8.609.2.2 uint8_t pack_pds_SetEventReportCallback_t::rptPosData

8.610 pack_pds_SetPDSDefaults_t Struct Reference

Data Fields

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

8.610.1 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)

8.610.2 Field Documentation

8.610.2.1 uint32_t pack_pds_SetPDSDefaults_t::accuracy

8.610.2.2 uint32_t pack_pds_SetPDSDefaults_t::interval

8.610.2.3 uint32_t pack_pds_SetPDSDefaults_t::operation

8.610.2.4 uint8_t pack_pds_SetPDSDefaults_t::timeout

8.611 pack_pds_SetPDSState_t Struct Reference

Data Fields

- uint32_t enable

8.611.1 Detailed Description

Structure contain PDS state parameters.

Parameters

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

8.611.2 Field Documentation

8.611.2.1 uint32_t pack_pds_SetPDSState_t::enable

8.612 pack_pds_SetPortAutomaticTracking_t Struct Reference

Data Fields

- uint32_t bAuto

8.612.1 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

8.612.2 Field Documentation

8.612.2.1 uint32_t pack_pds_SetPortAutomaticTracking_t::bAuto

8.613 pack_pds_SetServiceAutomaticTracking_t Struct Reference

Data Fields

- uint32_t bAuto

8.613.1 Detailed Description

Structure contain the parameter to Sets the automatic tracking state.

Parameters

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

8.613.2 Field Documentation

8.613.2.1 uint32_t pack_pds_SetServiceAutomaticTracking_t::bAuto

8.614 pack_pds_SetXTRAAutomaticDownload_t Struct Reference

Data Fields

- uint32_t bEnabled
- uint16_t interval

8.614.1 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

8.614.2 Field Documentation

8.614.2.1 uint32_t pack_pds_SetXTRAAutomaticDownload_t::bEnabled

8.614.2.2 uint16_t pack_pds_SetXTRAAutomaticDownload_t::interval

8.615 pack_pds_SetXTRANetwork_t Struct Reference

Data Fields

- uint32_t preference

8.615.1 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

8.615.2 Field Documentation

8.615.2.1 uint32_t pack_pds_SetXTRANetwork_t::preference

8.616 pack_pds_SLQSGetAGPSConfig_t Struct Reference

Data Fields

- uint8_t * pNetworkMode

8.616.1 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

8.616.2 Field Documentation

8.616.2.1 uint8_t* pack_pds_SLQSGetAGPSConfig_t::pNetworkMode

8.617 pack_pds_SLQSPDSInjectAbsoluteTimeReference_t Struct Reference

Data Fields

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

8.617.1 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

8.617.2 Field Documentation

8.617.2.1 uint8_t pack_pds_SLQSPDSInjectAbsoluteTimeReference_t::forceFlag

8.617.2.2 uint8_t pack_pds_SLQSPDSInjectAbsoluteTimeReference_t::timeBase

8.617.2.3 uint64_t pack_pds_SLQSPDSInjectAbsoluteTimeReference_t::timeMsec

8.617.2.4 uint32_t pack_pds_SLQSPDSInjectAbsoluteTimeReference_t::timeUncMsec

8.618 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

8.618.1 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

8.618.2 Field Documentation

8.618.2.1 uint32_t* pack_pds_SLQSPDSInjectPositionData_t::pAltitudeWrtEllipsoid

8.618.2.2 uint32_t* pack_pds_SLQSPDSInjectPositionData_t::pAltitudeWrtSealevel

8.618.2.3 uint8_t* pack_pds_SLQSPDSInjectPositionData_t::pHorizontalConfidence

8.618.2.4 uint32_t* pack_pds_SLQSPDSInjectPositionData_t::pHorizontalUncCircular

8.618.2.5 uint64_t* pack_pds_SLQSPDSInjectPositionData_t::pLatitude

8.618.2.6 uint64_t* pack_pds_SLQSPDSInjectPositionData_t::pLongitude

8.618.2.7 uint8_t* pack_pds_SLQSPDSInjectPositionData_t::pPositionSource

8.618.2.8 uint64_t* pack_pds_SLQSPDSInjectPositionData_t::pTimeStamp

8.618.2.9 uint8_t* pack_pds_SLQSPDSInjectPositionData_t::pTimeType

8.618.2.10 uint8_t* pack_pds_SLQSPDSInjectPositionData_t::pVerticalConfidence

8.618.2.11 uint32_t* pack_pds_SLQSPDSInjectPositionData_t::pVerticalUnc

8.619 pack_pds_SLQSSetAGPSConfig_t Struct Reference

Data Fields

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

8.619.1 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

8.619.2 Field Documentation

8.619.2.1 uint8_t* pack_pds_SLQSSetAGPSConfig_t::pNetworkMode

8.619.2.2 uint32_t* pack_pds_SLQSSetAGPSConfig_t::pServerAddress

8.619.2.3 uint32_t* pack_pds_SLQSSetAGPSConfig_t::pServerPort

8.619.2.4 uint8_t* pack_pds_SLQSSetAGPSConfig_t::pServerURL

8.619.2.5 uint8_t* pack_pds_SLQSSetAGPSConfig_t::pServerURLLength

8.620 pack_pds_SLQSSetPositionMethodState_t Struct Reference

Data Fields

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

8.620.1 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

8.620.2 Field Documentation

8.620.2.1 uint8_t* pack_pds_SLQSSetPositionMethodState_t::pWifiState

8.620.2.2 uint8_t* pack_pds_SLQSSetPositionMethodState_t::pXtraDataState

8.620.2.3 uint8_t* pack_pds_SLQSSetPositionMethodState_t::pXtraTimeState

8.621 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

8.621.1 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)

8.621.2 Field Documentation

8.621.2.1 uint32_t pack_pds_StartPDSTrackingSessionExt_t::fixAccuracy

8.621.2.2 uint32_t pack_pds_StartPDSTrackingSessionExt_t::fixCount

8.621.2.3 uint32_t pack_pds_StartPDSTrackingSessionExt_t::fixInterval

8.621.2.4 uint8_t pack_pds_StartPDSTrackingSessionExt_t::fixTimeout

8.621.2.5 uint8_t pack_pds_StartPDSTrackingSessionExt_t::sessionControl

8.621.2.6 uint8_t pack_pds_StartPDSTrackingSessionExt_t::sessionOperation

8.621.2.7 uint8_t pack_pds_StartPDSTrackingSessionExt_t::sessionServerOption

8.621.2.8 uint8_t pack_pds_StartPDSTrackingSessionExt_t::sessionType

8.622 pack_qmi_t Struct Reference

Data Fields

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

8.622.1 Detailed Description

qmi request context

Parameters

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

8.622.2 Field Documentation

8.622.2.1 uint16_t pack_qmi_t::msgid

8.622.2.2 uint8_t pack_qmi_t::svc

8.622.2.3 int pack_qmi_t::timeout

8.622.2.4 uint16_t pack_qmi_t::xid

8.623 pack_qos_SLQSQosSwiReadApnExtraParams_t Struct Reference

Data Fields

- uint32_t apnId

8.623.1 Detailed Description

Structure that contains the APN ID to obtain extra APN parameters

Parameters

apnId[IN]	• APN id

8.623.2 Field Documentation

8.623.2.1 uint32_t pack_qos_SLQSQosSwiReadApnExtraParams_t::apnId

8.624 pack_qos_SLQSQosSwiReadDataStats_t Struct Reference

Data Fields

- uint32_t apnId

8.624.1 Detailed Description

Structure that contains the APN ID to obtain data statistics

Parameters

apnId[IN]	• APN id

8.624.2 Field Documentation

8.624.2.1 uint32_t pack_qos_SLQSQosSwiReadDataStats_t::apnId

8.625 pack_qos_SLQSSetQosEventCallback_t Struct Reference

Data Fields

- uint8_t enable

8.625.1 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

8.625.2 Field Documentation

8.625.2.1 uint8_t pack_qos_SLQSSetQosEventCallback_t::enable

8.626 pack_rms_SetSMSWake_t Struct Reference

Data Fields

- uint32_t enabled
- uint32_t wake_mask

8.626.1 Detailed Description

Pack set SMS wake parameters

Parameters

enabled	• SMS wake functionality enabled – 0 - Disabled – 1 - Enabled
wake_mask	• SMS wake mask to search for incoming messages (only relevant when enabled)

8.626.2 Field Documentation

8.626.2.1 uint32_t pack_rms_SetSMSWake_t::enabled

8.626.2.2 uint32_t pack_rms_SetSMSWake_t::wake_mask

8.627 pack_sar_SLQSSetRfSarState_t Struct Reference

Data Fields

- uint32_t RfSarState

8.627.1 Detailed Description

This sturcture contain sets the specified RF SAR state parameter.

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

8.627.2 Field Documentation

8.627.2.1 uint32_t pack_sar_SLQSSetRfSarState_t::RfSarState

8.628 pack_sms_SaveSMS_t Struct Reference

Data Fields

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

8.628.1 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

8.628.2 Field Documentation

8.628.2.1 uint32_t pack_sms_SaveSMS_t::messageFormat

8.628.2.2 uint32_t pack_sms_SaveSMS_t::messageSize

8.628.2.3 uint8_t* pack_sms_SaveSMS_t::pMessage

8.628.2.4 uint32_t pack_sms_SaveSMS_t::storageType

8.629 pack_sms_SendSMS_t Struct Reference

Data Fields

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

8.629.1 Detailed Description

This structure contains pack send SMS parameters.

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

8.629.2 Field Documentation

8.629.2.1 uint32_t pack_sms_SendSMS_t::messageFormat

8.629.2.2 uint32_t pack_sms_SendSMS_t::messageSize

8.629.2.3 uint8_t* pack_sms_SendSMS_t::pLinktimer

8.629.2.4 uint8_t* pack_sms_SendSMS_t::pMessage

8.630 pack_sms_SetNewSMSCallback_t Struct Reference

Data Fields

- enum eqmiCbkSetStatus status

8.630.1 Detailed Description

This strucure contains pack set new SMS callback parameters.

Parameters

status	callback parameter

8.630.2 Field Documentation

8.630.2.1 enum eqmiCbkSetStatus pack_sms_SetNewSMSCallback_t::status

8.631 pack_sms_SetSMSCAddress_t Struct Reference

Data Fields

- uint8_t * pSMSCAddress
- uint8_t * pSMSCType

8.631.1 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).

8.631.2 Field Documentation

8.631.2.1 uint8_t* pack_sms_SetSMSCAddress_t::pSMSCAddress

8.631.2.2 uint8_t* pack_sms_SetSMSCAddress_t::pSMSCType

8.632 pack_sms_SLQSDeleteSMS_t Struct Reference

Data Fields

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

8.632.1 Detailed Description

This structure contains pack delete SMS parameters.

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)

8.632.2 Field Documentation

8.632.2.1 uint32_t* pack_sms_SLQSDeleteSMS_t::pMessageIndex

8.632.2.2 uint8_t* pack_sms_SLQSDeleteSMS_t::pMessageMode

8.632.2.3 uint32_t* pack_sms_SLQSDeleteSMS_t::pMessageTag

8.632.2.4 uint32_t pack_sms_SLQSDeleteSMS_t::storageType

8.633 pack_sms_SLQSGetSMS_t Struct Reference

Data Fields

- uint32_t storageType
- uint32_t messageIndex
- uint8_t * pMessageMode

8.633.1 Detailed Description

This sturcture contains pack get SMS parameters.

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)

8.633.2 Field Documentation

8.633.2.1 uint32_t pack_sms_SLQSGetSMS_t::messageIndex

8.633.2.2 uint8_t* pack_sms_SLQSGetSMS_t::pMessageMode

8.633.2.3 uint32_t pack_sms_SLQSGetSMS_t::storageType

8.634 pack_sms_SLQSGetSmsBroadcastConfig_t Struct Reference

Data Fields

- uint8_t mode

8.634.1 Detailed Description

This structure contains get SMS boardcast configure parameter.

Parameters

mode	• Mode – 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_SLQSGetSmsBroadcastConfig_t::mode

8.635 pack_sms_SLQSGetSMSList_t Struct Reference

Data Fields

- uint32_t storageType
- uint32_t * pRequestedTag
- uint8_t * pMessageMode

8.635.1 Detailed Description

This structure contains pack get SMS list parameters.

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)

8.635.2 Field Documentation

8.635.2.1 uint8_t* pack_sms_SLQSGetSMSList_t::pMessageMode

8.635.2.2 uint32_t* pack_sms_SLQSGetSMSList_t::pRequestedTag

8.635.2.3 uint32_t pack_sms_SLQSGetSMSList_t::storageType

8.636 pack_sms_SLQSModifySMSStatus_t Struct Reference

Data Fields

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

8.636.1 Detailed Description

This structure contains pack modify SMS status parameters.

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)

8.636.2 Field Documentation

8.636.2.1 uint32_t pack_sms_SLQSModifySMSStatus_t::messageIndex

8.636.2.2 uint32_t pack_sms_SLQSModifySMSStatus_t::messageTag

8.636.2.3 uint8_t* pack_sms_SLQSModifySMSStatus_t::pMessageMode

8.636.2.4 uint32_t pack_sms_SLQSModifySMSStatus_t::storageType

8.637 pack_sms_SLQSSendAsyncSMS_t Struct Reference

Data Fields

- sms_sendAsyncsmsParams * pSendSmsParams

8.637.1 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

8.637.2 Field Documentation

8.637.2.1 sms_sendAsyncsmsParams∗ pack_sms_SLQSSendAsyncSMS_t::pSendSmsParams

8.638 pack_sms_SLQSSetIndicationRegister_t Struct Reference

Data Fields

- sms_setIndicationReg ∗ pSetIndicationRegReq

8.638.1 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

8.638.2 Field Documentation

8.638.2.1 sms_setIndicationReg∗ pack_sms_SLQSSetIndicationRegister_t::pSetIndicationRegReq

8.639 pack_sms_SLQSSetSmsBroadcastActivation_t Struct Reference

Data Fields

- uint8_t mode
- uint8_t broadcastActivate

8.639.1 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

8.639.2 Field Documentation

8.639.2.1 uint8_t pack_sms_SLQSSetSmsBroadcastActivation_t::broadcastActivate

8.639.2.2 uint8_t pack_sms_SLQSSetSmsBroadcastActivation_t::mode

8.640 pack_sms_SLQSSetSmsBroadcastConfig_t Struct Reference

Data Fields

- uint8_t mode
- sms_qaQmi3GPPBroadcastCfgInfo * pBroadcastConfig
- sms_qaQmi3GPP2BroadcastCfgInfo * pCDMABroadcastConfig

8.640.1 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).

8.640.2 Field Documentation

8.640.2.1 uint8_t pack_sms_SLQSSetSmsBroadcastConfig_t::mode

8.640.2.2 sms_qaQmi3GPPBroadcastCfgInfo* pack_sms_SLQSSetSmsBroadcastConfig_t::pBroadcastConfig

8.640.2.3 sms_qaQmi3GPP2BroadcastCfgInfo* pack_sms_SLQSSetSmsBroadcastConfig_t::pCDMABroadcastConfig

8.641 pack_sms_SLQSSetSmsStorage_t Struct Reference

Data Fields

- uint8_t smsStorage

8.641.1 Detailed Description

Structure contain Parameter to set the SMS Storage on the device

Parameters

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

8.641.2 Field Documentation

8.641.2.1 uint8_t pack_sms_SLQSSetSmsStorage_t::smsStorage

8.642 pack_sms_SLQSSmsGetMaxStorageSize_t Struct Reference

Data Fields

- sms_maxStorageSizeReq * pMaxStorageSizeReq

8.642.1 Detailed Description

This structure contains get maximum storage size.

Parameters

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

8.642.2 Field Documentation

8.642.2.1 sms_maxStorageSizeReq* pack_sms_SLQSSmsGetMaxStorageSize_t::pMaxStorageSizeReq

8.643 pack_sms_SLQSSmsSetRoutes_t Struct Reference

Data Fields

- sms_setRoutesReq * pSetRoutesReq

8.643.1 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

8.643.2 Field Documentation

8.643.2.1 sms_setRoutesReq* pack_sms_SLQSSmsSetRoutes_t::pSetRoutesReq

8.644 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

8.644.1 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

8.644.2 Field Documentation

8.644.2.1 uint32_t pack_swiloc_SwiLocSetAutoStart_t::fix_rate

8.644.2.2 uint8_t pack_swiloc_SwiLocSetAutoStart_t::fix_type

8.644.2.3 uint8_t pack_swiloc_SwiLocSetAutoStart_t::function

8.644.2.4 uint32_t pack_swiloc_SwiLocSetAutoStart_t::max_dist

8.644.2.5 uint8_t pack_swiloc_SwiLocSetAutoStart_t::max_time

8.644.2.6 int pack_swiloc_SwiLocSetAutoStart_t::set_fix_rate

8.644.2.7 int pack_swiloc_SwiLocSetAutoStart_t::set_fix_type

8.644.2.8 int pack_swiloc_SwiLocSetAutoStart_t::set_function

8.644.2.9 int pack_swiloc_SwiLocSetAutoStart_t::set_max_dist

8.644.2.10 int pack_swiloc_SwiLocSetAutoStart_t::set_max_time

8.645 pack_swioma_SLQSOMADMCancelSession_t Struct Reference

Data Fields

- uint32_t sessionType

8.645.1 Detailed Description

Structure that contains the session type for OMA cancel session command

Parameters

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

8.645.2 Field Documentation

8.645.2.1 uint32_t pack_swioma_SLQSOMADMCancelSession_t::sessionType

8.646 pack_swioma_SLQSOMADMGetSessionInfo_t Struct Reference

Data Fields

- uint32_t SessionType

8.646.1 Detailed Description

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

Parameters

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

8.646.2 Field Documentation

8.646.2.1 uint32_t pack_swioma_SLQSOMADMGetSessionInfo_t::SessionType

8.647 pack_swioma_SLQSOMADMSendSelection_t Struct Reference

Data Fields

- uint32_t selection
- uint32_t * pDeferTime
- uint32_t * pRejectReason

8.647.1 Detailed Description

Structure containing the OMA DM selection

Parameters

selection	[in] • OMA-DM NIA Selection – 0x01 - Accept – 0x02 - Reject – 0x03 - Defer
pDeferTime	[in] • Defer time in minutes. A value of 0 will cause the prompt to be resent immediately. • This TLV is mandatory if selection is set to 0x03.
pRejectReason	[in] • Reject Reason • This TLV is processed if selection is set to 0x02. If it is not present, the reject reason 0 is used as default.

8.647.2 Field Documentation

8.647.2.1 uint32_t* pack_swioma_SLQSOMADMSendSelection_t::pDeferTime

8.647.2.2 uint32_t* pack_swioma_SLQSOMADMSendSelection_t::pRejectReason

8.647.2.3 uint32_t pack_swioma_SLQSOMADMSendSelection_t::selection

8.648 pack_swioma_SLQSOMADMSetSettings_t Struct Reference

Data Fields

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

8.648.1 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

8.648.2 Field Documentation

8.648.2.1 uint8_t pack_swioma_SLQSOMADMSetSettings_t::FOTAdownload

8.648.2.2 uint8_t pack_swioma_SLQSOMADMSetSettings_t::FOTAUpdate

8.648.2.3 uint8_t* pack_swioma_SLQSOMADMSetSettings_t::pAutosdm

8.648.2.4 uint8_t* pack_swioma_SLQSOMADMSetSettings_t::pFwAutoCheck

8.649 pack_swioma_SLQSOMADMStartSession_t Struct Reference

Data Fields

- uint32_t sessionType

8.649.1 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

8.649.2 Field Documentation

8.649.2.1 uint32_t pack_swioma_SLQSOMADMStartSession_t::sessionType

8.650 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

8.650.1 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.

8.650.2 Field Documentation

8.650.2.1 uim_changeUIMPIN pack_uim_ChangePin_t::changePIN

8.650.2.2 uim_encryptedPIN1 pack_uim_ChangePin_t::EncryptedPIN1

8.650.2.3 uint32_t* pack_uim_ChangePin_t::pIndicationToken

8.650.2.4 uint8_t* pack_uim_ChangePin_t::pKeyReferenceID

8.650.2.5 uim_sessionInformation pack_uim_ChangePin_t::sessionInfo

8.650.2.6 uint16_t pack_uim_ChangePin_t::Tlvresult

8.651 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

8.651.1 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.

8.651.2 Field Documentation

8.651.2.1 uim_fileInfo pack_uim_ReadTransparent_t::fileIndex

8.651.2.2 uint8_t* pack_uim_ReadTransparent_t::pEncryptData

8.651.2.3 uint32_t* pack_uim_ReadTransparent_t::pIndicationToken

8.651.2.4 uim_readTransparentInfo pack_uim_ReadTransparent_t::readTransparent

8.651.2.5 uim_sessionInformation pack_uim_ReadTransparent_t::sessionInfo

8.651.2.6 uint16_t pack_uim_ReadTransparent_t::TlvResult

8.652 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

8.652.1 Detailed Description

This structure contains information of the request parameters associated with a set pin protection API.

Parameters

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

8.652.2 Field Documentation

8.652.2.1 uim_encryptedPIN1 pack_uim_SetPinProtection_t::EncryptedPIN1

8.652.2.2 uint32_t* pack_uim_SetPinProtection_t::pIndicationToken

8.652.2.3 uim_setPINProtection pack_uim_SetPinProtection_t::pinProtection

8.652.2.4 uint8_t* pack_uim_SetPinProtection_t::pKeyReferenceID

8.652.2.5 uim_sessionInformation pack_uim_SetPinProtection_t::sessionInfo

8.652.2.6 uint16_t pack_uim_SetPinProtection_t::TlvResult

8.653 pack_uim_SLQSUIMAuthenticate_t Struct Reference

Data Fields

- uim_sessionInformation sessionInfo
- uim_authenticationData authData
- uint32_t * pIndicationToken

8.653.1 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.

8.653.2 Field Documentation

8.653.2.1 uim_authenticationData pack_uim_SLQSUIMAuthenticate_t::authData

8.653.2.2 uint32_t* pack_uim_SLQSUIMAuthenticate_t::pIndicationToken

8.653.2.3 uim_sessionInformation pack_uim_SLQSUIMAuthenticate_t::sessionInfo

8.654 pack_uim_SLQSUIMDepersonalization_t Struct Reference

Data Fields

- uim_depersonalizationInformation depersonilisationInfo

8.654.1 Detailed Description

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

Parameters

depersonilisation-Info	• See uim_depersonalizationInformation for more information.

8.654.2 Field Documentation

8.654.2.1 uim_depersonalizationInformation pack_uim_SLQSUIMDepersonalization_t::depersonilisationInfo

8.655 pack_uim_SLQSUIMEventRegister_t Struct Reference

Data Fields

- uint32_t eventMask

8.655.1 Detailed Description

This structure contains pack event register parameter.

Parameters

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

8.655.2 Field Documentation

8.655.2.1 uint32_t pack_uim_SLQSUIMEventRegister_t::eventMask

8.656 pack_uim_SLQSUIMGetConfiguration_t Struct Reference

Data Fields

- uint32_t * pConfigurationMask

8.656.1 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.

8.656.2 Field Documentation

8.656.2.1 uint32_t* pack_uim_SLQSUIMGetConfiguration_t::pConfigurationMask

8.657 pack_uim_SLQSUIMGetFileAttributes_t Struct Reference

Data Fields

- uim_sessionInformation sessionInfo
- uim_fileInfo fileIndex
- uint32_t * pIndicationToken

8.657.1 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.

8.657.2 Field Documentation

8.657.2.1 uim_fileInfo pack_uim_SLQSUIMGetFileAttributes_t::fileIndex

8.657.2.2 uint32_t* pack_uim_SLQSUIMGetFileAttributes_t::pIndicationToken

8.657.2.3 uim_sessionInformation pack_uim_SLQSUIMGetFileAttributes_t::sessionInfo

8.658 pack_uim_SLQSUIMPowerDown_t Struct Reference

Data Fields

- uint8_t slot

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

8.658.2 Field Documentation

8.658.2.1 uint8_t pack_uim_SLQSUIMPowerDown_t::slot

8.659 pack_uim_SLQSUIMPowerUp_t Struct Reference

Data Fields

- uint8_t slot
- uint8_t * pIgnoreHotSwapSwitch

8.659.1 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

8.659.2 Field Documentation

8.659.2.1 uint8_t* pack_uim_SLQSUIMPowerUp_t::pIgnoreHotSwapSwitch

8.659.2.2 uint8_t pack_uim_SLQSUIMPowerUp_t::slot

8.660 pack_uim_SLQSUIMRefreshComplete_t Struct Reference

Data Fields

- uim_sessionInformation sessionInfo
- uint8_t refreshComplete

8.660.1 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

8.660.2 Field Documentation

8.660.2.1 uint8_t pack_uim_SLQSUIMRefreshComplete_t::refreshComplete

8.660.2.2 uim_sessionInformation pack_uim_SLQSUIMRefreshComplete_t::sessionInfo

8.661 pack_uim_SLQSUIMRefreshGetLastEvent_t Struct Reference

Data Fields

- uim_sessionInformation sessionInfo

8.661.1 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.

8.661.2 Field Documentation

8.661.2.1 uim_sessionInformation pack_uim_SLQSUIMRefreshGetLastEvent_t::sessionInfo

8.662 pack_uim_SLQSUIMRefreshOK_t Struct Reference

Data Fields

- uim_sessionInformation sessionInfo
- uint8_t OKtoRefresh

8.662.1 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

8.662.2 Field Documentation

8.662.2.1 uint8_t pack_uim_SLQSUIMRefreshOK_t::OKtoRefresh

8.662.2.2 uim_sessionInformation pack_uim_SLQSUIMRefreshOK_t::sessionInfo

8.663 pack_uim_SLQSUIMRefreshRegister_t Struct Reference

Data Fields

- uim_sessionInformation sessionInfo
- uim_registerRefresh regRefresh

8.663.1 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

8.663.2 Field Documentation

8.663.2.1 uim_registerRefresh pack_uim_SLQSUIMRefreshRegister_t::regRefresh

8.663.2.2 uim_sessionInformation pack_uim_SLQSUIMRefreshRegister_t::sessionInfo

8.664 pack_uim_SLQSUIMSwitchSlot_t Struct Reference

Data Fields

- uint8_t bLogicalSlot
- uint32_t ulPhysicalSlot

8.664.1 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

8.664.2 Field Documentation

8.664.2.1 uint8_t pack_uim_SLQSUIMSwitchSlot_t::bLogicalSlot

8.664.2.2 uint32_t pack_uim_SLQSUIMSwitchSlot_t::ulPhysicalSlot

8.665 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

8.665.1 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

8.665.2 Field Documentation

8.665.2.1 uim_encryptedPIN1 pack_uim_UnblockPin_t::EncryptedPIN1

8.665.2.2 uint32_t* pack_uim_UnblockPin_t::pIndicationToken

8.665.2.3 uim_unblockUIMPIN pack_uim_UnblockPin_t::pinProtection

8.665.2.4 uint8_t* pack_uim_UnblockPin_t::pKeyReferenceID

8.665.2.5 uim_sessionInformation pack_uim_UnblockPin_t::sessionInfo

8.665.2.6 uint16_t pack_uim_UnblockPin_t::Tlvresult

8.666 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

8.666.1 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.

8.666.2 Field Documentation

8.666.2.1 uim_encryptedPIN1∗ pack_uim_VerifyPin_t::pEncryptedPIN1

8.666.2.2 uint32_t∗ pack_uim_VerifyPin_t::pIndicationToken

8.666.2.3 uint8_t∗ pack_uim_VerifyPin_t::pKeyReferenceID

8.666.2.4 uim_sessionInformation pack_uim_VerifyPin_t::sessionInfo

8.666.2.5 uint16_t pack_uim_VerifyPin_t::Tlvresult

8.666.2.6 uim_verifyUIMPIN pack_uim_VerifyPin_t::verifyPIN

8.667 pack_voice_AnswerUSSD_t Struct Reference

Data Fields

- uint8_t ∗ pInfo

8.667.1 Detailed Description

This structure contains pack anser USSD parameter.

Parameters

pInfo	• USS information • See voice_USSInfo for more details

8.667.2 Field Documentation

8.667.2.1 uint8_t* pack_voice_AnswerUSSD_t::pInfo

8.668 pack_voice_OriginateUSSD_t Struct Reference

Data Fields

- uint8_t * pInfo

8.668.1 Detailed Description

This structure contains pack orginate USSD parameter.

Parameters

pInfo	• USS information • See voice_USSInfo for more details

8.668.2 Field Documentation

8.668.2.1 uint8_t* pack_voice_OriginateUSSD_t::pInfo

8.669 pack_voice_SLQSOriginateUSSD_t Struct Reference

Data Fields

- uint8_t ussDCS
- uint8_t ussLen
- uint8_t ussData [182]

8.669.1 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

8.669.2 Field Documentation

8.669.2.1 uint8_t pack_voice_SLQSOriginateUSSD_t::ussData[182]

8.669.2.2 uint8_t pack_voice_SLQSOriginateUSSD_t::ussDCS

8.669.2.3 uint8_t pack_voice_SLQSOriginateUSSD_t::ussLen

8.670 pack_voice_SLQSVoiceALSSelectLine_t Struct Reference

Data Fields

- uint8_t lineValue

8.670.1 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

8.670.2 Field Documentation

8.670.2.1 uint8_t pack_voice_SLQSVoiceALSSelectLine_t::lineValue

8.671 pack_voice_SLQSVoiceALSSetLineSwitching_t Struct Reference

Data Fields

- uint8_t switchOption

8.671.1 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

8.671.2 Field Documentation

8.671.2.1 uint8_t pack_voice_SLQSVoiceALSSetLineSwitching_t::switchOption

8.672 pack_voice_SLQSVoiceAnswerCall_t Struct Reference

Data Fields

- uint8_t * pCallId

8.672.1 Detailed Description

Contains the parameters passed for pack voice Answer Call.

Parameters

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

8.672.2 Field Documentation

8.672.2.1 uint8_t* pack_voice_SLQSVoiceAnswerCall_t::pCallId

8.673 pack_voice_SLQSVoiceBindSubscription_t Struct Reference

Data Fields

- uint8_t subsType

8.673.1 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

8.673.2 Field Documentation

8.673.2.1 uint8_t pack_voice_SLQSVoiceBindSubscription_t::subsType

8.674 pack_voice_SLQSVoiceBurstDTMF_t Struct Reference

Data Fields

- voice_burstDTMFInfo BurstDTMFInfo
- voice_DTMFLengths * pBurstDTMFLengths

8.674.1 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 • NULL pointer - Invalid data.

8.674.2 Field Documentation

8.674.2.1 voice_burstDTMFInfo pack_voice_SLQSVoiceBurstDTMF_t::BurstDTMFInfo

8.674.2.2 voice_DTMFLengths∗ pack_voice_SLQSVoiceBurstDTMF_t::pBurstDTMFLengths

8.675 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

8.675.1 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 voice_UUSInfo for more information
pCUG-Info(optional)	• Pointer to structure of CUGInfo – See voice_CUGInfo for more information
pEmergency-Category(optional)	• Bit mask of emergency number categories. This is only applicable when the call type is set to Emergency. – 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 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

8.675.2 Field Documentation

8.675.2.1 uint8_t pack_voice_SLQSVoiceDialCall_t::callNumber[81]

8.675.2.2 voice_calledPartySubAdd* pack_voice_SLQSVoiceDialCall_t::pCallPartySubAdd

8.675.2.3 uint8_t* pack_voice_SLQSVoiceDialCall_t::pCallType

8.675.2.4 uint8_t* pack_voice_SLQSVoiceDialCall_t::pCLIRType

8.675.2.5 voice_CUGInfo* pack_voice_SLQSVoiceDialCall_t::pCUGInfo

8.675.2.6 uint8_t* pack_voice_SLQSVoiceDialCall_t::pEmergencyCategory

8.675.2.7 uint8_t* pack_voice_SLQSVoiceDialCall_t::pSvcType

8.675.2.8 voice_UUSInfo* pack_voice_SLQSVoiceDialCall_t::pUUSInFo

8.676 pack_voice_SLQSVoiceEndCall_t Struct Reference

Data Fields

- uint8_t * pCallId

8.676.1 Detailed Description

This structure contain pack voice end call parameter.

Parameters

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

8.676.2 Field Documentation

8.676.2.1 uint8_t* pack_voice_SLQSVoiceEndCall_t::pCallId

8.677 pack_voice_SLQSVoiceGetCallBarring_t Struct Reference

Data Fields

- uint8_t reason
- uint8_t * pSvcClass

8.677.1 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

8.677.2 Field Documentation

8.677.2.1 uint8_t* pack_voice_SLQSVoiceGetCallBarring_t::pSvcClass

8.677.2.2 uint8_t pack_voice_SLQSVoiceGetCallBarring_t::reason

8.678 pack_voice_SLQSVoiceGetCallForwardingStatus_t Struct Reference

Data Fields

- uint8_t Reason
- uint8_t * pSvcClass

8.678.1 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.

8.678.2 Field Documentation

8.678.2.1 uint8_t* pack_voice_SLQSVoiceGetCallForwardingStatus_t::pSvcClass

8.678.2.2 uint8_t pack_voice_SLQSVoiceGetCallForwardingStatus_t::Reason

8.679 pack_voice_SLQSVoiceGetCallInfo_t Struct Reference

Data Fields

- uint8_t callID

8.679.1 Detailed Description

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

Parameters

callID	• Call identifier for the call queried for information.

8.679.2 Field Documentation

8.679.2.1 uint8_t pack_voice_SLQSVoiceGetCallInfo_t::callID

8.680 pack_voice_SLQSVoiceGetCallWaiting_t Struct Reference

Data Fields

- uint8_t * pSvcClass

8.680.1 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

8.680.2 Field Documentation

8.680.2.1 uint8_t* pack_voice_SLQSVoiceGetCallWaiting_t::pSvcClass

8.681 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

8.681.1 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.

8.681.2 Field Documentation

8.681.2.1 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pAirTimer

8.681.2.2 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pAMRStatus

8.681.2.3 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pAutoAnswer

8.681.2.4 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pNamID

8.681.2.5 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pPrefVoicePrivacy

8.681.2.6 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pPrefVoiceSO

8.681.2.7 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pRoamTimer

8.681.2.8 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pTTYMode

8.681.2.9 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pVoiceDomainPref

8.682 pack_voice_SLQSVoiceIndicationRegister_t Struct Reference

Data Fields

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

8.682.1 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.

8.682.2 Field Documentation

8.682.2.1 uint8_t* pack_voice_SLQSVoiceIndicationRegister_t::pRegDTMFEvents

8.682.2.2 uint8_t* pack_voice_SLQSVoiceIndicationRegister_t::pRegVoicePrivacyEvents

8.682.2.3 uint8_t* pack_voice_SLQSVoiceIndicationRegister_t::pSuppsNotifEvents

8.683 pack_voice_SLQSVoiceManageCalls_t Struct Reference

Data Fields

- uint8_t SUPSType

- uint8_t * pCallID

8.683.1 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 • NULL pointer - Invalid data.

8.683.2 Field Documentation

8.683.2.1 uint8_t* pack_voice_SLQSVoiceManageCalls_t::pCallID

8.683.2.2 uint8_t pack_voice_SLQSVoiceManageCalls_t::SUPSType

8.684 pack_voice_SLQSVoiceOrigUSSDNoWait_t Struct Reference

Data Fields

- struct voice_USSInfo USSInformation

8.684.1 Detailed Description

This structure contains Orig USSD No Wait Information Parameters.

Parameters

USSInformation	• See voice_USSInfo for more information.

8.684.2 Field Documentation

8.684.2.1 struct voice_USSInfo pack_voice_SLQSVoiceOrigUSSDNoWait_t::USSInformation

8.685 pack_voice_SLQSVoiceSendFlash_t Struct Reference

Data Fields

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

8.685.1 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

8.685.2 Field Documentation

8.685.2.1 uint8_t* pack_voice_SLQSVoiceSendFlash_t::pCallID

8.685.2.2 uint8_t* pack_voice_SLQSVoiceSendFlash_t::pFlashPayLd

8.685.2.3 uint8_t* pack_voice_SLQSVoiceSendFlash_t::pFlashType

8.686 pack_voice_SLQSVoiceSetCallBarringPassword_t Struct Reference

Data Fields

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

- uint8_t newPasswdAgain [4]

8.686.1 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.

8.686.2 Field Documentation

8.686.2.1 uint8_t pack_voice_SLQSVoiceSetCallBarringPassword_t::newPasswd[4]

8.686.2.2 uint8_t pack_voice_SLQSVoiceSetCallBarringPassword_t::newPasswdAgain[4]

8.686.2.3 uint8_t pack_voice_SLQSVoiceSetCallBarringPassword_t::oldPasswd[4]

8.686.2.4 uint8_t pack_voice_SLQSVoiceSetCallBarringPassword_t::Reason

8.687 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

8.687.1 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.

8.687.2 Field Documentation

8.687.2.1 voice_airTimer* pack_voice_SLQSVoiceSetConfig_t::pAirTimerConfig

8.687.2.2 uint8_t* pack_voice_SLQSVoiceSetConfig_t::pAutoAnswer

8.687.2.3 uint8_t* pack_voice_SLQSVoiceSetConfig_t::pPrefVoiceDomain

8.687.2.4 voice_prefVoiceSO* pack_voice_SLQSVoiceSetConfig_t::pPrefVoiceSO

8.687.2.5 voice_roamTimer* pack_voice_SLQSVoiceSetConfig_t::pRoamTimerConfig

8.687.2.6 uint8_t* pack_voice_SLQSVoiceSetConfig_t::pTTYMode

8.688 pack_voice_SLQSVoiceSetPreferredPrivacy_t Struct Reference

Data Fields

- uint8_t privacyPref

8.688.1 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

8.688.2 Field Documentation

8.688.2.1 uint8_t pack_voice_SLQSVoiceSetPreferredPrivacy_t::privacyPref

8.689 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

8.689.1 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

8.689.2 Field Documentation

8.689.2.1 uint8_t* pack_voice_SLQSVoiceSetSUPSService_t::pCallBarringPasswd

8.689.2.2 uint8_t* pack_voice_SLQSVoiceSetSUPSService_t::pCallForwardingNumber

8.689.2.3 voice_callFwdTypeAndPlan* pack_voice_SLQSVoiceSetSUPSService_t::pCallFwdTypeAndPlan

8.689.2.4 uint8_t* pack_voice_SLQSVoiceSetSUPSService_t::pServiceClass

8.689.2.5 uint8_t* pack_voice_SLQSVoiceSetSUPSService_t::pTimerVal

8.689.2.6 uint8_t pack_voice_SLQSVoiceSetSUPSService_t::reason

8.689.2.7 uint8_t pack_voice_SLQSVoiceSetSUPSService_t::voiceSvc

8.690 pack_voice_SLQSVoiceStartContDTMF_t Struct Reference

Data Fields

- uint8_t * pCallID
- uint8_t DTMFdigit

8.690.1 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.

8.690.2 Field Documentation

8.690.2.1 uint8_t pack_voice_SLQSVoiceStartContDTMF_t::DTMFdigit

8.690.2.2 uint8_t* pack_voice_SLQSVoiceStartContDTMF_t::pCallID

8.691 pack_voice_SLQSVoiceStopContDTMF_t Struct Reference

Data Fields

- uint8_t callID

8.691.1 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

8.691.2 Field Documentation

8.691.2.1 uint8_t pack_voice_SLQSVoiceStopContDTMF_t::callID

8.692 pack_wds_DHCPv4ClientLeaseChange_t Struct Reference

Data Fields

- uint8_t * pEnableNotification

8.692.1 Detailed Description

WDS SWI DHCPv4 Client Lease Change Structure

Parameters

pEnable-Notification	• Enable Notification or not

8.692.2 Field Documentation

8.692.2.1 uint8_t* pack_wds_DHCPv4ClientLeaseChange_t::pEnableNotification

8.693 pack_wds_GetDefaultProfile_t Struct Reference

Data Fields

- uint32_t profiletype

8.693.1 Detailed Description

This structure contains pack get default profile information.

Parameters

profiletype	profile type • 0 - WDS_PROFILE_TYPE_3GPP (0x00) - 3GPP • 1 - WDS_PROFILE_TYPE_3GPP (0x01) - 3GPP2 • 2 - WDS_PROFILE_TYPE_EPC (0x02) - EPC

8.693.2 Field Documentation

8.693.2.1 uint32_t pack_wds_GetDefaultProfile_t::profiletype

8.694 pack_wds_GetDefaultProfileNum_t Struct Reference

Data Fields

- uint8_t type
- uint8_t family

8.694.1 Detailed Description

This structure contains pack Get Default Profile Number information.

Parameters

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

8.694.2 Field Documentation

8.694.2.1 uint8_t pack_wds_GetDefaultProfileNum_t::family

8.694.2.2 uint8_t pack_wds_GetDefaultProfileNum_t::type

8.695 pack_wds_GetDormancyState_t Struct Reference

8.695.1 Detailed Description

This structure contains pack get dormancy state information.

Parameters

NULL	• this is a dummy structure

8.696 pack_wds_GetLastMobileIPError_t Struct Reference

8.696.1 Detailed Description

This structure contains pack get last mobile IP error information.

Parameters

NULL	• this is a dummy structure

8.697 pack_wds_GetMobileIP_t Struct Reference

8.697.1 Detailed Description

This structure contains pack get mobile IP information.

Parameters

NULL	• this is a dummy structure

8.698 pack_wds_GetMobileIPProfile_t Struct Reference

Data Fields

- uint8_t index

8.698.1 Detailed Description

This structure contains pack get mobile IP profile information.

Parameters

index	• Mobile IP profile ID

8.698.2 Field Documentation

8.698.2.1 uint8_t pack_wds_GetMobileIPProfile_t::index

8.699 pack_wds_GetPacketStatistics_t Struct Reference

Data Fields

- uint32_t * pStatMask

8.699.1 Detailed Description

This structure contains pack get packet statistics information.

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

8.699.2 Field Documentation

8.699.2.1 uint32_t* pack_wds_GetPacketStatistics_t::pStatMask

8.700 pack_wds_GetPacketStatus_t Struct Reference

Data Fields

- uint32_t statmask

8.700.1 Detailed Description

This structure contains pack get packet status information.

Parameters

statmask	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 • 0x00000100 - Tx packets dropped • 0x00000200 - Rx packets dropped

8.700.2 Field Documentation

8.700.2.1 uint32_t pack_wds_GetPacketStatus_t::statmask

8.701 pack_wds_GetSessionDuration_t Struct Reference

8.701.1 Detailed Description

This structure contains pack get session duration information.

Parameters

NULL	• this is a dummy structure

8.702 pack_wds_RMSetTransferStatistics_t Struct Reference

Data Fields

- rmTrasnferStaticsReq RmTrasnferStaticsReq

8.702.1 Detailed Description

This structure contains pack fetch current data system transfer statistics information.

Parameters

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

8.702.2 Field Documentation

8.702.2.1 rmTrasnferStaticsReq pack_wds_RMSetTransferStatistics_t::RmTrasnferStaticsReq

8.703 pack_wds_SetAutoconnect_t Struct Reference

Data Fields

- uint8_t acsetting
- uint8_t acroamsetting

8.703.1 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

8.703.2 Field Documentation

8.703.2.1 uint8_t pack_wds_SetAutoconnect_t::acroamsetting

8.703.2.2 uint8_t pack_wds_SetAutoconnect_t::acsetting

8.704 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

8.704.1 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)
pUsername	• Username used during network authentication (optional)
pAPNName	• Access point name. NULL-terminated string parameter that is a logical name used to select GGSN and external packet data network (optional) • If value is NULL or omitted, then subscription default value will be requested.
pPassword	• Password used during network authentication (optional)

8.704.2 Field Documentation

8.704.2.1 uint32_t pack_wds_SetDefaultProfile_t::authentication

8.704.2.2 uint32_t pack_wds_SetDefaultProfile_t::ipAddress

8.704.2.3 uint8_t* pack_wds_SetDefaultProfile_t::pApnname

8.704.2.4 uint32_t pack_wds_SetDefaultProfile_t::pdpType

8.704.2.5 uint8_t* pack_wds_SetDefaultProfile_t::pName

8.704.2.6 uint8_t* pack_wds_SetDefaultProfile_t::pPassword

8.704.2.7 uint32_t pack_wds_SetDefaultProfile_t::primaryDNS

8.704.2.8 uint32_t pack_wds_SetDefaultProfile_t::profileType

8.704.2.9 uint8_t* pack_wds_SetDefaultProfile_t::pUsername

8.704.2.10 uint32_t pack_wds_SetDefaultProfile_t::secondaryDNS

8.705 pack_wds_SetDefaultProfileNum_t Struct Reference

Data Fields

- uint8_t type
- uint8_t family
- uint8_t index

8.705.1 Detailed Description

This structure to hold Set default profile number

Parameters

type	Identifies the technology type of the profile • 0 - 3GPP • 1 - 3GPP2
family	Identifies the family of profile • 0 - Embedded • 1 - Tethered
index	Profile number to be set as default profile.

8.705.2 Field Documentation

8.705.2.1 uint8_t pack_wds_SetDefaultProfileNum_t::family

8.705.2.2 uint8_t pack_wds_SetDefaultProfileNum_t::index

8.705.2.3 uint8_t pack_wds_SetDefaultProfileNum_t::type

8.706 pack_wds_SetMobileIP_t Struct Reference

Data Fields

- uint32_t mode

8.706.1 Detailed Description

This structure contains set mobile IP pack information.

Parameters

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

8.706.2 Field Documentation

8.706.2.1 uint32_t pack_wds_SetMobileIP_t::mode

8.707 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

8.707.1 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

8.707.2 Field Documentation

8.707.2.1 uint8_t* pack_wds_SetMobileIPParameters_t::pHA2002bis

8.707.2.2 uint8_t* pack_wds_SetMobileIPParameters_t::pHAAuthenticator

8.707.2.3 uint32_t* pack_wds_SetMobileIPParameters_t::pMode

8.707.2.4 uint8_t* pack_wds_SetMobileIPParameters_t::pReRegPeriod

8.707.2.5 uint8_t* pack_wds_SetMobileIPParameters_t::pReRegTraffic

8.707.2.6 uint8_t* pack_wds_SetMobileIPParameters_t::pRetryInterval

8.707.2.7 uint8_t* pack_wds_SetMobileIPParameters_t::pRetryLimit

8.707.2.8 char* pack_wds_SetMobileIPParameters_t::pSPC

8.708 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

8.708.1 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

8.708.2 Field Documentation

8.708.2.1 uint8_t pack_wds_SetMobileIPProfile_t::index

8.708.2.2 uint32_t* pack_wds_SetMobileIPProfile_t::pAAASPI

8.708.2.3 uint32_t* pack_wds_SetMobileIPProfile_t::pAddress

8.708.2.4 uint8_t* pack_wds_SetMobileIPProfile_t::pEnabled

8.708.2.5 uint32_t* pack_wds_SetMobileIPProfile_t::pHASPI

8.708.2.6 int8_t* pack_wds_SetMobileIPProfile_t::pMNAAA

8.708.2.7 int8_t* pack_wds_SetMobileIPProfile_t::pMNHA

8.708.2.8 int8_t* pack_wds_SetMobileIPProfile_t::pNAI

8.708.2.9 uint32_t* pack_wds_SetMobileIPProfile_t::pPrimaryHA

8.708.2.10 uint8_t* pack_wds_SetMobileIPProfile_t::pRevTunneling

8.708.2.11 uint32_t* pack_wds_SetMobileIPProfile_t::pSecondaryHA

8.708.2.12 int8_t pack_wds_SetMobileIPProfile_t::spc[10]

8.709 pack_wds_SLQSCreateProfile_t Struct Reference

Data Fields

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

8.709.1 Detailed Description

This structure contains pack create profile.

Parameters

pProfileId	• 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
pProfileType	• Identifies the technology type of the profile – 0x00 - 3GPP – 0x01 - 3GPP2 – NULL is not allowed
pCurProfile	• union of 3GPP and 3GPP2 profile • See wds_profileInfo

Note

- If profileID is NULL, 3GPP profile will be created and index will be assigned based on availability in device.
- If profileID is not NULL depending on pProfileType 3GPP/3GPP2 relevant profile will be created

8.709.2 Field Documentation

8.709.2.1 wds_profileInfo* pack_wds_SLQSCreateProfile_t::pCurProfile

8.709.2.2 uint8_t* pack_wds_SLQSCreateProfile_t::pProfileId

8.709.2.3 uint8_t* pack_wds_SLQSCreateProfile_t::pProfileType

8.710 pack_wds_SLQSDeleteProfile_t Struct Reference

Data Fields

- uint8_t profileType
- uint8_t profileIndex

8.710.1 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)

8.710.2 Field Documentation

8.710.2.1 uint8_t pack_wds_SLQSDeleteProfile_t::profileIndex

8.710.2.2 uint8_t pack_wds_SLQSDeleteProfile_t::profileType

8.711 pack_wds_SLQSGetCurrDataSystemStat_t Struct Reference

8.711.1 Detailed Description

This structure contains pack get current data system state information.

Parameters

NULL	• this is a dummy structure

8.712 pack_wds_SLQSGetDataBearerTechnology_t Struct Reference

8.712.1 Detailed Description

This structure contains pack get data bearer technology information.

Parameters

NULL	• this is a dummy structure

8.713 pack_wds_SLQSGetDUNCallInfo_t Struct Reference

Data Fields

- uint32_t Mask
- uint8_t * pReportConnStatus
- wds_transferStatInd * pTransferStatInd
- uint8_t * pReportDormStatus
- uint8_t * pReportDataBearerTech
- uint8_t * pReportChannelRate

8.713.1 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 wds_transferStatInd for more information

pReportDormStatus	• Dormancy Status Indicator – 0 - Do not report – 1 - Report traffic channel state of interface used for data connection
pReportDataBearerTech	• Current Data Bearer Technology Indicator – 0 - Do not report – 1 - Report radio interface used for data transfer when it changes
pReportChannelRate	• Channel Rate Indicator – 0 - Do not report – 1 - Report channel rate

8.713.2 Field Documentation

8.713.2.1 uint32_t pack_wds_SLQSGetDUNCallInfo_t::Mask

8.713.2.2 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportChannelRate

8.713.2.3 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportConnStatus

8.713.2.4 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportDataBearerTech

8.713.2.5 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportDormStatus

8.713.2.6 wds_transferStatInd* pack_wds_SLQSGetDUNCallInfo_t::pTransferStatInd

8.714 pack_wds_SLQSGetProfileSettings_t Struct Reference

Data Fields

- uint8_t ProfileId
- uint8_t ProfileType

8.714.1 Detailed Description

This structure contains pack get profile settings information.

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

8.714.2 Field Documentation

8.714.2.1 uint8_t pack_wds_SLQSGetProfileSettings_t::ProfileId

8.714.2.2 uint8_t pack_wds_SLQSGetProfileSettings_t::ProfileType

8.715 pack_wds_SLQSGetRuntimeSettings_t Struct Reference

Data Fields

- uint32_t * pReqSettings

8.715.1 Detailed Description

This structure contains pack get runtime settings information

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)

8.715.2 Field Documentation

8.715.2.1 uint32_t* pack_wds_SLQSGetRuntimeSettings_t::pReqSettings

8.716 pack_wds_SLQSModifyProfile_t Struct Reference

Data Fields

- uint8_t * pProfileId
- uint8_t * pProfileType
- wds_profileInfo curProfile

8.716.1 Detailed Description

This structure contains pack modify profile information.

Parameters

pProfileId	• 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
pProfileType	• Identifies the technology type of the profile – 0x00 - 3GPP – 0x01 - 3GPP2 – NULL is not allowed
curProfile	• union of 3GPP and 3GPP2 profile • See wds_profileInfo

Note

- If profileID is NULL, 3GPP profile will be created and index will be assigned based on availability in device.
- If profileID is not NULL depending on pProfileType 3GPP/3GPP2 relevant profile will be created

8.716.2 Field Documentation

8.716.2.1 wds_profileInfo pack_wds_SLQSModifyProfile_t::curProfile

8.716.2.2 uint8_t* pack_wds_SLQSModifyProfile_t::pProfileId

8.716.2.3 uint8_t* pack_wds_SLQSModifyProfile_t::pProfileType

8.717 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

8.717.1 Detailed Description

This structure contains pack set 3GPP configuration item information.

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

8.717.2 Field Documentation

8.717.2.1 uint16_t pack_wds_SLQSSet3GPPConfigItem_t::LTEAttachProfileListLen

8.717.2.2 uint8_t* pack_wds_SLQSSet3GPPConfigItem_t::p3gppRelease

8.717.2.3 uint8_t* pack_wds_SLQSSet3GPPConfigItem_t::pDefaultPDNEnabled

8.717.2.4 uint16_t* pack_wds_SLQSSet3GPPConfigItem_t::pLTEAttachProfile

8.717.2.5 uint16_t* pack_wds_SLQSSet3GPPConfigItem_t::pLTEAttachProfileList

8.717.2.6 uint16_t* pack_wds_SLQSSet3GPPConfigItem_t::pProfileList

8.718 pack_wds_SLQSSetIPFamilyPreference_t Struct Reference

Data Fields

- uint8_t IPFamilyPreference

8.718.1 Detailed Description

This structure contains pack set IP family preference information

Parameters

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

8.718.2 Field Documentation

8.718.2.1 uint8_t pack_wds_SLQSSetIPFamilyPreference_t::IPFamilyPreference

8.719 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

8.719.1 Detailed Description

This structure contains set WDS event callback information.

Parameters

dataBearer	data bearer
dormancyStatus	dormancy status
mobileIP	mobile IP
currentData-Bearer	current data bearer
dataSystem-Status	data system status
dataBearerTech-Ext	data Bearer Technology Extended
interval	interval

8.719.2 Field Documentation

8.719.2.1 uint8_t pack_wds_SLQSSetWdsEventCallback_t::currentDataBearer

8.719.2.2 uint8_t pack_wds_SLQSSetWdsEventCallback_t::dataBearer

8.719.2.3 uint8_t pack_wds_SLQSSetWdsEventCallback_t::dataBearerTechExt

8.719.2.4 uint8_t pack_wds_SLQSSetWdsEventCallback_t::dataSystemStatus

8.719.2.5 uint8_t pack_wds_SLQSSetWdsEventCallback_t::dormancyStatus

8.719.2.6 uint8_t pack_wds_SLQSSetWdsEventCallback_t::interval

8.719.2.7 uint8_t pack_wds_SLQSSetWdsEventCallback_t::mobileIP

8.719.2.8 uint8_t pack_wds_SLQSSetWdsEventCallback_t::transferStats

8.720 pack_wds_SLQSSGetDHCPv4ClientConfig_t Struct Reference

Data Fields

- wdsDhcpv4ProfileId * pProfileId

8.720.1 Detailed Description

This structure contain get DHCPv4 client configure.

Parameters

pProfileId	pointer to Profile Id structure • See wdsDhcpv4ProfileId

8.720.2 Field Documentation

8.720.2.1 wdsDhcpv4ProfileId* pack_wds_SLQSSGetDHCPv4ClientConfig_t::pProfileId

8.721 pack_wds_SLQSSSetDHCPv4ClientConfig_t Struct Reference

Data Fields

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

8.721.1 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

8.721.2 Field Documentation

8.721.2.1 wds_DHCPv4HWConfig∗ pack_wds_SLQSSSetDHCPv4ClientConfig_t::pHwConfig

8.721.2.2 wds_DHCPv4ProfileId∗ pack_wds_SLQSSSetDHCPv4ClientConfig_t::pProfileId

8.721.2.3 wds_DHCPv4OptionList∗ pack_wds_SLQSSSetDHCPv4ClientConfig_t::pRequestOptionList

8.722 pack_wds_SLQSSSetLoopback_t Struct Reference

Data Fields

- uint8_t loopbackMode
- uint8_t loopbackMultiplier

8.722.1 Detailed Description

This structure contains pack set loopback information.

Parameters

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

8.722.2 Field Documentation

8.722.2.1 uint8_t pack_wds_SLQSSSetLoopback_t::loopbackMode

8.722.2.2 uint8_t pack_wds_SLQSSSetLoopback_t::loopbackMultiplier

8.723 pack_wds_SLQSStartDataSession_t Struct Reference

Data Fields

- uint8_t ∗ pTech

- uint32_t * pprofileid3gpp
- uint32_t * pprofileid3gpp2
- uint32_t * pAuth
- char * pUser
- char * pPass

8.723.1 Detailed Description

This structure contains pack Start Data Session Information.

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

8.723.2 Field Documentation

8.723.2.1 uint32_t* pack_wds_SLQSStartDataSession_t::pAuth

8.723.2.2 char* pack_wds_SLQSStartDataSession_t::pPass

8.723.2.3 uint32_t* pack_wds_SLQSStartDataSession_t::pprofileid3gpp

8.723.2.4 uint32_t* pack_wds_SLQSStartDataSession_t::pprofileid3gpp2

8.723.2.5 uint8_t* pack_wds_SLQSStartDataSession_t::pTech

8.723.2.6 char* pack_wds_SLQSStartDataSession_t::pUser

8.724 pack_wds_SLQSStopDataSession_t Struct Reference

Data Fields

- uint32_t * psid

8.724.1 Detailed Description

This structure contains pack stop data session information.

Parameters

psid	session id

8.724.2 Field Documentation

8.724.2.1 uint32_t* pack_wds_SLQSStopDataSession_t::psid

8.725 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

8.725.1 Detailed Description

This structure contains the information about the Set Event Report Request parameters.

Parameters

pCurrChannel-RateInd	(optional) • Current Channel Rate Indicator. – 0 - Do not report – 1 - Report channel rate when it changes
pTransferStatInd	(optional) • See wds_TrStatInd for more information.
pDataBearer-TechInd	(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.

8.725.2 Field Documentation

8.725.2.1 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pCurrChannelRateInd

8.725.2.2 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pCurrDataBearerTechInd

8.725.2.3 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pCurrPrefDataSysInd

8.725.2.4 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pDataBearerTechInd

8.725.2.5 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pDataCallStatusChangeInd

8.725.2.6 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pDataSystemStatusChangeInd

8.725.2.7 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pDormancyStatusInd

8.725.2.8 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pEVDOPageMonPerChangeInd

8.725.2.9 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pMIPStatusInd

8.725.2.10 wds_TrStatInd* pack_wds_SLQSWdsSetEventReport_t::pTransferStatInd

8.726 pack_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference

Data Fields

- uint8_t contextId
- uint8_t contextType

8.726.1 Detailed Description

This structure contains the PDP Runtime Settings Request parameters.

Parameters

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

8.726.2 Field Documentation

8.726.2.1 uint8_t pack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextId

8.726.2.2 uint8_t pack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextType

8.727 PackCreateProfileOut Struct Reference

Data Fields

- uint8_t ProfileType
- uint8_t ProfileIndex
- uint16_t ExtErrorCode

8.727.1 Detailed Description

This structure contains out parameter Information

Parameters

ProfileType	• Identifies the type of profile 0x00 = 3GPP 0x01 = 3GPP2
ProfileIndex	• Index identifying the profile that was created
ExtErrorCode	• The extended error code received from DS Profile subsystem

8.727.2 Field Documentation

8.727.2.1 uint16_t PackCreateProfileOut::ExtErrorCode

8.727.2.2 uint8_t PackCreateProfileOut::ProfileIndex

8.727.2.3 uint8_t PackCreateProfileOut::ProfileType

8.728 packgetDyingGaspCfg Struct Reference

Data Fields

- uint8_t * pDestSMSNum
- uint8_t * pDestSMSContent

8.728.1 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.

8.728.2 Field Documentation

8.728.2.1 uint8_t* packgetDyingGaspCfg::pDestSMSContent

8.728.2.2 uint8_t* packgetDyingGaspCfg::pDestSMSNum

8.729 packgetDyingGaspStatistics Struct Reference

Data Fields

- uint32_t * pTimeStamp

- uint8_t * pSMSAttemptedFlag

8.729.1 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).
pSMSAttemptedFlag[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.

8.729.2 Field Documentation

8.729.2.1 uint8_t* packgetDyingGaspStatistics::pSMSAttemptedFlag

8.729.2.2 uint32_t* packgetDyingGaspStatistics::pTimeStamp

8.730 PCMparams Struct Reference

Data Fields

- BYTE iFaceTabLen
- BYTE iFaceTab [255]

8.730.1 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

8.730.2 Field Documentation

8.730.2.1 BYTE PCMparams::iFaceTab[255]

8.730.2.2 BYTE PCMparams::iFaceTabLen

8.731 PCSCFFQDNAddress Struct Reference

Data Fields

- WORD fqdnLen
- CHAR fqdnAddr [256]

8.731.1 Detailed Description

This structure contains the PCSCFFQDNAddress Information

Parameters

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

8.731.2 Field Documentation

8.731.2.1 CHAR PCSCFFQDNAddress::fqdnAddr[256]

8.731.2.2 WORD PCSCFFQDNAddress::fqdnLen

8.732 PCSCFFQDNAddressList Struct Reference

Data Fields

- BYTE numInstances
- struct PCSCFFQDNAddress pcsfFQDNAddress [10]

8.732.1 Detailed Description

This structure contains the PCSCFFQDNAddressList Information

Parameters

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

8.732.2 Field Documentation

8.732.2.1 BYTE PCSCFFQDNAddressList::numInstances

8.732.2.2 struct PCSCFFQDNAddress PCSCFFQDNAddressList::pcsfFQDNAddress[10]

8.733 PCSCFIPv4ServerAddressList Struct Reference

Data Fields

- BYTE numInstances
- ULONG pscsfIPv4Addr [64]

8.733.1 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)

8.733.2 Field Documentation

8.733.2.1 BYTE PCSCFIPv4ServerAddressList::numInstances

8.733.2.2 ULONG PCSCFIPv4ServerAddressList::pscsfIPv4Addr[64]

8.734 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

8.734.1 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

8.734.2 Field Documentation

8.734.2.1 ULONG* PDSPositionData::pAltitudeWrtEllipsoid

8.734.2.2 ULONG* PDSPositionData::pAltitudeWrtSealevel

8.734.2.3 BYTE* PDSPositionData::pHorizontalConfidence

8.734.2.4 ULONG* PDSPositionData::pHorizontalUncCircular

8.734.2.5 ULONGLONG* PDSPositionData::pLatitude

8.734.2.6 ULONGLONG* PDSPositionData::pLongitude

8.734.2.7 BYTE* PDSPositionData::pPositionSource

8.734.2.8 ULONGLONG* PDSPositionData::pTimeStamp

8.734.2.9 BYTE* PDSPositionData::pTimeType

8.734.2.10 BYTE* PDSPositionData::pVerticalConfidence

8.734.2.11 ULONG* PDSPositionData::pVerticalUnc

8.735 PDSPosMethodStateReq Struct Reference

Data Fields

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

8.735.1 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

8.735.2 Field Documentation

8.735.2.1 BYTE* PDSPosMethodStateReq::pWifiState

8.735.2.2 BYTE* PDSPosMethodStateReq::pXtraDataState

8.735.2.3 BYTE* PDSPosMethodStateReq::pXtraTimeState

8.736 peerNumberInfo Struct Reference

Data Fields

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

8.736.1 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.

8.736.2 Field Documentation

8.736.2.1 BYTE peerNumberInfo::callID

8.736.2.2 BYTE peerNumberInfo::number[81]

8.736.2.3 BYTE peerNumberInfo::numLen

8.736.2.4 BYTE peerNumberInfo::numPI

8.736.2.5 BYTE peerNumberInfo::numPlan

8.736.2.6 BYTE peerNumberInfo::numSI

8.736.2.7 BYTE peerNumberInfo::numType

8.737 personalizationStatus Struct Reference

Data Fields

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

8.737.1 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.

8.737.2 Field Documentation

8.737.2.1 BYTE personalizationStatus::feature[12]

8.737.2.2 BYTE personalizationStatus::numFeatures

8.737.2.3 BYTE personalizationStatus::unblockLeft[12]

8.737.2.4 BYTE personalizationStatus::verifyLeft[12]

8.738 PhyCaAggPcellInfo Struct Reference

Data Fields

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

8.738.1 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.

8.738.2 Field Documentation

8.738.2.1 NAS_LTE_CPHY_CA_BW_NRB PhyCaAggPcellInfo::dl_bw_value

8.738.2.2 int PhyCaAggPcellInfo::freq

8.738.2.3 int PhyCaAggPcellInfo::iLTEbandValue

8.738.2.4 int PhyCaAggPcellInfo::pci

8.738.2.5 BYTE PhyCaAggPcellInfo::TlvPresent

8.739 PhyCaAggScellDlBw Struct Reference

Data Fields

- NAS_LTE_CPHY_CA_BW_NRB dl_bw_value

- BYTE TlvPresent

8.739.1 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.

8.739.2 Field Documentation

8.739.2.1 NAS_LTE_CPHY_CA_BW_NRB PhyCaAggScellDlBw::dl_bw_value

8.739.2.2 BYTE PhyCaAggScellDlBw::TlvPresent

8.740 PhyCaAggScellIndex Struct Reference

Data Fields

- BYTE scell_idx
- BYTE TlvPresent

8.740.1 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.

8.740.2 Field Documentation

8.740.2.1 BYTE PhyCaAggScellIndex::scell_idx

8.740.2.2 BYTE PhyCaAggScellIndex::TlvPresent

8.741 PhyCaAggScellIndType Struct Reference

Data Fields

- int pci
- int freq

- 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 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.

8.741.2 Field Documentation

8.741.2.1 int PhyCaAggScellIndType::freq

8.741.2.2 int PhyCaAggScellIndType::pci

8.741.2.3 NAS_LTE_CPHY_SCELL_STATE PhyCaAggScellIndType::scell_state

8.741.2.4 BYTE PhyCaAggScellIndType::TlvPresent

8.742 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

8.742.1 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.

8.742.2 Field Documentation

8.742.2.1 NAS_LTE_CPHY_CA_BW_NRB PhyCaAggScellInfo::dl_bw_value

8.742.2.2 int PhyCaAggScellInfo::freq

8.742.2.3 int PhyCaAggScellInfo::iLTEbandValue

8.742.2.4 int PhyCaAggScellInfo::pci

8.742.2.5 NAS_LTE_CPHY_SCELL_STATE PhyCaAggScellInfo::scell_state

8.742.2.6 BYTE PhyCaAggScellInfo::TlvPresent

8.743 PilotSetData Struct Reference

Data Fields

- BYTE NumPilots
- PilotSetParams * pPilotSetInfo

8.743.1 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.

8.743.2 Field Documentation

8.743.2.1 BYTE PilotSetData::NumPilots

8.743.2.2 PilotSetParams* PilotSetData::pPilotSetInfo

8.744 PilotSetParams Struct Reference

Data Fields

- ULONG PilotType
- WORD PilotPN
- WORD PilotStrength

8.744.1 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)

8.744.2 Field Documentation

8.744.2.1 WORD PilotSetParams::PilotPN

8.744.2.2 WORD PilotSetParams::PilotStrength

8.744.2.3 ULONG PilotSetParams::PilotType

8.745 pktErrRate Struct Reference

Data Fields

- WORD multiplier
- WORD exponent

8.745.1 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)

8.745.2 Field Documentation

8.745.2.1 WORD pktErrRate::exponent

8.745.2.2 WORD pktErrRate::multiplier

8.746 PlmnID Struct Reference

Data Fields

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

8.746.1 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

8.746.2 Field Documentation

8.746.2.1 WORD PlmnID::mcc

8.746.2.2 WORD PlmnID::mnc

8.746.2.3 BOOL PlmnID::pcsDigit

8.746.2.4 BYTE PlmnID::TlvPresent

8.747 PLMNNetworkName Struct Reference

Data Fields

- BYTE numInstance
- PLMNNetworkNameData PLMNNetName [255]

8.747.1 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).

8.747.2 Field Documentation

8.747.2.1 BYTE PLMNNetworkName::numInstance

8.747.2.2 PLMNNetworkNameData PLMNNetworkName::PLMNNetName[255]

8.748 PLMNNetworkNameData Struct Reference

Data Fields

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

8.748.1 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.

8.748.2 Field Documentation

8.748.2.1 BYTE PLMNNetworkNameData::codingScheme

8.748.2.2 BYTE PLMNNetworkNameData::countryInitials

8.748.2.3 BYTE PLMNNetworkNameData::longName[255]

8.748.2.4 BYTE PLMNNetworkNameData::longNameLen

8.748.2.5 BYTE PLMNNetworkNameData::longNameSpareBits

8.748.2.6 BYTE PLMNNetworkNameData::shortName[255]

8.748.2.7 BYTE PLMNNetworkNameData::shortNameLen

8.748.2.8 BYTE PLMNNetworkNameData::shortNameSpareBits

8.749 Port Struct Reference

Data Fields

- WORD port
- WORD range

8.749.1 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

8.749.2 Field Documentation

8.749.2.1 WORD Port::port

8.749.2.2 WORD Port::range

8.750 precisionDilution_s Struct Reference

Data Fields

- ULONG PDOP
- ULONG HDOP
- ULONG VDOP

8.750.1 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)

8.750.2 Field Documentation

8.750.2.1 ULONG precisionDilution_s::HDOP

8.750.2.2 ULONG precisionDilution_s::PDOP

8.750.2.3 ULONG precisionDilution_s::VDOP

8.751 PrefImageList Struct Reference

Data Fields

- BYTE listSize
- struct ImageElement listEntries [2]

8.751.1 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

8.751.2 Field Documentation

8.751.2.1 struct ImageElement PrefImageList::listEntries[2]

8.751.2.2 BYTE PrefImageList::listSize

8.752 prefVoiceSO Struct Reference

Data Fields

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

8.752.1 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

8.752.2 Field Documentation

8.752.2.1 BYTE prefVoiceSO::evrcCapability

8.752.2.2 WORD prefVoiceSO::homeOrigVoiceSO

8.752.2.3 WORD prefVoiceSO::homePageVoiceSO

8.752.2.4 BYTE prefVoiceSO::namID

8.752.2.5 WORD prefVoiceSO::roamOrigVoiceSO

8.753 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

8.753.1 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().

8.753.2 Field Documentation

8.753.2.1 BYTE* Profile3GPP::pAddrAllocPref

8.753.2.2 BYTE* Profile3GPP::pAPNClass

8.753.2.3 BYTE* Profile3GPP::pAPNDisabledFlag

8.753.2.4 CHAR* Profile3GPP::pAPNName

8.753.2.5 WORD* Profile3GPP::pAPNnameSize

8.753.2.6 BYTE* Profile3GPP::pAuthenticationPref

8.753.2.7 struct GPRSRequestedQoS* Profile3GPP::pGPRSMinimumQoS

8.753.2.8 struct GPRSRequestedQoS* Profile3GPP::pGPRSRequestedQos

8.753.2.9 BYTE* Profile3GPP::pImCnFlag

8.753.2.10 ULONG* Profile3GPP::pIPv4AddrPref

8.753.2.11 USHORT* Profile3GPP::pIPv6AddPref

8.753.2.12 CHAR* Profile3GPP::pPassword

8.753.2.13 WORD* Profile3GPP::pPasswordSize

8.753.2.14 BYTE* Profile3GPP::pPcscfAddrUsingDhcp

8.753.2.15 BYTE* Profile3GPP::pPcscfAddrUsingPCO

8.753.2.16 ULONG* Profile3GPP::pPDNInactivTimeout

8.753.2.17 BYTE* Profile3GPP::pPdpAccessConFlag

8.753.2.18 BYTE* Profile3GPP::pPdpContext

8.753.2.19 BYTE* Profile3GPP::pPdpDataCompType

8.753.2.20 BYTE* Profile3GPP::pPdpHdrCompType

8.753.2.21 BYTE* Profile3GPP::pPDPtype

8.753.2.22 ULONG* Profile3GPP::pPriDNSIPv4AddPref

8.753.2.23 USHORT* Profile3GPP::pPriDNSIPv6addpref

8.753.2.24 **BYTE*** Profile3GPP::pPrimaryID

8.753.2.25 **CHAR*** Profile3GPP::pProfilename

8.753.2.26 **WORD*** Profile3GPP::pProfilenameSize

8.753.2.27 struct **QosClassID*** Profile3GPP::pQosClassID

8.753.2.28 **ULONG*** Profile3GPP::pSecDNSIPv4AddPref

8.753.2.29 **USHORT*** Profile3GPP::pSecDNSIPv6addpref

8.753.2.30 **BYTE*** Profile3GPP::pSecondaryFlag

8.753.2.31 struct **TFTIDParams*** Profile3GPP::pTFTID1Params

8.753.2.32 struct **TFTIDParams*** Profile3GPP::pTFTID2Params

8.753.2.33 struct **UMTSQoS*** Profile3GPP::pUMTSMinQoS

8.753.2.34 struct **UMTSReqQoSSigInd*** Profile3GPP::pUMTSMinQosSigInd

8.753.2.35 struct **UMTSQoS*** Profile3GPP::pUMTSReqQoS

8.753.2.36 struct **UMTSReqQoSSigInd*** Profile3GPP::pUMTSReqQoSSigInd

8.753.2.37 **CHAR*** Profile3GPP::pUsername

8.753.2.38 **WORD*** Profile3GPP::pUsernameSize

8.754 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

8.754.1 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().

8.754.2 Field Documentation

8.754.2.1 BYTE* Profile3GPP2::pAllowLinger

8.754.2.2 BYTE* Profile3GPP2::pAPNClass3GPP2

8.754.2.3 BYTE* Profile3GPP2::pAPNEnabled3GPP2

8.754.2.4 CHAR* Profile3GPP2::pApnString

8.754.2.5 WORD* Profile3GPP2::pApnStringSize

8.754.2.6 BYTE* Profile3GPP2::pAppPriority

8.754.2.7 ULONG* Profile3GPP2::pAppType

8.754.2.8 CHAR* Profile3GPP2::pAuthPassword

8.754.2.9 WORD* Profile3GPP2::pAuthPasswordSize

8.754.2.10 BYTE* Profile3GPP2::pAuthProtocol

8.754.2.11 BYTE* Profile3GPP2::pAuthRetryCount

8.754.2.12 USHORT* Profile3GPP2::pAuthTimeout

8.754.2.13 BYTE* Profile3GPP2::pDataMode

8.754.2.14 BYTE* Profile3GPP2::pDataRate

8.754.2.15 USHORT* Profile3GPP2::pIpcpAckTimeout

8.754.2.16 BYTE* Profile3GPP2::pIpcpCreqRetryCount

8.754.2.17 BYTE* Profile3GPP2::pIsPcscfAddressNedded

8.754.2.18 USHORT* Profile3GPP2::pLcpAckTimeout

8.754.2.19 BYTE* Profile3GPP2::pLcpCreqRetryCount

8.754.2.20 BYTE* Profile3GPP2::pNegoDnsSrvrPref

8.754.2.21 ULONG* Profile3GPP2::pPDNInactivTimeout3GPP2

8.754.2.22 BYTE* Profile3GPP2::pPdnType

8.754.2.23 ULONG* Profile3GPP2::pPppSessCloseTimer1x

8.754.2.24 ULONG* Profile3GPP2::pPppSessCloseTimerDO

8.754.2.25 ULONG* Profile3GPP2::pPrimaryV4DnsAddress

8.754.2.26 USHORT* Profile3GPP2::pPriV6DnsAddress

8.754.2.27 BYTE* Profile3GPP2::pRATType

8.754.2.28 ULONG* Profile3GPP2::pSecondaryV4DnsAddress

8.754.2.29 USHORT* Profile3GPP2::pSecV6DnsAddress

8.754.2.30 CHAR* Profile3GPP2::pUserId

8.754.2.31 WORD* Profile3GPP2::pUserIdSize

8.755 ProfileIdentifier Struct Reference

Data Fields

- BYTE profileType
- BYTE profileIndex

8.755.1 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

8.755.2 Field Documentation

8.755.2.1 BYTE ProfileIdentifier::profileIndex

8.755.2.2 BYTE ProfileIdentifier::profileType

8.756 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

8.756.1 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

8.756.2 Field Documentation

8.756.2.1 WORD protocolSubtypeElement::AccessMac

8.756.2.2 WORD protocolSubtypeElement::AuthProt

8.756.2.3 WORD protocolSubtypeElement::ControlMac

8.756.2.4 WORD protocolSubtypeElement::EncryptProt

8.756.2.5 WORD protocolSubtypeElement::ForwardMac

8.756.2.6 WORD protocolSubtypeElement::IdleState

8.756.2.7 WORD protocolSubtypeElement::KeyExchange

8.756.2.8 WORD protocolSubtypeElement::MultDisc

8.756.2.9 WORD protocolSubtypeElement::PhysicalLayer

8.756.2.10 WORD protocolSubtypeElement::ReverseMac

8.756.2.11 WORD protocolSubtypeElement::SecProt

8.756.2.12 WORD protocolSubtypeElement::VirtStream

8.757 PSDetachReq Struct Reference

Data Fields

- BYTE * pDetachAction

8.757.1 Detailed Description

This structure contains information about the SLQSSwiPSDetach request parameters.

Parameters

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

8.757.2 Field Documentation

8.757.2.1 BYTE* PSDetachReq::pDetachAction

8.758 qaQmi3Gpp2TimeZone Struct Reference

Data Fields

- BYTE leapSeconds
- BYTE localTimeOffset
- BYTE daylightSavings

8.758.1 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)

8.758.2 Field Documentation

8.758.2.1 BYTE qaQmi3Gpp2TimeZone::daylightSavings

8.758.2.2 BYTE qaQmi3Gpp2TimeZone::leapSeconds

8.758.2.3 BYTE qaQmi3Gpp2TimeZone::localTimeOffset

8.759 qaQmiInterfaceInfo Struct Reference

Data Fields

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

8.759.1 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

8.759.2 Field Documentation

8.759.2.1 BYTE qaQmiInterfaceInfo::qaQmiinstanceid

8.759.2.2 eQaQMIService qaQmiInterfaceInfo::qaQmisvctype

8.759.2.3 ULONG qaQmiInterfaceInfo::v4sessionId

8.759.2.4 ULONG qaQmiInterfaceInfo::v6sessionId

8.760 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

8.760.1 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

8.760.2 Field Documentation

8.760.2.1 WORD qaQmiServingSystemParam::BasestationID

8.760.2.2 ULONG qaQmiServingSystemParam::BasestationLatitude

8.760.2.3 ULONG qaQmiServingSystemParam::BasestationLongitude

8.760.2.4 callBarStatus qaQmiServingSystemParam::CallBarStatus

8.760.2.5 BYTE qaQmiServingSystemParam::CDMA_P_Rev

8.760.2.6 CDMASysInfoExt qaQmiServingSystemParam::CDMASystemInfoExt

8.760.2.7 ULONG qaQmiServingSystemParam::CellID

8.760.2.8 BYTE qaQmiServingSystemParam::concSvcInfo

8.760.2.9 currentPLMN qaQmiServingSystemParam::CurrentPLMN

8.760.2.10 dataSrvCapabilities qaQmiServingSystemParam::DataSrvCapabilities

8.760.2.11 BYTE qaQmiServingSystemParam::defaultRoamInd

8.760.2.12 detailSvcInfo qaQmiServingSystemParam::DetailedSvcInfo

8.760.2.13 BYTE qaQmiServingSystemParam::DTMInd

8.760.2.14 qaQmi3Gpp2TimeZone qaQmiServingSystemParam::Gpp2TimeZone

8.760.2.15 BYTE qaQmiServingSystemParam::GppNetworkDSTAdjustment

8.760.2.16 BYTE qaQmiServingSystemParam::GppTimeZone

8.760.2.17 BYTE qaQmiServingSystemParam::hdrPersonality

8.760.2.18 WORD qaQmiServingSystemParam::Lac

8.760.2.19 WORD qaQmiServingSystemParam::NetworkID

8.760.2.20 BYTE qaQmiServingSystemParam::PRLInd

8.760.2.21 BYTE qaQmiServingSystemParam::roamIndicatorVal

8.760.2.22 roamIndList qaQmiServingSystemParam::RoamingIndicatorList

8.760.2.23 servSystem qaQmiServingSystemParam::ServingSystem

8.760.2.24 WORD qaQmiServingSystemParam::SystemID

8.760.2.25 WORD qaQmiServingSystemParam::trackAreaCode

8.761 QmiCbkCatEventStatusReportInd Struct Reference

Data Fields

- BYTE event_Index
- struct CatCommonEventTlv CCETlv [11]

8.761.1 Field Documentation

8.761.1.1 struct CatCommonEventTlv QmiCbkCatEventStatusReportInd::CCETlv[11]

8.761.1.2 BYTE QmiCbkCatEventStatusReportInd::event_Index

8.762 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

8.762.1 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

8.762.2 Field Documentation

8.762.2.1 ULONG* QmiCbkLocBestAvailPosInd::pAltitudeWrtEllipsoid

8.762.2.2 ULONG* QmiCbkLocBestAvailPosInd::pAltitudeWrtMeanSeaLevel

8.762.2.3 gpsTime* QmiCbkLocBestAvailPosInd::pGpsTime

8.762.2.4 ULONG* QmiCbkLocBestAvailPosInd::pHeading

8.762.2.5 ULONG* QmiCbkLocBestAvailPosInd::pHeadingUnc

8.762.2.6 BYTE* QmiCbkLocBestAvailPosInd::pHorCirConf

8.762.2.7 BYTE* QmiCbkLocBestAvailPosInd::pHorEllpConf

8.762.2.8 ULONG* QmiCbkLocBestAvailPosInd::pHorReliability

8.762.2.9 ULONG* QmiCbkLocBestAvailPosInd::pHorUncCircular

8.762.2.10 ULONG* QmiCbkLocBestAvailPosInd::pHorUncEllipseOrientAzimuth

8.762.2.11 ULONG* QmiCbkLocBestAvailPosInd::pHorUncEllipseSemiMajor

8.762.2.12 **ULONG*** QmiCbkLocBestAvailPosInd::pHorUncEllipseSemiMinor

8.762.2.13 **ULONGLONG*** QmiCbkLocBestAvailPosInd::pLatitude

8.762.2.14 **ULONGLONG*** QmiCbkLocBestAvailPosInd::pLongitude

8.762.2.15 **ULONG*** QmiCbkLocBestAvailPosInd::pMagneticDeviation

8.762.2.16 **precisionDilution*** QmiCbkLocBestAvailPosInd::pPrecisionDilution

8.762.2.17 **sensorDataUsage*** QmiCbkLocBestAvailPosInd::pSensorDataUsage

8.762.2.18 **ULONG*** QmiCbkLocBestAvailPosInd::pSpeedHorizontal

8.762.2.19 **ULONG*** QmiCbkLocBestAvailPosInd::pSpeedUnc

8.762.2.20 **ULONG*** QmiCbkLocBestAvailPosInd::pSpeedVertical

8.762.2.21 **ULONG*** QmiCbkLocBestAvailPosInd::pSpeedVerticalUnc

8.762.2.22 **svUsedforFix*** QmiCbkLocBestAvailPosInd::pSvUsedforFix

8.762.2.23 **ULONG*** QmiCbkLocBestAvailPosInd::pTechnologyMask

8.762.2.24 **ULONG*** QmiCbkLocBestAvailPosInd::pTimeSrc

8.762.2.25 **ULONGLONG*** QmiCbkLocBestAvailPosInd::pTimestampUtc

8.762.2.26 **ULONG*** QmiCbkLocBestAvailPosInd::pTimeUnc

8.762.2.27 **BYTE*** QmiCbkLocBestAvailPosInd::pVertConfidence

8.762.2.28 **ULONG*** QmiCbkLocBestAvailPosInd::pVertReliability

8.762.2.29 **ULONG*** QmiCbkLocBestAvailPosInd::pVertUnc

8.762.2.30 **ULONG*** QmiCbkLocBestAvailPosInd::pXid

8.762.2.31 **ULONG** QmiCbkLocBestAvailPosInd::status

8.763 QmiCbkLocCradleMountInd Struct Reference

Data Fields

- ULONG cradleMountConfigStatus

8.763.1 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

8.763.2 Field Documentation

8.763.2.1 ULONG QmiCbkLocCradleMountInd::cradleMountConfigStatus

8.764 QmiCbkLocEngineStateInd Struct Reference

Data Fields

- ULONG engineState

8.764.1 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

8.764.2 Field Documentation

8.764.2.1 ULONG QmiCbkLocEngineStateInd::engineState

8.765 QmiCbkLocEventTimeSyncInd Struct Reference

Data Fields

- ULONG timeSyncRefCounter

8.765.1 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.

8.765.2 Field Documentation

8.765.2.1 ULONG QmiCbkLocEventTimeSyncInd::timeSyncRefCounter

8.766 QmiCbkLocGetServerInd Struct Reference

Data Fields

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

8.766.1 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.

8.766.2 Field Documentation

8.766.2.1 IPv4Info∗ QmiCbkLocGetServerInd::pIPv4AddrInfo

8.766.2.2 IPv6Info∗ QmiCbkLocGetServerInd::pIPv6AddrInfo

8.766.2.3 urlAddr∗ QmiCbkLocGetServerInd::pURL

8.766.2.4 ULONG QmiCbkLocGetServerInd::serverStatus

8.766.2.5 ULONG QmiCbkLocGetServerInd::serverType

8.767 QmiCbkLocInjectPositionInd Struct Reference

Data Fields

- ULONG status

8.767.1 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

8.767.2 Field Documentation

8.767.2.1 ULONG QmiCbkLocInjectPositionInd::status

8.768 QmiCbkLocInjectSensorDataInd Struct Reference

Data Fields

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

8.768.1 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.

8.768.2 Field Documentation

8.768.2.1 ULONG QmiCbkLocInjectSensorDataInd::injectSensorDataStatus

8.768.2.2 BYTE* QmiCbkLocInjectSensorDataInd::pAccelSamplesAccepted

8.768.2.3 BYTE* QmiCbkLocInjectSensorDataInd::pAccelTempSamplesAccepted

8.768.2.4 BYTE* QmiCbkLocInjectSensorDataInd::pGyroSamplesAccepted

8.768.2.5 BYTE* QmiCbkLocInjectSensorDataInd::pGyroTempSamplesAccepted

8.768.2.6 ULONG* QmiCbkLocInjectSensorDataInd::pOpaqueIdentifier

8.769 QmiCbkLocInjectTimeInd Struct Reference

Data Fields

- ULONG injectTimeSyncStatus

8.769.1 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

8.769.2 Field Documentation

8.769.2.1 ULONG QmiCbkLocInjectTimeInd::injectTimeSyncStatus

8.770 QmiCbkLocInjectUTCTimeInd Struct Reference

Data Fields

- ULONG status

8.770.1 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

8.770.2 Field Documentation

8.770.2.1 ULONG QmiCbkLocInjectUTCTimeInd::status

8.771 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

8.771.1 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

8.771.2 Field Documentation

8.771.2.1 BYTE* QmiCbkLocPositionReportInd::pAltitudeAssumed

8.771.2.2 ULONG* QmiCbkLocPositionReportInd::pAltitudeWrtEllipsoid

8.771.2.3 ULONG* QmiCbkLocPositionReportInd::pAltitudeWrtMeanSeaLevel

8.771.2.4 ULONG* QmiCbkLocPositionReportInd::pFixId

8.771.2.5 gpsTime* QmiCbkLocPositionReportInd::pGpsTime

8.771.2.6 **ULONG*** QmiCbkLocPositionReportInd::pHeading

8.771.2.7 **ULONG*** QmiCbkLocPositionReportInd::pHeadingUnc

8.771.2.8 **BYTE*** QmiCbkLocPositionReportInd::pHorConfidence

8.771.2.9 **ULONG*** QmiCbkLocPositionReportInd::pHorReliability

8.771.2.10 **ULONG*** QmiCbkLocPositionReportInd::pHorUncCircular

8.771.2.11 **ULONG*** QmiCbkLocPositionReportInd::pHorUncEllipseOrientAzimuth

8.771.2.12 **ULONG*** QmiCbkLocPositionReportInd::pHorUncEllipseSemiMajor

8.771.2.13 **ULONG*** QmiCbkLocPositionReportInd::pHorUncEllipseSemiMinor

8.771.2.14 **ULONGLONG*** QmiCbkLocPositionReportInd::pLatitude

8.771.2.15 **BYTE*** QmiCbkLocPositionReportInd::pLeapSeconds

8.771.2.16 **ULONGLONG*** QmiCbkLocPositionReportInd::pLongitude

8.771.2.17 **ULONG*** QmiCbkLocPositionReportInd::pMagneticDeviation

8.771.2.18 **precisionDilution*** QmiCbkLocPositionReportInd::pPrecisionDilution

8.771.2.19 **sensorDataUsage*** QmiCbkLocPositionReportInd::pSensorDataUsage

8.771.2.20 **ULONG*** QmiCbkLocPositionReportInd::pSpeedHorizontal

8.771.2.21 **ULONG*** QmiCbkLocPositionReportInd::pSpeedUnc

8.771.2.22 **ULONG*** QmiCbkLocPositionReportInd::pSpeedVertical

8.771.2.23 **svUsedforFix*** QmiCbkLocPositionReportInd::pSvUsedforFix

8.771.2.24 **ULONG*** QmiCbkLocPositionReportInd::pTechnologyMask

8.771.2.25 **ULONG*** QmiCbkLocPositionReportInd::pTimeSrc

8.771.2.26 **ULONGLONG*** QmiCbkLocPositionReportInd::pTimestampUtc

8.771.2.27 **ULONG*** QmiCbkLocPositionReportInd::pTimeUnc

8.771.2.28 **BYTE*** QmiCbkLocPositionReportInd::pVertConfidence

8.771.2.29 **ULONG*** QmiCbkLocPositionReportInd::pVertReliability

8.771.2.30 **ULONG*** QmiCbkLocPositionReportInd::pVertUnc

8.771.2.31 **BYTE** QmiCbkLocPositionReportInd::sessionId

8.771.2.32 **ULONG** QmiCbkLocPositionReportInd::sessionStatus

8.772 QmiCbkLocSensorStreamingInd Struct Reference

Data Fields

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

8.772.1 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

8.772.2 Field Documentation

8.772.2.1 accelAcceptReady* QmiCbkLocSensorStreamingInd::pAccelAcceptReady

8.772.2.2 accelTempAcceptReady* QmiCbkLocSensorStreamingInd::pAccelTempAcceptReady

8.772.2.3 gyroAcceptReady* QmiCbkLocSensorStreamingInd::pGyroAcceptReady

8.772.2.4 gyroTempAcceptReady* QmiCbkLocSensorStreamingInd::pGyroTempAcceptReady

8.773 QmiCbkLocSetExtPowerConfigInd Struct Reference

Data Fields

- ULONG status

8.773.1 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

8.773.2 Field Documentation

8.773.2.1 ULONG QmiCbkLocSetExtPowerConfigInd::status

8.774 QmiCbkNasLTECphyCaInfo Struct Reference

Data Fields

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

8.774.1 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.

8.774.2 Field Documentation

8.774.2.1 PhyCaAggPcellInfo QmiCbkNasLTEphyCaInfo::sPhyCaAggPcellInfo

8.774.2.2 PhyCaAggScellDlBw QmiCbkNasLTEphyCaInfo::sPhyCaAggScellDlBw

8.774.2.3 PhyCaAggScellIndex QmiCbkNasLTEphyCaInfo::sPhyCaAggScellIndex

8.774.2.4 PhyCaAggScellIndType QmiCbkNasLTEphyCaInfo::sPhyCaAggScellIndType

8.774.2.5 PhyCaAggScellInfo QmiCbkNasLTEphyCaInfo::sPhyCaAggScellInfo

8.775 QmiCbkSwiOmaDmEventStatusReportInd Struct Reference

Data Fields

- struct sessionInfoTlv SITlv

8.775.1 Field Documentation

8.775.1.1 struct sessionInfoTlv QmiCbkSwiOmaDmEventStatusReportInd::SITlv

8.776 QmiCbkSwiOmaDmEventStatusReportIndExt Struct Reference

Data Fields

- struct sessionInfoTlvExt SITlv

8.776.1 Field Documentation

8.776.1.1 struct sessionInfoTlvExt QmiCbkSwiOmaDmEventStatusReportIndExt::SITlv

8.777 QmiCbkTmdMitiLvlRptInd Struct Reference

Data Fields

- _MitigationDevInfo MitigationDevInfo
- BYTE currentMitigationLvl

8.777.1 Detailed Description

This structure contains LOC Cradle Mount Config Status

Parameters

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

8.777.2 Field Documentation

8.777.2.1 BYTE QmiCbkTmdMitiLvlRptInd::currentMitigationLvl

8.777.2.2 _MitigationDevInfo QmiCbkTmdMitiLvlRptInd::MitigationDevInfo

8.778 QmiCbkWdsStatisticsIndState Struct Reference

Data Fields

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

8.778.1 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.

8.778.2 Field Documentation

8.778.2.1 DataULongTlv QmiCbkWdsStatisticsIndState::RxDropConutTlv

8.778.2.2 DataULongLongTlv QmiCbkWdsStatisticsIndState::RxOkByteCountTlv

8.778.2.3 DataULongTlv QmiCbkWdsStatisticsIndState::RxOkConutTlv

8.778.2.4 DataULongTlv QmiCbkWdsStatisticsIndState::TxDropConutTlv

8.778.2.5 DataULongLongTlv QmiCbkWdsStatisticsIndState::TxOkByteCountTlv

8.778.2.6 DataULongTlv QmiCbkWdsStatisticsIndState::TxOkConutTlv

8.779 qmifwinfo_s Struct Reference

Data Fields

- union {
  struct fwinfo_s g
  struct slqsfwinfo_s s
  } dev

8.779.1 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

8.779.2 Field Documentation

8.779.2.1 union { ... } qmifwinfo_s::dev

8.779.2.2 struct fwinfo_s qmifwinfo_s::g

8.779.2.3 struct slqsfwinfo_s qmifwinfo_s::s

8.780 QmiNas3GppNetworkInfo Struct Reference

Data Fields

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

8.780.1 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

8.780.2 Field Documentation

8.780.2.1 CHAR QmiNas3GppNetworkInfo::pDesription[255]

8.780.2.2 ULONG QmiNas3GppNetworkInfo::pForbidden

8.780.2.3 ULONG QmiNas3GppNetworkInfo::pInUse

8.780.2.4 WORD QmiNas3GppNetworkInfo::pMCC

8.780.2.5 WORD QmiNas3GppNetworkInfo::pMNC

8.780.2.6 ULONG QmiNas3GppNetworkInfo::pPreferred

8.780.2.7 ULONG QmiNas3GppNetworkInfo::pRoaming

8.781 QmiNasGetRFBandInfoResp Struct Reference

Data Fields

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

8.781.1 Field Documentation

8.781.1.1 BYTE* QmiNasGetRFBandInfoResp::pInstancesSize

8.781.1.2 struct RFBandInfoElements* QmiNasGetRFBandInfoResp::pRFBandInfoElements

8.781.1.3 struct qmTlvResult QmiNasGetRFBandInfoResp::results

8.782 QmiNasPerformNetworkScanResp Struct Reference

Data Fields

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

8.782.1 Field Documentation

8.782.1.1 struct QmiNas3GppNetworkInfo* QmiNasPerformNetworkScanResp::pInstances

8.782.1.2 BYTE* QmiNasPerformNetworkScanResp::pInstanceSize

8.782.1.3 struct qmTlvResult QmiNasPerformNetworkScanResp::results

8.783 qmiSmsMessageList Struct Reference

Data Fields

- uint32_t messageIndex
- uint32_t messageTag

8.783.1 Detailed Description

This structure contains SMS message list.

Parameters

messageIndex	• Message index of each matched message
messageTag	• Messagetag

8.783.2 Field Documentation

8.783.2.1 uint32_t qmiSmsMessageList::messageIndex

8.783.2.2 uint32_t qmiSmsMessageList::messageTag

8.784 qmiWDSDataBearerTechnology Struct Reference

Data Fields

- uint8_t currentNetwork
- uint32_t ratMask
- uint32_t soMask

8.784.1 Detailed Description

Structure to hold the current data bearer technology values

Parameters

currentNetwork[-OUT]	• current selected network – 0 - UNKNOWN – 1 - 3GPP2 – 2 - 3GPP – 0xff - Invalid data.

ratMask[OUT]	• Radio Access Technology (RAT) mask to indicate the type of technology (RAT mask value of zero indicates that this field is ignored) – 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+ – 0xffffffff - Invalid data.
soMask[OUT]	• Service Option (SO) mask to indicate the SO or type of application (SO mask value of zero indicates that this field is ignored) – 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 – 0xffffffff - Invalid data.

8.784.2 Field Documentation

8.784.2.1 uint8_t qmiWDSDataBearerTechnology::currentNetwork

8.784.2.2 uint32_t qmiWDSDataBearerTechnology::ratMask

8.784.2.3 uint32_t qmiWDSDataBearerTechnology::soMask

8.785 QmiWdsIpAddressInfo Struct Reference

Data Fields

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

8.785.1 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

8.785.2 Field Documentation

8.785.2.1 ULONG* QmiWdsIpAddressInfo::pIPAddressV4

8.785.2.2 USHORT* QmiWdsIpAddressInfo::pIPAddressV6

8.785.2.3 BYTE* QmiWdsIpAddressInfo::pIPv6prefixlen

8.786 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

8.786.1 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
pIPV6Address-Info	• IPV6 Address Information
pIPV6GW-AddressInfo	• IPV6 Gateway Address Information

8.786.2 Field Documentation

8.786.2.1 CHAR* qmiWdsRunTimeSettings::pAPNName

8.786.2.2 ULONG* qmiWdsRunTimeSettings::pAuthentication

8.786.2.3 struct DomainNameList* qmiWdsRunTimeSettings::pDomainList

8.786.2.4 struct GPRSQoS* qmiWdsRunTimeSettings::pGPRSGrantedQoS

8.786.2.5 ULONG* qmiWdsRunTimeSettings::pGWAddressV4

8.786.2.6 BYTE* qmiWdsRunTimeSettings::pIMCNflag

8.786.2.7 ULONG* qmiWdsRunTimeSettings::pIPAddressV4

8.786.2.8 BYTE* qmiWdsRunTimeSettings::pIPFamilyPreference

8.786.2.9 struct IPV6AddressInfo* qmiWdsRunTimeSettings::pIPV6AddrInfo

8.786.2.10 struct IPV6GWAddressInfo* qmiWdsRunTimeSettings::pIPV6GWAddrInfo

8.786.2.11 ULONG* qmiWdsRunTimeSettings::pMtu

8.786.2.12 BYTE* qmiWdsRunTimeSettings::pPCSCFAddrPCO

8.786.2.13 struct PCSCFFQDNAddressList* qmiWdsRunTimeSettings::pPCSCFFQDNAddrList

8.786.2.14 ULONG* qmiWdsRunTimeSettings::pPDPType

8.786.2.15 ULONG* qmiWdsRunTimeSettings::pPrimaryDNSV4

8.786.2.16 USHORT* qmiWdsRunTimeSettings::pPrimaryDNSV6

8.786.2.17 struct ProfileIdentifier* qmiWdsRunTimeSettings::pProfileID

8.786.2.18 CHAR* qmiWdsRunTimeSettings::pProfileName

8.786.2.19 ULONG* qmiWdsRunTimeSettings::pSecondaryDNSV4

8.786.2.20 USHORT* qmiWdsRunTimeSettings::pSecondaryDNSV6

8.786.2.21 struct PCSCFIPv4ServerAddressList* qmiWdsRunTimeSettings::pServerAddrList

8.786.2.22 ULONG* qmiWdsRunTimeSettings::pSubnetMaskV4

8.786.2.23 WORD* qmiWdsRunTimeSettings::pTechnology

8.786.2.24 struct UMTSQoS* qmiWdsRunTimeSettings::pUMTSGrantedQoS

8.786.2.25 CHAR* qmiWdsRunTimeSettings::pUsername

8.787 qmuxIPTable Struct Reference

Data Fields

- uint8_t muxID
- uint32_t ipAddress

8.787.1 Field Documentation

8.787.1.1 uint32_t qmuxIPTable::ipAddress

8.787.1.2 uint8_t qmuxIPTable::muxID

8.788 QosClassID Struct Reference

Data Fields

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

8.788.1 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

8.788.2 Field Documentation

8.788.2.1 ULONG QosClassID::gDlBitRate

8.788.2.2 ULONG QosClassID::gUlBitRate

8.788.2.3 ULONG QosClassID::maxDlBitRate

8.788.2.4 ULONG QosClassID::maxUlBitRate

8.788.2.5 BYTE QosClassID::QCI

8.789 QosEventInfo Struct Reference

Data Fields

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

8.789.1 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)

8.789.2 Field Documentation

8.789.2.1 ULONG* QosEventInfo::pDataBearer

8.789.2.2 ULONG* QosEventInfo::pPacketsCountRX

8.789.2.3 ULONG* QosEventInfo::pPacketsCountTX

8.789.2.4 ULONGLONG* QosEventInfo::pTotalBytesRX

8.789.2.5 ULONGLONG* QosEventInfo::pTotalBytesTX

8.790 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

8.790.1 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.

8.790.2 Field Documentation

8.790.2.1 BYTE* QosFlowInfo::pBearerID

8.790.2.2 QosFlowInfoState* QosFlowInfo::pQFlowState

8.790.2.3 swiQosFilter* QosFlowInfo::pRxQFilter[MAX_QOS_FILTER_TLV]

8.790.2.4 swiQosFlow* QosFlowInfo::pRxQFlowGranted

8.790.2.5 swiQosFilter* QosFlowInfo::pTxQFilter[MAX_QOS_FILTER_TLV]

8.790.2.6 swiQosFlow* QosFlowInfo::pTxQFlowGranted

8.791 QosFlowInfoState Struct Reference

Data Fields

- ULONG id
- BYTE isNewFlow
- BYTE state

8.791.1 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

8.791.2 Field Documentation

8.791.2.1 ULONG QosFlowInfoState::id

8.791.2.2 BYTE QosFlowInfoState::isNewFlow

8.791.2.3 BYTE QosFlowInfoState::state

8.792 QosMap Struct Reference

Data Fields

- BYTE dscp
- ULONG qos_id
- BYTE state

8.792.1 Detailed Description

This structure contains the SLQSQosDumpMap Information

Parameters

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

8.792.2 Field Documentation

8.792.2.1 BYTE QosMap::dscp

8.792.2.2 ULONG QosMap::qos_id

8.792.2.3 BYTE QosMap::state

8.793 RankIndicatorInd Struct Reference

Data Fields

- WORD Count1
- WORD Count2

8.793.1 Field Documentation

8.793.1.1 WORD RankIndicatorInd::Count1

8.793.1.2 WORD RankIndicatorInd::Count2

8.794 readResult Struct Reference

Data Fields

- WORD contentLen
- BYTE content [255+1]

8.794.1 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.

8.794.2 Field Documentation

8.794.2.1 BYTE readResult::content[255+1]

8.794.2.2 WORD readResult::contentLen

8.795 readTransparentInfo Struct Reference

Data Fields

- WORD offset
- WORD length

8.795.1 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.

8.795.2 Field Documentation

8.795.2.1 WORD readTransparentInfo::length

8.795.2.2 WORD readTransparentInfo::offset

8.796 redirNumInfo Struct Reference

Data Fields

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

8.796.1 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.

8.796.2 Field Documentation

8.796.2.1 BYTE redirNumInfo::number[255]

8.796.2.2 BYTE redirNumInfo::numLen

8.796.2.3 BYTE redirNumInfo::numPlan

8.796.2.4 BYTE redirNumInfo::numType

8.796.2.5 BYTE redirNumInfo::PI

8.796.2.6 BYTE redirNumInfo::reason

8.796.2.7 BYTE redirNumInfo::SI

8.797 registerRefresh Struct Reference

Data Fields

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

8.797.1 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

8.797.2 Field Documentation

8.797.2.1 fileInfo registerRefresh::arrfileInfo[255]

8.797.2.2 WORD registerRefresh::numFiles

8.797.2.3 BYTE registerRefresh::registerFlag

8.797.2.4 BYTE registerRefresh::voteForInit

8.798 remainingRetries Struct Reference

Data Fields

- BYTE verifyLeft
- BYTE unblockLeft

8.798.1 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.

8.798.2 Field Documentation

8.798.2.1 BYTE remainingRetries::unblockLeft

8.798.2.2 BYTE remainingRetries::verifyLeft

8.799 remotePartyName Struct Reference

Data Fields

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

8.799.1 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.

8.799.2 Field Documentation

8.799.2.1 BYTE remotePartyName::callerName[255]

8.799.2.2 BYTE remotePartyName::codingScheme

8.799.2.3 BYTE remotePartyName::nameLen

8.799.2.4 BYTE remotePartyName::namePI

8.800 remotePartyNum Struct Reference

Data Fields

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

8.800.1 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.

8.800.2 Field Documentation

8.800.2.1 BYTE remotePartyNum::numLen

8.800.2.2 BYTE remotePartyNum::presentationInd

8.800.2.3 BYTE remotePartyNum::remPartyNumber[81]

8.801 ReqFieldsList Struct Reference

Data Fields

- BYTE requestFieldsLen
- BYTE requestFields [256]

8.801.1 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.

8.801.2 Field Documentation

8.801.2.1 BYTE ReqFieldsList::requestFields[256]

8.801.2.2 BYTE ReqFieldsList::requestFieldsLen

8.802 RespFieldsList Struct Reference

Data Fields

- BYTE responseFieldsLen
- BYTE responseFields [256]

8.802.1 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.

8.802.2 Field Documentation

8.802.2.1 BYTE RespFieldsList::responseFields[256]

8.802.2.2 BYTE RespFieldsList::responseFieldsLen

8.803 RFBandInfoElements Struct Reference

Data Fields

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

8.803.1 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)

8.803.2 Field Documentation

8.803.2.1 WORD RFBandInfoElements::activeBandClass

8.803.2.2 uint16_t RFBandInfoElements::activeBandClass

8.803.2.3 WORD RFBandInfoElements::activeChannel

8.803.2.4 uint16_t RFBandInfoElements::activeChannel

8.803.2.5 BYTE RFBandInfoElements::radioInterface

8.803.2.6 uint8_t RFBandInfoElements::radioInterface

8.804 rmTrasnferStaticsReq Struct Reference

Data Fields

- uint8_t bResetStatistics
- uint32_t ulMask

8.804.1 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

8.804.2 Field Documentation

8.804.2.1 uint8_t rmTrasnferStaticsReq::bResetStatistics

8.804.2.2 uint32_t rmTrasnferStaticsReq::ulMask

8.805 roamIndList Struct Reference

Data Fields

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

8.805.1 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

8.805.2 Field Documentation

8.805.2.1 BYTE roamIndList::numInstances

8.805.2.2 BYTE roamIndList::radioInterface[0x0A]

8.805.2.3 BYTE roamIndList::roamIndicator[0x0A]

8.806 RoamingInfo Struct Reference

Data Fields

- BYTE TlvPresent
- BYTE roaming_ind

8.806.1 Field Documentation

8.806.1.1 BYTE RoamingInfo::roaming_ind

8.806.1.2 BYTE RoamingInfo::TlvPresent

8.807 roamTimer Struct Reference

Data Fields

- BYTE namID
- ULONG roamTimerValue

8.807.1 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.

8.807.2 Field Documentation

8.807.2.1 BYTE roamTimer::namID

8.807.2.2 ULONG roamTimer::roamTimerValue

8.808 RSRPThresh Struct Reference

Data Fields

- BYTE RSRPThresListLen
- SHORT * pRSRPThresList

8.808.1 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

8.808.2 Field Documentation

8.808.2.1 SHORT* RSRPThresh::pRSRPThresList

8.808.2.2 BYTE RSRPThresh::RSRPThresListLen

8.809 rsrqInformation Struct Reference

Data Fields

- INT8 rsrq
- BYTE radioIf

8.809.1 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

8.809.2 Field Documentation

8.809.2.1 BYTE rsrqInformation::radioIf

8.809.2.2 INT8 rsrqInformation::rsrq

8.810 RSRQThresh Struct Reference

Data Fields

- BYTE RSRQThresListLen
- INT8 * pRSRQThresList

8.810.1 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

8.810.2 Field Documentation

8.810.2.1 INT8* RSRQThresh::pRSRQThresList

8.810.2.2 BYTE RSRQThresh::RSRQThresListLen

8.811 RSSIThresh Struct Reference

Data Fields

- BYTE RSSIThresListLen
- INT8 * pRSSIThresList

8.811.1 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.

8.811.2 Field Documentation

8.811.2.1 INT8* RSSIThresh::pRSSIThresList

8.811.2.2 BYTE RSSIThresh::RSSIThresListLen

8.812 RXAGCList Struct Reference

Data Fields

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

8.812.1 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

8.812.2 Field Documentation

8.812.2.1 WORD* RXAGCList::pRXAIG

8.812.2.2 WORD* RXAGCList::pRXComprSlope

8.812.2.3 WORD* RXAGCList::pRXComprThres

8.812.2.4 WORD* RXAGCList::pRXExpSlope

8.812.2.5 WORD* RXAGCList::pRXExpThres

8.812.2.6 WORD* RXAGCList::pRXStaticGain

8.813 RXAVCList Struct Reference

Data Fields

- WORD * pAVRXAVCSens
- WORD * pAVRXAVCHeadroom

8.813.1 Detailed Description

This structure contains the SLQSGetAudioPathConfig parameters related to AV_RXAVCLIST.

Parameters

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

8.813.2 Field Documentation

8.813.2.1 WORD* RXAVCList::pAVRXAVCHeadroom

8.813.2.2 WORD* RXAVCList::pAVRXAVCSens

8.814 rxInfo Struct Reference

Data Fields

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

8.814.1 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.

8.814.2 Field Documentation

8.814.2.1 INT32 rxInfo::ecio

8.814.2.2 BYTE rxInfo::isRadioTuned

8.814.2.3 ULONG rxInfo::phase

8.814.2.4 INT32 rxInfo::rscp

8.814.2.5 INT32 rxInfo::rsrp

8.814.2.6 INT32 rxInfo::rxPower

8.815 RXPCMIIRFltr Struct Reference

Data Fields

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

8.815.1 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

8.815.2 Field Documentation

8.815.2.1 WORD* RXPCMIIRFItr::pFlag

8.815.2.2 BYTE* RXPCMIIRFItr::pStage0Val

8.815.2.3 BYTE* RXPCMIIRFItr::pStage1Val

8.815.2.4 BYTE* RXPCMIIRFItr::pStage2Val

8.815.2.5 BYTE* RXPCMIIRFItr::pStage3Val

8.815.2.6 BYTE* RXPCMIIRFItr::pStage4Val

8.815.2.7 WORD* RXPCMIIRFItr::pStageCnt

8.816 RxSigInfo Struct Reference

Data Fields

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

8.816.1 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

8.816.2 Field Documentation

8.816.2.1 BYTE RxSigInfo::isRadioTuned

8.816.2.2 INT32 RxSigInfo::rsrp

8.816.2.3 BYTE RxSigInfo::rxChainIndex

8.816.2.4 INT32 RxSigInfo::rxPower

8.817 rxSignalStrengthListElement Struct Reference

Data Fields

- SHORT rxSignalStrength
- BYTE radioIf

8.817.1 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.

8.817.2 Field Documentation

8.817.2.1 BYTE rxSignalStrengthListElement::radioIf

8.817.2.2 SHORT rxSignalStrengthListElement::rxSignalStrength

8.818 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

8.818.1 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

8.818.2 Field Documentation

8.818.2.1 BYTE sApnExtraParams::ambr_dl

8.818.2.2 BYTE sApnExtraParams::ambr_dl_ext

8.818.2.3 BYTE sApnExtraParams::ambr_dl_ext2

8.818.2.4 BYTE sApnExtraParams::ambr_ul

8.818.2.5 BYTE sApnExtraParams::ambr_ul_ext

8.818.2.6 BYTE sApnExtraParams::ambr_ul_ext2

8.818.2.7 ULONG sApnExtraParams::apnId

8.819 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

8.819.1 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

8.819.2 Field Documentation

8.819.2.1 FLOAT satelliteInfo::azimuth

8.819.2.2 FLOAT satelliteInfo::elevation

8.819.2.3 WORD satelliteInfo::gnssSvId

8.819.2.4 BYTE satelliteInfo::healthStatus

8.819.2.5 FLOAT satelliteInfo::snr

8.819.2.6 BYTE satelliteInfo::svInfoMask

8.819.2.7 BYTE satelliteInfo::svListLen

8.819.2.8 ULONG satelliteInfo::svStatus

8.819.2.9 ULONG satelliteInfo::system

8.819.2.10 ULONG satelliteInfo::validMask

8.820 SccRxInfo Struct Reference

Data Fields

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

8.820.1 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.

8.820.2 Field Documentation

8.820.2.1 BYTE SccRxInfo::numInstances

8.820.2.2 INT32 SccRxInfo::rsrq

8.820.2.3 RxSigInfo SccRxInfo::sigInfo[255]

8.820.2.4 SHORT SccRxInfo::snr

8.820.2.5 BYTE SccRxInfo::TlvPresent

8.821 sensorData Struct Reference

Data Fields

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

8.821.1 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

8.821.2 Field Documentation

8.821.2.1 BYTE sensorData::flags

8.821.2.2 BYTE sensorData::sensorDataLen

8.821.2.3 ULONG sensorData::timeOfFirstSample

8.821.2.4 WORD sensorData::timeOffset[64]

8.821.2.5 ULONG sensorData::xAxis[64]

8.821.2.6 ULONG sensorData::yAxis[64]

8.821.2.7 ULONG sensorData::zAxis[64]

8.822 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]

8.822.1 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

8.822.2 Field Documentation

8.822.2.1 uint8_t sensorData_t::flags

8.822.2.2 uint8_t sensorData_t::sensorDataLen

8.822.2.3 uint32_t sensorData_t::timeOfFirstSample

8.822.2.4 uint16_t sensorData_t::timeOffset[64]

8.822.2.5 uint32_t sensorData_t::xAxis[64]

8.822.2.6 uint32_t sensorData_t::yAxis[64]

8.822.2.7 uint32_t sensorData_t::zAxis[64]

8.823 sensorDataUsage_s Struct Reference

Data Fields

- ULONG usageMask
- ULONG aidingIndicatorMask

8.823.1 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

8.823.2 Field Documentation

8.823.2.1 ULONG sensorDataUsage_s::aidingIndicatorMask

8.823.2.2 ULONG sensorDataUsage_s::usageMask

8.824 serialNumbersInfo Struct Reference

Data Fields

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

8.824.1 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.

8.824.2 Field Documentation

8.824.2.1 BYTE serialNumbersInfo::esnSize

8.824.2.2 BYTE serialNumbersInfo::imeiSize

8.824.2.3 BYTE serialNumbersInfo::imeiSvnSize

8.824.2.4 BYTE serialNumbersInfo::meidSize

8.824.2.5 CHAR* serialNumbersInfo::pESNString

8.824.2.6 CHAR* serialNumbersInfo::pIMEIString

8.824.2.7 CHAR* serialNumbersInfo::pImeiSvnString

8.824.2.8 CHAR* serialNumbersInfo::pMEIDString

8.825 serviceProviderName Struct Reference

Data Fields

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

8.825.1 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.

8.825.2 Field Documentation

8.825.2.1 BYTE serviceProviderName::displayCondition

8.825.2.2 BYTE serviceProviderName::spn[255]

8.825.2.3 BYTE serviceProviderName::spnLength

8.826 ServingSystemInfo Struct Reference

Data Fields

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

8.826.1 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

8.826.2 Field Documentation

8.826.2.1 BYTE ServingSystemInfo::csAttachState

8.826.2.2 BYTE ServingSystemInfo::hdrPersonality

8.826.2.3 BYTE ServingSystemInfo::psAttachState

8.826.2.4 BYTE ServingSystemInfo::radioInterfaceList[255]

8.826.2.5 BYTE ServingSystemInfo::radioInterfaceNo

8.826.2.6 BYTE ServingSystemInfo::registrationState

8.826.2.7 BYTE ServingSystemInfo::selectedNetwork

8.827 servSystem Struct Reference

Data Fields

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

8.827.1 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

8.827.2 Field Documentation

8.827.2.1 BYTE servSystem::csAttachState

8.827.2.2 BYTE servSystem::numRadioInterfaces

8.827.2.3 BYTE servSystem::psAttachState

8.827.2.4 BYTE servSystem::radioInterface[0x0A]

8.827.2.5 BYTE servSystem::regState

8.827.2.6 BYTE servSystem::selNetwork

8.828 sessionInfo Union Reference

Data Fields

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

8.828.1 Detailed Description

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

8.828.2 Field Documentation

8.828.2.1 struct omaDmConfigTlv sessionInfo::omaDmConfig

8.828.2.2 struct omaDmFotaTlv sessionInfo::omaDmFota

8.828.2.3 struct omaDmNotificationsTlv sessionInfo::omaDmNotifications

8.829 sessionInfoExt Union Reference

Data Fields

- struct omaDmFotaTlvExt omaDmFota
- struct omaDmConfigTlvExt omaDmConfig

8.829.1 Detailed Description

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

8.829.2 Field Documentation

8.829.2.1 struct omaDmConfigTlvExt sessionInfoExt::omaDmConfig

8.829.2.2 struct omaDmFotaTlvExt sessionInfoExt::omaDmFota

8.830 sessionInfoTlv Struct Reference

Data Fields

- BYTE TlvPresent
- ULONG sessionType
- sessionInformation sessionInfo

8.830.1 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

8.830.2 Field Documentation

8.830.2.1 sessionInformation sessionInfoTlv::sessionInfo

8.830.2.2 ULONG sessionInfoTlv::sessionType

8.830.2.3 BYTE sessionInfoTlv::TlvPresent

8.831 sessionInfoTlvExt Struct Reference

Data Fields

- BYTE TlvPresent
- ULONG sessionType
- sessionInformationExt sessionInfo

8.831.1 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

8.831.2 Field Documentation

8.831.2.1 sessionInformationExt sessionInfoTlvExt::sessionInfo

8.831.2.2 ULONG sessionInfoTlvExt::sessionType

8.831.2.3 BYTE sessionInfoTlvExt::TlvPresent

8.832 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

8.832.1 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

8.832.2 Field Documentation

8.832.2.1 WORD* SetAudioPathConfigReq::pCodecSTGain

8.832.2.2 WORD* SetAudioPathConfigReq::pDTMFTXGain

8.832.2.3 BYTE* SetAudioPathConfigReq::pECMode

8.832.2.4 BYTE* SetAudioPathConfigReq::pNSEnable

8.832.2.5 BYTE SetAudioPathConfigReq::Profile

8.832.2.6 RXAGCList* SetAudioPathConfigReq::pRXAGCList

8.832.2.7 BYTE* SetAudioPathConfigReq::pRXAVCAGCSwitch

8.832.2.8 RXAVCList* SetAudioPathConfigReq::pRXAVCList

8.832.2.9 RXPCMIIRFltr* SetAudioPathConfigReq::pRXPCMIIRFltr

8.832.2.10 TXAGCList* SetAudioPathConfigReq::pTXAGCList

8.832.2.11 BYTE* SetAudioPathConfigReq::pTXAVCSwitch

8.832.2.12 WORD* SetAudioPathConfigReq::pTXGain

8.832.2.13 TXPCMIIRFltr* SetAudioPathConfigReq::pTXPCMIIRFltr

8.833 SetAudioProfileReq Struct Reference

Data Fields

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

8.833.1 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

8.833.2 Field Documentation

8.833.2.1 BYTE SetAudioProfileReq::EarMute

8.833.2.2 BYTE SetAudioProfileReq::Generator

8.833.2.3 BYTE SetAudioProfileReq::MicMute

8.833.2.4 BYTE SetAudioProfileReq::Profile

8.833.2.5 BYTE SetAudioProfileReq::Volume

8.834 SetAudioVolTLBConfigReq Struct Reference

Data Fields

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

8.834.1 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

8.834.2 Field Documentation

8.834.2.1 BYTE SetAudioVolTLBConfigReq::Generator

8.834.2.2 BYTE SetAudioVolTLBConfigReq::Item

8.834.2.3 BYTE SetAudioVolTLBConfigReq::Profile

8.834.2.4 BYTE SetAudioVolTLBConfigReq::Volume

8.834.2.5 WORD SetAudioVolTLBConfigReq::VolValue

8.835 SetAudioVolTLBConfigResp Struct Reference

Data Fields

- WORD ResCode

8.835.1 Detailed Description

This structure contains the SLQSSetAudioVolTLBConfig response parameters.

Parameters

ResCode	• Result of requested item

8.835.2 Field Documentation

8.835.2.1 WORD SetAudioVolTLBConfigResp::ResCode

8.836 setCustomSettingV2 Struct Reference

Data Fields

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

8.836.1 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)

8.836.2 Field Documentation

8.836.2.1 CHAR setCustomSettingV2::cust_id[64+1]

8.836.2.2 BYTE setCustomSettingV2::cust_value[8+1]

8.836.2.3 WORD setCustomSettingV2::value_length

8.837 setDyingGaspCfg Struct Reference

Data Fields

- BYTE * pDestSMSNum
- BYTE * pDestSMSContent

8.837.1 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.

8.837.2 Field Documentation

8.837.2.1 BYTE* setDyingGaspCfg::pDestSMSContent

8.837.2.2 BYTE* setDyingGaspCfg::pDestSMSNum

8.838 SetIMSSMSConfigReq Struct Reference

Data Fields

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

8.838.1 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

8.838.2 Field Documentation

8.838.2.1 BYTE* SetIMSSMSConfigReq::pPhoneCtxtURI

8.838.2.2 BYTE* SetIMSSMSConfigReq::pPhoneCtxtURILen

8.838.2.3 BYTE* SetIMSSMSConfigReq::pSMSFormat

8.838.2.4 BYTE* SetIMSSMSConfigReq::pSMSOverIPNwInd

8.839 SetIMSSMSConfigResp Struct Reference

Data Fields

- BYTE * pSettingResp

8.839.1 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

8.839.2 Field Documentation

8.839.2.1 BYTE* SetIMSSMSConfigResp::pSettingResp

8.840 SetIMSUserConfigReq Struct Reference

Data Fields

- BYTE * pIMSDomainLen
- BYTE * pIMSDomain

8.840.1 Detailed Description

This structure contains the SLQSSetIMSUserConfig request parameters.

Parameters

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

8.840.2 Field Documentation

8.840.2.1 BYTE* SetIMSUserConfigReq::pIMSDomain

8.840.2.2 BYTE* SetIMSUserConfigReq::pIMSDomainLen

8.841 SetIMSUserConfigResp Struct Reference

Data Fields

- BYTE * pSettingResp

8.841.1 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

8.841.2 Field Documentation

8.841.2.1 BYTE* SetIMSUserConfigResp::pSettingResp

8.842 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

8.842.1 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.

8.842.2 Field Documentation

8.842.2.1 BYTE* SetIMSVoIPConfigReq::pAmrMode

8.842.2.2 BYTE* SetIMSVoIPConfigReq::pAmrOctetAligned

8.842.2.3 BYTE* SetIMSVoIPConfigReq::pAmrWbEnable

8.842.2.4 WORD* SetIMSVoIPConfigReq::pAmrWBMode

8.842.2.5 BYTE* SetIMSVoIPConfigReq::pAmrWBOctetAligned

8.842.2.6 WORD* SetIMSVoIPConfigReq::pMinSessionExpiryTimer

8.842.2.7 WORD* SetIMSVoIPConfigReq::pRingBackTimer

8.842.2.8 WORD* SetIMSVoIPConfigReq::pRingingTimer

8.842.2.9 WORD* SetIMSVoIPConfigReq::pRTPRTCPInactTimer

8.842.2.10 BYTE* SetIMSVoIPConfigReq::pScrAmrEnable

8.842.2.11 BYTE* SetIMSVoIPConfigReq::pScrAmrWbEnable

8.842.2.12 WORD* SetIMSVoIPConfigReq::pSessionExpiryTimer

8.843 SetIMSVoIPConfigResp Struct Reference

Data Fields

- BYTE * pSettingResp

8.843.1 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

8.843.2 Field Documentation

8.843.2.1 BYTE* SetIMSVoIPConfigResp::pSettingResp

8.844 SetM2MAudioAVCFGReq Struct Reference

Data Fields

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

8.844.1 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

8.844.2 Field Documentation

8.844.2.1 BYTE SetM2MAudioAVCFGReq::Device

8.844.2.2 BYTE SetM2MAudioAVCFGReq::PIFACEId

8.844.2.3 PCMparams* SetM2MAudioAVCFGReq::pPCMParams

8.844.2.4 BYTE SetM2MAudioAVCFGReq::Profile

8.845 SetM2MAudioLPBKReq Struct Reference

Data Fields

- BYTE Enable

8.845.1 Detailed Description

This structure contains the SLQSSetM2MAudioLPBK request parameters.

Parameters

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

8.845.2 Field Documentation

8.845.2.1 BYTE SetM2MAudioLPBKReq::Enable

8.846 SetM2MAudioProfileReq Struct Reference

Data Fields

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

8.846.1 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

8.846.2 Field Documentation

8.846.2.1 BYTE* SetM2MAudioProfileReq::pCwtMute

8.846.2.2 BYTE* SetM2MAudioProfileReq::pEarMute

8.846.2.3 BYTE* SetM2MAudioProfileReq::pGenerator

8.846.2.4 BYTE* SetM2MAudioProfileReq::pMicMute

8.846.2.5 BYTE SetM2MAudioProfileReq::Profile

8.846.2.6 BYTE* SetM2MAudioProfileReq::pVolume

8.847 SetM2MAudioVolumeReq Struct Reference

Data Fields

- BYTE Profile
- BYTE Generator
- BYTE Level

8.847.1 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

8.847.2 Field Documentation

8.847.2.1 BYTE SetM2MAudioVolumeReq::Generator

8.847.2.2 BYTE SetM2MAudioVolumeReq::Level

8.847.2.3 BYTE SetM2MAudioVolumeReq::Profile

8.848 SetM2MAVMuteReq Struct Reference

Data Fields

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

8.848.1 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

8.848.2 Field Documentation

8.848.2.1 BYTE SetM2MAVMuteReq::EarMute

8.848.2.2 BYTE SetM2MAVMuteReq::MicMute

8.848.2.3 BYTE* SetM2MAVMuteReq::pCwtMute

8.848.2.4 BYTE SetM2MAVMuteReq::Profile

8.849 SetM2MSpkrGainReq Struct Reference

Data Fields

- BYTE Profile
- WORD Value

8.849.1 Detailed Description

This structure contains the SLQSSetM2MSpkrGain request parameters.

Parameters

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

8.849.2 Field Documentation

8.849.2.1 BYTE SetM2MSpkrGainReq::Profile

8.849.2.2 WORD SetM2MSpkrGainReq::Value

8.850 setPINProtection Struct Reference

Data Fields

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

8.850.1 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.

8.850.2 Field Documentation

8.850.2.1 BYTE setPINProtection::pinID

8.850.2.2 BYTE setPINProtection::pinLength

8.850.2.3 BYTE setPINProtection::pinOperation

8.850.2.4 BYTE setPINProtection::pinValue[255]

8.851 SetRegMgrConfigReq Struct Reference

Data Fields

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

8.851.1 Detailed Description

This structure contains the SLQSSetRegMgrConfig request parameters.

Parameters

pPriCSCFPort	• Primary call session control function port
pCSCFPort-NameLen	• Length of the CSCF Port name parameter to follow

pCSCFPort-Name	• Call Session control port, fully qualified domain name • Length of this string must be specified in pCSCFPortNameLen parameter
pIMSTestMode	• IMS Test mode Enabled. – TRUE - Enable, no IMS registration – FALSE - Disable, IMS registration is initiated

8.851.2 Field Documentation

8.851.2.1 BYTE* SetRegMgrConfigReq::pCSCFPortName

8.851.2.2 BYTE* SetRegMgrConfigReq::pCSCFPortNameLen

8.851.2.3 BYTE* SetRegMgrConfigReq::pIMSTestMode

8.851.2.4 WORD* SetRegMgrConfigReq::pPriCSCFPort

8.852 SetRegMgrConfigResp Struct Reference

Data Fields

- BYTE * pSettingResp

8.852.1 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

8.852.2 Field Documentation

8.852.2.1 BYTE* SetRegMgrConfigResp::pSettingResp

8.853 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

8.853.1 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.

8.853.2 Field Documentation

8.853.2.1 WORD* setSignalStrengthInfo::pCDMAECIODelta

8.853.2.2 CDMAECIOThresh* setSignalStrengthInfo::pCDMAECIOThresh

8.853.2.3 WORD* setSignalStrengthInfo::pCDMARSSIDelta

8.853.2.4 CDMARSSIThresh* setSignalStrengthInfo::pCDMARSSIThresh

8.853.2.5 WORD* setSignalStrengthInfo::pGSMRSSIDelta

8.853.2.6 GSMRSSIThresh* setSignalStrengthInfo::pGSMRSSIThresh

8.853.2.7 WORD* setSignalStrengthInfo::pHDRECIODelta

8.853.2.8 HDRECIOThresh* setSignalStrengthInfo::pHDRECIOThresh

8.853.2.9 WORD* setSignalStrengthInfo::pHDRIODelta

8.853.2.10 HDRIOThresh* setSignalStrengthInfo::pHDRIOThresh

8.853.2.11 WORD* setSignalStrengthInfo::pHDRRSSIDelta

8.853.2.12 HDRRSSIThresh* setSignalStrengthInfo::pHDRRSSIThresh

8.853.2.13 WORD* setSignalStrengthInfo::pHDRSINRDelta

8.853.2.14 HDRSINRThreshold* setSignalStrengthInfo::pHDRSINRThresh

8.853.2.15 WORD* setSignalStrengthInfo::pLTERSRPDelta

8.853.2.16 LTERSRPThresh* setSignalStrengthInfo::pLTERSRPThresh

8.853.2.17 WORD* setSignalStrengthInfo::pLTERSRQDelta

8.853.2.18 LTERSRQThresh∗ setSignalStrengthInfo::pLTERSRQThresh

8.853.2.19 WORD∗ setSignalStrengthInfo::pLTERSSIDelta

8.853.2.20 LTERSSIThresh∗ setSignalStrengthInfo::pLTERSSIThresh

8.853.2.21 LTESigRptConfig∗ setSignalStrengthInfo::pLTESigRptConfig

8.853.2.22 WORD∗ setSignalStrengthInfo::pLTESNRDelta

8.853.2.23 LTESNRThreshold∗ setSignalStrengthInfo::pLTESNRThresh

8.853.2.24 ULONG∗ setSignalStrengthInfo::pTDSCDMAECIODelta

8.853.2.25 TDSCDMAECIOThresh∗ setSignalStrengthInfo::pTDSCDMAECIOThresh

8.853.2.26 WORD∗ setSignalStrengthInfo::pTDSCDMARSCPDelta

8.853.2.27 TDSCDMARSCPThresh∗ setSignalStrengthInfo::pTDSCDMARSCPThresh

8.853.2.28 ULONG∗ setSignalStrengthInfo::pTDSCDMARSSIDelta

8.853.2.29 TDSCDMARSSIThresh∗ setSignalStrengthInfo::pTDSCDMARSSIThresh

8.853.2.30 ULONG∗ setSignalStrengthInfo::pTDSCDMASINRDelta

8.853.2.31 TDSCDMASINRThresh∗ setSignalStrengthInfo::pTDSCDMASINRThresh

8.853.2.32 WORD∗ setSignalStrengthInfo::pWCDMAECIODelta

8.853.2.33 WCDMAECIOThresh∗ setSignalStrengthInfo::pWCDMAECIOThresh

8.853.2.34 WORD∗ setSignalStrengthInfo::pWCDMARSSIDelta

8.853.2.35 WCDMARSSIThresh∗ setSignalStrengthInfo::pWCDMARSSIThresh

8.854 SetSIPConfigReq Struct Reference

Data Fields

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

8.854.1 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

8.854.2 Field Documentation

8.854.2.1 BYTE* SetSIPConfigReq::pSigCompEnabled

8.854.2.2 WORD* SetSIPConfigReq::pSIPLocalPort

8.854.2.3 ULONG* SetSIPConfigReq::pSubscribeTimer

8.854.2.4 ULONG* SetSIPConfigReq::pTimerSIPReg

8.854.2.5 ULONG* SetSIPConfigReq::pTimerT1

8.854.2.6 ULONG* SetSIPConfigReq::pTimerT2

8.854.2.7 ULONG* SetSIPConfigReq::pTimerTf

8.855 SetSIPConfigResp Struct Reference

Data Fields

- BYTE * pSettingResp

8.855.1 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

8.855.2 Field Documentation

8.855.2.1 BYTE* SetSIPConfigResp::pSettingResp

8.856 sGetDeviceSeriesResult Struct Reference

Data Fields

- enum eGobiDeviceSeries eDevice
- ULONG uResult

8.856.1 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

8.856.2 Field Documentation

8.856.2.1 enum eGobiDeviceSeries sGetDeviceSeriesResult::eDevice

8.856.2.2 ULONG sGetDeviceSeriesResult::uResult

8.857 sidNid Struct Reference

Data Fields

- WORD nid
- WORD sid

8.857.1 Detailed Description

This structure contains the parameters for SidNid Information

Parameters

nid	• Network ID
sid	• System ID

8.857.2 Field Documentation

8.857.2.1 WORD sidNid::nid

8.857.2.2 WORD sidNid::sid

8.858 sigInfo Struct Reference

Data Fields

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

8.858.1 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

8.858.2 Field Documentation

8.858.2.1 ECIOThresh∗ sigInfo::pECIOThresh

8.858.2.2 HDRSINRThresh∗ sigInfo::pHDRSINRThresh

8.858.2.3 IOThresh∗ sigInfo::pIOThresh

8.858.2.4 LTESigRptCfg∗ sigInfo::pLTESigRptCfg

8.858.2.5 LTESNRThresh∗ sigInfo::pLTESNRThresh

8.858.2.6 RSRPThresh∗ sigInfo::pRSRPThresh

8.858.2.7 RSRQThresh∗ sigInfo::pRSRQThresh

8.858.2.8 RSSIThresh∗ sigInfo::pRSSIThresh

8.858.2.9 TDSCDMASINRCONFThresh∗ sigInfo::pTDSCDMASINRCONFThresh

8.859 signalInfo Struct Reference

Data Fields

- BYTE signalType
- BYTE alertPitch
- BYTE signal

8.859.1 Detailed Description

This structure contains Signal Information

Parameters

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

8.859.2 Field Documentation

8.859.2.1 BYTE signalInfo::alertPitch

8.859.2.2 BYTE signalInfo::signal

8.859.2.3 BYTE signalInfo::signalType

8.860 SignalStrengthDataType Struct Reference

Data Fields

- BYTE thresholdsSize
- INT8 thresholds [5]

8.860.1 Field Documentation

8.860.1.1 INT8 SignalStrengthDataType::thresholds[5]

8.860.1.2 BYTE SignalStrengthDataType::thresholdsSize

8.861 slot_t Struct Reference

Data Fields

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

8.861.1 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.

8.861.2 Field Documentation

8.861.2.1 uint8_t slot_t::bICCID[255]

8.861.2.2 uint8_t slot_t::bICCIDLength

8.861.2.3 uint8_t slot_t::bLogicalSlot

8.861.2.4 uint32_t slot_t::uPhyCardStatus

8.861.2.5 uint32_t slot_t::uPhySlotStatus

8.862 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]

8.862.1 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.

8.862.2 Field Documentation

8.862.2.1 appStats slotInf::AppStatus[10]

8.862.2.2 uint8_t slotInf::cardState

8.862.2.3 uint8_t slotInf::errorState

8.862.2.4 uint8_t slotInf::numApp

8.862.2.5 uint8_t slotInf::upinRetries

8.862.2.6 uint8_t slotInf::upinState

8.862.2.7 uint8_t slotInf::upukRetries

8.863 slotInfo Struct Reference

Data Fields

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

8.863.1 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_APPL-ICATIONS]	• See appStatus for more information.

8.863.2 Field Documentation

8.863.2.1 appStatus slotInfo::AppStatus[10]

8.863.2.2 BYTE slotInfo::cardState

8.863.2.3 BYTE slotInfo::errorState

8.863.2.4 BYTE slotInfo::numApp

8.863.2.5 BYTE slotInfo::upinRetries

8.863.2.6 BYTE slotInfo::upinState

8.863.2.7 BYTE slotInfo::upukRetries

8.864 slots_t Struct Reference

Data Fields

- slot_t uimSlotStatus [255]

8.864.1 Detailed Description

This structure contains slots status.

Parameters

uimSlotStatus	• See slot_t

8.864.2 Field Documentation

8.864.2.1 slot_t slots_t::uimSlotStatus[255]

8.865 slqsautoconnect Struct Reference

Data Fields

- BOOL action
- BYTE acsetting
- BYTE acroamsetting

8.865.1 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

8.865.2 Field Documentation

8.865.2.1 BYTE slqsautoconnect::acroamsetting

8.865.2.2 BYTE slqsautoconnect::acsetting

8.865.2.3 BOOL slqsautoconnect::action

8.866 SLQSDeleteProfileParams Struct Reference

Data Fields

- BYTE profileType

- BYTE profileIndex

8.866.1 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)

8.866.2 Field Documentation

8.866.2.1 BYTE SLQSDeleteProfileParams::profileIndex

8.866.2.2 BYTE SLQSDeleteProfileParams::profileType

8.867 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]

8.867.1 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

8.867.2 Field Documentation

8.867.2.1 CHAR slqsfwinfo_s::appversion_str[85]

8.867.2.2 CHAR slqsfwinfo_s::bootversion_str[85]

8.867.2.3 CHAR slqsfwinfo_s::carrier_str[20]

8.867.2.4 CHAR slqsfwinfo_s::cur_carr_name[17]

8.867.2.5 CHAR slqsfwinfo_s::cur_carr_rev[13]

8.867.2.6 CHAR slqsfwinfo_s::modelid_str[20]

8.867.2.7 CHAR slqsfwinfo_s::packageid_str[85]

8.867.2.8 CHAR slqsfwinfo_s::priversion_str[16]

8.867.2.9 CHAR slqsfwinfo_s::sku_str[15]

8.868 SlqsNas3GppNetworkInfo Struct Reference

Data Fields

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

8.868.1 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.

8.868.2 Field Documentation

8.868.2.1 CHAR SlqsNas3GppNetworkInfo::Description[255]

8.868.2.2 ULONG SlqsNas3GppNetworkInfo::Forbidden

8.868.2.3 ULONG SlqsNas3GppNetworkInfo::InUse

8.868.2.4 WORD SlqsNas3GppNetworkInfo::MCC

8.868.2.5 WORD SlqsNas3GppNetworkInfo::MNC

8.868.2.6 ULONG SlqsNas3GppNetworkInfo::Preferred

8.868.2.7 ULONG SlqsNas3GppNetworkInfo::Roaming

8.869 SlqsNasPcsDigit Struct Reference

Data Fields

- WORD MCC
- WORD MNC
- BYTE includes_pcs_digit

8.869.1 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

8.869.2 Field Documentation

8.869.2.1 BYTE SlqsNasPcsDigit::includes_pcs_digit

8.869.2.2 WORD SlqsNasPcsDigit::MCC

8.869.2.3 WORD SlqsNasPcsDigit::MNC

8.870 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

8.870.1 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.

8.870.2 Field Documentation

8.870.2.1 ULONG slqssendasyncsmsparams_s::messageFormat

8.870.2.2 ULONG slqssendasyncsmsparams_s::messageSize

8.870.2.3 BYTE* slqssendasyncsmsparams_s::pFollowOnDC

8.870.2.4 BYTE* slqssendasyncsmsparams_s::pForceOnDC

8.870.2.5 BYTE* slqssendasyncsmsparams_s::pLinktimer

8.870.2.6 BYTE* slqssendasyncsmsparams_s::pMessage

8.870.2.7 BYTE* slqssendasyncsmsparams_s::pRetryMessage

8.870.2.8 ULONG* slqssendasyncsmsparams_s::pRetryMessageId

8.870.2.9 BYTE* slqssendasyncsmsparams_s::pServiceOption

8.870.2.10 BYTE* slqssendasyncsmsparams_s::pSmsOnIms

8.870.2.11 ULONG* slqssendasyncsmsparams_s::pUserData

8.871 slqssendsmsparams_s Struct Reference

Data Fields

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

8.871.1 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

8.871.2 Field Documentation

8.871.2.1 ULONG slqssendsmsparams_s::messageFailureCode

8.871.2.2 ULONG slqssendsmsparams_s::messageFormat

8.871.2.3 USHORT slqssendsmsparams_s::messageID

8.871.2.4 ULONG slqssendsmsparams_s::messageSize

8.871.2.5 BYTE* slqssendsmsparams_s::pLinktimer

8.871.2.6 BYTE* slqssendsmsparams_s::pMessage

8.871.2.7 BYTE* slqssendsmsparams_s::pSmsOnIms

8.872 slqsSessionStateInfo Struct Reference

Data Fields

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

8.872.1 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

8.872.2 Field Documentation

8.872.2.1 qaQmiInterfaceInfo* slqsSessionStateInfo::pQmiInterfaceInfo

8.872.2.2 ULONG slqsSessionStateInfo::reconfiguration_required

8.872.2.3 ULONG slqsSessionStateInfo::sessionEndReason

8.872.2.4 ULONG slqsSessionStateInfo::state

8.873 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

8.873.1 Detailed Description

This structure contains the Signal Strength Information

Parameters

signalStrength-ReqMask[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
rxSignal-StrengthList-Len[OUT]	• Number of elements in Receive Signal Strength List
rxSignal-StrengthList[OU-T]	• See rxSignalStrengthListElement for more information
ecioListLen[OU-T]	• Number of elements in ECIO List

ecioList[OUT]	• See ecioListElement for more information
Io[OUT]	• Received Io in dBm; IO is only applicable for 1xEV-DO
sinr[OUT]	• SINR level – 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

8.873.2 Field Documentation

8.873.2.1 struct ecioListElement slqsSignalStrengthInfo::ecioList[18]

8.873.2.2 USHORT slqsSignalStrengthInfo::ecioListLen

8.873.2.3 struct errorRateListElement slqsSignalStrengthInfo::errorRateList[18]

8.873.2.4 USHORT slqsSignalStrengthInfo::errorRateListLen

8.873.2.5 INT32 slqsSignalStrengthInfo::Io

8.873.2.6 SHORT slqsSignalStrengthInfo::ltersrp

8.873.2.7 SHORT slqsSignalStrengthInfo::ltesnr

8.873.2.8 struct rsrqInformation slqsSignalStrengthInfo::rsrqInfo

8.873.2.9 struct rxSignalStrengthListElement slqsSignalStrengthInfo::rxSignalStrengthList[18]

8.873.2.10 USHORT slqsSignalStrengthInfo::rxSignalStrengthListLen

8.873.2.11 USHORT slqsSignalStrengthInfo::signalStrengthReqMask

8.873.2.12 BYTE slqsSignalStrengthInfo::sinr

8.874 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

8.874.1 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

8.874.2 Field Documentation

8.874.2.1 BYTE SLQSSignalStrengthsIndReq::ecioDelta

8.874.2.2 SHORT SLQSSignalStrengthsIndReq::ecioThresholdList[10]

8.874.2.3 BYTE SLQSSignalStrengthsIndReq::ecioThresholdListLen

8.874.2.4 BYTE SLQSSignalStrengthsIndReq::ioDelta

8.874.2.5 BYTE SLQSSignalStrengthsIndReq::lteRsrpDelta

8.874.2.6 WORD SLQSSignalStrengthsIndReq::lteSnrDelta

8.874.2.7 BYTE SLQSSignalStrengthsIndReq::rsrqDelta

8.874.2.8 BYTE SLQSSignalStrengthsIndReq::rxSignalStrengthDelta

8.874.2.9 BYTE SLQSSignalStrengthsIndReq::sinrDelta

8.874.2.10 BYTE SLQSSignalStrengthsIndReq::sinrThresholdList[5]

8.874.2.11 BYTE SLQSSignalStrengthsIndReq::sinrThresholdListLen

8.875 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

8.875.1 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

8.875.2 Field Documentation

8.875.2.1 struct ecioListElement SLQSSignalStrengthsInformation::ecioInfo

8.875.2.2 struct errorRateListElement SLQSSignalStrengthsInformation::errorRateInfo

8.875.2.3 ULONG SLQSSignalStrengthsInformation::io

8.875.2.4 struct lteRsrpinformation SLQSSignalStrengthsInformation::lteRsrpinfo

8.875.2.5 struct lteSnrinformation SLQSSignalStrengthsInformation::lteSnrinfo

8.875.2.6 struct rsrqInformation SLQSSignalStrengthsInformation::rsrqInfo

8.875.2.7 struct rxSignalStrengthListElement SLQSSignalStrengthsInformation::rxSignalStrengthInfo

8.875.2.8 BYTE SLQSSignalStrengthsInformation::sinr

8.876 sIqsWdsEventInfo Struct Reference

Data Fields

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

8.876.1 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)

8.876.2 Field Documentation

8.876.2.1 ULONG* slqsWdsEventInfo::pDataBearer

8.876.2.2 ULONG* slqsWdsEventInfo::pDormancyStatus

8.876.2.3 ULONG* slqsWdsEventInfo::pPacketsCountRX

8.876.2.4 ULONG* slqsWdsEventInfo::pPacketsCountTX

8.876.2.5 qaQmiInterfaceInfo* slqsWdsEventInfo::pQmiInterfaceInfo

8.876.2.6 ULONGLONG* slqsWdsEventInfo::pTotalBytesRX

8.876.2.7 ULONGLONG* slqsWdsEventInfo::pTotalBytesTX

8.877 sms_BroadcastConfig Struct Reference

Data Fields

- uint16_t fromServiceId
- uint16_t toServiceId
- uint8_t selected

8.877.1 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

8.877.2 Field Documentation

8.877.2.1 uint16_t sms_BroadcastConfig::fromServiceId

8.877.2.2 uint8_t sms_BroadcastConfig::selected

8.877.2.3 uint16_t sms_BroadcastConfig::toServiceId

8.878 sms_CDMABroadcastConfig Struct Reference

Data Fields

- uint16_t serviceCategory
- uint16_t language
- uint8_t selected

8.878.1 Detailed Description

This structure contains CDMABroadcastConfig parameters

Parameters

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

8.878.2 Field Documentation

8.878.2.1 uint16_t sms_CDMABroadcastConfig::language

8.878.2.2 uint8_t sms_CDMABroadcastConfig::selected

8.878.2.3 uint16_t sms_CDMABroadcastConfig::serviceCategory

8.879 sms_getIndicationReg Struct Reference

Data Fields

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

8.879.1 Detailed Description

This structure contains Get Indication Register Response parameters

Parameters

pRegTransLayerInfoEvt	- • 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.
pRegTransNWRegInfoEvt	- • 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.

8.879.2 Field Documentation

8.879.2.1 uint8_t* sms_getIndicationReg::pRegCallStatInfoEvt

8.879.2.2 uint8_t* sms_getIndicationReg::pRegTransLayerInfoEvt

8.879.2.3 uint8_t* sms_getIndicationReg::pRegTransNWRegInfoEvt

8.880 sms_getMsgWaitingInfo Struct Reference

Data Fields

- uint8_t numInstances
- sms_messageWaitingInfoContent msgWaitInfo [0xFF]

8.880.1 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.

8.880.2 Field Documentation

8.880.2.1 sms_messageWaitingInfoContent sms_getMsgWaitingInfo::msgWaitInfo[0xFF]

8.880.2.2 uint8_t sms_getMsgWaitingInfo::numInstances

8.881 sms_getTransLayerInfo Struct Reference

Data Fields

- uint8_t * pRegInd
- sms_transLayerInfo * pTransLayerInfo

8.881.1 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. • NULL pointer - Invalid data.
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. • NULL pointer - Invalid data.

8.881.2 Field Documentation

8.881.2.1 uint8_t* sms_getTransLayerInfo::pRegInd

8.881.2.2 sms_transLayerInfo* sms_getTransLayerInfo::pTransLayerInfo

8.882 sms_getTransNWRegInfo Struct Reference

Data Fields

- uint8_t * pRegStatus

8.882.1 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.

8.882.2 Field Documentation

8.882.2.1 uint8_t* sms_getTransNWRegInfo::pRegStatus

8.883 sms_maxStorageSizeReq Struct Reference

Data Fields

- uint8_t storageType
- uint8_t * pMessageMode

8.883.1 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

8.883.2 Field Documentation

8.883.2.1 uint8_t* sms_maxStorageSizeReq::pMessageMode

8.883.2.2 uint8_t sms_maxStorageSizeReq::storageType

8.884 sms_maxStorageSizeResp Struct Reference

Data Fields

- uint32_t maxStorageSize
- uint32_t freeSlots

8.884.1 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.

8.884.2 Field Documentation

8.884.2.1 uint32_t sms_maxStorageSizeResp::freeSlots

8.884.2.2 uint32_t sms_maxStorageSizeResp::maxStorageSize

8.885 sms_messageWaitingInfoContent Struct Reference

Data Fields

- uint8_t msgType
- uint8_t activeInd
- uint8_t msgCount

8.885.1 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

8.885.2 Field Documentation

8.885.2.1 uint8_t sms_messageWaitingInfoContent::activeInd

8.885.2.2 uint8_t sms_messageWaitingInfoContent::msgCount

8.885.2.3 uint8_t sms_messageWaitingInfoContent::msgType

8.886 sms_msgProtocolResp Struct Reference

Data Fields

- uint8_t msgProtocol

8.886.1 Detailed Description

This structure contains get message protocol response parameters

Parameters

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

8.886.2 Field Documentation

8.886.2.1 uint8_t sms_msgProtocolResp::msgProtocol

8.887 sms_qaQmi3GPP2BroadcastCfgInfo Struct Reference

Data Fields

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

8.887.1 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

8.887.2 Field Documentation

8.887.2.1 uint8_t sms_qaQmi3GPP2BroadcastCfgInfo::activated_ind

8.887.2.2 sms_CDMABroadcastConfig sms_qaQmi3GPP2BroadcastCfgInfo::CDMABroadcastConfig[0x05]

8.887.2.3 uint16_t sms_qaQmi3GPP2BroadcastCfgInfo::num_instances

8.888 sms_qaQmi3GPPBroadcastCfgInfo Struct Reference

Data Fields

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

8.888.1 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

8.888.2 Field Documentation

8.888.2.1 uint8_t sms_qaQmi3GPPBroadcastCfgInfo::activated_ind

8.888.2.2 sms_BroadcastConfig sms_qaQmi3GPPBroadcastCfgInfo::broadcastConfig[0x05]

8.888.2.3 uint16_t sms_qaQmi3GPPBroadcastCfgInfo::num_instances

8.889 sms_routeEntry Struct Reference

Data Fields

- uint8_t messageType
- uint8_t messageClass

- uint8_t routeStorage
- uint8_t receiptAction

8.889.1 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)

8.889.2 Field Documentation

8.889.2.1 uint8_t sms_routeEntry::messageClass

8.889.2.2 uint8_t sms_routeEntry::messageType

8.889.2.3 uint8_t sms_routeEntry::receiptAction

8.889.2.4 uint8_t sms_routeEntry::routeStorage

8.890 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

8.890.1 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.

8.890.2 Field Documentation

8.890.2.1 uint32_t sms_sendAsyncsmsParams::messageFormat

8.890.2.2 uint32_t sms_sendAsyncsmsParams::messageSize

8.890.2.3 uint8_t* sms_sendAsyncsmsParams::pFollowOnDC

8.890.2.4 uint8_t* sms_sendAsyncsmsParams::pForceOnDC

8.890.2.5 uint8_t* sms_sendAsyncsmsParams::pLinktimer

8.890.2.6 uint8_t* sms_sendAsyncsmsParams::pMessage

8.890.2.7 uint8_t* sms_sendAsyncsmsParams::pRetryMessage

8.890.2.8 uint32_t* sms_sendAsyncsmsParams::pRetryMessageId

8.890.2.9 uint8_t* sms_sendAsyncsmsParams::pServiceOption

8.890.2.10 uint8_t* sms_sendAsyncsmsParams::pSmsOnIms

8.890.2.11 uint32_t* sms_sendAsyncsmsParams::pUserData

8.891 sms_setIndicationReg Struct Reference

Data Fields

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

8.891.1 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

8.891.2 Field Documentation

8.891.2.1 uint8_t* sms_setIndicationReg::pRegCallStatInfoEvt

8.891.2.2 uint8_t* sms_setIndicationReg::pRegTransLayerInfoEvt

8.891.2.3 uint8_t* sms_setIndicationReg::pRegTransNWRegInfoEvt

8.892 sms_setRoutesReq Struct Reference

Data Fields

- uint16_t numOfRoutes

- sms_routeEntry routeList [0x0A]
- uint8_t * pTransferStatusReport

8.892.1 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)

8.892.2 Field Documentation

8.892.2.1 uint16_t sms_setRoutesReq::numOfRoutes

8.892.2.2 uint8_t* sms_setRoutesReq::pTransferStatusReport

8.892.2.3 sms_routeEntry sms_setRoutesReq::routeList[0x0A]

8.893 sms_transLayerInfo Struct Reference

Data Fields

- uint8_t TransType
- uint8_t TransCap

8.893.1 Detailed Description

This structure contains Transport Layer Information

Parameters

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

8.893.2 Field Documentation

8.893.2.1 uint8_t sms_transLayerInfo::TransCap

8.893.2.2 uint8_t sms_transLayerInfo::TransType

8.894 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

8.894.1 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.

8.894.2 Field Documentation

8.894.2.1 BYTE SMSAsyncRawSend_s::alphaIDLen

8.894.2.2 WORD SMSAsyncRawSend_s::causeCode

8.894.2.3 BYTE SMSAsyncRawSend_s::errorClass

8.894.2.4 WORD SMSAsyncRawSend_s::messageID

8.894.2.5 BYTE SMSAsyncRawSend_s::msgDelFailureCause

8.894.2.6 BYTE SMSAsyncRawSend_s::msgDelFailureType

8.894.2.7 BYTE* SMSAsyncRawSend_s::pAlphaID

8.894.2.8 WORD SMSAsyncRawSend_s::RPCause

8.894.2.9 WORD SMSAsyncRawSend_s::sendStatus

8.894.2.10 BYTE SMSAsyncRawSend_s::TPCause

8.894.2.11 ULONG SMSAsyncRawSend_s::userData

8.895 SMSCAddress Struct Reference

Data Fields

- BYTE length
- BYTE data [256]

8.895.1 Detailed Description

This structure holds SMSC information

Parameters

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

8.895.2 Field Documentation

8.895.2.1 BYTE SMSCAddress::data[256]

8.895.2.2 BYTE SMSCAddress::length

8.896 sMSCAddressInfo Struct Reference

Data Fields

- uint8_t length
- uint8_t data [256]

8.896.1 Detailed Description

This structure contains SMSC information

Parameters

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

8.896.2 Field Documentation

8.896.2.1 uint8_t sMSCAddressInfo::data[256]

8.896.2.2 uint8_t sMSCAddressInfo::length

8.897 sMSCAddressTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- sMSCAddressInfo SMSCInfo

8.897.1 Detailed Description

This structure contains SMSC TLV information

Parameters

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

8.897.2 Field Documentation

8.897.2.1 sMSCAddressInfo sMSCAddressTlv::SMSCInfo

8.897.2.2 uint8_t sMSCAddressTlv::TlvPresent

8.898 SMSEtwsMessage Struct Reference

Data Fields

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

8.898.1 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

8.898.2 Field Documentation

8.898.2.1 BYTE SMSEtwsMessage::data[1254]

8.898.2.2 WORD SMSEtwsMessage::length

8.898.2.3 BYTE SMSEtwsMessage::notificationType

8.899 sMSEtwsMessageInfo Struct Reference

Data Fields

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

8.899.1 Detailed Description

This structure contains ETWS information

Parameters

notificationType	• Message mode 0x00 - Primary 0x01 - Secondary GSM 0x02 - Secondary UMTS
length	• Number of sets of following elements
data	• Raw message data

8.899.2 Field Documentation

8.899.2.1 uint8_t sMSEtwsMessageInfo::data[1254]

8.899.2.2 uint16_t sMSEtwsMessageInfo::length

8.899.2.3 uint8_t sMSEtwsMessageInfo::notificationType

8.900 sMSEtwsMessageTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- sMSEtwsMessageInfo EtwsMessageInfo

8.900.1 Detailed Description

This structure contains ETWS TLV information

Parameters

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

8.900.2 Field Documentation

8.900.2.1 sMSEtwsMessageInfo sMSEtwsMessageTlv::EtwsMessageInfo

8.900.2.2 uint8_t sMSEtwsMessageTlv::TlvPresent

8.901 SMSEtwsPlmn Struct Reference

Data Fields

- WORD mobileCountryCode
- WORD mobileNetworkCode

8.901.1 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

8.901.2 Field Documentation

8.901.2.1 WORD SMSEtwsPlmn::mobileCountryCode

8.901.2.2 WORD SMSEtwsPlmn::mobileNetworkCode

8.902 sMSEtwsPlmnInfo Struct Reference

Data Fields

- uint16_t mobileCountryCode
- uint16_t mobileNetworkCode

8.902.1 Detailed Description

This structure contains ETWS PLMN information

Parameters

mobileCountry-Code	• 16 bit representation of MCC value range : 0 -999
mobileNetwork-Code	• 16 bit representation of MNC value range : 0 -999

8.902.2 Field Documentation

8.902.2.1 uint16_t sMSEtwsPlmnInfo::mobileCountryCode

8.902.2.2 uint16_t sMSEtwsPlmnInfo::mobileNetworkCode

8.903 SMSEventInfo_s Struct Reference

Data Fields

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

8.903.1 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

8.903.2 Field Documentation

8.903.2.1 SMSEtwsMessageInfo∗ SMSEventInfo_s::pEtwsMessageInfo

8.903.2.2 SMSEtwsPlmnInfo∗ SMSEventInfo_s::pEtwsPlmnInfo

8.903.2.3 SMSMessageModeInfo∗ SMSEventInfo_s::pMessageModeInfo

8.903.2.4 SMSMTMessageInfo∗ SMSEventInfo_s::pMTMessageInfo

8.903.2.5 SMSCAddressInfo∗ SMSEventInfo_s::pSMSCAddressInfo

8.903.2.6 SMSOnIMSInfo∗ SMSEventInfo_s::pSMSOnIMSInfo

8.903.2.7 SMSTransferRouteMTMessageInfo∗ SMSEventInfo_s::pTransferRouteMTMessageInfo

8.903.2.8 BYTE SMSEventInfo_s::smsEventType

8.904 smsMaxStorageSizeReq Struct Reference

Data Fields

- BYTE storageType
- BYTE ∗ pMessageMode

8.904.1 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

8.904.2 Field Documentation

8.904.2.1 BYTE* smsMaxStorageSizeReq::pMessageMode

8.904.2.2 BYTE smsMaxStorageSizeReq::storageType

8.905 smsMaxStorageSizeResp Struct Reference

Data Fields

- ULONG maxStorageSize
- ULONG freeSlots

8.905.1 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.

8.905.2 Field Documentation

8.905.2.1 ULONG smsMaxStorageSizeResp::freeSlots

8.905.2.2 ULONG smsMaxStorageSizeResp::maxStorageSize

8.906 SMSMemoryInfo Struct Reference

Data Fields

- BYTE storageType
- BYTE messageMode

8.906.1 Detailed Description

This structure holds information related to memory

Parameters

storageType	• Indicates the type of memory storage 0x00 - STORAGE_TYPE_UIM 0x01 - STORAGE_TYPE-_NV
messageMode	• Indicates the type of memory mode 0x00 - MESSAGE_MODE_CDMA - CDMA 0x01 - MESSA-GE_MODE_GW - GW

8.906.2 Field Documentation

8.906.2.1 BYTE SMSMemoryInfo::messageMode

8.906.2.2 BYTE SMSMemoryInfo::storageType

8.907 SMSMessageMode Struct Reference

Data Fields

- BYTE messageMode

8.907.1 Detailed Description

This structure holds information related to message mode

Parameters

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

8.907.2 Field Documentation

8.907.2.1 BYTE SMSMessageMode::messageMode

8.908 sMSMessageModeInfo Struct Reference

Data Fields

- uint8_t messageMode

8.908.1 Detailed Description

This structure contains message mode information.

Parameters

messageMode	Message Mode

8.908.2 Field Documentation

8.908.2.1 uint8_t sMSMessageModeInfo::messageMode

8.909 smsMsgprotocolResp Struct Reference

Data Fields

- BYTE msgProtocol

8.909.1 Detailed Description

This structure contains get message protocol response parameters

Parameters

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

8.909.2 Field Documentation

8.909.2.1 BYTE smsMsgprotocolResp::msgProtocol

8.910 SMSMTMessage Struct Reference

Data Fields

- ULONG storageType
- ULONG messageIndex

8.910.1 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

8.910.2 Field Documentation

8.910.2.1 ULONG SMSMTMessage::messageIndex

8.910.2.2 ULONG SMSMTMessage::storageType

8.911 sMSMTMessageInfo Struct Reference

Data Fields

- uint32_t storageType
- uint32_t messageIndex

8.911.1 Detailed Description

This structure contains MT message information.

Parameters

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

8.911.2 Field Documentation

8.911.2.1 uint32_t sMSMTMessageInfo::messageIndex

8.911.2.2 uint32_t sMSMTMessageInfo::storageType

8.912 SMSOnIMS Struct Reference

Data Fields

- BYTE smsOnIMS

8.912.1 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.

8.912.2 Field Documentation

8.912.2.1 BYTE SMSOnIMS::smsOnIMS

8.913 sMSOnIMSInfo Struct Reference

Data Fields

- uint8_t smsOnIMS

8.913.1 Detailed Description

This structure contains SMS on IMS information

Parameters

smsOnIMS	SMS on IMS

8.913.2 Field Documentation

8.913.2.1 uint8_t sMSOnIMSInfo::smsOnIMS

8.914 sMSOnIMSTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- sMSOnIMSInfo IMSInfo

8.914.1 Detailed Description

This structure contains SMS on IMS TLV information.

Parameters

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

8.914.2 Field Documentation

8.914.2.1 sMSOnIMSInfo sMSOnIMSTlv::IMSInfo

8.914.2.2 uint8_t sMSOnIMSTlv::TlvPresent

8.915 smsRouteEntry Struct Reference

Data Fields

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

8.915.1 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)

8.915.2 Field Documentation

8.915.2.1 BYTE smsRouteEntry::messageClass

8.915.2.2 BYTE smsRouteEntry::messageType

8.915.2.3 BYTE smsRouteEntry::receiptAction

8.915.2.4 BYTE smsRouteEntry::routeStorage

8.916 smsSetRoutesReq Struct Reference

Data Fields

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

8.916.1 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)

8.916.2 Field Documentation

8.916.2.1 WORD smsSetRoutesReq::numOfRoutes

8.916.2.2 BYTE* smsSetRoutesReq::pTransferStatusReport

8.916.2.3 smsRouteEntry smsSetRoutesReq::routeList[0x0A]

8.917 SMSTransferRouteMTMessage Struct Reference

Data Fields

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

8.917.1 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

8.917.2 Field Documentation

8.917.2.1 BYTE SMSTransferRouteMTMessage::ackIndicator

8.917.2.2 BYTE SMSTransferRouteMTMessage::data[256]

8.917.2.3 BYTE SMSTransferRouteMTMessage::format

8.917.2.4 WORD SMSTransferRouteMTMessage::length

8.917.2.5 ULONG SMSTransferRouteMTMessage::transactionID

8.918 sMSTransferRouteMTMessageInfo Struct Reference

Data Fields

- uint8_t ackIndicator
- uint32_t transactionID
- uint8_t format
- uint16_t length
- uint8_t data [256]

8.918.1 Detailed Description

This structure contains Transfer route MT message information.

Parameters

ackIndicator	• Parameter to indicate if ACK must be sent by the control point 0x00 - Send ACK 0x01 - Do not send ACK
transactionID	• Transaction ID of the message
format	• Message format 0x00 - CDMA 0x02 - 0x05 - Reserved 0x06 - GW_PP 0x07 - GW_BC
length	• Length of the raw message. This length should not exceed the maximum WMS payload length of 256 bytes
data	• Raw message data

8.918.2 Field Documentation

8.918.2.1 uint8_t sMSTransferRouteMTMessageInfo::ackIndicator

8.918.2.2 uint8_t sMSTransferRouteMTMessageInfo::data[256]

8.918.2.3 uint8_t sMSTransferRouteMTMessageInfo::format

8.918.2.4 uint16_t sMSTransferRouteMTMessageInfo::length

8.918.2.5 uint32_t sMSTransferRouteMTMessageInfo::transactionID

8.919 sQosFlowStat Struct Reference

Data Fields

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

8.919.1 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

8.919.2 Field Documentation

8.919.2.1 ULONG sQosFlowStat::bearerId

8.919.2.2 ULONGLONG sQosFlowStat::tx_bytes

8.919.2.3 ULONGLONG sQosFlowStat::tx_bytes_drp

8.919.2.4 ULONG sQosFlowStat::tx_pkt

8.919.2.5 ULONG sQosFlowStat::tx_pkt_drp

8.920 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)]

8.920.1 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

8.920.2 Field Documentation

8.920.2.1 ULONG sQosStat::apnId

8.920.2.2 ULONG sQosStat::numQosFlow

8.920.2.3 sQosFlowStat sQosStat::qosFlow[(10)]

8.920.2.4 ULONGLONG sQosStat::total_rx_bytes

8.920.2.5 ULONG sQosStat::total_rx_pkt

8.920.2.6 ULONGLONG sQosStat::total_tx_bytes

8.920.2.7 ULONGLONG sQosStat::total_tx_bytes_drp

8.920.2.8 ULONG sQosStat::total_tx_pkt

8.920.2.9 ULONG sQosStat::total_tx_pkt_drp

8.921 SrvStatusInfo Struct Reference

Data Fields

- BYTE srvStatus
- BYTE isPrefDataPath

8.921.1 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

8.921.2 Field Documentation

8.921.2.1 BYTE SrvStatusInfo::isPrefDataPath

8.921.2.2 BYTE SrvStatusInfo::srvStatus

8.922 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

8.922.1 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

8.922.2 Field Documentation

8.922.2.1 BOOL ssdatasession_params::action

8.922.2.2 ULONG ssdatasession_params::failureReason

8.922.2.3 ULONG ssdatasession_params::failureReasonv4

8.922.2.4 ULONG ssdatasession_params::failureReasonv6

8.922.2.5 BYTE ssdatasession_params::instanceId

8.922.2.6 BYTE ssdatasession_params::ipfamily

8.922.2.7 ULONG* ssdatasession_params::pAuthentication

8.922.2.8 CHAR* ssdatasession_params::pPassword

8.922.2.9 ULONG* ssdatasession_params::pProfileId3GPP

8.922.2.10 ULONG* ssdatasession_params::pProfileId3GPP2

8.922.2.11 ULONG* ssdatasession_params::pTechnology

8.922.2.12 CHAR* ssdatasession_params::pUsername

8.922.2.13 ULONG ssdatasession_params::rcv4

8.922.2.14 ULONG ssdatasession_params::rcv6

8.922.2.15 ULONG ssdatasession_params::sessionId

8.922.2.16 ULONG ssdatasession_params::v4sessionId

8.922.2.17 ULONG ssdatasession_params::v6sessionId

8.922.2.18 ULONG ssdatasession_params::verbFailReason

8.922.2.19 ULONG ssdatasession_params::verbFailReasonType

8.923 SupportedMsgList Struct Reference

Data Fields

- WORD supportedMsgLen
- BYTE supportedMsgs [256]

8.923.1 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.

8.923.2 Field Documentation

8.923.2.1 WORD SupportedMsgList::supportedMsgLen

8.923.2.2 BYTE SupportedMsgList::supportedMsgs[256]

8.924 SUPSInfo Struct Reference

Data Fields

- BYTE svcType
- BYTE isModByCC

8.924.1 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

8.924.2 Field Documentation

8.924.2.1 BYTE SUPSInfo::isModByCC

8.924.2.2 BYTE SUPSInfo::svcType

8.925 SV Struct Reference

Data Fields

- WORD id
- ULONG system
- BYTE mask

8.925.1 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

8.925.2 Field Documentation

8.925.2.1 WORD SV::id

8.925.2.2 BYTE SV::mask

8.925.2.3 ULONG SV::system

8.926 SVInfo Struct Reference

Data Fields

- BYTE len
- SV * pSV

8.926.1 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

8.926.2 Field Documentation

8.926.2.1 BYTE SVInfo::len

8.926.2.2 SV* SVInfo::pSV

8.927 svUsedforFix_s Struct Reference

Data Fields

- BYTE gnssSvUsedList_len
- WORD gnssSvUsedList [255]

8.927.1 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

8.927.2 Field Documentation

8.927.2.1 WORD svUsedforFix_s::gnssSvUsedList[255]

8.927.2.2 BYTE svUsedforFix_s::gnssSvUsedList_len

8.928 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]

8.928.1 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

8.928.2 Field Documentation

8.928.2.1 BYTE SWI_STRUCT_CarrierImage::m_FwBuildId[100]

8.928.2.2 BYTE SWI_STRUCT_CarrierImage::m_FwImageId[16]

8.928.2.3 ULONG SWI_STRUCT_CarrierImage::m_nCarrierId

8.928.2.4 ULONG SWI_STRUCT_CarrierImage::m_nFolderId

8.928.2.5 ULONG SWI_STRUCT_CarrierImage::m_nStorage

8.928.2.6 BYTE SWI_STRUCT_CarrierImage::m_PriBuildId[100]

8.928.2.7 BYTE SWI_STRUCT_CarrierImage::m_PriImageId[16]

8.929 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

8.929.1 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

8.929.2 Field Documentation

8.929.2.1 ULONG SwiLocGetAutoStartResp::fix_rate

8.929.2.2 BOOL SwiLocGetAutoStartResp::fix_rate_reported

8.929.2.3 BYTE SwiLocGetAutoStartResp::fix_type

8.929.2.4 BOOL SwiLocGetAutoStartResp::fix_type_reported

8.929.2.5 BYTE SwiLocGetAutoStartResp::function

8.929.2.6 BOOL SwiLocGetAutoStartResp::function_reported

8.929.2.7 ULONG SwiLocGetAutoStartResp::max_dist

8.929.2.8 BOOL SwiLocGetAutoStartResp::max_dist_reported

8.929.2.9 BYTE SwiLocGetAutoStartResp::max_time

8.929.2.10 BOOL SwiLocGetAutoStartResp::max_time_reported

8.930 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

8.930.1 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

8.930.2 Field Documentation

8.930.2.1 ULONG SwiLocSetAutoStartReq::fix_rate

8.930.2.2 BYTE SwiLocSetAutoStartReq::fix_type

8.930.2.3 BYTE SwiLocSetAutoStartReq::function

8.930.2.4 ULONG SwiLocSetAutoStartReq::max_dist

8.930.2.5 BYTE SwiLocSetAutoStartReq::max_time

8.930.2.6 BOOL SwiLocSetAutoStartReq::set_fix_rate

8.930.2.7 BOOL SwiLocSetAutoStartReq::set_fix_type

8.930.2.8 BOOL SwiLocSetAutoStartReq::set_function

8.930.2.9 BOOL SwiLocSetAutoStartReq::set_max_dist

8.930.2.10 BOOL SwiLocSetAutoStartReq::set_max_time

8.931 swiModemStatusResp Struct Reference

Data Fields

- CommInfo commonInfo
- LTEInfo * pLTEInfo

8.931.1 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

8.931.2 Field Documentation

8.931.2.1 CommInfo swiModemStatusResp::commonInfo

8.931.2.2 LTEInfo* swiModemStatusResp::pLTEInfo

8.932 SwiOTAMsg_s Struct Reference

Data Fields

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

- ULONGLONG * pTime

8.932.1 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

8.932.2 Field Documentation

8.932.2.1 BYTE SwiOTAMsg_s::data[2048]

8.932.2.2 WORD SwiOTAMsg_s::data_len

8.932.2.3 LteNasReleaseInfo* SwiOTAMsg_s::pLteNasRelInfo

8.932.2.4 ULONGLONG* SwiOTAMsg_s::pTime

8.932.2.5 ULONG SwiOTAMsg_s::type

8.933 swiPDPRuntimeSettingsReq Struct Reference

Data Fields

- BYTE contextId
- BYTE contextType

8.933.1 Detailed Description

This structure contains the PDP Runtime Settings Request parameters.

Parameters

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

8.933.2 Field Documentation

8.933.2.1 BYTE swiPDPRuntimeSettingsReq::contextId

8.933.2.2 BYTE swiPDPRuntimeSettingsReq::contextType

8.934 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

8.934.1 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.

8.934.2 Field Documentation

8.934.2.1 CHAR* swiPDPRuntimeSettingsResp::pAPNName

8.934.2.2 BYTE* swiPDPRuntimeSettingsResp::pBearerId

8.934.2.3 BYTE* swiPDPRuntimeSettingsResp::pContextId

8.934.2.4 ULONG* swiPDPRuntimeSettingsResp::pIPv4Address

8.934.2.5 ULONG* swiPDPRuntimeSettingsResp::pIPv4GWAddress

8.934.2.6 struct IPV6AddressInfo* swiPDPRuntimeSettingsResp::pIPv6Address

8.934.2.7 struct IPV6AddressInfo* swiPDPRuntimeSettingsResp::pIPv6GWAddress

8.934.2.8 ULONG* swiPDPRuntimeSettingsResp::pPrDNSIPv4Address

8.934.2.9 WORD* swiPDPRuntimeSettingsResp::pPrDNSIPv6Address

8.934.2.10 ULONG* swiPDPRuntimeSettingsResp::pPrPCSCFIPv4Address

8.934.2.11 WORD* swiPDPRuntimeSettingsResp::pPrPCSCFIPv6Address

8.934.2.12 ULONG* swiPDPRuntimeSettingsResp::pSeDNSIPv4Address

8.934.2.13 WORD* swiPDPRuntimeSettingsResp::pSeDNSIPv6Address

8.934.2.14 ULONG* swiPDPRuntimeSettingsResp::pSePCSCFIPv4Address

8.934.2.15 WORD* swiPDPRuntimeSettingsResp::pSePCSCFIPv6Address

8.935 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

8.935.1 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

8.935.2 Field Documentation

8.935.2.1 BYTE swiQosFilter::index

8.935.2.2 ULONG* swiQosFilter::pEspSpi

8.935.2.3 WORD* swiQosFilter::pId

8.935.2.4 IPv4Addr* swiQosFilter::pIPv4DstAddr

8.935.2.5 IPv4Addr* swiQosFilter::pIPv4SrcAddr

8.935.2.6 IPv6Addr* swiQosFilter::pIPv6DstAddr

8.935.2.7 ULONG* swiQosFilter::pIPv6Label

8.935.2.8 IPv6Addr* swiQosFilter::pIPv6SrcAddr

8.935.2.9 IPv6TrafCls* swiQosFilter::pIPv6TrafCls

8.935.2.10 BYTE* swiQosFilter::pNxtHdrProto

8.935.2.11 WORD* swiQosFilter::pPrecedence

8.935.2.12 Port* swiQosFilter::pTCPDstPort

8.935.2.13 Port* swiQosFilter::pTCPSrcPort

8.935.2.14 Tos* swiQosFilter::pTos

8.935.2.15 Port* swiQosFilter::pTranDstPort

8.935.2.16 Port* swiQosFilter::pTranSrcPort

8.935.2.17 Port* swiQosFilter::pUDPDstPort

8.935.2.18 Port* swiQosFilter::pUDPSrcPort

8.935.2.19 BYTE swiQosFilter::version

8.936 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

8.936.1 Detailed Description

This structure contains the QoS Flow Request

Parameters

index	• IP flow index • Integer that uniquely identifies each flow instance • Unique index must be assigned by the control point to every flow_spec instance
pProfileId3GPP2	• IP flow 3GPP2 profile ID • A profile ID is shorthand for a defined set of QoS flow parameters specified by the network; to be present while requesting QoS for a CDMA device
p3GPP2Pri	• IP flow 3GPP2 flow priority • Flow priority used by the network in case of contention between flows with same QoS; this parameter applies for CDMA devices
pTrafficClass	• IP flow traffic class • Integer that designates the requested traffic class: • 0 – Conversational • 1 – Streaming • 2 – Interactive • 3 – Background

pDataRate	• IP flow data rate min max • See dataRate for more information
pTokenBucket	• IP flow data rate token bucket • See tokenBucket for more information
pLatency	• IP flow latency • Maximum delay (in milliseconds) that can be tolerated by an IP packet during transfer through the wireless link
pJitter	• IP flow jitter • Difference between the maximum and minimum latency (in milliseconds) that can be tolerated by an IP packet during the transfer through the wireless link
pPktErrRate	• IP flow packet error rate • See pktErrRate for more information
pMinPolicedPkt-Sz	• IP flow minimum policed packet size • Integer that defines the minimum packet size (in bytes) that will be policed for QoS guarantees; any IP packets that are smaller than the minimum specified policed size may not receive requested QoS
pMaxAllowed-PktSz	• IP flow maximum allowed packet size • Integer that defines the maximum packet size (in bytes) allowed in the IP flow; any IP packets greater in size than the maximum allowed packet size are not queued for transmission
p3GPPRes-ResidualBER	• IP flow 3GPP residual bit error rate • residual_bit_error_rate • 0 = 5∗10-2 residual BER • 1 = 1∗10-2 residual BER • 2 = 5∗10-3 residual BER • 3 = 4∗10-3 residual BER • 4 = 1∗10-3 residual BER • 5 = 1∗10-4 residual BER • 6 = 1∗10-5 residual BER • 7 = 1∗10-6 residual BER • 8 = 6∗10-8 residual BER • Integer that indicates the undetected BER for each IP flow in the delivered packets; Tapplies only to 3GPP networks

p3GPPTraHdlPri	• 3GPP traffic handling priority • 0 – Relative traffic handling priority 1 • 1 – Relative traffic handling priority 2 • 2 – Relative traffic handling priority 3 • Defines the relative priority of the flow; applies only to 3GPP networks
p3GPPImCn	• IP flow 3GPP IM CN flag • IM CN subsystem signaling flag: • 0x00 – FALSE • 0x01 – TRUE • This parameter applies only to 3GPP networks
p3GPPSigInd	• IP flow 3GPP signaling indication • 0x00 – FALSE • 0x01 – TRUE • This parameter applies only to 3GPP networks
pLteQci	• LTE QoS Class Identifier • QoS Class Identifier(QCI) is a required parameter to request QoS in LTE • QCI values: – QCI value 0 requests the network to assign the appropriate QCI value – QCI values 1-4 are associated with guaranteed bitrates – QCI values 5-9 are associated with nonguaranteed bitrates, so the values specified as guaranteed and maximum bitrates are ignored

8.936.2 Field Documentation

8.936.2.1 BYTE swiQosFlow::index

8.936.2.2 BYTE* swiQosFlow::p3GPP2Pri

8.936.2.3 BYTE* swiQosFlow::p3GPPImCn

8.936.2.4 WORD* swiQosFlow::p3GPPResResidualBER

8.936.2.5 BYTE* swiQosFlow::p3GPPSigInd

8.936.2.6 BYTE* swiQosFlow::p3GPPTraHdlPri

8.936.2.7 dataRate* swiQosFlow::pDataRate

8.936.2.8 ULONG* swiQosFlow::pJitter

8.936.2.9 ULONG* swiQosFlow::pLatency

8.936.2.10 BYTE* swiQosFlow::pLteQci

8.936.2.11 ULONG* swiQosFlow::pMaxAllowedPktSz

8.936.2.12 ULONG* swiQosFlow::pMinPolicedPktSz

8.936.2.13 pktErrRate* swiQosFlow::pPktErrRate

8.936.2.14 WORD* swiQosFlow::pProfileId3GPP2

8.936.2.15 tokenBucket* swiQosFlow::pTokenBucket

8.936.2.16 BYTE* swiQosFlow::pTrafficClass

8.937 swiQosGranted Struct Reference

Data Fields

- swiQosFlow * pTxFlow
- swiQosFlow * pRxFlow

8.937.1 Detailed Description

This structure contains the QoS granted flow

Parameters

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

8.937.2 Field Documentation

8.937.2.1 swiQosFlow* swiQosGranted::pRxFlow

8.937.2.2 swiQosFlow* swiQosGranted::pTxFlow

8.938 swiQosIds Struct Reference

Data Fields

- BYTE sz
- ULONG * pIds

8.938.1 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

8.938.2 Field Documentation

8.938.2.1 ULONG* swiQosIds::pIds

8.938.2.2 BYTE swiQosIds::sz

8.939 swiQosModifyReq Struct Reference

Data Fields

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

8.939.1 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

8.939.2 Field Documentation

8.939.2.1 ULONG swiQosModifyReq::id

8.939.2.2 swiQosFilter* swiQosModifyReq::pRxFilter

8.939.2.3 swiQosFlow* swiQosModifyReq::pRxFlow

8.939.2.4 swiQosFilter* swiQosModifyReq::pTxFilter

8.939.2.5 swiQosFlow* swiQosModifyReq::pTxFlow

8.940 swiQosReq Struct Reference

Data Fields

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

8.940.1 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

8.940.2 Field Documentation

8.940.2.1 BYTE swiQosReq::index

8.940.2.2 swiQosFilter* swiQosReq::pRxFilter

8.940.2.3 swiQosFlow* swiQosReq::pRxFlow

8.940.2.4 swiQosFilter* swiQosReq::pTxFilter

8.940.2.5 swiQosFlow* swiQosReq::pTxFlow

8.941 swiRMTrasnferStaticsReq Struct Reference

Data Fields

- BYTE bResetStatistics
- ULONG ulMask

8.941.1 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

8.941.2 Field Documentation

8.941.2.1 BYTE swiRMTrasnferStaticsReq::bResetStatistics

8.941.2.2 ULONG swiRMTrasnferStaticsReq::ulMask

8.942 sysInfoCommon Struct Reference

Data Fields

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

8.942.1 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

8.942.2 Field Documentation

8.942.2.1 BYTE sysInfoCommon::isSysForbidden

8.942.2.2 BYTE sysInfoCommon::isSysForbiddenValid

8.942.2.3 BYTE sysInfoCommon::roamStatus

8.942.2.4 BYTE sysInfoCommon::roamStatusValid

8.942.2.5 BYTE sysInfoCommon::srvCapability

8.942.2.6 BYTE sysInfoCommon::srvCapabilityValid

8.942.2.7 BYTE sysInfoCommon::srvDomain

8.942.2.8 BYTE sysInfoCommon::srvDomainValid

8.943 t_gpsTime Struct Reference

Data Fields

- USHORT gpsWeek
- ULONG gpsTimeOfWeekMs

8.943.1 Field Documentation

8.943.1.1 ULONG t_gpsTime::gpsTimeOfWeekMs

8.943.1.2 USHORT t_gpsTime::gpsWeek

8.944 t_sensor Struct Reference

Data Fields

- ULONG usageMask
- ULONG aidingIndicatorMask

8.944.1 Field Documentation

8.944.1.1 ULONG t_sensor::aidingIndicatorMask

8.944.1.2 ULONG t_sensor::usageMask

8.945 t_Sv Struct Reference

Data Fields

- BYTE len
- USHORT entries [255]

8.945.1 Field Documentation

8.945.1.1 USHORT t_Sv::entries[255]

8.945.1.2 BYTE t_Sv::len

8.946 TDSCDMAECIOThresh Struct Reference

Data Fields

- BYTE TDSCDMAECIOThreshListLen
- ULONG * pTDSCDMAECIOThreshList

8.946.1 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.

8.946.2 Field Documentation

8.946.2.1 ULONG* TDSCDMAECIOThresh::pTDSCDMAECIOThreshList

8.946.2.2 BYTE TDSCDMAECIOThresh::TDSCDMAECIOThreshListLen

8.947 TDSCDMARSCPThresh Struct Reference

Data Fields

- BYTE TDSCDMARSCPThreshListLen
- WORD * pTDSCDMARSCPThreshList

8.947.1 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).

8.947.2 Field Documentation

8.947.2.1 WORD* TDSCDMARSCPThresh::pTDSCDMARSCPThreshList

8.947.2.2 BYTE TDSCDMARSCPThresh::TDSCDMARSCPThreshListLen

8.948 TDSCDMARSSIThresh Struct Reference

Data Fields

- BYTE TDSCDMARSSIThreshListLen
- ULONG * pTDSCDMARSSIThreshList

8.948.1 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.

8.948.2 Field Documentation

8.948.2.1 ULONG* TDSCDMARSSIThresh::pTDSCDMARSSIThreshList

8.948.2.2 BYTE TDSCDMARSSIThresh::TDSCDMARSSIThreshListLen

8.949 TDSCDMASigInfoExt Struct Reference

Data Fields

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

8.949.1 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

8.949.2 Field Documentation

8.949.2.1 FLOAT TDSCDMASigInfoExt::ecio

8.949.2.2 FLOAT TDSCDMASigInfoExt::rscp

8.949.2.3 FLOAT TDSCDMASigInfoExt::rssi

8.949.2.4 FLOAT TDSCDMASigInfoExt::sinr

8.950 tdscdmaSigInfoExt Struct Reference

Data Fields

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

8.950.1 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

8.950.2 Field Documentation

8.950.2.1 float tdscdmaSigInfoExt::ecio

8.950.2.2 float tdscdmaSigInfoExt::rscp

8.950.2.3 float tdscdmaSigInfoExt::rssi

8.950.2.4 float tdscdmaSigInfoExt::sinr

8.951 TDSCDMASINRCONFThresh Struct Reference

Data Fields

- BYTE TDSCDMASINRCONFThreshListLen
- FLOAT * pTDSCDMASINRCONFThreshList

8.951.1 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

8.951.2 Field Documentation

8.951.2.1 FLOAT* TDSCDMASINRCONFThresh::pTDSCDMASINRCONFThreshList

8.951.2.2 BYTE TDSCDMASINRCONFThresh::TDSCDMASINRCONFThreshListLen

8.952 TDSCDMASINRThresh Struct Reference

Data Fields

- BYTE TDSCDMASINRThreshListLen
- ULONG * pTDSCDMASINRThreshList

8.952.1 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

8.952.2 Field Documentation

8.952.2.1 ULONG* TDSCDMASINRThresh::pTDSCDMASINRThreshList

8.952.2.2 BYTE TDSCDMASINRThresh::TDSCDMASINRThreshListLen

8.953 tempData_t Struct Reference

Data Fields

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

8.953.1 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

8.953.2 Field Documentation

8.953.2.1 uint32_t tempData_t::temperature[64]

8.953.2.2 uint8_t tempData_t::temperatureDataLen

8.953.2.3 uint32_t tempData_t::timeOfFirstSample

8.953.2.4 uint16_t tempData_t::timeOffset[64]

8.953.2.5 uint32_t tempData_t::timeSource

8.954 tempratureData Struct Reference

Data Fields

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

8.954.1 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

8.954.2 Field Documentation

8.954.2.1 ULONG tempratureData::temperature[64]

8.954.2.2 BYTE tempratureData::temperatureDataLen

8.954.2.3 ULONG tempratureData::timeOfFirstSample

8.954.2.4 WORD tempratureData::timeOffset[64]

8.954.2.5 ULONG tempratureData::timeSource

8.955 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

8.955.1 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

8.955.2 Field Documentation

8.955.2.1 WORD TFTIDParams::destPortRangeEnd

8.955.2.2 WORD TFTIDParams::destPortRangeStart

8.955.2.3 BYTE TFTIDParams::eValid

8.955.2.4 BYTE TFTIDParams::filterId

8.955.2.5 ULONG TFTIDParams::flowLabel

8.955.2.6 ULONG TFTIDParams::IPSECSPI

8.955.2.7 BYTE TFTIDParams::ipVersion

8.955.2.8 BYTE TFTIDParams::nextHeader

8.955.2.9 WORD* TFTIDParams::pSourceIP

8.955.2.10 BYTE TFTIDParams::sourceIPMask

8.955.2.11 WORD TFTIDParams::srcPortRangeEnd

8.955.2.12 WORD TFTIDParams::srcPortRangeStart

8.955.2.13 WORD TFTIDParams::tosMask

8.956 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

8.956.1 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.

8.956.2 Field Documentation

8.956.2.1 BYTE timeInfo::day

8.956.2.2 BYTE timeInfo::dayLtSavingAdj

8.956.2.3 BYTE timeInfo::dayOfWeek

8.956.2.4 BYTE timeInfo::hour

8.956.2.5 BYTE timeInfo::minute

8.956.2.6 BYTE timeInfo::month

8.956.2.7 BYTE timeInfo::radioInterface

8.956.2.8 BYTE timeInfo::second

8.956.2.9 INT8 timeInfo::timeZone

8.956.2.10 BYTE timeInfo::TlvPresent

8.956.2.11 WORD timeInfo::year

8.957 TmdDeRegNotMitigationLvlReq Struct Reference

Data Fields

- BYTE mitigationDevIDLen
- CHAR mitigationDevID [255]

8.957.1 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

8.957.2 Field Documentation

8.957.2.1 CHAR TmdDeRegNotMitigationLvlReq::mitigationDevID[255]

8.957.2.2 BYTE TmdDeRegNotMitigationLvlReq::mitigationDevIDLen

8.958 TmdGetMitigationDevListResp Struct Reference

Data Fields

- BYTE * pMitigationDevListLen
- mitigationDevList * pMitigationDevList

8.958.1 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.

8.958.2 Field Documentation

8.958.2.1 mitigationDevList* TmdGetMitigationDevListResp::pMitigationDevList

8.958.2.2 BYTE* TmdGetMitigationDevListResp::pMitigationDevListLen

8.959 TmdGetMitigationLvlReq Struct Reference

Data Fields

- BYTE mitigationDevIDLen
- CHAR mitigationDevID [255]

8.959.1 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

8.959.2 Field Documentation

8.959.2.1 CHAR TmdGetMitigationLvlReq::mitigationDevID[255]

8.959.2.2 BYTE TmdGetMitigationLvlReq::mitigationDevIDLen

8.960 TmdGetMitigationLvlResp Struct Reference

Data Fields

- BYTE * pCurrentmitigationLvl
- BYTE * pReqMitigationLvl

8.960.1 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.

8.960.2 Field Documentation

8.960.2.1 BYTE* TmdGetMitigationLvlResp::pCurrentmitigationLvl

8.960.2.2 BYTE* TmdGetMitigationLvlResp::pReqMitigationLvl

8.961 TmdMitigationLvlIndReq Struct Reference

Data Fields

- BYTE mitigationDevIDLen
- CHAR mitigationDevID [255]

8.961.1 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

8.961.2 Field Documentation

8.961.2.1 CHAR TmdMitigationLvlIndReq::mitigationDevID[255]

8.961.2.2 BYTE TmdMitigationLvlIndReq::mitigationDevIDLen

8.962 TmdRegNotMitigationLvlReq Struct Reference

Data Fields

- BYTE mitigationDevIDLen

- CHAR mitigationDevID [255]

8.962.1 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

8.962.2 Field Documentation

8.962.2.1 CHAR TmdRegNotMitigationLvlReq::mitigationDevID[255]

8.962.2.2 BYTE TmdRegNotMitigationLvlReq::mitigationDevIDLen

8.963 tokenBucket Struct Reference

Data Fields

- ULONG peakRate
- ULONG tokenRate
- ULONG bucketSz

8.963.1 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

8.963.2 Field Documentation

8.963.2.1 ULONG tokenBucket::bucketSz

8.963.2.2 ULONG tokenBucket::peakRate

8.963.2.3 ULONG tokenBucket::tokenRate

8.964 Tos Struct Reference

Data Fields

- BYTE val
- BYTE mask

8.964.1 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.

8.964.2 Field Documentation

8.964.2.1 BYTE Tos::mask

8.964.2.2 BYTE Tos::val

8.965 transferRouteMessageTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- sMSTransferRouteMTMessageInfo TransferRouteMTMessageInfo

8.965.1 Detailed Description

This structure contains Transfer route MT message TLV information.

Parameters

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

8.965.2 Field Documentation

8.965.2.1 uint8_t transferRouteMessageTlv::TlvPresent

8.965.2.2 sMSTransferRouteMTMessageInfo transferRouteMessageTlv::TransferRouteMTMessageInfo

8.966 TransferStatInd Struct Reference

Data Fields

- BYTE StatsPeriod
- ULONG StatsMask

8.966.1 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

8.966.2 Field Documentation

8.966.2.1 ULONG TransferStatInd::StatsMask

8.966.2.2 BYTE TransferStatInd::StatsPeriod

8.967 TransferStatsDataType Struct Reference

Data Fields

- BYTE interval

8.967.1 Field Documentation

8.967.1.1 BYTE TransferStatsDataType::interval

8.968 TrStatInd Struct Reference

Data Fields

- BYTE statsPeriod
- ULONG statsMask

8.968.1 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.

8.968.2 Field Documentation

8.968.2.1 ULONG TrStatInd::statsMask

8.968.2.2 BYTE TrStatInd::statsPeriod

8.969 trueIMSI Struct Reference

Data Fields

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

8.969.1 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

8.969.2 Field Documentation

8.969.2.1 WORD trueIMSI::imsiT1112

8.969.2.2 BYTE trueIMSI::imsiTaddrNum

8.969.2.3 BYTE trueIMSI::imsiTS1[7]

8.969.2.4 BYTE trueIMSI::imsiTS2[3]

8.969.2.5 BYTE trueIMSI::mccT[3]

8.970 TXAGCList Struct Reference

Data Fields

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

8.970.1 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

8.970.2 Field Documentation

8.970.2.1 WORD* TXAGCList::pTXAIG

8.970.2.2 WORD* TXAGCList::pTXComprSlope

8.970.2.3 WORD* TXAGCList::pTXComprThres

8.970.2.4 WORD* TXAGCList::pTXExpSlope

8.970.2.5 WORD* TXAGCList::pTXExpThres

8.970.2.6 WORD* TXAGCList::pTXStaticGain

8.971 txInfo Struct Reference

Data Fields

- BYTE isInTraffic
- INT32 txPower

8.971.1 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.

8.971.2 Field Documentation

8.971.2.1 BYTE txInfo::isInTraffic

8.971.2.2 INT32 txInfo::txPower

8.972 TXPCMIIRFltr Struct Reference

Data Fields

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

8.972.1 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

8.972.2 Field Documentation

8.972.2.1 WORD* TXPCMIIRFItr::pFlag

8.972.2.2 BYTE* TXPCMIIRFItr::pStage0Val

8.972.2.3 BYTE* TXPCMIIRFItr::pStage1Val

8.972.2.4 BYTE* TXPCMIIRFItr::pStage2Val

8.972.2.5 BYTE* TXPCMIIRFItr::pStage3Val

8.972.2.6 BYTE* TXPCMIIRFItr::pStage4Val

8.972.2.7 WORD* TXPCMIIRFItr::pStageCnt

8.973 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

8.973.1 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_DE-SCRIPTION_LE-NGTH]	• 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.

8.973.2 Field Documentation

8.973.2.1 uint8_t uim_appStatus::aidLength

8.973.2.2 uint8_t uim_appStatus::aidVal[255]

8.973.2.3 uint8_t uim_appStatus::appState

8.973.2.4 uint8_t uim_appStatus::appType

8.973.2.5 uint8_t uim_appStatus::persoFeature

8.973.2.6 uint8_t uim_appStatus::persoRetries

8.973.2.7 uint8_t uim_appStatus::persoState

8.973.2.8 uint8_t uim_appStatus::persoUnblockRetries

8.973.2.9 uint8_t uim_appStatus::pin1Retries

8.973.2.10 uint8_t uim_appStatus::pin1State

8.973.2.11 uint8_t uim_appStatus::pin2Retries

8.973.2.12 uint8_t uim_appStatus::pin2State

8.973.2.13 uint8_t uim_appStatus::puk1Retries

8.973.2.14 uint8_t uim_appStatus::puk2Retries

8.973.2.15 uint8_t uim_appStatus::univPin

8.974 uim_authenticateResult Struct Reference

Data Fields

- uint16_t contentLen
- uint8_t content [1024]

8.974.1 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.

8.974.2 Field Documentation

8.974.2.1 uint8_t uim_authenticateResult::content[1024]

8.974.2.2 uint16_t uim_authenticateResult::contentLen

8.975 uim_authenticationData Struct Reference

Data Fields

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

8.975.1 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.

8.975.2 Field Documentation

8.975.2.1 uint8_t uim_authenticationData::context

8.975.2.2 uint8_t uim_authenticationData::data[1024]

8.975.2.3 uint16_t uim_authenticationData::dataLen

8.976 uim_cardResult Struct Reference

Data Fields

- uint8_t sw1
- uint8_t sw2

8.976.1 Detailed Description

This structure contains the information about the card result.

Parameters

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

8.976.2 Field Documentation

8.976.2.1 uint8_t uim_cardResult::sw1

8.976.2.2 uint8_t uim_cardResult::sw2

8.977 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]

8.977.1 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.

8.977.2 Field Documentation

8.977.2.1 uint16_t uim_cardStatus::index1xPri

8.977.2.2 uint16_t uim_cardStatus::index1xSec

8.977.2.3 uint16_t uim_cardStatus::indexGwPri

8.977.2.4 uint16_t uim_cardStatus::indexGwSec

8.977.2.5 uint8_t uim_cardStatus::numSlot

8.977.2.6 uim_slotInfo uim_cardStatus::SlotInfo[5]

8.978 uim_changeUIMPIN Struct Reference

Data Fields

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

8.978.1 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.

8.978.2 Field Documentation

8.978.2.1 uint8_t uim_changeUIMPIN::oldPINLen

8.978.2.2 uint8_t uim_changeUIMPIN::oldPINVal[255]

8.978.2.3 uint8_t uim_changeUIMPIN::pinID

8.978.2.4 uint8_t uim_changeUIMPIN::pinLen

8.978.2.5 uint8_t uim_changeUIMPIN::pinVal[255]

8.979 uim_depersonalizationInformation Struct Reference

Data Fields

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

8.979.1 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.

8.979.2 Field Documentation

8.979.2.1 uint8_t uim_depersonalizationInformation::ckLen

8.979.2.2 uint8_t uim_depersonalizationInformation::ckVal[255]

8.979.2.3 uint8_t uim_depersonalizationInformation::feature

8.979.2.4 uint8_t uim_depersonalizationInformation::operation

8.980 uim_encryptedPIN1 Struct Reference

Data Fields

- uint8_t pin1Len
- uint8_t pin1Val [255]

8.980.1 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.

8.980.2 Field Documentation

8.980.2.1 uint8_t uim_encryptedPIN1::pin1Len

8.980.2.2 uint8_t uim_encryptedPIN1::pin1Val[255]

8.981 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]

8.981.1 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.

8.981.2 Field Documentation

8.981.2.1 uint16_t uim_fileAttributes::fileID

8.981.2.2 uint16_t uim_fileAttributes::fileSize

8.981.2.3 uint8_t uim_fileAttributes::fileType

8.981.2.4 uint16_t uim_fileAttributes::rawLen

8.981.2.5 uint8_t uim_fileAttributes::rawValue[255]

8.981.2.6 uint16_t uim_fileAttributes::recordCount

8.981.2.7 uint16_t uim_fileAttributes::recordSize

8.981.2.8 uint8_t uim_fileAttributes::secActivate

8.981.2.9 uint16_t uim_fileAttributes::secActivateMask

8.981.2.10 uint8_t uim_fileAttributes::secDeactivate

8.981.2.11 uint16_t uim_fileAttributes::secDeactivateMask

8.981.2.12 uint8_t uim_fileAttributes::secIncrease

8.981.2.13 uint16_t uim_fileAttributes::secIncreaseMask

8.981.2.14 uint8_t uim_fileAttributes::secRead

8.981.2.15 uint16_t uim_fileAttributes::secReadMask

8.981.2.16 uint8_t uim_fileAttributes::secWrite

8.981.2.17 uint16_t uim_fileAttributes::secWriteMask

8.982 uim_fileInfo Struct Reference

Data Fields

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

8.982.1 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).

8.982.2 Field Documentation

8.982.2.1 uint16_t uim_fileInfo::fileID

8.982.2.2 uint16_t uim_fileInfo::path[255]

8.982.2.3 uint8_t uim_fileInfo::pathLen

8.983 uim_hotSwapStatus Struct Reference

Data Fields

- uint8_t hotSwapLength
- uint8_t hotSwap [255]

8.983.1 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

8.983.2 Field Documentation

8.983.2.1 uint8_t uim_hotSwapStatus::hotSwap[255]

8.983.2.2 uint8_t uim_hotSwapStatus::hotSwapLength

8.984 uim_personalizationStatus Struct Reference

Data Fields

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

8.984.1 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.

8.984.2 Field Documentation

8.984.2.1 uint8_t uim_personalizationStatus::feature[12]

8.984.2.2 uint8_t uim_personalizationStatus::numFeatures

8.984.2.3 uint8_t uim_personalizationStatus::unblockLeft[12]

8.984.2.4 uint8_t uim_personalizationStatus::verifyLeft[12]

8.985 uim_readResult Struct Reference

Data Fields

- uint16_t contentLen
- uint8_t content [255]

8.985.1 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.

8.985.2 Field Documentation

8.985.2.1 uint8_t uim_readResult::content[255]

8.985.2.2 uint16_t uim_readResult::contentLen

8.986 uim_readTransparentInfo Struct Reference

Data Fields

- uint16_t offset
- uint16_t length

8.986.1 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.

8.986.2 Field Documentation

8.986.2.1 uint16_t uim_readTransparentInfo::length

8.986.2.2 uint16_t uim_readTransparentInfo::offset

8.987 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]

8.987.1 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

8.987.2 Field Documentation

8.987.2.1 uint8_t uim_refreshevent::aid[255]

8.987.2.2 uint8_t uim_refreshevent::aidLength

8.987.2.3 uim_fileInfo uim_refreshevent::arrfileInfo[255]

8.987.2.4 uint8_t uim_refreshevent::mode

8.987.2.5 uint16_t uim_refreshevent::numOfFiles

8.987.2.6 uint8_t uim_refreshevent::sessionType

8.987.2.7 uint8_t uim_refreshevent::stage

8.988 uim_registerRefresh Struct Reference

Data Fields

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

8.988.1 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

8.988.2 Field Documentation

8.988.2.1 uim_fileInfo uim_registerRefresh::arrfileInfo[255]

8.988.2.2 uint16_t uim_registerRefresh::numFiles

8.988.2.3 uint8_t uim_registerRefresh::registerFlag

8.988.2.4 uint8_t uim_registerRefresh::voteForInit

8.989 uim_remainingRetries Struct Reference

Data Fields

- uint8_t verifyLeft
- uint8_t unblockLeft

8.989.1 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.

8.989.2 Field Documentation

8.989.2.1 uint8_t uim_remainingRetries::unblockLeft

8.989.2.2 uint8_t uim_remainingRetries::verifyLeft

8.990 uim_sessionInformation Struct Reference

Data Fields

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

8.990.1 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.

8.990.2 Field Documentation

8.990.2.1 uint8_t uim_sessionInformation::aid[255]

8.990.2.2 uint8_t uim_sessionInformation::aidLength

8.990.2.3 uint8_t uim_sessionInformation::sessionType

8.991 uim_setPINProtection Struct Reference

Data Fields

- uint8_t pinID
- uint8_t pinOperation

- uint8_t pinLength
- uint8_t pinValue [255]

8.991.1 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.

8.991.2 Field Documentation

8.991.2.1 uint8_t uim_setPINProtection::pinID

8.991.2.2 uint8_t uim_setPINProtection::pinLength

8.991.2.3 uint8_t uim_setPINProtection::pinOperation

8.991.2.4 uint8_t uim_setPINProtection::pinValue[255]

8.992 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]

8.992.1 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.

8.992.2 Field Documentation

8.992.2.1 uim_appStatus uim_slotInfo::AppStatus[10]

8.992.2.2 uint8_t uim_slotInfo::cardState

8.992.2.3 uint8_t uim_slotInfo::errorState

8.992.2.4 uint8_t uim_slotInfo::numApp

8.992.2.5 uint8_t uim_slotInfo::upinRetries

8.992.2.6 uint8_t uim_slotInfo::upinState

8.992.2.7 uint8_t uim_slotInfo::upukRetries

8.993 uim_UIMSessionInformation Struct Reference

Data Fields

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

8.993.1 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.

8.993.2 Field Documentation

8.993.2.1 uint8_t uim_UIMSessionInformation::aid[255]

8.993.2.2 uint8_t uim_UIMSessionInformation::aidLength

8.993.2.3 uint8_t uim_UIMSessionInformation::sessionType

8.994 uim_unblockUIMPIN Struct Reference

Data Fields

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

8.994.1 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.

8.994.2 Field Documentation

8.994.2.1 uint8_t uim_unblockUIMPIN::newPINLen

8.994.2.2 uint8_t uim_unblockUIMPIN::newPINVal[255]

8.994.2.3 uint8_t uim_unblockUIMPIN::pinID

8.994.2.4 uint8_t uim_unblockUIMPIN::pukLen

8.994.2.5 uint8_t uim_unblockUIMPIN::pukVal[255]

8.995 uim_verifyUIMPIN Struct Reference

Data Fields

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

8.995.1 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.

8.995.2 Field Documentation

8.995.2.1 uint8_t uim_verifyUIMPIN::pinID

8.995.2.2 uint8_t uim_verifyUIMPIN::pinLen

8.995.2.3 uint8_t uim_verifyUIMPIN::pinVal[255]

8.996 UIMAuthenticateReq Struct Reference

Data Fields

- UIMSessionInformation sessionInfo
- authenticationData authData
- ULONG * pIndicationToken

8.996.1 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.

8.996.2 Field Documentation

8.996.2.1 authenticationData UIMAuthenticateReq::authData

8.996.2.2 ULONG* UIMAuthenticateReq::pIndicationToken

8.996.2.3 UIMSessionInformation UIMAuthenticateReq::sessionInfo

8.997 UIMAuthenticateResp Struct Reference

Data Fields

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

8.997.1 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.

8.997.2 Field Documentation

8.997.2.1 authenticateResult* UIMAuthenticateResp::pAuthenticateResult

8.997.2.2 cardResult* UIMAuthenticateResp::pCardResult

8.997.2.3 ULONG* UIMAuthenticateResp::pIndicationToken

8.998 UIMChangePinReq Struct Reference

Data Fields

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

8.998.1 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.

8.998.2 Field Documentation

8.998.2.1 changeUIMPIN UIMChangePinReq::changePIN

8.998.2.2 ULONG* UIMChangePinReq::pIndicationToken

8.998.2.3 BYTE* UIMChangePinReq::pKeyReferenceID

8.998.2.4 UIMSessionInformation UIMChangePinReq::sessionInfo

8.999 UIMDepersonalizationReq Struct Reference

Data Fields

- depersonalizationInformation depersonilisationInfo

8.999.1 Detailed Description

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

Parameters

depersonilisation-Info	• See depersonalizationInformation for more information.

8.999.2 Field Documentation

8.999.2.1 depersonalizationInformation UIMDepersonalizationReq::depersonilisationInfo

8.1000 UIMDepersonalizationResp Struct Reference

Data Fields

- remainingRetries * pRemainingRetries

8.1000.1 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.

8.1000.2 Field Documentation

8.1000.2.1 remainingRetries* UIMDepersonalizationResp::pRemainingRetries

8.1001 UIMEventRegisterReqResp Struct Reference

Data Fields

- ULONG eventMask

8.1001.1 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

8.1001.2 Field Documentation

8.1001.2.1 ULONG UIMEventRegisterReqResp::eventMask

8.1002 UIMGetCardStatusResp Struct Reference

Data Fields

- cardStatus * pCardStatus
- hotSwapStatus * pHotSwapStatus

8.1002.1 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.

8.1002.2 Field Documentation

8.1002.2.1 cardStatus* UIMGetCardStatusResp::pCardStatus

8.1002.2.2 hotSwapStatus* UIMGetCardStatusResp::pHotSwapStatus

8.1003 UIMGetConfigurationReq Struct Reference

Data Fields

- ULONG * pConfigurationMask

8.1003.1 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.

8.1003.2 Field Documentation

8.1003.2.1 ULONG* UIMGetConfigurationReq::pConfigurationMask

8.1004 UIMGetConfigurationResp Struct Reference

Data Fields

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

8.1004.1 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

8.1004.2 Field Documentation

8.1004.2.1 BYTE* UIMGetConfigurationResp::pAutoSelection

8.1004.2.2 BYTE* UIMGetConfigurationResp::pHaltSubscription

8.1004.2.3 personalizationStatus* UIMGetConfigurationResp::pPersonalizationStatus

8.1005 UIMGetFileAttributesReq Struct Reference

Data Fields

- UIMSessionInformation sessionInfo
- fileInfo fileIndex
- ULONG * pIndicationToken

8.1005.1 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.

8.1005.2 Field Documentation

8.1005.2.1 fileInfo UIMGetFileAttributesReq::fileIndex

8.1005.2.2 ULONG* UIMGetFileAttributesReq::pIndicationToken

8.1005.2.3 UIMSessionInformation UIMGetFileAttributesReq::sessionInfo

8.1006 UIMGetFileAttributesResp Struct Reference

Data Fields

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

8.1006.1 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.

8.1006.2 Field Documentation

8.1006.2.1 cardResult∗ UIMGetFileAttributesResp::pCardResult

8.1006.2.2 fileAttributes∗ UIMGetFileAttributesResp::pFileAttributes

8.1006.2.3 ULONG∗ UIMGetFileAttributesResp::pIndicationToken

8.1007 UIMGetSlotsStatusResp Struct Reference

Data Fields

- BYTE ∗ pNumberOfPhySlot
- UIMSlotsStatus ∗ pUimSlotsStatus

8.1007.1 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..

8.1007.2 Field Documentation

8.1007.2.1 BYTE∗ UIMGetSlotsStatusResp::pNumberOfPhySlot

8.1007.2.2 UIMSlotsStatus* UIMGetSlotsStatusResp::pUimSlotsStatus

8.1008 UIMPinResp Struct Reference

Data Fields

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

8.1008.1 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.

8.1008.2 Field Documentation

8.1008.2.1 encryptedPIN1* UIMPinResp::pEncryptedPIN1

8.1008.2.2 ULONG* UIMPinResp::pIndicationToken

8.1008.2.3 remainingRetries* UIMPinResp::pRemainingRetries

8.1009 UIMPowerDownReq Struct Reference

Data Fields

- BYTE slot

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

8.1009.2 Field Documentation

8.1009.2.1 BYTE UIMPowerDownReq::slot

8.1010 UIMPowerUpReq Struct Reference

Data Fields

- BYTE slot
- BYTE * pIgnoreHotSwapSwitch

8.1010.1 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

8.1010.2 Field Documentation

8.1010.2.1 BYTE* UIMPowerUpReq::pIgnoreHotSwapSwitch

8.1010.2.2 BYTE UIMPowerUpReq::slot

8.1011 UIMReadTransparentReq Struct Reference

Data Fields

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

8.1011.1 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.

8.1011.2 Field Documentation

8.1011.2.1 fileInfo UIMReadTransparentReq::fileIndex

8.1011.2.2 BYTE* UIMReadTransparentReq::pEncryptData

8.1011.2.3 ULONG* UIMReadTransparentReq::pIndicationToken

8.1011.2.4 readTransparentInfo UIMReadTransparentReq::readTransparent

8.1011.2.5 UIMSessionInformation UIMReadTransparentReq::sessionInfo

8.1012 UIMReadTransparentResp Struct Reference

Data Fields

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

8.1012.1 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.

8.1012.2 Field Documentation

8.1012.2.1 cardResult* UIMReadTransparentResp::pCardResult

8.1012.2.2 BYTE* UIMReadTransparentResp::pEncryptedData

8.1012.2.3 ULONG* UIMReadTransparentResp::pIndicationToken

8.1012.2.4 readResult* UIMReadTransparentResp::pReadResult

8.1013 UIMRefreshCompleteReq Struct Reference

Data Fields

- UIMSessionInformation sessionInfo
- BYTE refreshComplete

8.1013.1 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

8.1013.2 Field Documentation

8.1013.2.1 BYTE UIMRefreshCompleteReq::refreshComplete

8.1013.2.2 UIMSessionInformation UIMRefreshCompleteReq::sessionInfo

8.1014 UIMRefreshEvent Struct Reference

Data Fields

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

8.1014.1 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

8.1014.2 Field Documentation

8.1014.2.1 BYTE UIMRefreshEvent::aid[255]

8.1014.2.2 BYTE UIMRefreshEvent::aidLength

8.1014.2.3 fileInfo UIMRefreshEvent::arrfileInfo[255]

8.1014.2.4 BYTE UIMRefreshEvent::mode

8.1014.2.5 WORD UIMRefreshEvent::numOfFiles

8.1014.2.6 BYTE UIMRefreshEvent::sessionType

8.1014.2.7 BYTE UIMRefreshEvent::stage

8.1015 UIMRefreshGetLastEventReq Struct Reference

Data Fields

- UIMSessionInformation sessionInfo

8.1015.1 Detailed Description

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

Parameters

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

8.1015.2 Field Documentation

8.1015.2.1 UIMSessionInformation UIMRefreshGetLastEventReq::sessionInfo

8.1016 UIMRefreshGetLastEventResp Struct Reference

Data Fields

- UIMRefreshEvent * pRefreshEvent

8.1016.1 Detailed Description

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

Parameters

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

8.1016.2 Field Documentation

8.1016.2.1 UIMRefreshEvent* UIMRefreshGetLastEventResp::pRefreshEvent

8.1017 UIMRefreshOKReq Struct Reference

Data Fields

- UIMSessionInformation sessionInfo
- BYTE OKtoRefresh

8.1017.1 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

8.1017.2 Field Documentation

8.1017.2.1 BYTE UIMRefreshOKReq::OKtoRefresh

8.1017.2.2 UIMSessionInformation UIMRefreshOKReq::sessionInfo

8.1018 UIMRefreshRegisterReq Struct Reference

Data Fields

- UIMSessionInformation sessionInfo
- registerRefresh regRefresh

8.1018.1 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

8.1018.2 Field Documentation

8.1018.2.1 registerRefresh UIMRefreshRegisterReq::regRefresh

8.1018.2.2 UIMSessionInformation UIMRefreshRegisterReq::sessionInfo

8.1019 UIMSessionInformation Struct Reference

Data Fields

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

8.1019.1 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.

8.1019.2 Field Documentation

8.1019.2.1 BYTE UIMSessionInformation::aid[255]

8.1019.2.2 BYTE UIMSessionInformation::aidLength

8.1019.2.3 BYTE UIMSessionInformation::sessionType

8.1020 UIMSetPinProtectionReq Struct Reference

Data Fields

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

8.1020.1 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.

8.1020.2 Field Documentation

8.1020.2.1 ULONG* UIMSetPinProtectionReq::pIndicationToken

8.1020.2.2 setPINProtection UIMSetPinProtectionReq::pinProtection

8.1020.2.3 BYTE* UIMSetPinProtectionReq::pKeyReferenceID

8.1020.2.4 UIMSessionInformation UIMSetPinProtectionReq::sessionInfo

8.1021 UIMSlotsStatus Struct Reference

Data Fields

- UIMSlotStatus uimSlotStatus [255]

8.1021.1 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.

8.1021.2 Field Documentation

8.1021.2.1 UIMSlotStatus UIMSlotsStatus::uimSlotStatus[255]

8.1022 UIMSlotStatus Struct Reference

Data Fields

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

8.1022.1 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.

8.1022.2 Field Documentation

8.1022.2.1 BYTE UIMSlotStatus::bICCID[255]

8.1022.2.2 BYTE UIMSlotStatus::bICCIDLength

8.1022.2.3 BYTE UIMSlotStatus::bLogicalSlot

8.1022.2.4 ULONG UIMSlotStatus::uPhyCardStatus

8.1022.2.5 ULONG UIMSlotStatus::uPhySlotStatus

8.1023 UIMSlotStatusChangeInfo Struct Reference

Data Fields

- UIMSlotsStatus slotsstatusChange
- BYTE bNumberOfPhySlots

8.1023.1 Detailed Description

Structure consist of cardstatus params

Parameters

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

8.1023.2 Field Documentation

8.1023.2.1 BYTE UIMSlotStatusChangeInfo::bNumberOfPhySlots

8.1023.2.2 UIMSlotsStatus UIMSlotStatusChangeInfo::slotsstatusChange

8.1024 UIMStatusChangeInfo Struct Reference

Data Fields

- cardStatus statusChange

8.1024.1 Detailed Description

Structure consist of cardstatus params

Parameters

statusChange	• See cardStatus for more information

8.1024.2 Field Documentation

8.1024.2.1 cardStatus UIMStatusChangeInfo::statusChange

8.1025 UIMSwitchSlotReq Struct Reference

Data Fields

- BYTE bLogicalSlot
- ULONG ulPhysicalSlot

8.1025.1 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

8.1025.2 Field Documentation

8.1025.2.1 BYTE UIMSwitchSlotReq::bLogicalSlot

8.1025.2.2 ULONG UIMSwitchSlotReq::ulPhysicalSlot

8.1026 UIMUnblockPinReq Struct Reference

Data Fields

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

8.1026.1 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.

8.1026.2 Field Documentation

8.1026.2.1 ULONG* UIMUnblockPinReq::pIndicationToken

8.1026.2.2 BYTE* UIMUnblockPinReq::pKeyReferenceID

8.1026.2.3 UIMSessionInformation UIMUnblockPinReq::sessionInfo

8.1026.2.4 unblockUIMPIN UIMUnblockPinReq::unblockPIN

8.1027 UIMVerifyPinReq Struct Reference

Data Fields

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

8.1027.1 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.

8.1027.2 Field Documentation

8.1027.2.1 encryptedPIN1∗ UIMVerifyPinReq::pEncryptedPIN1

8.1027.2.2 ULONG∗ UIMVerifyPinReq::pIndicationToken

8.1027.2.3 BYTE∗ UIMVerifyPinReq::pKeyReferenceID

8.1027.2.4 UIMSessionInformation UIMVerifyPinReq::sessionInfo

8.1027.2.5 verifyUIMPIN UIMVerifyPinReq::verifyPIN

8.1028 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]

8.1028.1 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.

8.1028.2 Field Documentation

8.1028.2.1 WORD UMTSInfo::cellID

8.1028.2.2 SHORT UMTSInfo::ecio

8.1028.2.3 BYTE UMTSInfo::geranInst

8.1028.2.4 geranInstInfo UMTSInfo::GeranInstInfo[255]

8.1028.2.5 WORD UMTSInfo::lac

8.1028.2.6 BYTE UMTSInfo::plmn[3]

8.1028.2.7 WORD UMTSInfo::psc

8.1028.2.8 SHORT UMTSInfo::rscp

8.1028.2.9 WORD UMTSInfo::uarfcn

8.1028.2.10 BYTE UMTSInfo::umtsInst

8.1028.2.11 UMTSinstInfo UMTSInfo::UMTSInstInfo[255]

8.1029 UMTSinstInfo Struct Reference

Data Fields

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

8.1029.1 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).

8.1029.2 Field Documentation

8.1029.2.1 SHORT UMTSinstInfo::umtsEcio

8.1029.2.2 WORD UMTSinstInfo::umtsPsc

8.1029.2.3 SHORT UMTSinstInfo::umtsRscp

8.1029.2.4 WORD UMTSinstInfo::umtsUarfcn

8.1030 umtsLTENbrCell Struct Reference

Data Fields

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

8.1030.1 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.

8.1030.2 Field Documentation

8.1030.2.1 BYTE umtsLTENbrCell::cellIsTDD

8.1030.2.2 WORD umtsLTENbrCell::earfcn

8.1030.2.3 WORD umtsLTENbrCell::pci

8.1030.2.4 ULONG umtsLTENbrCell::rsrp

8.1030.2.5 ULONG umtsLTENbrCell::rsrq

8.1030.2.6 SHORT umtsLTENbrCell::srxlev

8.1031 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

8.1031.1 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.

8.1031.2 Field Documentation

8.1031.2.1 BYTE UMTSMinQoS::deliveryErrSDU

8.1031.2.2 ULONG UMTSMinQoS::grntDownlinkBitrate

8.1031.2.3 ULONG UMTSMinQoS::grntUplinkBitrate

8.1031.2.4 ULONG UMTSMinQoS::maxDownlinkBitrate

8.1031.2.5 ULONG UMTSMinQoS::maxSDUSize

8.1031.2.6 ULONG UMTSMinQoS::maxUplinkBitrate

8.1031.2.7 BYTE UMTSMinQoS::qosDeliveryOrder

8.1031.2.8 BYTE UMTSMinQoS::resBerRatio

8.1031.2.9 BYTE UMTSMinQoS::sduErrorRatio

8.1031.2.10 BYTE UMTSMinQoS::trafficClass

8.1031.2.11 ULONG UMTSMinQoS::trafficPriority

8.1031.2.12 ULONG UMTSMinQoS::transferDelay

8.1032 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

8.1032.1 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.

8.1032.2 Field Documentation

8.1032.2.1 BYTE UMTSQoS::deliveryErrSDU

8.1032.2.2 ULONG UMTSQoS::grntDownlinkBitrate

8.1032.2.3 ULONG UMTSQoS::grntUplinkBitrate

8.1032.2.4 ULONG UMTSQoS::maxDownlinkBitrate

8.1032.2.5 ULONG UMTSQoS::maxSDUSize

8.1032.2.6 ULONG UMTSQoS::maxUplinkBitrate

8.1032.2.7 BYTE UMTSQoS::qosDeliveryOrder

8.1032.2.8 BYTE UMTSQoS::resBerRatio

8.1032.2.9 BYTE UMTSQoS::sduErrorRatio

8.1032.2.10 BYTE UMTSQoS::trafficClass

8.1032.2.11 ULONG UMTSQoS::trafficPriority

8.1032.2.12 ULONG UMTSQoS::transferDelay

8.1033 UMTSReqQoSSigInd Struct Reference

Data Fields

- struct UMTSQoS UMTSReqQoS
- BYTE SigInd

8.1033.1 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

8.1033.2 Field Documentation

8.1033.2.1 BYTE UMTSReqQoSSigInd::SigInd

8.1033.2.2 struct UMTSQoS UMTSReqQoSSigInd::UMTSReqQoS

8.1034 unblockUIMPIN Struct Reference

Data Fields

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

8.1034.1 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.

8.1034.2 Field Documentation

8.1034.2.1 BYTE unblockUIMPIN::newPINLen

8.1034.2.2 BYTE unblockUIMPIN::newPINVal[255]

8.1034.2.3 BYTE unblockUIMPIN::pinID

8.1034.2.4 BYTE unblockUIMPIN::pukLen

8.1034.2.5 BYTE unblockUIMPIN::pukVal[255]

8.1035 UniversalTime Struct Reference

Data Fields

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

8.1035.1 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.

8.1035.2 Field Documentation

8.1035.2.1 BYTE UniversalTime::day

8.1035.2.2 BYTE UniversalTime::dayOfWeek

8.1035.2.3 BYTE UniversalTime::hour

8.1035.2.4 BYTE UniversalTime::minute

8.1035.2.5 BYTE UniversalTime::month

8.1035.2.6 BYTE UniversalTime::second

8.1035.2.7 WORD UniversalTime::year

8.1036 unpack_dms_GetActivationState_t Struct Reference

Data Fields

- uint8_t state

8.1036.1 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 0xff - Invalid data.

8.1036.2 Field Documentation

8.1036.2.1 uint8_t unpack_dms_GetActivationState_t::state

8.1037 unpack_dms_GetBandCapability_t Struct Reference

Data Fields

- uint32_t BandCapability
- uint16_t Tlvresult

8.1037.1 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. • eQCWWAN_ERR_NULL_TLV - Invalid Tlv Data.

8.1037.2 Field Documentation

8.1037.2.1 uint32_t unpack_dms_GetBandCapability_t::BandCapability

8.1037.2.2 uint16_t unpack_dms_GetBandCapability_t::Tlvresult

8.1038 unpack_dms_GetCrashAction_t Struct Reference

Data Fields

- uint8_t DevCrashState
- uint16_t Tlvresult

8.1038.1 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 – 0xff - Invaild data.
Tlvresult	• unpack Tlv result

8.1038.2 Field Documentation

8.1038.2.1 uint8_t unpack_dms_GetCrashAction_t::DevCrashState

8.1038.2.2 uint16_t unpack_dms_GetCrashAction_t::Tlvresult

8.1039 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

8.1039.1 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 – 0xff - Invaild data.
DisableIMSI	• optional 1 byte parameter • describes if IMSI display is enabled or disabled • values: – 0x00 - Allow display of IMSI – 0x01 - Do not display IMSI – 0xff - Invaild data.
IPFamSupport	• optional 2 byte BitMask • bitmask representing the IP families supported • values: – 0x01 - IPv4 – 0x02 - IPv6 – 0x04 - IPv4v6 – 0xff - Invaild data.
RMAutoConnect	• optional 1 byte parameter • QMI Mode RM Net Auto Connect Support • values: – 0x00 - Not Supported – 0x01 - Supported – 0xff - Invaild data.
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 – 0xff - Invaild data.

SMSSupport	• optional 1 byte parameter • SMS support • values: – 0x00 - Not supported – 0x01 - supported – 0xff - Invaild data. • 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 – 0xff - Invaild data.
DHCPRelay-Enabled	• optional 1 byte parameter • DHCP Relay support • values: – 0x00 - Disable DHCP relay – 0x01 - Enable DHCP relay – 0xff - Invaild data.
GPSLPM	• optional 1 byte parameter • GPSLPM support • values: – 0x00 - Enable GPS in Low Power Mode – 0x01 - Disable GPS in Low Power Mode – 0xff - Invaild data.
Tlvresult	• unpack Tlv Result.

8.1039.2 Field Documentation

8.1039.2.1 uint8_t unpack_dms_GetCustFeature_t::DHCPRelayEnabled

8.1039.2.2 uint8_t unpack_dms_GetCustFeature_t::DisableIMSI

8.1039.2.3 uint32_t unpack_dms_GetCustFeature_t::GpsEnable

8.1039.2.4 uint8_t unpack_dms_GetCustFeature_t::GPSLPM

8.1039.2.5 uint8_t unpack_dms_GetCustFeature_t::GPSSel

8.1039.2.6 uint16_t unpack_dms_GetCustFeature_t::IPFamSupport

8.1039.2.7 uint8_t unpack_dms_GetCustFeature_t::IsVoiceEnabled

8.1039.2.8 uint8_t unpack_dms_GetCustFeature_t::RMAutoConnect

8.1039.2.9 uint8_t unpack_dms_GetCustFeature_t::SMSSupport

8.1039.2.10 uint16_t unpack_dms_GetCustFeature_t::Tlvresult

8.1040 unpack_dms_GetCustFeaturesV2_t Struct Reference

Data Fields

- DMSgetCustomFeatureV2 GetCustomFeatureV2
- uint16_t Tlvresult

8.1040.1 Detailed Description

This structure contains customization settings set to modem unpack

Parameters

GetCustom-FeatureV2	• See DMSgetCustomFeatureV2 for more information
Tlvresult	• Unpack Result

8.1040.2 Field Documentation

8.1040.2.1 DMSgetCustomFeatureV2 unpack_dms_GetCustFeaturesV2_t::GetCustomFeatureV2

8.1040.2.2 uint16_t unpack_dms_GetCustFeaturesV2_t::Tlvresult

8.1041 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

8.1041.1 Detailed Description

Structure to store the device capabilities

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
DataService-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	• 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 • eQCWWAN_ERR_NULL_TLV - Invalid Tlv Data.

8.1041.2 Field Documentation

8.1041.2.1 uint32_t unpack_dms_GetDeviceCap_t::DataServiceCapability

8.1041.2.2 uint32_t unpack_dms_GetDeviceCap_t::MaxRXChannelRate

8.1041.2.3 uint32_t unpack_dms_GetDeviceCap_t::MaxTXChannelRate

8.1041.2.4 uint8_t unpack_dms_GetDeviceCap_t::RadioIfaces[64]

8.1041.2.5 uint32_t unpack_dms_GetDeviceCap_t::RadioIfacesSize

8.1041.2.6 uint32_t unpack_dms_GetDeviceCap_t::SimCapability

8.1041.2.7 uint16_t unpack_dms_GetDeviceCap_t::Tlvresult

8.1042 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]

8.1042.1 Detailed Description

Gets the device capabilities structure

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
dataServiceCaCapability	• 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 – 0xff - Invalid Device capablitityes Data.
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[OUT]	• Radio interface array. This is a structure of array containing the elements below. – See qaGobiApiTableRadioInterfaces.h for Radio Interfaces

8.1042.2 Field Documentation

8.1042.2.1 uint32_t unpack_dms_GetDeviceCapabilities_t::dataServiceCaCapability

8.1042.2.2 uint32_t unpack_dms_GetDeviceCapabilities_t::maxRxChannelRate

8.1042.2.3 uint32_t unpack_dms_GetDeviceCapabilities_t::maxTxChannelRate

8.1042.2.4 uint8_t unpack_dms_GetDeviceCapabilities_t::RadioIfaces[255]

8.1042.2.5 uint32_t unpack_dms_GetDeviceCapabilities_t::radioIfacesSize

8.1042.2.6 uint32_t unpack_dms_GetDeviceCapabilities_t::simCapability

8.1043 unpack_dms_GetDeviceHardwareRev_t Struct Reference

Data Fields

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

8.1043.1 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 • eQCWWAN_ERR_NULL_TLV - Invalid Tlv Data.

8.1043.2 Field Documentation

8.1043.2.1 char unpack_dms_GetDeviceHardwareRev_t::String[255]

8.1043.2.2 uint8_t unpack_dms_GetDeviceHardwareRev_t::stringSize

8.1043.2.3 uint16_t unpack_dms_GetDeviceHardwareRev_t::Tlvresult

8.1044 unpack_dms_GetDeviceMfr_t Struct Reference

Data Fields

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

8.1044.1 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 • eQCWWAN_ERR_NULL_TLV - Invalid Tlv Data.

8.1044.2 Field Documentation

8.1044.2.1 char unpack_dms_GetDeviceMfr_t::String[255]

8.1044.2.2 uint8_t unpack_dms_GetDeviceMfr_t::stringSize

8.1044.2.3 uint16_t unpack_dms_GetDeviceMfr_t::Tlvresult

8.1045 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

8.1045.1 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.

8.1045.2 Field Documentation

8.1045.2.1 uint8_t unpack_dms_GetDeviceSerialNumbers_t::esnSize

8.1045.2.2 char unpack_dms_GetDeviceSerialNumbers_t::ESNString[255]

8.1045.2.3 uint8_t unpack_dms_GetDeviceSerialNumbers_t::imeiSize

8.1045.2.4 char unpack_dms_GetDeviceSerialNumbers_t::IMEIString[255]

8.1045.2.5 uint8_t unpack_dms_GetDeviceSerialNumbers_t::imeiSvnSize

8.1045.2.6 char unpack_dms_GetDeviceSerialNumbers_t::ImeiSvnString[255]

8.1045.2.7 uint8_t unpack_dms_GetDeviceSerialNumbers_t::meidSize

8.1045.2.8 char unpack_dms_GetDeviceSerialNumbers_t::MEIDString[255]

8.1045.2.9 uint16_t unpack_dms_GetDeviceSerialNumbers_t::Tlvresult

8.1046 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

8.1046.1 Detailed Description

This structure used to store unpack_dms_GetFirmwareInfo parameters

Parameters

modelid_str	- Mode ID String. • NULL-terminated Mode ID String. • Invalid data : index 0 equal 0x7F
bootversion_str	- Boot Version. • NULL-terminated Boot Version String. • Invalid data : index 0 equal 0x7F
appversion_str	- Application Version String. • NULL-terminated Application Version String. • Invalid data : index 0 equal 0x7F
sku_str	- SKU String. • NULL-terminated SKU String. • Invalid data : index 0 equal 0x7F
packageid_str	- Package ID String. • NULL-terminated Package ID String. • deprecated on EM/MC74xx(9x30) devices • Invalid data : index 0 equal 0x7F
carrier_str	- Carrier String. • NULL-terminated Carrier String. • Invalid data : index 0 equal 0x7F
priversion_str	- PRI Version String. • NULL-terminated PRI Version String. • Invalid data : index 0 equal 0x7F
cur_carr_name	- Current Carrier Name String. • NULL-terminated Current Carrier Name String. • Invalid data : index 0 equal 0x7F

cur_carr_rev	- Current Carrier Revision String. • NULL-terminated Current Carrier Revision String. • Invalid data : index 0 equal 0x7F
Tlvresult	- unpack Tlv Result.

8.1046.2 Field Documentation

8.1046.2.1 char unpack_dms_GetFirmwareInfo_t::appversion_str[85]

8.1046.2.2 char unpack_dms_GetFirmwareInfo_t::bootversion_str[85]

8.1046.2.3 char unpack_dms_GetFirmwareInfo_t::carrier_str[20]

8.1046.2.4 char unpack_dms_GetFirmwareInfo_t::cur_carr_name[17]

8.1046.2.5 char unpack_dms_GetFirmwareInfo_t::cur_carr_rev[13]

8.1046.2.6 char unpack_dms_GetFirmwareInfo_t::modelid_str[20]

8.1046.2.7 char unpack_dms_GetFirmwareInfo_t::packageid_str[85]

8.1046.2.8 char unpack_dms_GetFirmwareInfo_t::priversion_str[16]

8.1046.2.9 char unpack_dms_GetFirmwareInfo_t::sku_str[15]

8.1046.2.10 uint16_t unpack_dms_GetFirmwareInfo_t::Tlvresult

8.1047 unpack_dms_GetFirmwareRevision_t Struct Reference

Data Fields

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

8.1047.1 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 • Invalid data : index 0 equal 0x7F

PRIString[OUT]	• NULL-terminated PRI revision string. • Maximum Length is 255 Bytes • Invalid data : index 0 equal 0x7F
Tlvresult	• unpack Tlv Result.

8.1047.2 Field Documentation

8.1047.2.1 uint8_t unpack_dms_GetFirmwareRevision_t::amssSize

8.1047.2.2 char unpack_dms_GetFirmwareRevision_t::AMSSString[255]

8.1047.2.3 char unpack_dms_GetFirmwareRevision_t::PRIString[255]

8.1047.2.4 uint16_t unpack_dms_GetFirmwareRevision_t::Tlvresult

8.1048 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

8.1048.1 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 • Invalid data : index 0 equal 0x7F and amssSize is equal 0.
bootSize	• Maximum number of characters (including NULL terminator) that the boot string array can contain

BootString	• NULL-terminated boot code revision string • Maximum Length is 255 Bytes • Invalid data : index 0 equal 0x7F and bootSize is equal 0.
priSize	• Maximum number of characters (including NULL terminator) that the PRI string array can contain
PRIString	• NULL-terminated PRI revision string • Maximum Length is 255 Bytes • Invalid data : index 0 equal 0x7F and priSize is equal 0.
Tlvresult	• unpack Tlv Result

8.1048.2 Field Documentation

8.1048.2.1 uint8_t unpack_dms_GetFirmwareRevisions_t::amssSize

8.1048.2.2 char unpack_dms_GetFirmwareRevisions_t::AMSSString[255]

8.1048.2.3 uint8_t unpack_dms_GetFirmwareRevisions_t::bootSize

8.1048.2.4 char unpack_dms_GetFirmwareRevisions_t::BootString[255]

8.1048.2.5 uint8_t unpack_dms_GetFirmwareRevisions_t::priSize

8.1048.2.6 char unpack_dms_GetFirmwareRevisions_t::PRIString[255]

8.1048.2.7 uint16_t unpack_dms_GetFirmwareRevisions_t::Tlvresult

8.1049 unpack_dms_GetFSN_t Struct Reference

Data Fields

- char String [255]
- uint16_t Tlvresult

8.1049.1 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 - eQCWWAN_ERR_NULL_TLV - Invalid Tlv Data.

8.1049.2 Field Documentation

8.1049.2.1 char unpack_dms_GetFSN_t::String[255]

8.1049.2.2 uint16_t unpack_dms_GetFSN_t::Tlvresult

8.1050 unpack_dms_GetHardwareRevision_t Struct Reference

Data Fields

- char hwVer [255]

8.1050.1 Detailed Description

This structure used to store unpack_dms_GetHardwareRevision parameters

Parameters

hwVer	- Hardware version • NULL-terminated string • Maximum Length is 255 Bytes • Invalid data : index 0 equal 0x7F.

8.1050.2 Field Documentation

8.1050.2.1 char unpack_dms_GetHardwareRevision_t::hwVer[255]

8.1051 unpack_dms_GetIMSI_t Struct Reference

Data Fields

- char imsi [255]
- uint16_t Tlvresult

8.1051.1 Detailed Description

This structure used to store unpack_dms_GetIMSI parameters

Parameters

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

8.1051.2 Field Documentation

8.1051.2.1 char unpack_dms_GetIMSI_t::imsi[255]

8.1051.2.2 uint16_t unpack_dms_GetIMSI_t::Tlvresult

8.1052 unpack_dms_GetManufacturer_t Struct Reference

Data Fields

- char manufacturer [255]
- uint16_t Tlvresult

8.1052.1 Detailed Description

This structure is used to store device manufacturer information.

Parameters

manufacturer[OUT]	• NULL terminated string
Tlvresult	• Unpack Result • eQCWWAN_ERR_NULL_TLV - Invalid Tlv Data.

8.1052.2 Field Documentation

8.1052.2.1 char unpack_dms_GetManufacturer_t::manufacturer[255]

8.1052.2.2 uint16_t unpack_dms_GetManufacturer_t::Tlvresult

8.1053 unpack_dms_GetModelID_t Struct Reference

Data Fields

- char modelid [255]
- uint16_t Tlvresult

8.1053.1 Detailed Description

This structure used to store unpack_dms_GetModelID parameters

Parameters

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

8.1053.2 Field Documentation

8.1053.2.1 char unpack_dms_GetModelID_t::modelid[255]

8.1053.2.2 uint16_t unpack_dms_GetModelID_t::Tlvresult

8.1054 unpack_dms_GetNetworkTime_t Struct Reference

Data Fields

- uint16_t source
- uint64_t timestamp
- uint16_t Tlvresult

8.1054.1 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 • eQCWWAN_ERR_NULL_TLV - Invalid Tlv Data.

Note

The source of the timestamp provided specifies how the timestamp was determined. The first network time that is available will be returned. If no network time is available, the timestamp is taken from the 32 kHz slow-clock of the device.

8.1054.2 Field Documentation

8.1054.2.1 uint16_t unpack_dms_GetNetworkTime_t::source

8.1054.2.2 uint64_t unpack_dms_GetNetworkTime_t::timestamp

8.1054.2.3 uint16_t unpack_dms_GetNetworkTime_t::Tlvresult

8.1055 unpack_dms_GetOfflineReason_t Struct Reference

Data Fields

- uint32_t * pReasonMask
- uint32_t * pbPlatform
- uint16_t Tlvresult

8.1055.1 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 – NULL pointer - Invalid data.
pbPlatform[OUT]	• Optional parameter • Is the device offline due to a platform restriction? – 0 - No – 1 - Yes – NULL pointer - Invalid data.
Tlvresult	• Unpack Result

8.1055.2 Field Documentation

8.1055.2.1 uint32_t* unpack_dms_GetOfflineReason_t::pbPlatform

8.1055.2.2 uint32_t* unpack_dms_GetOfflineReason_t::pReasonMask

8.1055.2.3 uint16_t unpack_dms_GetOfflineReason_t::Tlvresult

8.1056 unpack_dms_GetPower_t Struct Reference

Data Fields

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

8.1056.1 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. • 0xff - Invalid data.

Note

Valid transitions for Power Modes

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

Parameters

OfflineReason	- offline reason. • 0x0001 - Host image misconfiguration. • 0x0002 - PRI image misconfiguration. • 0x0004 - PRI version incompatible. • 0x0008 - Device memory is full,cannot copy PRI information. • 0xffff - Invalid data.
Hardware-ControlledMode	- hardware restricted mode. • 0x00 - FALSE. • 0x01 - TRUE. • 0xFF - Invalid data.
Tlvresult	- unpack Tlv Result. • eQCWWAN_ERR_NULL_TLV - Invalid Tlv Data.

8.1056.2 Field Documentation

8.1056.2.1 uint32_t unpack_dms_GetPower_t::HardwareControlledMode

8.1056.2.2 uint32_t unpack_dms_GetPower_t::OfflineReason

8.1056.2.3 uint32_t unpack_dms_GetPower_t::OperationMode

8.1056.2.4 uint16_t unpack_dms_GetPower_t::Tlvresult

8.1057 unpack_dms_GetPRLVersion_t Struct Reference

Data Fields

- uint8_t u8PRLPreference
- uint16_t u16PRLVersion
- uint16_t Tlvresult

8.1057.1 Detailed Description

This structure used to store unpack_dms_GetPRLVersion parameters

Parameters

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

8.1057.2 Field Documentation

8.1057.2.1 uint16_t unpack_dms_GetPRLVersion_t::Tlvresult

8.1057.2.2 uint16_t unpack_dms_GetPRLVersion_t::u16PRLVersion

8.1057.2.3 uint8_t unpack_dms_GetPRLVersion_t::u8PRLPreference

8.1058 unpack_dms_GetSerialNumbers_t Struct Reference

Data Fields

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

8.1058.1 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. • Invalid data : index 0 equal 0x7F.
imei_no	- International Mobile Equipment Identity of the device. • NULL terminated IMEI string. Empty string is returned when IMEI is not supported/programmed. • Invalid data : index 0 equal 0x7F.
meid	- Mobile Equipment Identifier of the device. • NULL-terminated MEID string. Empty string is returned when MEID is not supported/programmed. • Invalid data : index 0 equal 0x7F.
imeisv_svn	- NULL-terminated IMEI SVN string. Empty string is returned when IMEI SVN is not supported/programmed.imei software version revision. • Invalid data : index 0 equal 0x7F.

8.1058.2 Field Documentation

8.1058.2.1 char unpack_dms_GetSerialNumbers_t::esn[255]

8.1058.2.2 char unpack_dms_GetSerialNumbers_t::imei_no[255]

8.1058.2.3 char unpack_dms_GetSerialNumbers_t::imeisv_svn[255]

8.1058.2.4 char unpack_dms_GetSerialNumbers_t::meid[255]

8.1059 unpack_dms_GetUSBComp_t Struct Reference

Data Fields

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

8.1059.1 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 • Invalid data : USBComp index 0 equal 0x7F and NumSupUSBComps is equal 0.

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 – 255 - invalid data.
Tlvresult	• unpack Tlv Result.

8.1059.2 Field Documentation

8.1059.2.1 uint8_t unpack_dms_GetUSBComp_t::NumSupUSBComps

8.1059.2.2 uint8_t unpack_dms_GetUSBComp_t::SupUSBComps

8.1059.2.3 uint16_t unpack_dms_GetUSBComp_t::Tlvresult

8.1059.2.4 uint8_t unpack_dms_GetUSBComp_t::USBComp[255]

8.1060 unpack_dms_GetVoiceNumber_t Struct Reference

Data Fields

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

8.1060.1 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 • Invalid data : index 0 equal 0x7F and voiceNumberSize is equal 0.
minSize	• Maximum number of characters (including NULL terminator) that the MIN array can contain.
MIN	• Optional Parameter • MIN string: Empty string returned when MIN is not supported/ programmed. • Maximum Length is 255 Bytes • Invalid data : index 0 equal 0x7F and minSize is equal 0.
Tlvresult	• Unpack Result

8.1060.2 Field Documentation

8.1060.2.1 char unpack_dms_GetVoiceNumber_t::MIN[255]

8.1060.2.2 uint8_t unpack_dms_GetVoiceNumber_t::minSize

8.1060.2.3 uint16_t unpack_dms_GetVoiceNumber_t::Tlvresult

8.1060.2.4 char unpack_dms_GetVoiceNumber_t::VoiceNumber[255]

8.1060.2.5 uint8_t unpack_dms_GetVoiceNumber_t::voiceNumberSize

8.1061 unpack_dms_ResetToFactoryDefaults_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1061.1 Detailed Description

This structure contains reset to factory default unpack

Parameters

Tlvresult	• Unpack Result

8.1061.2 Field Documentation

8.1061.2.1 uint16_t unpack_dms_ResetToFactoryDefaults_t::Tlvresult

8.1062 unpack_dms_SetActivationStatusCallback_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1062.1 Detailed Description

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

Parameters

Tlvresult	• Unpack Result

8.1062.2 Field Documentation

8.1062.2.1 uint16_t unpack_dms_SetActivationStatusCallback_t::Tlvresult

8.1063 unpack_dms_SetCrashAction_t Struct Reference

Data Fields

- uint8_t notused

8.1063.1 Detailed Description

This structure is used to store unpack_dms_SetCrashAction parameters

Parameters

notused	• unpack Tlv result

8.1063.2 Field Documentation

8.1063.2.1 uint8_t unpack_dms_SetCrashAction_t::notused

8.1064 unpack_dms_SetCustFeature_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1064.1 Detailed Description

This structure is used to store unpack_dms_SetCustFeature parameters

Parameters

Tlvresult	• unpack Tlv Result.

8.1064.2 Field Documentation

8.1064.2.1 uint16_t unpack_dms_SetCustFeature_t::Tlvresult

8.1065 unpack_dms_SetCustFeaturesV2_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1065.1 Detailed Description

This structure contains customization settings set to modem unpack

Parameters

Tlvresult	• Unpack Result

8.1065.2 Field Documentation

8.1065.2.1 uint16_t unpack_dms_SetCustFeaturesV2_t::Tlvresult

8.1066 unpack_dms_SetEventReport_ind_t Struct Reference

Data Fields

- dms_ActivationStatusTlv ActivationStatusTlv
- dms_OperatingModeTlv OperatingModeTlv
- uint16_t Tlvresult

8.1066.1 Detailed Description

DMS Event Report indication structure

Parameters

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

Tlvresult	• Unpack Result

8.1066.2 Field Documentation

8.1066.2.1 dms_ActivationStatusTlv unpack_dms_SetEventReport_ind_t::ActivationStatusTlv

8.1066.2.2 dms_OperatingModeTlv unpack_dms_SetEventReport_ind_t::OperatingModeTlv

8.1066.2.3 uint16_t unpack_dms_SetEventReport_ind_t::Tlvresult

8.1067 unpack_dms_SetEventReport_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1067.1 Detailed Description

This structure is used to store unpack_dms_SetEventReport parameters.

Parameters

Tlvresult	• unpack Tlv result

8.1067.2 Field Documentation

8.1067.2.1 uint16_t unpack_dms_SetEventReport_t::Tlvresult

8.1068 unpack_dms_SetFirmwarePreference_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1068.1 Detailed Description

This structure is used to store unpack_dms_SetFirmwarePreference parameters

Parameters

Tlvresult	• unpack Tlv result

8.1068.2 Field Documentation

8.1068.2.1 uint16_t unpack_dms_SetFirmwarePreference_t::Tlvresult

8.1069 unpack_dms_SetPower_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1069.1 Detailed Description

Structure to store unpack the operating mode of the device.

Parameters

Tlvresult	• unpack Tlv Result.

8.1069.2 Field Documentation

8.1069.2.1 uint16_t unpack_dms_SetPower_t::Tlvresult

8.1070 unpack_dms_SetUSBComp_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1070.1 Detailed Description

This structure is used to store unpack USB composition information

Parameters

Tlvresult	• unpack Tlv Result.

8.1070.2 Field Documentation

8.1070.2.1 uint16_t unpack_dms_SetUSBComp_t::Tlvresult

8.1071 unpack_dms_SLQSDmsSwiGetResetInfo_Ind_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[OUT]	• Unpack Result 8 - eQCWWAN_ERR_NULL_TLV - Invalid Tlv Data.

8.1071.2 Field Documentation

8.1071.2.1 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t::source

8.1071.2.2 uint16_t unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t::Tlvresult

8.1071.2.3 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t::type

8.1072 unpack_dms_SLQSDmsSwiGetResetInfo_t Struct Reference

Data Fields

- uint8_t type
- uint8_t source
- uint16_t Tlvresult

8.1072.1 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 – 255 - invalid data.
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 – 255 - invalid data.
Tlvresult	• Unpack Result • eQCWWAN_ERR_NULL_TLV - Invalid Tlv Data.

8.1072.2 Field Documentation

8.1072.2.1 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_t::source

8.1072.2.2 uint16_t unpack_dms_SLQSDmsSwiGetResetInfo_t::Tlvresult

8.1072.2.3 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_t::type

8.1073 unpack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1073.1 Detailed Description

This structure contains set registration state for different indication unpack

Parameters

Tlvresult	• Unpack Result

8.1073.2 Field Documentation

8.1073.2.1 uint16_t unpack_dms_SLQSDmsSwiIndicationRegister_t::TlvResult

8.1074 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

8.1074.1 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 • 0xffffffffffffffff - invalid data
is_LteBand-Capability_-Available[OUT]	LTE Band Capabilities Availability. • 0 : Unavailable. • 1 : Available.

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
is_TdsBand-Capability_-Available[OUT]	TDS Band Capabilities Availability. • 0 : Unavailable. • 1 : Available.

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

8.1074.2 Field Documentation

8.1074.2.1 uint64_t unpack_dms_SLQSGetBandCapability_t::bandCapability

8.1074.2.2 int unpack_dms_SLQSGetBandCapability_t::is_LteBandCapability_Available

8.1074.2.3 int unpack_dms_SLQSGetBandCapability_t::is_TdsBandCapability_Available

8.1074.2.4 uint64_t unpack_dms_SLQSGetBandCapability_t::LteBandCapability

8.1074.2.5 uint64_t unpack_dms_SLQSGetBandCapability_t::TdsBandCapability

8.1075 unpack_dms_SLQSGetERIFile_t Struct Reference

Data Fields

- uint16_t Tlvresult
- eriDataparams eriFile

8.1075.1 Detailed Description

This structure contains Get ERI file unpack

Parameters

Tlvresult	• Unpack Result • eQCWWAN_ERR_NULL_TLV - Invalid TLV data.
eriFile	• Pointer to structure ERIFileparams • See eriDataparams for more information

8.1075.2 Field Documentation

8.1075.2.1 eriDataparams unpack_dms_SLQSGetERIFile_t::eriFile

8.1075.2.2 uint16_t unpack_dms_SLQSGetERIFile_t::Tlvresult

8.1076 unpack_dms_SLQSSwiClearDyingGaspStatistics_t Struct Reference

Data Fields

- uint16_t TlvResult

8.1076.1 Detailed Description

This structure contains Clear Dying GASP unpack

Parameters

TlvResult	• Unpack Result

8.1076.2 Field Documentation

8.1076.2.1 uint16_t unpack_dms_SLQSSwiClearDyingGaspStatistics_t::TlvResult

8.1077 unpack_dms_SLQSSwiGetCrashInfo_t Struct Reference

Data Fields

- uint16_t TlvResult
- crashInfoParams crashInfoParam

8.1077.1 Detailed Description

This structure contains SWI get crash information unpack information

Parameters

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

8.1077.2 Field Documentation

8.1077.2.1 crashInfoParams unpack_dms_SLQSSwiGetCrashInfo_t::crashInfoParam

8.1077.2.2 uint16_t unpack_dms_SLQSSwiGetCrashInfo_t::TlvResult

8.1078 unpack_dms_SLQSSwiGetDyingGaspCfg_t Struct Reference

Data Fields

- packgetDyingGaspCfg * pGetDyingGaspCfg
- uint16_t TlvResult

8.1078.1 Detailed Description

This structure contains Get Dying GASP Config unpack

Parameters

pGetDyingGasp-Cfg	• See packgetDyingGaspCfg • NULL pDestSMSNum for invalid data. • NULL pDestSMSContent for invalid data.
Tlvresult	• Unpack Result

8.1078.2 Field Documentation

8.1078.2.1 packgetDyingGaspCfg* unpack_dms_SLQSSwiGetDyingGaspCfg_t::pGetDyingGaspCfg

8.1078.2.2 uint16_t unpack_dms_SLQSSwiGetDyingGaspCfg_t::Tlvresult

8.1079 unpack_dms_SLQSSwiGetDyingGaspStatistics_t Struct Reference

Data Fields

- packgetDyingGaspStatistics * pGetDyingGaspStatistics
- uint16_t Tlvresult

8.1079.1 Detailed Description

This structure contains Get Dying GASP Statistics unpack.

Parameters

pGetDyingGasp-Statistics	• See packgetDyingGaspStatistics • NULL pTimeStamp for invalid Timestamp data. • NULL pSMSAttemptedFlag for invalid SMS attempted data.
Tlvresult	• Unpack Result

8.1079.2 Field Documentation

8.1079.2.1 packgetDyingGaspStatistics* unpack_dms_SLQSSwiGetDyingGaspStatistics_t::pGetDyingGaspStatistics

8.1079.2.2 uint16_t unpack_dms_SLQSSwiGetDyingGaspStatistics_t::Tlvresult

8.1080 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]

8.1080.1 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 • Invalid Data - index 0 equal 0x7F.
pkgver[OUT]	• Package version of the currently running firmware • Invalid Data - index 0 equal 0x7F.
fwvers[OUT]	• firmware version of the currently running firmware • Invalid Data - index 0 equal 0x7F.
carrier[OUT]	• Carrier string of the currently running firmware • Invalid Data - index 0 equal 0x7F.

8.1080.2 Field Documentation

8.1080.2.1 char unpack_dms_SLQSSwiGetFirmwareCurr_t::carrier[16]

8.1080.2.2 char unpack_dms_SLQSSwiGetFirmwareCurr_t::fwvers[16]

8.1080.2.3 uint8_t unpack_dms_SLQSSwiGetFirmwareCurr_t::numEntries

8.1080.2.4 image_info_t* unpack_dms_SLQSSwiGetFirmwareCurr_t::pCurrImgInfo

8.1080.2.5 char unpack_dms_SLQSSwiGetFirmwareCurr_t::pkgver[16]

8.1080.2.6 char unpack_dms_SLQSSwiGetFirmwareCurr_t::priver[16]

8.1081 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

8.1081.1 Detailed Description

This structure is used to store Firmware Update Status

Parameters

ResCode	• FW Update Result Code • Values: – 0x00000001 - Successful – 0x000000FF - Invalid Data. – 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
imgType	• Optional parameter • Firmware image type that failed the update • 0xff - Invalid Data.
refData	• Optional parameter • Failed image reference data • This is normally the offset of the image that caused the failure • 0xff - Invalid Data.
refString	• Optional parameter • Failed image reference string. This is normally the partition name of the image that caused the failure if applicable. • Invalid Data - index 0 equal 0xFF.

logString	• Optional parameter • Failed image reference string. This is normally the partition name of the image that caused the failure if applicable. • Invalid Data - index 0 equal 0xFF.
Tlvresult	• Unpack Result

8.1081.2 Field Documentation

8.1081.2.1 uint8_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::imgType

8.1081.2.2 uint8_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::logString[255]

8.1081.2.3 uint32_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::refData

8.1081.2.4 uint8_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::refString[15]

8.1081.2.5 uint32_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::ResCode

8.1081.2.6 uint16_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::Tlvresult

8.1082 unpack_dms_SLQSSwiGetHostDevInfo_t Struct Reference

Data Fields

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

8.1082.1 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 • Invalid Data - index 0 equal 0x7F.
modelString[OUT]	• Host Device Model String • Null terminated ASCII String • Invalid Data - index 0 equal 0x7F.

swVerString[OUT]	• Host Device Software Version String • Null terminated ASCII string • Invalid Data - index 0 equal 0x7F.
plasmaIDString[OUT]	• Host Device Plasma ID String • Null terminated alphanumeric ASCII String. • Invalid Data - index 0 equal 0x7F.

8.1082.2 Field Documentation

8.1082.2.1 char unpack_dms_SLQSSwiGetHostDevInfo_t::manString[255]

8.1082.2.2 char unpack_dms_SLQSSwiGetHostDevInfo_t::modelString[255]

8.1082.2.3 char unpack_dms_SLQSSwiGetHostDevInfo_t::plasmaIDString[255]

8.1082.2.4 char unpack_dms_SLQSSwiGetHostDevInfo_t::swVerString[255]

8.1082.2.5 uint16_t unpack_dms_SLQSSwiGetHostDevInfo_t::TlvResult

8.1083 unpack_dms_SLQSSwiGetOSInfo_t Struct Reference

Data Fields

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

8.1083.1 Detailed Description

This structure contains SWI get host os info unpack information

Parameters

TlvResult	• Unpack Result
nameString[OUT]	• Host operating system name • Invalid Data - index 0 equal 0xFF.
VersionString[OUT]	• Host operating system version • Invalid Data - index 0 equal 0xFF.

8.1083.2 Field Documentation

8.1083.2.1 char unpack_dms_SLQSSwiGetOSInfo_t::nameString[255]

8.1083.2.2 uint16_t unpack_dms_SLQSSwiGetOSInfo_t::Tlvresult

8.1083.2.3 char unpack_dms_SLQSSwiGetOSInfo_t::versionString[255]

8.1084 unpack_dms_SLQSSwiGetSerialNoExt_t Struct Reference

Data Fields

- uint16_t Tlvresult
- char meidString [8]

8.1084.1 Detailed Description

This structure contains SWI get device serial number extension unpack information

Parameters

Tlvresult	• pack Result
meidString[OUT]	• optional parameter, mobile equipment identifier • Invalid Data - index 0 equal 0x7F.

8.1084.2 Field Documentation

8.1084.2.1 char unpack_dms_SLQSSwiGetSerialNoExt_t::meidString[8]

8.1084.2.2 uint16_t unpack_dms_SLQSSwiGetSerialNoExt_t::Tlvresult

8.1085 unpack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1085.1 Detailed Description

This structure contains set Dying GASP Config unpack

Parameters

Tlvresult	• Unpack Result

8.1085.2 Field Documentation

8.1085.2.1 uint16_t unpack_dms_SLQSSwiSetDyingGaspCfg_t::Tlvresult

8.1086 unpack_dms_SLQSSwiSetHostDevInfo_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1086.1 Detailed Description

This structure contains SWI set host dev info unpack

Parameters

Tlvresult	• Unpack Result

8.1086.2 Field Documentation

8.1086.2.1 uint16_t unpack_dms_SLQSSwiSetHostDevInfo_t::Tlvresult

8.1087 unpack_dms_SLQSSwiSetOSInfo_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1087.1 Detailed Description

This structure contains SWI set host OS info unpack

Parameters

Tlvresult	• Unpack Result

8.1087.2 Field Documentation

8.1087.2.1 uint16_t unpack_dms_SLQSSwiSetOSInfo_t::Tlvresult

8.1088 unpack_dms_SLQSUIMGetState_t Struct Reference

Data Fields

- uint8_t state
- uint16_t Tlvresult

8.1088.1 Detailed Description

This structure contains UIM get state unpack information

Parameters

state[OUT]	• UIM state
Tlvresult	• Unpack Result

8.1088.2 Field Documentation

8.1088.2.1 uint8_t unpack_dms_SLQSUIMGetState_t::state

8.1088.2.2 uint16_t unpack_dms_SLQSUIMGetState_t::Tlvresult

8.1089 unpack_dms_UIMGetControlKeyStatus_t Struct Reference

Data Fields

- uint8_t facilityState
- uint8_t verifyRetriesLeft
- uint8_t unblockRetriesLeft
- uint16_t Tlvresult

8.1089.1 Detailed Description

This structure contains PIN retries status

Parameters

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

8.1089.2 Field Documentation

8.1089.2.1 uint8_t unpack_dms_UIMGetControlKeyStatus_t::facilityState

8.1089.2.2 uint16_t unpack_dms_UIMGetControlKeyStatus_t::Tlvresult

8.1089.2.3 uint8_t unpack_dms_UIMGetControlKeyStatus_t::unblockRetriesLeft

8.1089.2.4 uint8_t unpack_dms_UIMGetControlKeyStatus_t::verifyRetriesLeft

8.1090 unpack_dms_UIMGetICCID_t Struct Reference

Data Fields

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

8.1090.1 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 - eQCWWAN_ERR_NULL_TLV - Invalid Tlv Data.

8.1090.2 Field Documentation

8.1090.2.1 uint8_t unpack_dms_UIMGetICCID_t::String[255]

8.1090.2.2 uint8_t unpack_dms_UIMGetICCID_t::stringSize

8.1090.2.3 uint16_t unpack_dms_UIMGetICCID_t::Tlvresult

8.1091 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

8.1091.1 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 – 0xFF - Invalid data.
p1VerifyRetries-Left[OUT]	• Optional parameter • Upon operational failure, this will indicate number of retries left of PIN1, after which PIN will be blocked. • 0xFF - Invalid data.
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. • 0xFF - Invalid data.
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 – 0xFF - Invalid data.
p2VerifyRetries-Left[OUT]	• Optional parameter • Upon operational failure, this will indicate number of retries left of PIN2, after which PIN will be blocked. • 0xFF - Invalid data.
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. • 0xFF - Invalid data.

Tlvresult	• Unpack Result

8.1091.2 Field Documentation

8.1091.2.1 uint8_t unpack_dms_UIMGetPINStatus_t::p1Status

8.1091.2.2 uint8_t unpack_dms_UIMGetPINStatus_t::p1UnblockRetriesLeft

8.1091.2.3 uint8_t unpack_dms_UIMGetPINStatus_t::p1VerifyRetriesLeft

8.1091.2.4 uint8_t unpack_dms_UIMGetPINStatus_t::p2Status

8.1091.2.5 uint8_t unpack_dms_UIMGetPINStatus_t::p2UnblockRetriesLeft

8.1091.2.6 uint8_t unpack_dms_UIMGetPINStatus_t::p2VerifyRetriesLeft

8.1091.2.7 uint16_t unpack_dms_UIMGetPINStatus_t::Tlvresult

8.1092 unpack_dms_UIMSetControlKeyProtection_t Struct Reference

Data Fields

- uint8_t verifyRetriesLeft
- uint16_t Tlvresult

8.1092.1 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

8.1092.2 Field Documentation

8.1092.2.1 uint16_t unpack_dms_UIMSetControlKeyProtection_t::Tlvresult

8.1092.2.2 uint8_t unpack_dms_UIMSetControlKeyProtection_t::verifyRetriesLeft

8.1093 unpack_dms_UIMSetPINProtection_t Struct Reference

Data Fields

- uint8_t verifyRetriesLeft

- uint8_t unblockRetriesLeft
- uint16_t Tlvresult

8.1093.1 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. • 0xFF - Invalid data.
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. • 0xFF - Invalid data.
Tlvresult[OUT]	• Unpack Result

8.1093.2 Field Documentation

8.1093.2.1 uint16_t unpack_dms_UIMSetPINProtection_t::Tlvresult

8.1093.2.2 uint8_t unpack_dms_UIMSetPINProtection_t::unblockRetriesLeft

8.1093.2.3 uint8_t unpack_dms_UIMSetPINProtection_t::verifyRetriesLeft

8.1094 unpack_dms_UIMUnblockControlKey_t Struct Reference

Data Fields

- uint8_t unblockRetriesLeft
- uint16_t Tlvresult

8.1094.1 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 • 0xff - Invalid data.
Tlvresult[OUT]	• Unpack Result

8.1094.2 Field Documentation

8.1094.2.1 uint16_t unpack_dms_UIMUnblockControlKey_t::TlvResult

8.1094.2.2 uint8_t unpack_dms_UIMUnblockControlKey_t::unblockRetriesLeft

8.1095 unpack_fms_GetImagesPreference_t Struct Reference

Data Fields

- uint32_t ImageListSize
- FMSPrefImageList * pImageList
- uint16_t TlvResult

8.1095.1 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

8.1095.2 Field Documentation

8.1095.2.1 uint32_t unpack_fms_GetImagesPreference_t::ImageListSize

8.1095.2.2 FMSPrefImageList* unpack_fms_GetImagesPreference_t::pImageList

8.1095.2.3 uint16_t unpack_fms_GetImagesPreference_t::TlvResult

8.1096 unpack_fms_GetStoredImages_t Struct Reference

Data Fields

- uint32_t imagelistSize
- FMSImageList imageList
- uint16_t TlvResult

8.1096.1 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

8.1096.2 Field Documentation

8.1096.2.1 FMSImageList unpack_fms_GetStoredImages_t::imageList

8.1096.2.2 uint32_t unpack_fms_GetStoredImages_t::imagelistSize

8.1096.2.3 uint16_t unpack_fms_GetStoredImages_t::TlvResult

8.1097 unpack_fms_SetImagesPreference_t Struct Reference

Data Fields

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

8.1097.1 Detailed Description

This structure contains the Set Images Preference unpack

Parameters

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

8.1097.2 Field Documentation

8.1097.2.1 uint8_t unpack_fms_SetImagesPreference_t::ImageTypes[255]

8.1097.2.2 uint32_t unpack_fms_SetImagesPreference_t::ImageTypesSize

8.1097.2.3 uint16_t unpack_fms_SetImagesPreference_t::TlvResult

8.1098 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

8.1098.1 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.
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 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

8.1098.2 Field Documentation

8.1098.2.1 uint32_t* unpack_loc_BestAvailPos_Ind_t::pAltitudeWrtEllipsoid

8.1098.2.2 uint32_t* unpack_loc_BestAvailPos_Ind_t::pAltitudeWrtMeanSeaLevel

8.1098.2.3 loc_gpsTime* unpack_loc_BestAvailPos_Ind_t::pGpsTime

8.1098.2.4 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHeading

8.1098.2.5 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHeadingUnc

8.1098.2.6 uint8_t* unpack_loc_BestAvailPos_Ind_t::pHorCirConf

8.1098.2.7 uint8_t* unpack_loc_BestAvailPos_Ind_t::pHorEllpConf

8.1098.2.8 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorReliability

8.1098.2.9 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncCircular

8.1098.2.10 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncEllipseOrientAzimuth

8.1098.2.11 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncEllipseSemiMajor

8.1098.2.12 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncEllipseSemiMinor

8.1098.2.13 uint64_t* unpack_loc_BestAvailPos_Ind_t::pLatitude

8.1098.2.14 uint64_t* unpack_loc_BestAvailPos_Ind_t::pLongitude

8.1098.2.15 uint32_t* unpack_loc_BestAvailPos_Ind_t::pMagneticDeviation

8.1098.2.16 loc_precisionDilution* unpack_loc_BestAvailPos_Ind_t::pPrecisionDilution

8.1098.2.17 loc_sensorDataUsage* unpack_loc_BestAvailPos_Ind_t::pSensorDataUsage

8.1098.2.18 uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedHorizontal

8.1098.2.19 uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedUnc

8.1098.2.20 uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedVertical

8.1098.2.21 uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedVerticalUnc

8.1098.2.22 loc_svUsedforFix* unpack_loc_BestAvailPos_Ind_t::pSvUsedforFix

8.1098.2.23 uint32_t* unpack_loc_BestAvailPos_Ind_t::pTechnologyMask

8.1098.2.24 uint32_t* unpack_loc_BestAvailPos_Ind_t::pTimeSrc

8.1098.2.25 uint64_t* unpack_loc_BestAvailPos_Ind_t::pTimestampUtc

8.1098.2.26 uint32_t* unpack_loc_BestAvailPos_Ind_t::pTimeUnc

8.1098.2.27 uint8_t* unpack_loc_BestAvailPos_Ind_t::pVertConfidence

8.1098.2.28 uint32_t* unpack_loc_BestAvailPos_Ind_t::pVertReliability

8.1098.2.29 uint32_t* unpack_loc_BestAvailPos_Ind_t::pVertUnc

8.1098.2.30 uint32_t* unpack_loc_BestAvailPos_Ind_t::pXid

8.1098.2.31 uint32_t unpack_loc_BestAvailPos_Ind_t::status

8.1098.2.32 uint16_t unpack_loc_BestAvailPos_Ind_t::Tlvresult

8.1099 unpack_loc_Delete_Assist_Data_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1099.1 Detailed Description

This structure contains LOC delete assist data unpack

Parameters

Tlvresult	• Unpack result.

8.1099.2 Field Documentation

8.1099.2.1 uint16_t unpack_loc_Delete_Assist_Data_t::Tlvresult

8.1100 unpack_loc_DeleteAssistData_Ind_t Struct Reference

Data Fields

- uint32_t status
- uint16_t Tlvresult

8.1100.1 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 – 0xffffffff - Invalid data.
Tlvresult	• unpack result

8.1100.2 Field Documentation

8.1100.2.1 uint32_t unpack_loc_DeleteAssistData_Ind_t::status

8.1100.2.2 uint16_t unpack_loc_DeleteAssistData_Ind_t::TlvResult

8.1101 unpack_loc_EngineState_Ind_t Struct Reference

Data Fields

- uint32_t engineState
- uint16_t TlvResult

8.1101.1 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 • 0xffffffff - Invalid data.
TlvResult	• unpack result

8.1101.2 Field Documentation

8.1101.2.1 uint32_t unpack_loc_EngineState_Ind_t::engineState

8.1101.2.2 uint16_t unpack_loc_EngineState_Ind_t::TlvResult

8.1102 unpack_loc_EventNMEA_Ind_t Struct Reference

Data Fields

- char NMEAData [201]
- uint16_t TlvResult

8.1102.1 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
TlvResult	• unpack result

8.1102.2 Field Documentation

8.1102.2.1 char unpack_loc_EventNMEA_Ind_t::NMEAData[201]

8.1102.2.2 uint16_t unpack_loc_EventNMEA_Ind_t::TlvResult

8.1103 unpack_loc_EventRegister_t Struct Reference

Data Fields

- uint16_t TlvResult

8.1103.1 Detailed Description

This structure contains Event Register unpack

Parameters

TlvResult	• Unpack result.

8.1103.2 Field Documentation

8.1103.2.1 uint16_t unpack_loc_EventRegister_t::TlvResult

8.1104 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

8.1104.1 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 – 0xffffffff - Invalid data.
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 – 0xffffffff - Invalid data.
pIPv4AddrInfo	• See loc_IPv4Info for more information.
pIPv6AddrInfo	• See loc_IPv6Info for more information.
pURL	• See loc_urlAddr for more information.

8.1104.2 Field Documentation

8.1104.2.1 loc_IPv4Info* unpack_loc_GetServer_Ind_t::pIPv4AddrInfo

8.1104.2.2 loc_IPv6Info* unpack_loc_GetServer_Ind_t::pIPv6AddrInfo

8.1104.2.3 loc_urlAddr* unpack_loc_GetServer_Ind_t::pURL

8.1104.2.4 uint32_t unpack_loc_GetServer_Ind_t::serverStatus

8.1104.2.5 uint32_t unpack_loc_GetServer_Ind_t::serverType

8.1104.2.6 uint16_t unpack_loc_GetServer_Ind_t::TlvResult

8.1105 unpack_loc_GnssSvInfo_Ind_t Struct Reference

Data Fields

- uint8_t altitudeAssumed
- loc_satelliteInfo * pSatelliteInfo
- uint16_t TlvResult

8.1105.1 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 – 0xff - Invalid data.
pSatelliteInfo	• See loc_satelliteInfo for more information. • NULL pointer - Invalid data.
TlvResult	• unpack result

8.1105.2 Field Documentation

8.1105.2.1 uint8_t unpack_loc_GnssSvInfo_Ind_t::altitudeAssumed

8.1105.2.2 loc_satelliteInfo* unpack_loc_GnssSvInfo_Ind_t::pSatelliteInfo

8.1105.2.3 uint16_t unpack_loc_GnssSvInfo_Ind_t::TlvResult

8.1106 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

8.1106.1 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 – 0xffffffff - Invalid data.
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 • NULL pointer - Invalid data.
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 • NULL pointer - Invalid data.
pHorUncCircular	• Horizontal position uncertainty. • Units - Meters • NULL pointer - Invalid data.
pHorUncEllipse-SemiMinor	• Semi-minor axis of horizontal elliptical uncertainty. • Units - Meters • NULL pointer - Invalid data.
pHorUncEllipse-SemiMajor	• Semi-major axis of horizontal elliptical uncertainty. • Units: Meters • NULL pointer - Invalid data.
pHorUncEllipse-OrientAzimuth	• Elliptical horizontal uncertainty azimuth of orientation. • Units - Decimal degrees • Range - 0 to 180 • NULL pointer - Invalid data.
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 • NULL pointer - Invalid data.

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 – NULL pointer - Invalid data.
pSpeed-Horizontal	• Horizontal speed. • Units - Meters/second • NULL pointer - Invalid data.
pSpeedUnc	• 3-D Speed uncertainty. • Units - Meters/second. • NULL pointer - Invalid data.
pAltitudeWrt-Ellipsoid	• Altitude With Respect to WGS84 Ellipsoid. • Units - Meters • Range -500 to 15883 • NULL pointer - Invalid data.
pAltitudeWrt-MeanSeaLevel	• Altitude With Respect to Sea Level. • Units - Meters • NULL pointer - Invalid data.
pVertUnc	• Vertical uncertainty. • Units - Meters • NULL pointer - Invalid data.
pVertConfidence	• Vertical uncertainty confidence. • Units - Percentage • Range 0 to 99 • NULL pointer - Invalid data.
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 – NULL pointer - Invalid data.

pSpeedVertical	• Vertical speed. • Units - Meters/second • NULL pointer - Invalid data.
pHeading	• Heading. • Units - Degree • Range 0 to 359.999 • NULL pointer - Invalid data.
pHeadingUnc	• Heading uncertainty. • Units - Degree • Range 0 to 359.999 • NULL pointer - Invalid data.
pMagnetic-Deviation	• Difference between the bearing to true north and the bearing shown on a magnetic compass. The deviation is positive when the magnetic north is east of true north. • NULL pointer - Invalid data.
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 – NULL pointer - Invalid data.
-pPrecision-Dilution	• See loc_precisionDilution for more information • NULL pointer - Invalid data.
pTimestampUtc	• UTC timestamp • Units - Milliseconds since Jan. 1, 1970 • NULL pointer - Invalid data.
pLeapSeconds	• Leap second information. If leapSeconds is not available, timestampUtc is calculated based on a hard-coded value for leap seconds. • Units - Seconds • NULL pointer - Invalid data.

-pGpsTime	• See loc_gpsTime for more information • NULL pointer - Invalid data.
pTimeUnc	• Time uncertainty. • Units - Milliseconds • NULL pointer - Invalid data.
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. – NULL pointer - Invalid data.
-pSensorData-Usage	• See loc_sensorDataUsage for more information – NULL pointer - Invalid data.
pFixId	• Fix count for the session. Starts with 0 and increments by one for each successive position report for a particular session. • NULL pointer - Invalid data.
-pSvUsedforFix	• See loc_svUsedforFix for more information – NULL pointer - Invalid data.
pAltitude-Assumed	• Indicates whether altitude is assumed or calculated.

- Value
  - 0x00 - Altitude is calculated

- 0x01 - Altitude is assumed
- NULL pointer - Invalid data.

8.1106.2 Field Documentation

8.1106.2.1 uint8_t* unpack_loc_PositionRpt_Ind_t::pAltitudeAssumed

8.1106.2.2 uint32_t* unpack_loc_PositionRpt_Ind_t::pAltitudeWrtEllipsoid

8.1106.2.3 uint32_t* unpack_loc_PositionRpt_Ind_t::pAltitudeWrtMeanSeaLevel

8.1106.2.4 uint32_t* unpack_loc_PositionRpt_Ind_t::pFixId

8.1106.2.5 loc_gpsTime* unpack_loc_PositionRpt_Ind_t::pGpsTime

8.1106.2.6 uint32_t* unpack_loc_PositionRpt_Ind_t::pHeading

8.1106.2.7 uint32_t* unpack_loc_PositionRpt_Ind_t::pHeadingUnc

8.1106.2.8 uint8_t* unpack_loc_PositionRpt_Ind_t::pHorConfidence

8.1106.2.9 uint32_t* unpack_loc_PositionRpt_Ind_t::pHorReliability

8.1106.2.10 uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncCircular

8.1106.2.11 uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncEllipseOrientAzimuth

8.1106.2.12 uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncEllipseSemiMajor

8.1106.2.13 uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncEllipseSemiMinor

8.1106.2.14 uint64_t* unpack_loc_PositionRpt_Ind_t::pLatitude

8.1106.2.15 uint8_t* unpack_loc_PositionRpt_Ind_t::pLeapSeconds

8.1106.2.16 uint64_t* unpack_loc_PositionRpt_Ind_t::pLongitude

8.1106.2.17 uint32_t* unpack_loc_PositionRpt_Ind_t::pMagneticDeviation

8.1106.2.18 loc_precisionDilution* unpack_loc_PositionRpt_Ind_t::pPrecisionDilution

8.1106.2.19 loc_sensorDataUsage* unpack_loc_PositionRpt_Ind_t::pSensorDataUsage

8.1106.2.20 uint32_t* unpack_loc_PositionRpt_Ind_t::pSpeedHorizontal

8.1106.2.21 uint32_t* unpack_loc_PositionRpt_Ind_t::pSpeedUnc

8.1106.2.22 uint32_t* unpack_loc_PositionRpt_Ind_t::pSpeedVertical

8.1106.2.23 loc_svUsedforFix* unpack_loc_PositionRpt_Ind_t::pSvUsedforFix

8.1106.2.24 uint32_t* unpack_loc_PositionRpt_Ind_t::pTechnologyMask

8.1106.2.25 uint32_t* unpack_loc_PositionRpt_Ind_t::pTimeSrc

8.1106.2.26 uint64_t* unpack_loc_PositionRpt_Ind_t::pTimestampUtc

8.1106.2.27 uint32_t* unpack_loc_PositionRpt_Ind_t::pTimeUnc

8.1106.2.28 uint8_t* unpack_loc_PositionRpt_Ind_t::pVertConfidence

8.1106.2.29 uint32_t* unpack_loc_PositionRpt_Ind_t::pVertReliability

8.1106.2.30 uint32_t* unpack_loc_PositionRpt_Ind_t::pVertUnc

8.1106.2.31 uint8_t unpack_loc_PositionRpt_Ind_t::sessionId

8.1106.2.32 uint32_t unpack_loc_PositionRpt_Ind_t::sessionStatus

8.1106.2.33 uint16_t unpack_loc_PositionRpt_Ind_t::Tlvresult

8.1107 unpack_loc_SetExtPowerConfig_Ind_t Struct Reference

Data Fields

- uint32_t status
- uint16_t Tlvresult

8.1107.1 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 - 0xffffffff - Invalid data.
Tlvresult	• unpack result

8.1107.2 Field Documentation

8.1107.2.1 uint32_t unpack_loc_SetExtPowerConfig_Ind_t::status

8.1107.2.2 uint16_t unpack_loc_SetExtPowerConfig_Ind_t::Tlvresult

8.1108 unpack_loc_SetExtPowerState_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1108.1 Detailed Description

This structure contains Set Ext Power State unpack

Parameters

Tlvresult	• Unpack result.

8.1108.2 Field Documentation

8.1108.2.1 uint16_t unpack_loc_SetExtPowerState_t::Tlvresult

8.1109 unpack_loc_SetOperationMode_Ind_t Struct Reference

Data Fields

- uint32_t status
- uint16_t Tlvresult

8.1109.1 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 – 0xffffffff - Invalid data.

Tlvresult	• unpack result

8.1109.2 Field Documentation

8.1109.2.1 uint32_t unpack_loc_SetOperationMode_Ind_t::status

8.1109.2.2 uint16_t unpack_loc_SetOperationMode_Ind_t::Tlvresult

8.1110 unpack_loc_SetOperationMode_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_SetOperationMode_t::Tlvresult

8.1111 unpack_loc_SLQSLOCGetBestAvailPos_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1111.1 Detailed Description

This structure contains Set Operation Mode unpack

Parameters

Tlvresult	• Unpack result.

8.1111.2 Field Documentation

8.1111.2.1 uint16_t unpack_loc_SLQSLOCGetBestAvailPos_t::Tlvresult

8.1112 unpack_loc_Start_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1112.1 Detailed Description

This structure contains Start LOC unpack

Parameters

Tlvresult	• Unpack result.

8.1112.2 Field Documentation

8.1112.2.1 uint16_t unpack_loc_Start_t::Tlvresult

8.1113 unpack_loc_Stop_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1113.1 Detailed Description

This structure contains Stop LOC unpack

Parameters

Tlvresult	• Unpack result.

8.1113.2 Field Documentation

8.1113.2.1 uint16_t unpack_loc_Stop_t::Tlvresult

8.1114 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

8.1114.1 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

8.1114.2 Field Documentation

8.1114.2.1 uint32_t unpack_nas_GetCDMANetworkParameters_t::Application

8.1114.2.2 uint32_t unpack_nas_GetCDMANetworkParameters_t::Broadcast

8.1114.2.3 uint8_t unpack_nas_GetCDMANetworkParameters_t::CustomSCP

8.1114.2.4 uint8_t unpack_nas_GetCDMANetworkParameters_t::ForceRev0

8.1114.2.5 uint32_t unpack_nas_GetCDMANetworkParameters_t::Protocol

8.1114.2.6 uint8_t unpack_nas_GetCDMANetworkParameters_t::RegForeignNID

8.1114.2.7 uint8_t unpack_nas_GetCDMANetworkParameters_t::RegForeignSID

8.1114.2.8 uint8_t unpack_nas_GetCDMANetworkParameters_t::RegHomeSID

8.1114.2.9 uint32_t unpack_nas_GetCDMANetworkParameters_t::Roaming

8.1114.2.10 uint8_t unpack_nas_GetCDMANetworkParameters_t::SCI

8.1114.2.11 uint8_t unpack_nas_GetCDMANetworkParameters_t::SCM

8.1115 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

8.1115.1 Detailed Description

Structure for HomeNetwork3GPP2 unpack.

Parameters

pMCC	• Mobile country code (UMTS only). • NULL for invalid data
pMNC	• Mobile network code (UMTS only). • NULL for invalid data
nameSize	• Maximum number of characters (including NULL terminator) that network name array can contain (UMTS only).
pName	• Network name or description represented as a NULL terminated string (empty string returned when unknown) (UMTS only). • NULL for invalid data
pSID	• Home network system ID – 0xFFFF - Unknown. – Only applies to cdma2000 • NULL for invalid data

pNID	• Home network ID – 0xFFFF - Unknown. – Only applies to cdma2000 • NULL for invalid data
pNw2MCC	• Mobile country code (3GPP2 only). • Range : 0 to 999 • NULL for invalid data
pNw2MNC	• Mobile network code (3GPP2 only). • Range : 0 to 999 • NULL for invalid data
pNw2DescDisp	• Network Name Display (3GPP2 only). -Valid Value – 0x00 - Do not display – 0x01 - Display – 0xFF - Unknown • NULL for invalid data
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 • NULL for invalid data
pNw2DescLen	• Network Description Length (3GPP2 only). • NULL for invalid data
pNw2Name	• Network Name (3GPP2 only). • NULL for invalid data

8.1115.2 Field Documentation

8.1115.2.1 uint8_t unpack_nas_GetHomeNetwork3GPP2_t::nameSize

8.1115.2.2 uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pMCC

8.1115.2.3 uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pMNC

8.1115.2.4 uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pName

8.1115.2.5 uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pNID

8.1115.2.6 uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2DescDisp

8.1115.2.7 uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2DescEnc

8.1115.2.8 uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2DescLen

8.1115.2.9 uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2MCC

8.1115.2.10 uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2MNC

8.1115.2.11 uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2Name

8.1115.2.12 uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pSID

8.1116 unpack_nas_GetHomeNetwork_t Struct Reference

Data Fields

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

8.1116.1 Detailed Description

This structure contains unpack get get home network parameters.

Parameters

mcc	• Mobile country code (UMTS only). • A 16-bit integer representation of MCC. Range: 0 to 999. • 0xFFFF - Invalid data.
mnc	• Mobile network code (UMTS only). • A 16-bit integer representation of MNC. Range: 0 to 999. • 0xFFFF - Invalid data.
name	• Network name or description represented as a NULL terminated string (empty string returned when unknown) (UMTS only).
sid	• Home network system ID – 0xFFFF - Unknown. – Only applies to cdma2000

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

8.1116.2 Field Documentation

8.1116.2.1 uint16_t unpack_nas_GetHomeNetwork_t::mcc

8.1116.2.2 uint16_t unpack_nas_GetHomeNetwork_t::mnc

8.1116.2.3 char unpack_nas_GetHomeNetwork_t::name[255]

8.1116.2.4 uint16_t unpack_nas_GetHomeNetwork_t::nid

8.1116.2.5 uint16_t unpack_nas_GetHomeNetwork_t::sid

8.1117 unpack_nas_GetNetworkPreference_t Struct Reference

Data Fields

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

8.1117.1 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 – 0xffffffff - invalid data.

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 – 0xffffffff - invalid data.
Persistent-TechnologyPref[-OUT]	• Bit field representing persistent radio technology preference – Same representation as the pTechnologyPref parameter • 0xffffffff - invalid data.
Tlvresult	• unpack result

8.1117.2 Field Documentation

8.1117.2.1 uint32_t unpack_nas_GetNetworkPreference_t::ActiveTechPref

8.1117.2.2 uint32_t unpack_nas_GetNetworkPreference_t::Duration

8.1117.2.3 uint32_t unpack_nas_GetNetworkPreference_t::PersistentTechPref

8.1117.2.4 uint16_t unpack_nas_GetNetworkPreference_t::Tlvresult

8.1118 unpack_nas_GetRFInfo_t Struct Reference

Data Fields

- uint8_t instancesSize
- RFBandInfoElements RFBandInfoElements [255]

8.1118.1 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

8.1118.2 Field Documentation

8.1118.2.1 uint8_t unpack_nas_GetRFInfo_t::instancesSize

8.1118.2.2 RFBandInfoElements unpack_nas_GetRFInfo_t::RFBandInfoElements[255]

8.1119 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]

8.1119.1 Detailed Description

This structure contains unpack get get serving network parameters.

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 • 0xffffffff - invaild data.
MCC	• Mobile country code • 0xffff - invalid dta.
MNC	• Mobile network code • 0xffff - invalid dta.
nameSize	• Maximum number of characters (including NULL terminator) that network name array can contain; applicable only for UMTS networks
Name	• Network name or description represented as a NULL terminated string; empty string is returned when unknown; applicable only for UMTS networks
DataCapsLen[IN/OUT]	• Upon input, the maximum number of elements the data capabilities array can contain. • 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 uint32_t unpack_nas_GetServingNetwork_t::CSDomain

8.1119.2.2 uint8_t unpack_nas_GetServingNetwork_t::DataCaps[255]

8.1119.2.3 uint8_t unpack_nas_GetServingNetwork_t::DataCapsLen

8.1119.2.4 uint16_t unpack_nas_GetServingNetwork_t::MCC

8.1119.2.5 uint16_t unpack_nas_GetServingNetwork_t::MNC

8.1119.2.6 uint8_t unpack_nas_GetServingNetwork_t::Name[255]

8.1119.2.7 uint8_t unpack_nas_GetServingNetwork_t::nameSize

8.1119.2.8 uint32_t unpack_nas_GetServingNetwork_t::PSDomain

8.1119.2.9 uint8_t unpack_nas_GetServingNetwork_t::RadioIfaces[255]

8.1119.2.10 uint8_t unpack_nas_GetServingNetwork_t::RadioIfacesSize

8.1119.2.11 uint32_t unpack_nas_GetServingNetwork_t::RAN

8.1119.2.12 uint32_t unpack_nas_GetServingNetwork_t::RegistrationState

8.1119.2.13 uint32_t unpack_nas_GetServingNetwork_t::Roaming

8.1120 unpack_nas_GetServingNetworkCapabilities_t Struct Reference

Data Fields

- uint8_t DataCapsLen
- uint8_t DataCaps [255]

8.1120.1 Detailed Description

This structure contains unpack get serving network capabilities parameters.

Parameters

DataCapsLen	• Upon input, the maximum number of elements the data capabilities array can contain. • 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

8.1120.2 Field Documentation

8.1120.2.1 uint8_t unpack_nas_GetServingNetworkCapabilities_t::DataCaps[255]

8.1120.2.2 uint8_t unpack_nas_GetServingNetworkCapabilities_t::DataCapsLen

8.1121 unpack_nas_GetSignalStrengths_t Struct Reference

Data Fields

- uint32_t len
- signed char rssi [8]
- uint32_t radio [8]

8.1121.1 Detailed Description

This structure contains unpack get signal strengths parameters.

Parameters

len	number of rssi & radio items following
rssi	signal strength array
radio	radio interface array

8.1121.2 Field Documentation

8.1121.2.1 uint32_t unpack_nas_GetSignalStrengths_t::len

8.1121.2.2 uint32_t unpack_nas_GetSignalStrengths_t::radio[8]

8.1121.2.3 signed char unpack_nas_GetSignalStrengths_t::rssi[8]

8.1122 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

8.1122.1 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 • 0xffffffff - invalid data.

8.1122.2 Field Documentation

8.1122.2.1 nas_QmiNas3GppNetworkInfo* unpack_nas_PerformNetworkScan_t::p3GppNetworkInfoInstances

8.1122.2.2 uint8_t* unpack_nas_PerformNetworkScan_t::p3GppNetworkInstanceSize

8.1122.2.3 nas_QmisNasPcsDigit* unpack_nas_PerformNetworkScan_t::pPCSInstance

8.1122.2.4 uint8_t* unpack_nas_PerformNetworkScan_t::pPCSInstanceSize

8.1122.2.5 nas_QmiNas3GppNetworkRAT* unpack_nas_PerformNetworkScan_t::pRATInstance

8.1122.2.6 uint8_t* unpack_nas_PerformNetworkScan_t::pRATInstanceSize

8.1122.2.7 uint32_t* unpack_nas_PerformNetworkScan_t::pScanResult

8.1123 unpack_nas_SetDataCapabilitiesCallback_ind_t Struct Reference

Data Fields

- uint8_t dataCapsSize
- uint8_t dataCaps [255]

8.1123.1 Detailed Description

This structure contains unpack set data capabilities callback indication parameters.

Parameters

dataCapsSize	Number of Data Capabilities
dataCaps	Data Capabilities

8.1123.2 Field Documentation

8.1123.2.1 uint8_t unpack_nas_SetDataCapabilitiesCallback_ind_t::dataCaps[255]

8.1123.2.2 uint8_t unpack_nas_SetDataCapabilitiesCallback_ind_t::dataCapsSize

8.1124 unpack_nas_SetEventReportInd_t Struct Reference

Data Fields

- nas_SignalStrengthTlv SSTlv
- nas_RFInfoTlv RFTlv
- nas_RejectReasonTlv RRTlv
- nas_SLQSSignalStrengthsTlv SLQSSSTlv

8.1124.1 Detailed Description

This structure contains unpack set event report indication parameters.

Parameters

SSTlv	signal strength tlv
RFTlv	RF tlv
RRTlv	RR tlv
SLQSSSTlv	signal strength complete info tlv

8.1124.2 Field Documentation

8.1124.2.1 nas_RFInfoTlv unpack_nas_SetEventReportInd_t::RFTlv

8.1124.2.2 nas_RejectReasonTlv unpack_nas_SetEventReportInd_t::RRTlv

8.1124.2.3 nas_SLQSSignalStrengthsTlv unpack_nas_SetEventReportInd_t::SLQSSSTlv

8.1124.2.4 nas_SignalStrengthTlv unpack_nas_SetEventReportInd_t::SSTlv

8.1125 unpack_nas_SetNasLTECphyCaIndCallback_ind_t Struct Reference

Data Fields

- nas_PhyCaAggScellIndType sPhyCaAggScellIndType
- nas_PhyCaAggScellDlBw sPhyCaAggScellDlBw
- nas_PhyCaAggScellInfo sPhyCaAggScellInfo
- nas_PhyCaAggPcellInfo sPhyCaAggPcellInfo
- nas_PhyCaAggScellIndex sPhyCaAggScellIndex

8.1125.1 Detailed Description

Structure for storing the LTE PHY CA indication parameters.

Parameters

pPhyCaAgg-ScellIndType	• See nas_PhyCaAggScellIndType for more information.
sPhyCaAgg-ScellDlBw	• See nas_PhyCaAggScellDlBw for more information.
sPhyCaAgg-ScellInfo	• See nas_PhyCaAggScellInfo for more information.
sPhyCaAgg-PcellInfo	• See nas_PhyCaAggPcellInfo for more information.
sPhyCaAgg-ScellIndex	• See nas_PhyCaAggScellIndex for more information.

8.1125.2 Field Documentation

8.1125.2.1 nas_PhyCaAggPcellInfo unpack_nas_SetNasLTECphyCaIndCallback_ind_t::sPhyCaAggPcellInfo

8.1125.2.2 nas_PhyCaAggSceIIDlBw unpack_nas_SetNasLTECphyCaIndCallback_ind_t::sPhyCaAggSceIIDlBw

8.1125.2.3 nas_PhyCaAggSceIIIndex unpack_nas_SetNasLTECphyCaIndCallback_ind_t::sPhyCaAggSceIIIndex

8.1125.2.4 nas_PhyCaAggSceIIIndType unpack_nas_SetNasLTECphyCaIndCallback_ind_t::sPhyCaAggSceIIIndType

8.1125.2.5 nas_PhyCaAggSceIIInfo unpack_nas_SetNasLTECphyCaIndCallback_ind_t::sPhyCaAggSceIIInfo

8.1126 unpack_nas_SetNetworkPreference_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1126.1 Detailed Description

This structure contains unpack set netwokr preference parameters.

Parameters

Tlvresult	• unpack result

8.1126.2 Field Documentation

8.1126.2.1 uint16_t unpack_nas_SetNetworkPreference_t::Tlvresult

8.1127 unpack_nas_SetRoamingIndicatorCallback_ind_t Struct Reference

Data Fields

- uint8_t roaming

8.1127.1 Detailed Description

This structure contains unpack set roaming indicator callback indication parameters.

Parameters

roaming	• Roaming Indication – 0 - Roaming – 1 - Home – 2 - Roaming partner – >2 - Operator defined values

8.1127.2 Field Documentation

8.1127.2.1 uint8_t unpack_nas_SetRoamingIndicatorCallback_ind_t::roaming

8.1128 unpack_nas_SetServingSystemCallback_ind_t Struct Reference

Data Fields

- NASServingSystemInfo SSInfo
- uint16_t Tlvresult

8.1128.1 Detailed Description

This structure contains unpack set serving system callback indication parameters.

Parameters

SSInfo	• Serving system parameters information – See NASServingSystemInfo for more details
Tlvresult	• unpack result

8.1128.2 Field Documentation

8.1128.2.1 NASServingSystemInfo unpack_nas_SetServingSystemCallback_ind_t::SSInfo

8.1128.2.2 uint16_t unpack_nas_SetServingSystemCallback_ind_t::Tlvresult

8.1129 unpack_nas_SLQSGetErrorRate_t Struct Reference

Data Fields

- uint16_t * pCDMAFrameErrRate
- uint16_t * pHDRPackErrRate
- uint8_t * pGSMBER
- uint8_t * pWCDMABER

8.1129.1 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. • NULL for invalid data

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. • NULL for invalid data
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. • NULL for invalid 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. • NULL for invalid data

8.1129.2 Field Documentation

8.1129.2.1 uint16_t* unpack_nas_SLQSGetErrorRate_t::pCDMAFrameErrRate

8.1129.2.2 uint8_t* unpack_nas_SLQSGetErrorRate_t::pGSMBER

8.1129.2.3 uint16_t* unpack_nas_SLQSGetErrorRate_t::pHDRPackErrRate

8.1129.2.4 uint8_t* unpack_nas_SLQSGetErrorRate_t::pWCDMABER

8.1130 unpack_nas_SlqsGetLTECphyCAInfo_t Struct Reference

Data Fields

- NasGetLTECphyCaInfo LTECphyCaInfo
- uint16_t Tlvresult

8.1130.1 Detailed Description

This structure contains unpack get carrier aggregation event information parameters.

Parameters

LTECphyCa	• Carrier aggregation event information – See NasGetLTECphyCaInfo for more details
Tlvresult	• unpack result

8.1130.2 Field Documentation

8.1130.2.1 NasGetLTECphyCaInfo unpack_nas_SlqsGetLTECphyCAInfo_t::LTECphyCaInfo

8.1130.2.2 uint16_t unpack_nas_SlqsGetLTECphyCAInfo_t::Tlvresult

8.1131 unpack_nas_SLQSGetNetworkTime_t Struct Reference

Data Fields

- nas_timeInfo * p3GPP2TimeInfo
- nas_timeInfo * p3GPPTimeInfo

8.1131.1 Detailed Description

This structure contains information about the GetNetworkTime response parameters.

Parameters

p3GPP2Time-Info	[Optional] • See nas_timeInfo for more information • NULL for invalid data.
p3GPPTimeInfo	[Optional] • See nas_timeInfo for more information • NULL for invalid data.

8.1131.2 Field Documentation

8.1131.2.1 nas_timeInfo* unpack_nas_SLQSGetNetworkTime_t::p3GPP2TimeInfo

8.1131.2.2 nas_timeInfo* unpack_nas_SLQSGetNetworkTime_t::p3GPPTimeInfo

8.1132 unpack_nas_SLQSGetOperatorNameData_t Struct Reference

Data Fields

- nas_serviceProviderName * pSrvcProviderName
- nas_operatorPLMNList * pOperatorPLMNList
- nas_PLMNNetworkName * pPLMNNetworkName
- nas_operatorNameString * pOperatorNameString
- nas_PLMNNetworkNameData * pNITZInformation

8.1132.1 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. • NULL for invalid data
pOperatorPLM-NList	• Refer nas_operatorPLMNList for details (Optional). • Can provide NULL if this parameter is not required. • NULL for invalid data
pPLMNNetwork-Name	• Refer nas_PLMNNetworkName for details (Optional). • Can provide NULL if this parameter is not required. • NULL for invalid data
pOperatorName-String	• Refer nas_operatorNameString for details (Optional). • Can provide NULL if this parameter is not required. • NULL for invalid data
pNITZ-Information	• Refer nas_PLMNNetworkNameData for details (Optional). • Can provide NULL if this parameter is not required. • NULL for invalid data

8.1132.2 Field Documentation

8.1132.2.1 nas_PLMNNetworkNameData∗ unpack_nas_SLQSGetOperatorNameData_t::pNITZInformation

8.1132.2.2 nas_operatorNameString∗ unpack_nas_SLQSGetOperatorNameData_t::pOperatorNameString

8.1132.2.3 nas_operatorPLMNList∗ unpack_nas_SLQSGetOperatorNameData_t::pOperatorPLMNList

8.1132.2.4 nas_PLMNNetworkName∗ unpack_nas_SLQSGetOperatorNameData_t::pPLMNNetworkName

8.1132.2.5 nas_serviceProviderName∗ unpack_nas_SLQSGetOperatorNameData_t::pSrvcProviderName

8.1133 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]

8.1133.1 Detailed Description

This structure contains unpack get operator name for specificed network parameters.

Parameters

spnEncoding	• Coding scheme used for service provider name. This value is ignored if spn_len is zero Values: – 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 – Note: This value is ignored if spnLength is zero.
spnLength	• Length of SPN which follows • Note: This SPN value is ignored if spnLength is zero and spnEncoding is 0xff.
spn	• Service Provider name string
shortNameEn	• Coding scheme used for PLMN short name. This value is ignored if PLMN short name length is zero Values: – 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 – Note: This value is ignored if shortNameLen is zero.
shortNameCI	• Indicates whether the country initials are to be added to the shortName. This value is ignored if shortNameLen is zero. Values: – 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 – Note: This value is ignored if shortNameLen is zero.
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 – Note: This value is ignored if shortNameLen is zero.
shortNameLen	• Length of shortName which follows • Note: This shortName value is ignored if shortNameLen is zero and shortNameEn is 0xff.

shortName	• PLMN short name
longNameEn	• Coding scheme used for PLMN long name. This value is ignored if PLMN long name length is zero Values: – 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 – Note: This value is ignored if shortNameLen is zero.
longNameCI	• Indicates whether the country initials are to be added to the longName. This value is ignored if longNameLen is zero. Values: – 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 – Note: This value is ignored if shortNameLen is zero.
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 – Note: This value is ignored if shortNameLen is zero.
longNameLen	• Length of longName which follows • Note: This longName value is ignored if longNameLen is zero and longNameEn is 0xff.
longName	• PLMN long name

8.1133.2 Field Documentation

8.1133.2.1 char unpack_nas_SLQSGetPLMNName_t::longName[255]

8.1133.2.2 uint8_t unpack_nas_SLQSGetPLMNName_t::longNameCI

8.1133.2.3 uint8_t unpack_nas_SLQSGetPLMNName_t::longNameEn

8.1133.2.4 uint8_t unpack_nas_SLQSGetPLMNName_t::longNameLen

8.1133.2.5 uint8_t unpack_nas_SLQSGetPLMNName_t::longNameSB

8.1133.2.6 uint8_t unpack_nas_SLQSGetPLMNName_t::shortName[255]

8.1133.2.7 uint8_t unpack_nas_SLQSGetPLMNName_t::shortNameCI

8.1133.2.8 uint8_t unpack_nas_SLQSGetPLMNName_t::shortNameEn

8.1133.2.9 char unpack_nas_SLQSGetPLMNName_t::shortNameLen

8.1133.2.10 uint8_t unpack_nas_SLQSGetPLMNName_t::shortNameSB

8.1133.2.11 char unpack_nas_SLQSGetPLMNName_t::spn[255]

8.1133.2.12 uint8_t unpack_nas_SLQSGetPLMNName_t::spnEncoding

8.1133.2.13 uint8_t unpack_nas_SLQSGetPLMNName_t::spnLength

8.1134 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

8.1134.1 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
RoamIndicator-Val	• 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

8.1134.2 Field Documentation

8.1134.2.1 uint16_t unpack_nas_SLQSGetServingSystem_t::BasestationID

8.1134.2.2 uint32_t unpack_nas_SLQSGetServingSystem_t::BasestationLatitude

8.1134.2.3 uint32_t unpack_nas_SLQSGetServingSystem_t::BasestationLongitude

8.1134.2.4 nas_callBarStatus unpack_nas_SLQSGetServingSystem_t::CallBarStatus

8.1134.2.5 uint8_t unpack_nas_SLQSGetServingSystem_t::CDMA_P_Rev

8.1134.2.6 nas_CDMASysInfoExt unpack_nas_SLQSGetServingSystem_t::CDMASystemInfoExt

8.1134.2.7 uint32_t unpack_nas_SLQSGetServingSystem_t::CellID

8.1134.2.8 uint8_t unpack_nas_SLQSGetServingSystem_t::ConcSvcInfo

8.1134.2.9 nas_currentPLMN unpack_nas_SLQSGetServingSystem_t::CurrentPLMN

8.1134.2.10 nas_dataSrvCapabilities unpack_nas_SLQSGetServingSystem_t::DataSrvCapabilities

8.1134.2.11 uint8_t unpack_nas_SLQSGetServingSystem_t::DefaultRoamInd

8.1134.2.12 nas_detailSvcInfo unpack_nas_SLQSGetServingSystem_t::DetailedSvcInfo

8.1134.2.13 uint8_t unpack_nas_SLQSGetServingSystem_t::DTMInd

8.1134.2.14 nas_qaQmi3Gpp2TimeZone unpack_nas_SLQSGetServingSystem_t::Gpp2TimeZone

8.1134.2.15 uint8_t unpack_nas_SLQSGetServingSystem_t::GppNetworkDSTAdjustment

8.1134.2.16 uint8_t unpack_nas_SLQSGetServingSystem_t::GppTimeZone

8.1134.2.17 uint8_t unpack_nas_SLQSGetServingSystem_t::HdrPersonality

8.1134.2.18 uint16_t unpack_nas_SLQSGetServingSystem_t::Lac

8.1134.2.19 uint16_t unpack_nas_SLQSGetServingSystem_t::NetworkID

8.1134.2.20 uint8_t unpack_nas_SLQSGetServingSystem_t::PRLInd

8.1134.2.21 uint8_t unpack_nas_SLQSGetServingSystem_t::RoamIndicatorVal

8.1134.2.22 nas_roamIndList unpack_nas_SLQSGetServingSystem_t::RoamingIndicatorList

8.1134.2.23 nas_servSystem unpack_nas_SLQSGetServingSystem_t::ServingSystem

8.1134.2.24 uint16_t unpack_nas_SLQSGetServingSystem_t::SystemID

8.1134.2.25 uint16_t unpack_nas_SLQSGetServingSystem_t::TrackAreaCode

8.1135 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

8.1135.1 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

8.1135.2 Field Documentation

8.1135.2.1 nas_ecioListElement unpack_nas_SLQSGetSignalStrength_t::ecioList[18]

8.1135.2.2 uint16_t unpack_nas_SLQSGetSignalStrength_t::ecioListLen

8.1135.2.3 nas_errorRateListElement unpack_nas_SLQSGetSignalStrength_t::errorRateList[18]

8.1135.2.4 uint16_t unpack_nas_SLQSGetSignalStrength_t::errorRateListLen

8.1135.2.5 int32_t unpack_nas_SLQSGetSignalStrength_t::lo

8.1135.2.6 int16_t unpack_nas_SLQSGetSignalStrength_t::ltersrp

8.1135.2.7 int16_t unpack_nas_SLQSGetSignalStrength_t::ltesnr

8.1135.2.8 nas_rsrqInformation unpack_nas_SLQSGetSignalStrength_t::rsrqInfo

8.1135.2.9 nas_rxSignalStrengthListElement unpack_nas_SLQSGetSignalStrength_t::rxSignalStrengthList[18]

8.1135.2.10 uint16_t unpack_nas_SLQSGetSignalStrength_t::rxSignalStrengthListLen

8.1135.2.11 uint16_t unpack_nas_SLQSGetSignalStrength_t::signalStrengthReqMask

8.1135.2.12 uint8_t unpack_nas_SLQSGetSignalStrength_t::sinr

8.1136 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

8.1136.1 Detailed Description

This structure contains unpack get system information parameters.

Parameters

pCDMASrvStatusInfo	• See nas_SrvStatusInfo for more information. • NULL Pointer for invalid data.
pHDRSrvStatusInfo	• See nas_SrvStatusInfo for more information. • NULL Pointer for invalid data.
pGSMSrvStatusInfo	• See nas_GSMSrvStatusInfo for more information. • NULL Pointer for invalid data.
pWCDMASrvStatusInfo	• See nas_GSMSrvStatusInfo for more information. • NULL Pointer for invalid data.
pLTESrvStatusInfo	• See nas_GSMSrvStatusInfo for more information. • NULL Pointer for invalid data.
pCDMASysInfo	• See nas_CDMASysInfo for more information. • NULL Pointer for invalid data.
pHDRSysInfo	• See nas_HDRSysInfo for more information. • NULL Pointer for invalid data.
pGSMSysInfo	• See nas_GSMSysInfo for more information. • NULL Pointer for invalid data.

pWCDMASysInfo	• See nas_WCDMASysInfo for more information. • NULL Pointer for invalid data.
pLTESysInfo	• See nas_LTESysInfo for more information. • NULL Pointer for invalid data.
pAddCDMASysInfo	• See nas_AddCDMASysInfo for more information. • NULL Pointer for invalid data.
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. • NULL Pointer for invalid data.
pAddGSMSysInfo	• See nas_AddSysInfo for more information. • NULL Pointer for invalid data.
pAddWCDMASysInfo	• See nas_AddSysInfo for more information. • NULL Pointer for invalid data.
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. • NULL Pointer for invalid data.
pGSMCallBarringSysInfo	• See nas_CallBarringSysInfo for more information. • NULL Pointer for invalid data.
pWCDMACallBarringSysInfo	• See nas_CallBarringSysInfo for more information. • NULL Pointer for invalid data.
pLTEVoiceSupportSysInfo	• Indicates voice support status on LTE. – 0x00 - Voice is not supported – 0x01 - Voice is supported • NULL Pointer for invalid data.

pGSMCipher-DomainSysInfo	• Ciphering on the service domain. – 0x00 - No service – 0x01 - Circuit-switched only – 0x02 - Packet-switched only – 0x03 - Circuit-switched and packet-switched • NULL Pointer for invalid data.
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 • NULL Pointer for invalid data.

8.1136.2 Field Documentation

8.1136.2.1 nas_AddCDMASysInfo∗ unpack_nas_SLQSGetSysInfo_t::pAddCDMASysInfo

8.1136.2.2 nas_AddSysInfo∗ unpack_nas_SLQSGetSysInfo_t::pAddGSMSysInfo

8.1136.2.3 uint16_t∗ unpack_nas_SLQSGetSysInfo_t::pAddHDRSysInfo

8.1136.2.4 uint16_t∗ unpack_nas_SLQSGetSysInfo_t::pAddLTESysInfo

8.1136.2.5 nas_AddSysInfo∗ unpack_nas_SLQSGetSysInfo_t::pAddWCDMASysInfo

8.1136.2.6 nas_SrvStatusInfo∗ unpack_nas_SLQSGetSysInfo_t::pCDMASrvStatusInfo

8.1136.2.7 nas_CDMASysInfo∗ unpack_nas_SLQSGetSysInfo_t::pCDMASysInfo

8.1136.2.8 nas_CallBarringSysInfo∗ unpack_nas_SLQSGetSysInfo_t::pGSMCallBarringSysInfo

8.1136.2.9 uint8_t∗ unpack_nas_SLQSGetSysInfo_t::pGSMCipherDomainSysInfo

8.1136.2.10 nas_GSMSrvStatusInfo∗ unpack_nas_SLQSGetSysInfo_t::pGSMSrvStatusInfo

8.1136.2.11 nas_GSMSysInfo∗ unpack_nas_SLQSGetSysInfo_t::pGSMSysInfo

8.1136.2.12 nas_SrvStatusInfo∗ unpack_nas_SLQSGetSysInfo_t::pHDRSrvStatusInfo

8.1136.2.13 nas_HDRSysInfo∗ unpack_nas_SLQSGetSysInfo_t::pHDRSysInfo

8.1136.2.14 nas_GSMSrvStatusInfo∗ unpack_nas_SLQSGetSysInfo_t::pLTESrvStatusInfo

8.1136.2.15 nas_LTESysInfo∗ unpack_nas_SLQSGetSysInfo_t::pLTESysInfo

8.1136.2.16 uint8_t∗ unpack_nas_SLQSGetSysInfo_t::pLTEVoiceSupportSysInfo

8.1136.2.17 nas_CallBarringSysInfo∗ unpack_nas_SLQSGetSysInfo_t::pWCDMACallBarringSysInfo

8.1136.2.18 uint8_t* unpack_nas_SLQSGetSysInfo_t::pWCDMACipherDomainSysInfo

8.1136.2.19 nas_GSMSrvStatusInfo* unpack_nas_SLQSGetSysInfo_t::pWCDMASrvStatusInfo

8.1136.2.20 nas_WCDMASysInfo* unpack_nas_SLQSGetSysInfo_t::pWCDMASysInfo

8.1137 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

8.1137.1 Detailed Description

This structure contains unpack get get system selection preferences parameters.

Parameters

pEmerMode	• Optional parameter specifying the emergency Mode • Values: – 0 - OFF (normal) – 1 - ON (Emergency) • NULL Pointer for invalid data.
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 • NULL Pointer for invalid data.

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 • NULL Pointer for invalid data.

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 • NULL Pointer for invalid data.
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 • NULL Pointer for invalid data.

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 • NULL Pointer for invalid data.
pNetSelPref	• Optional parameter indicating network selection preference • Values: – 0x00 - Automatic network selection – 0x01 - Manual network selection. • NULL Pointer for invalid data.

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 • NULL Pointer for invalid data.
pGWAcqOrder-Pref	• Optional parameter indicating GSM/WCDMA Acquisition order Preference • Values: – 0x00 - Automatic – 0x01 - GSM then WCDMA – 0x02 - WCDMA then GSM • NULL Pointer for invalid data.

8.1137.2 Field Documentation

8.1137.2.1 uint64_t* unpack_nas_SLQSGetSysSelectionPref_t::pBandPref

8.1137.2.2 uint8_t* unpack_nas_SLQSGetSysSelectionPref_t::pEmerMode

8.1137.2.3 uint32_t* unpack_nas_SLQSGetSysSelectionPref_t::pGWAcqOrderPref

8.1137.2.4 uint64_t* unpack_nas_SLQSGetSysSelectionPref_t::pLTEBandPref

8.1137.2.5 uint16_t* unpack_nas_SLQSGetSysSelectionPref_t::pModePref

8.1137.2.6 uint8_t* unpack_nas_SLQSGetSysSelectionPref_t::pNetSelPref

8.1137.2.7 uint16_t* unpack_nas_SLQSGetSysSelectionPref_t::pPRLPref

8.1137.2.8 uint16_t* unpack_nas_SLQSGetSysSelectionPref_t::pRoamPref

8.1137.2.9 uint32_t* unpack_nas_SLQSGetSysSelectionPref_t::pSrvDomainPref

8.1138 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

8.1138.1 Detailed Description

Structure for 3GPP2Subscription unpack.

Parameters

pNAMNameInfo	[Optional] • See nas_namName for more information • NULL for invalid data
pDirNum	[Optional] • See nas_dirNum for more information • NULL for invalid data
pHomeSIDNID	[Optional] • See nas_homeSIDNID for more information • NULL for invalid data
pMinBasedIMSI	[Optional] • See nas_minBasedIMSI for more information • NULL for invalid data
pTrueIMSI	[Optional] • See nas_trueIMSI for more information • NULL for invalid data
pCDMAChannel	[Optional] • See nas_CDMAChannel for more information • NULL for invalid data

8.1138.2 Field Documentation

8.1138.2.1 nas_CDMAChannel∗ unpack_nas_SLQSNasGet3GPP2Subscription_t::pCDMAChannel

8.1138.2.2 nas_dirNum∗ unpack_nas_SLQSNasGet3GPP2Subscription_t::pDirNum

8.1138.2.3 nas_homeSIDNID∗ unpack_nas_SLQSNasGet3GPP2Subscription_t::pHomeSIDNID

8.1138.2.4 nas_minBasedIMSI∗ unpack_nas_SLQSNasGet3GPP2Subscription_t::pMinBasedIMSI

8.1138.2.5 nas_namName∗ unpack_nas_SLQSNasGet3GPP2Subscription_t::pNAMNameInfo

8.1138.2.6 nas_trueIMSI∗ unpack_nas_SLQSNasGet3GPP2Subscription_t::pTrueIMSI

8.1139 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

8.1139.1 Detailed Description

This structure contains information about the Get Cell Location response parameters.

Parameters

pGERANInfo	• See nas_GERANInfo for more information. • NULL for invalid data
pUMTSInfo	• See nas_UMTSInfo for more information. • NULL for invalid data
pCDMAInfo	• See nas_CDMAInfo for more information. • NULL for invalid data
pLTEInfo-Intrafreq	• See nas_LTEInfoIntrafreq for more information. • NULL for invalid data
pLTEInfo-Interfreq	• See nas_LTEInfoInterfreq for more information. • NULL for invalid data
pLTEInfo-NeighboringGS-M	• See nas_LTEInfoNeighboringGSM for more information. • NULL for invalid data
pLTEInfo-NeighboringWC-DMA	• See nas_LTEInfoNeighboringWCDMA for more information. • NULL for invalid data
pUMTSCellID	• Cell ID. • 0xFFFFFFFF indicates cell ID information is not present. • NULL for invalid data
pWCDMAInfoLT-ENeighborCell	• See nas_WCDMAInfoLTENeighborCell for more information. • NULL for invalid data

8.1139.2 Field Documentation

8.1139.2.1 nas_CDMAInfo* unpack_nas_SLQSNasGetCellLocationInfo_t::pCDMAInfo

8.1139.2.2 **nas_GERANInfo**∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pGERANInfo

8.1139.2.3 **nas_LTEInfoInterfreq**∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoInterfreq

8.1139.2.4 **nas_LTEInfoIntrafreq**∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoIntrafreq

8.1139.2.5 **nas_LTEInfoNeighboringGSM**∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoNeighboringGSM

8.1139.2.6 **nas_LTEInfoNeighboringWCDMA**∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoNeighboringWCDMA

8.1139.2.7 uint32_t∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pUMTSCellID

8.1139.2.8 **nas_UMTSInfo**∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pUMTSInfo

8.1139.2.9 **nas_WCDMAInfoLTENeighborCell**∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pWCDMAInfoLTENeighborCell

8.1140 unpack_nas_SLQSNasGetHDRColorCode_t Struct Reference

Data Fields

- uint8_t ∗ pColorCode

8.1140.1 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 • NULL for invalid data

8.1140.2 Field Documentation

8.1140.2.1 uint8_t∗ unpack_nas_SLQSNasGetHDRColorCode_t::pColorCode

8.1141 unpack_nas_SLQSNasGetSigInfo_t Struct Reference

Data Fields

- cdmaSSInfo CDMASSInfo
- hdrSSInfo HDRSSInfo
- int8_t GSMSSInfo
- cdmaSSInfo WCDMASSInfo
- lteSSInfo LTESSInfo

8.1141.1 Detailed Description

This structure contains the SLQSNasGetSigInfo response parameters.

Parameters

CDMASSInfo	• See cdmaSSInfo for more information
HDRSSInfo	• See hdrSSInfo for more information
GSMSSInfo	• GSM signal strength is the RSSI in dBm (signed value). • A value of -125 dBm or lower is used to indicate No Signal.
WCDMASSInfo	• See cdmaSSInfo for more information
LTESSInfo	• See lteSSInfo for more information

8.1141.2 Field Documentation

8.1141.2.1 cdmaSSInfo unpack_nas_SLQSNasGetSigInfo_t::CDMASSInfo

8.1141.2.2 int8_t unpack_nas_SLQSNasGetSigInfo_t::GSMSSInfo

8.1141.2.3 hdrSSInfo unpack_nas_SLQSNasGetSigInfo_t::HDRSSInfo

8.1141.2.4 lteSSInfo unpack_nas_SLQSNasGetSigInfo_t::LTESSInfo

8.1141.2.5 cdmaSSInfo unpack_nas_SLQSNasGetSigInfo_t::WCDMASSInfo

8.1142 unpack_nas_SLQSNasGetTxRxInfo_t Struct Reference

Data Fields

- nas_rxInfo * pRXChain0Info
- nas_rxInfo * pRXChain1Info
- nas_txInfo * pTXInfo

8.1142.1 Detailed Description

Structure for TxRx Info unpack

Parameters

pRXChain0Info	[Optional] • See nas_rxInfo for more information. • NULL for invalid data.

pRXChain1Info	[Optional] • See nas_rxInfo for more information. • NULL for invalid data.
pTXInfo	[Optional] • See nas_txInfo for more information. • NULL for invalid data.

8.1142.2 Field Documentation

8.1142.2.1 nas_rxInfo* unpack_nas_SLQSNasGetTxRxInfo_t::pRXChain0Info

8.1142.2.2 nas_rxInfo* unpack_nas_SLQSNasGetTxRxInfo_t::pRXChain1Info

8.1142.2.3 nas_txInfo* unpack_nas_SLQSNasGetTxRxInfo_t::pTXInfo

8.1143 unpack_nas_SLQSNasNetworkTimeCallBack_ind_t Struct Reference

Data Fields

- nas_UniversalTime universalTime
- uint8_t * pTimeZone
- uint8_t * pDayltSavAdj
- uint8_t * pRadioInterface

8.1143.1 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). • NULL for invalid data.
pDayltSavAdj	• Daylight Saving Adjustment. • Daylight saving adjustment in hr. – Possible values: 0, 1, and 2. • NULL for invalid data.

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 • NULL for invalid data.

8.1143.2 Field Documentation

8.1143.2.1 uint8_t* unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::pDayltSavAdj

8.1143.2.2 uint8_t* unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::pRadioInterface

8.1143.2.3 uint8_t* unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::pTimeZone

8.1143.2.4 nas_UniversalTime unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::universalTime

8.1144 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

8.1144.1 Detailed Description

Structure for storing the nasSigInfo indication parameters.

Parameters

pCDMASigInfo	• See cdmaSSInfo for more information. • NULL Pointer for invalid data.
pHDRSigInfo	• See hdrSSInfo for more information. • NULL Pointer for invalid data.
pGSMSigInfo	• one byte value, GSM signal strength is the RSSI in dBm (signed value). A value of -125 dBm or lower is used to indicate No Signal • NULL Pointer for invalid data.

pWCDMASigInfo	• See cdmaSSInfo for more information. • NULL Pointer for invalid data.
pLTESigInfo	• See lteSSInfo for more information. • NULL Pointer for invalid data.
pRscp	• RSCP of the Primary Common Control Physical Channel (PCCPCH) in dBm. Measurement range: -120 dBm to -25 dBm. • NULL Pointer for invalid data.
pTDSCDMASig-InfoExt	• See tdscdmaSigInfoExt for more information. • NULL Pointer for invalid data.

8.1144.2 Field Documentation

8.1144.2.1 cdmaSSInfo∗ unpack_nas_SLQSNasSigInfoCallback_ind_t::pCDMASigInfo

8.1144.2.2 int8_t∗ unpack_nas_SLQSNasSigInfoCallback_ind_t::pGSMSigInfo

8.1144.2.3 hdrSSInfo∗ unpack_nas_SLQSNasSigInfoCallback_ind_t::pHDRSigInfo

8.1144.2.4 lteSSInfo∗ unpack_nas_SLQSNasSigInfoCallback_ind_t::pLTESigInfo

8.1144.2.5 int8_t∗ unpack_nas_SLQSNasSigInfoCallback_ind_t::pRscp

8.1144.2.6 tdscdmaSigInfoExt∗ unpack_nas_SLQSNasSigInfoCallback_ind_t::pTDSCDMASigInfoExt

8.1144.2.7 cdmaSSInfo∗ unpack_nas_SLQSNasSigInfoCallback_ind_t::pWCDMASigInfo

8.1145 unpack_nas_SLQSNASSwiGetChannelLock_t Struct Reference

Data Fields

- nas_wcdmaUARFCN ∗ pWcdmaUARFCN
- nas_lteEARFCN ∗ pLteEARFCN
- nas_ltePCI ∗ pLtePCI

8.1145.1 Detailed Description

Structure for Get Channel Lock unpack.

Parameters

pWcdmaUARF-CN	[Optional] • See nas_wcdmaUARFCN for more information • NULL for invalid data

pLteEARFCN	[Optional] • See nas_lteEARFCN for more information • NULL for invalid data
pLtePCI	[Optional] • See nas_ltePCI for more information • NULL for invalid data

8.1145.2 Field Documentation

8.1145.2.1 nas_lteEARFCN∗ unpack_nas_SLQSNASSwiGetChannelLock_t::pLteEARFCN

8.1145.2.2 nas_ltePCI∗ unpack_nas_SLQSNASSwiGetChannelLock_t::pLtePCI

8.1145.2.3 nas_wcdmaUARFCN∗ unpack_nas_SLQSNASSwiGetChannelLock_t::pWcdmaUARFCN

8.1146 unpack_nas_SLQSNasSwiModemStatus_t Struct Reference

Data Fields

- nas_CommInfo commonInfo
- nas_LTEInfo ∗ pLTEInfo

8.1146.1 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 • NULL for invalid data.

8.1146.2 Field Documentation

8.1146.2.1 nas_CommInfo unpack_nas_SLQSNasSwiModemStatus_t::commonInfo

8.1146.2.2 nas_LTEInfo∗ unpack_nas_SLQSNasSwiModemStatus_t::pLTEInfo

8.1147 unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t Struct Reference

Data Fields

- NASQmiCbkNasSwiOTAMessageInd Info
- uint16_t Tlvresult

8.1147.1 Detailed Description

This structure contains unpack OTA message indication parameters.

Parameters

Info	• Structure used to store all QMI Notification Info. – See NASQmiCbkNasSwiOTAMessageInd for more details
Tlvresult	• unpack result

8.1147.2 Field Documentation

8.1147.2.1 NASQmiCbkNasSwiOTAMessageInd unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t::Info

8.1147.2.2 uint16_t unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t::Tlvresult

8.1148 unpack_nas_SLQSNasTimerCallback_ind_t Struct Reference

Data Fields

- char t3396_apn [101]
- uint8_t t3396_plmn_id [3]
- uint32_t t3396_val

8.1148.1 Detailed Description

Structure for Network Timer indication parameters.

Parameters

t3396_apn	• apn
t3396_plmn_id	• plmn id
t3396_val	• timer value

8.1148.2 Field Documentation

8.1148.2.1 char unpack_nas_SLQSNasTimerCallback_ind_t::t3396_apn[101]

8.1148.2.2 uint8_t unpack_nas_SLQSNasTimerCallback_ind_t::t3396_plmn_id[3]

8.1148.2.3 uint32_t unpack_nas_SLQSNasTimerCallback_ind_t::t3396_val

8.1149 unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t Struct Reference

Data Fields

- NASQmiCbkNasSystemSelPrefInd Info
- uint16_t TlvResult

8.1149.1 Detailed Description

This structure contains unpack set system selection preference callback indication parameters.

Parameters

Info	• Structure used to store all QMI Notification Info. – See NASQmiCbkNasSystemSelPrefInd for more details
TlvResult	• unpack result

8.1149.2 Field Documentation

8.1149.2.1 NASQmiCbkNasSystemSelPrefInd unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t::Info

8.1149.2.2 uint16_t unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t::TlvResult

8.1150 unpack_nas_SLQSSwiGetHDRPersonality_t Struct Reference

Data Fields

- uint16_t * pCurrentPersonality
- uint8_t * pPersonalityListLength
- nas_protocolSubtypeElement * pProtocolSubtypeElement

8.1150.1 Detailed Description

Structure for HDRPersonality unpack.

Parameters

pCurrentPersonality[Out]	• Current active personality index. • NULL for invalid data
pPersonalityListLength[In/Out]	• Number of Personality Protocol Subtype contains in this response. • maximum input value is 3 • NULL for invalid data
pProtocolSubtypeElement[Out]	• See nas_protocolSubtypeElement for more information. • NULL for invalid data

8.1150.2 Field Documentation

8.1150.2.1 uint16_t* unpack_nas_SLQSSwiGetHDRPersonality_t::pCurrentPersonality

8.1150.2.2 uint8_t* unpack_nas_SLQSSwiGetHDRPersonality_t::pPersonalityListLength

8.1150.2.3 nas_protocolSubtypeElement* unpack_nas_SLQSSwiGetHDRPersonality_t::pProtocolSubtypeElement

8.1151 unpack_nas_SLQSSwiGetHDRProtSubtype_t Struct Reference

Data Fields

- uint16_t * pCurrentPrsnlty
- uint8_t * pPersonalityListLength
- nas_protocolSubtypeElement * pProtoSubTypElmnt
- uint64_t * pAppSubType

8.1151.1 Detailed Description

Structure for HDRProtSubtype unpack.

Parameters

pCurrent-Personality	• Current active personality index. • NULL for invalid data
pPersonalityList-Length	• Number of Personality Protocol Subtype contains in this response. • maximum input value is 4 • NULL for invalid data
pProtocol-SubtypeElement	• See nas_protocolSubtypeElement for more information. • NULL for invalid data
pAppSubType	• Stream application subtype • Application subtype for each stream • NULL for invalid data

8.1151.2 Field Documentation

8.1151.2.1 uint64_t* unpack_nas_SLQSSwiGetHDRProtSubtype_t::pAppSubType

8.1151.2.2 uint16_t* unpack_nas_SLQSSwiGetHDRProtSubtype_t::pCurrentPrsnlty

8.1151.2.3 uint8_t* unpack_nas_SLQSSwiGetHDRProtSubtype_t::pPersonalityListLength

8.1151.2.4 nas_protocolSubtypeElement* unpack_nas_SLQSSwiGetHDRProtSubtype_t::pProtoSubTypElmnt

8.1152 unpack_nas_SLQSSwiGetHRPDStats_t Struct Reference

Data Fields

- nas_DRCParams * pDRCParams
- uint8_t * pUATI
- nas_PilotSetData * pPilotSetData

8.1152.1 Detailed Description

Structure for HRPD Stats unpack.

Parameters

pDRCParams	• See nas_DRCParams for more information. • NULL for invalid data
pUATI	• A 128-bit address that includes the access terminal identifier and subnet ID • Size must be 16 bytes • NULL for invalid data
pPilotSetData	• See nas_PilotSetData for more information. • NULL for invalid data

8.1152.2 Field Documentation

8.1152.2.1 nas_DRCParams* unpack_nas_SLQSSwiGetHRPDStats_t::pDRCParams

8.1152.2.2 nas_PilotSetData* unpack_nas_SLQSSwiGetHRPDStats_t::pPilotSetData

8.1152.2.3 uint8_t* unpack_nas_SLQSSwiGetHRPDStats_t::pUATI

8.1153 unpack_nas_SLQSSwiGetLteCQI_t Struct Reference

Data Fields

- uint8_t ValidityCW0
- uint8_t CQIValueCW0
- uint8_t ValidityCW1
- uint8_t CQIValueCW1

8.1153.1 Detailed Description

This structure contains paramaters unpack fetch CQI parameters for LTE data session.

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

8.1153.2 Field Documentation

8.1153.2.1 uint8_t unpack_nas_SLQSSwiGetLteCQI_t::CQIValueCW0

8.1153.2.2 uint8_t unpack_nas_SLQSSwiGetLteCQI_t::CQIValueCW1

8.1153.2.3 uint8_t unpack_nas_SLQSSwiGetLteCQI_t::ValidityCW0

8.1153.2.4 uint8_t unpack_nas_SLQSSwiGetLteCQI_t::ValidityCW1

8.1154 unpack_nas_SLQSSwiGetLteSccRxInfo_t Struct Reference

Data Fields

- nas_SccRxInfo ∗ pSccRxInfo

8.1154.1 Detailed Description

This structure contains unpack get LTE Secondary carrier Rx signal level information parameters.

Parameters

pSccRxInfo	Secondary carrier Rx signal level info • See nas_SccRxInfo • NULL for invalid data

8.1154.2 Field Documentation

8.1154.2.1 nas_SccRxInfo∗ unpack_nas_SLQSSwiGetLteSccRxInfo_t::pSccRxInfo

8.1155 unpack_nas_SLQSSwiNetworkDebug_t Struct Reference

Data Fields

- uint8_t * pObjectVer
- nas_NetworkStat1x * pNetworkStat1x
- nas_NetworkStatEVDO * pNetworkStatEVDO
- nas_DeviceConfigDetail * pDeviceConfigDetail
- nas_DataStatusDetail * pDataStatusDetail

8.1155.1 Detailed Description

Structure for SwiNetworkDebug unpack.

Parameters

pObjectVer	• Object's version number for the host to handle – 0xFF - NA – Others - shows in decimal • NULL for invalid data
pNetworkStat1x	• See nas_NetworkStat1x for more information • NULL for invalid data
pNetworkStatEVDO	• See nas_NetworkStatEVDO for more information. • NULL for invalid data
pDeviceConfigDetail	• See nas_DeviceConfigDetail for more information. • NULL for invalid data
pDataStatusDetail	• See nas_DataStatusDetail for more information. • NULL for invalid data

8.1155.2 Field Documentation

8.1155.2.1 nas_DataStatusDetail* unpack_nas_SLQSSwiNetworkDebug_t::pDataStatusDetail

8.1155.2.2 nas_DeviceConfigDetail* unpack_nas_SLQSSwiNetworkDebug_t::pDeviceConfigDetail

8.1155.2.3 nas_NetworkStat1x* unpack_nas_SLQSSwiNetworkDebug_t::pNetworkStat1x

8.1155.2.4 nas_NetworkStatEVDO* unpack_nas_SLQSSwiNetworkDebug_t::pNetworkStatEVDO

8.1155.2.5 uint8_t* unpack_nas_SLQSSwiNetworkDebug_t::pObjectVer

8.1156 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

8.1156.1 Detailed Description

This structure contains unpack system information callback indication parameters.

Parameters

pCDMASrvStatusInfo	• See nas_SrvStatusInfo for more information.
pHDRSrvStatusInfo	• See nas_SrvStatusInfo for more information.
pGSMSrvStatusInfo	• See nas_GSMSrvStatusInfo for more information.
pWCDMASrvStatusInfo	• See nas_GSMSrvStatusInfo for more information.
pLTESrvStatusInfo	• See nas_GSMSrvStatusInfo for more information.
pCDMASysInfo	• See nas_CDMASysInfo for more information.
pHDRSysInfo	• See nas_HDRSysInfo for more information.
pGSMSysInfo	• See nas_GSMSysInfo for more information.

pWCDMASysInfo	• See nas_WCDMASysInfo for more information.
pLTESysInfo	• See nas_LTESysInfo for more information.
pAddCDMASysInfo	• See nas_AddCDMASysInfo for more information.
pAddHDRSysInfo	• System table index referencing the beginning of the geo in which the current serving system is present. • 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

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

8.1156.2 Field Documentation

8.1156.2.1 nas_AddCDMASysInfo∗ unpack_nas_SLQSSysInfoCallback_ind_t::pAddCDMASysInfo

8.1156.2.2 nas_AddSysInfo∗ unpack_nas_SLQSSysInfoCallback_ind_t::pAddGSMSysInfo

8.1156.2.3 uint16_t∗ unpack_nas_SLQSSysInfoCallback_ind_t::pAddHDRSysInfo

8.1156.2.4 uint16_t∗ unpack_nas_SLQSSysInfoCallback_ind_t::pAddLTESysInfo

8.1156.2.5 nas_AddSysInfo∗ unpack_nas_SLQSSysInfoCallback_ind_t::pAddWCDMASysInfo

8.1156.2.6 nas_SrvStatusInfo∗ unpack_nas_SLQSSysInfoCallback_ind_t::pCDMASrvStatusInfo

8.1156.2.7 nas_CDMASysInfo∗ unpack_nas_SLQSSysInfoCallback_ind_t::pCDMASysInfo

8.1156.2.8 nas_CallBarringSysInfo∗ unpack_nas_SLQSSysInfoCallback_ind_t::pGSMCallBarringSysInfo

8.1156.2.9 uint8_t∗ unpack_nas_SLQSSysInfoCallback_ind_t::pGSMCipherDomainSysInfo

8.1156.2.10 nas_GSMSrvStatusInfo∗ unpack_nas_SLQSSysInfoCallback_ind_t::pGSMSrvStatusInfo

8.1156.2.11 nas_GSMSysInfo∗ unpack_nas_SLQSSysInfoCallback_ind_t::pGSMSysInfo

8.1156.2.12 nas_SrvStatusInfo∗ unpack_nas_SLQSSysInfoCallback_ind_t::pHDRSrvStatusInfo

8.1156.2.13 nas_HDRSysInfo∗ unpack_nas_SLQSSysInfoCallback_ind_t::pHDRSysInfo

8.1156.2.14 nas_GSMSrvStatusInfo∗ unpack_nas_SLQSSysInfoCallback_ind_t::pLTESrvStatusInfo

8.1156.2.15 nas_LTESysInfo∗ unpack_nas_SLQSSysInfoCallback_ind_t::pLTESysInfo

8.1156.2.16 uint8_t∗ unpack_nas_SLQSSysInfoCallback_ind_t::pLTEVoiceSupportSysInfo

8.1156.2.17 uint8_t∗ unpack_nas_SLQSSysInfoCallback_ind_t::pSysInfoNoChange

8.1156.2.18 nas_CallBarringSysInfo∗ unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMACallBarringSysInfo

8.1156.2.19 uint8_t∗ unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMACipherDomainSysInfo

8.1156.2.20 nas_GSMSrvStatusInfo∗ unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMASrvStatusInfo

8.1156.2.21 nas_WCDMASysInfo∗ unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMASysInfo

8.1157 unpack_omaDmConfigTlv_t Struct Reference

Data Fields

- uint8_t state
- uint8_t userInputReq
- uint16_t userInputTimeout
- uint16_t alertmsglength
- uint8_t alertmsg [256]

8.1157.1 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)

8.1157.2 Field Documentation

8.1157.2.1 uint8_t unpack_omaDmConfigTlv_t::alertmsg[256]

8.1157.2.2 uint16_t unpack_omaDmConfigTlv_t::alertmsglength

8.1157.2.3 uint8_t unpack_omaDmConfigTlv_t::state

8.1157.2.4 uint8_t unpack_omaDmConfigTlv_t::userInputReq

8.1157.2.5 uint16_t unpack_omaDmConfigTlv_t::userInputTimeout

8.1158 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

8.1158.1 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

8.1158.2 Field Documentation

8.1158.2.1 uint8_t unpack_omaDmFotaTlv_t::description[256]

8.1158.2.2 uint16_t unpack_omaDmFotaTlv_t::descriptionlength

8.1158.2.3 uint32_t unpack_omaDmFotaTlv_t::fwdloadsize

8.1158.2.4 uint32_t unpack_omaDmFotaTlv_t::fwloadComplete

8.1158.2.5 uint16_t unpack_omaDmFotaTlv_t::namelength

8.1158.2.6 uint8_t unpack_omaDmFotaTlv_t::package_name[256]

8.1158.2.7 uint8_t unpack_omaDmFotaTlv_t::sessionType

8.1158.2.8 uint8_t unpack_omaDmFotaTlv_t::severity

8.1158.2.9 uint8_t unpack_omaDmFotaTlv_t::state

8.1158.2.10 uint16_t unpack_omaDmFotaTlv_t::updateCompleteStatus

8.1158.2.11 uint8_t unpack_omaDmFotaTlv_t::userInputReq

8.1158.2.12 uint16_t unpack_omaDmFotaTlv_t::userInputTimeout

8.1158.2.13 uint8_t unpack_omaDmFotaTlv_t::version[256]

8.1158.2.14 uint16_t unpack_omaDmFotaTlv_t::versionlength

8.1159 unpack_omaDmNotificationsTlv_t Struct Reference

Data Fields

- uint8_t notification
- uint16_t sessionStatus

8.1159.1 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.

8.1159.2 Field Documentation

8.1159.2.1 uint8_t unpack_omaDmNotificationsTlv_t::notification

8.1159.2.2 uint16_t unpack_omaDmNotificationsTlv_t::sessionStatus

8.1160 unpack_pds_GetPDSDefaults_t Struct Reference

Data Fields

- uint32_t * pOperation
- uint8_t * pTimeout
- uint32_t * pInterval
- uint32_t * pAccuracy

8.1160.1 Detailed Description

Structure contain session configuration parameters.

Parameters

pOperation	• Current session operating mode – 0 - Standalone – 1 - MS based – 2 - MS assisted • NULL pointer - Invalid data.
pTimeout	• Maximum amount of time (seconds) to work on each fix, maximum is 255 • NULL pointer - Invalid data.
pInterval	• Interval (seconds) between fix requests • NULL pointer - Invalid data.
pAccuracy	• Preferred accuracy threshold (meters) • NULL pointer - Invalid data.

8.1160.2 Field Documentation

8.1160.2.1 uint32_t* unpack_pds_GetPDSDefaults_t::pAccuracy

8.1160.2.2 uint32_t* unpack_pds_GetPDSDefaults_t::pInterval

8.1160.2.3 uint32_t* unpack_pds_GetPDSDefaults_t::pOperation

8.1160.2.4 uint8_t* unpack_pds_GetPDSDefaults_t::pTimeout

8.1161 unpack_pds_GetPDSState_t Struct Reference

Data Fields

- uint32_t * pEnabledStatus
- uint32_t * pTrackingStatus

8.1161.1 Detailed Description

Structure contain parameters of current PDS state.

Parameters

pEnabledStatus	• Current PDS state – 0 - disable – 1 - enable • NULL pointer - Invalid data.

pTrackingStatus	• Current PDS tracking session state • Values: – 0x00 - Unknown – 0x01 - Inactive – 0x02 - Active • NULL pointer - Invalid data.

8.1161.2 Field Documentation

8.1161.2.1 uint32_t* unpack_pds_GetPDSState_t::pEnabledStatus

8.1161.2.2 uint32_t* unpack_pds_GetPDSState_t::pTrackingStatus

8.1162 unpack_pds_GetPortAutomaticTracking_t Struct Reference

Data Fields

- uint32_t * pbAuto

8.1162.1 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

- NULL pointer - Invalid data.

8.1162.2 Field Documentation

8.1162.2.1 uint32_t* unpack_pds_GetPortAutomaticTracking_t::pbAuto

8.1163 unpack_pds_GetServiceAutomaticTracking_t Struct Reference

Data Fields

- uint32_t * pbAuto

8.1163.1 Detailed Description

Structure contain the parameter for automatic tracking state.

Parameters

pbAuto	• Automatic tracking session started for service – 0x00 - Disabled – 0x01 - Enabled • NULL pointer - Invalid data.

8.1163.2 Field Documentation

8.1163.2.1 uint32_t* unpack_pds_GetServiceAutomaticTracking_t::pbAuto

8.1164 unpack_pds_GetXTRAAutomaticDownload_t Struct Reference

Data Fields

- uint32_t * pbEnabled
- uint16_t * pInterval

8.1164.1 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 • NULL pointer - Invalid data.
pInterval	• Interval (hours) between XTRA downloads • NULL pointer - Invalid data.

8.1164.2 Field Documentation

8.1164.2.1 uint32_t* unpack_pds_GetXTRAAutomaticDownload_t::pbEnabled

8.1164.2.2 uint16_t* unpack_pds_GetXTRAAutomaticDownload_t::pInterval

8.1165 unpack_pds_GetXTRANetwork_t Struct Reference

Data Fields

- uint32_t * pPreference

8.1165.1 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 • NULL pointer - Invalid data.

8.1165.2 Field Documentation

8.1165.2.1 uint32_t* unpack_pds_GetXTRANetwork_t::pPreference

8.1166 unpack_pds_GetXTRAValidity_t Struct Reference

Data Fields

- uint16_t * pGPSWeek
- uint16_t * pGPSWeekOffset
- uint16_t * pDuration

8.1166.1 Detailed Description

Structure contain the parameter for XTRA database validity period

Parameters

pGPSWeek	• Starting GPS week of validity period • NULL pointer - Invalid data.
pGPSWeek-Offset	• Starting GPS week offset (minutes) of validity period • NULL pointer - Invalid data.
pDuration	• Length of validity period (hours) • NULL pointer - Invalid data.

8.1166.2 Field Documentation

8.1166.2.1 uint16_t* unpack_pds_GetXTRAValidity_t::pDuration

8.1166.2.2 uint16_t* unpack_pds_GetXTRAValidity_t::pGPSWeek

8.1166.2.3 uint16_t* unpack_pds_GetXTRAValidity_t::pGPSWeekOffset

8.1167 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

8.1167.1 Detailed Description

Structure contain the parameter for the Set event report Indication. Please check has_<Param_Name> field for presence of optional parameters

Parameters

PositionDataNMEA	• Position Data NMEA. • 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

8.1167.2 Field Documentation

8.1167.2.1 double unpack_pds_SetEventReport_Ind_t::dLatitude

8.1167.2.2 double unpack_pds_SetEventReport_Ind_t::dLongitude

8.1167.2.3 uint8_t unpack_pds_SetEventReport_Ind_t::has_dLatitude

8.1167.2.4 uint8_t unpack_pds_SetEventReport_Ind_t::has_dLongitude

8.1167.2.5 uint8_t unpack_pds_SetEventReport_Ind_t::has_PositionDataNMEA

8.1167.2.6 uint8_t unpack_pds_SetEventReport_Ind_t::has_posSrc

8.1167.2.7 uint8_t unpack_pds_SetEventReport_Ind_t::has_SessionStatus

8.1167.2.8 char unpack_pds_SetEventReport_Ind_t::PositionDataNMEA[200]

8.1167.2.9 uint32_t unpack_pds_SetEventReport_Ind_t::posSrc

8.1167.2.10 uint8_t unpack_pds_SetEventReport_Ind_t::SessionStatus

8.1168 unpack_pds_SetPdsState_Ind_t Struct Reference

Data Fields

- uint16_t TlvPresent
- uint32_t EnabledStatus
- uint32_t TrackingStatus

8.1168.1 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

8.1168.2 Field Documentation

8.1168.2.1 uint32_t unpack_pds_SetPdsState_Ind_t::EnabledStatus

8.1168.2.2 uint16_t unpack_pds_SetPdsState_Ind_t::TlvPresent

8.1168.2.3 uint32_t unpack_pds_SetPdsState_Ind_t::TrackingStatus

8.1169 unpack_pds_SLQSGetAGPSConfig_t Struct Reference

Data Fields

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

8.1169.1 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

8.1169.2 Field Documentation

8.1169.2.1 uint32_t* unpack_pds_SLQSGetAGPSConfig_t::pServerAddress

8.1169.2.2 uint32_t* unpack_pds_SLQSGetAGPSConfig_t::pServerPort

8.1169.2.3 uint8_t* unpack_pds_SLQSGetAGPSConfig_t::pServerURL

8.1169.2.4 uint8_t* unpack_pds_SLQSGetAGPSConfig_t::pServerURLLength

8.1170 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

8.1170.1 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

8.1170.2 Field Documentation

8.1170.2.1 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::Altitude

8.1170.2.2 uint8_t unpack_pds_SLQSGetGPSStateInfo_t::EngineState

8.1170.2.3 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::glo_almanac_sv_msk

8.1170.2.4 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::glo_ephemeris_sv_msk

8.1170.2.5 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::glo_health_sv_msk

8.1170.2.6 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::glo_visible_sv_msk

8.1170.2.7 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::gps_almanac_sv_msk

8.1170.2.8 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::gps_ephemeris_sv_msk

8.1170.2.9 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::gps_health_sv_msk

8.1170.2.10 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::gps_visible_sv_msk

8.1170.2.11 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::HorizontalUncertainty

8.1170.2.12 uint8_t unpack_pds_SLQSGetGPSStateInfo_t::Iono_valid

8.1170.2.13 uint64_t unpack_pds_SLQSGetGPSStateInfo_t::Latitude

8.1170.2.14 uint64_t unpack_pds_SLQSGetGPSStateInfo_t::Longitude

8.1170.2.15 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::sbas_almanac_sv_msk

8.1170.2.16 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::sbas_ephemeris_sv_msk

8.1170.2.17 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::sbas_health_sv_msk

8.1170.2.18 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::sbas_visible_sv_msk

8.1170.2.19 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::Time_uncert_ms

8.1170.2.20 uint16_t unpack_pds_SLQSGetGPSStateInfo_t::TimeStmp_gps_week

8.1170.2.21 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::TimeStmp_tow_ms

8.1170.2.22 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::ValidMask

8.1170.2.23 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::VerticalUncertainty

8.1170.2.24 uint16_t unpack_pds_SLQSGetGPSStateInfo_t::xtra_start_gps_minutes

8.1170.2.25 uint16_t unpack_pds_SLQSGetGPSStateInfo_t::xtra_start_gps_week

8.1170.2.26 uint16_t unpack_pds_SLQSGetGPSStateInfo_t::xtra_valid_duration_hours

8.1171 unpack_qmi_t Struct Reference

Data Fields

- enum msgtype type
- uint16_t msgid
- uint16_t xid

8.1171.1 Detailed Description

qmi response context

Parameters

out	*type*	message type
out	*msgid*	message id
out	*xid*	transaction id

8.1171.2 Field Documentation

8.1171.2.1 uint16_t unpack_qmi_t::msgid

8.1171.2.2 enum msgtype unpack_qmi_t::type

8.1171.2.3 uint16_t unpack_qmi_t::xid

8.1172 unpack_qos_dataRate_t Struct Reference

Data Fields

- uint32_t dataRateMax
- uint32_t guaranteedRate

8.1172.1 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)

8.1172.2 Field Documentation

8.1172.2.1 uint32_t unpack_qos_dataRate_t::dataRateMax

8.1172.2.2 uint32_t unpack_qos_dataRate_t::guaranteedRate

8.1173 unpack_qos_IPv4Addr_t Struct Reference

Data Fields

- uint32_t addr
- uint32_t subnetMask

8.1173.1 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

8.1173.2 Field Documentation

8.1173.2.1 uint32_t unpack_qos_IPv4Addr_t::addr

8.1173.2.2 uint32_t unpack_qos_IPv4Addr_t::subnetMask

8.1174 unpack_qos_IPv6Addr_t Struct Reference

Data Fields

- uint8_t addr [16]
- uint8_t prefixLen

8.1174.1 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.

8.1174.2 Field Documentation

8.1174.2.1 uint8_t unpack_qos_IPv6Addr_t::addr[16]

8.1174.2.2 uint8_t unpack_qos_IPv6Addr_t::prefixLen

8.1175 unpack_qos_IPv6TrafCls_t Struct Reference

Data Fields

- uint8_t val
- uint8_t mask

8.1175.1 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

8.1175.2 Field Documentation

8.1175.2.1 uint8_t unpack_qos_IPv6TrafCls_t::mask

8.1175.2.2 uint8_t unpack_qos_IPv6TrafCls_t::val

8.1176 unpack_qos_pktErrRate_t Struct Reference

Data Fields

- uint16_t multiplier
- uint16_t exponent

8.1176.1 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)

8.1176.2 Field Documentation

8.1176.2.1 uint16_t unpack_qos_pktErrRate_t::exponent

8.1176.2.2 uint16_t unpack_qos_pktErrRate_t::multiplier

8.1177 unpack_qos_Port_t Struct Reference

Data Fields

- uint16_t port
- uint16_t range

8.1177.1 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

8.1177.2 Field Documentation

8.1177.2.1 uint16_t unpack_qos_Port_t::port

8.1177.2.2 uint16_t unpack_qos_Port_t::range

8.1178 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

8.1178.1 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_TxQFlow-Granted_-Available	• TRUE if optional TxQFlowGranted is available

TxQFlow-Granted	• The Tx Qos flow granted, please check unpack_qos_swiQosFlow_t for more information
is_RxQFlow-Granted_-Available	• TRUE if optional RxQFlowGranted is available
RxQFlow-Granted	• 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.

8.1178.2 Field Documentation

8.1178.2.1 uint8_t unpack_qos_QosFlowInfo_t::BearerID

8.1178.2.2 uint8_t unpack_qos_QosFlowInfo_t::is_RxQFlowGranted_Available

8.1178.2.3 uint8_t unpack_qos_QosFlowInfo_t::is_TxQFlowGranted_Available

8.1178.2.4 uint8_t unpack_qos_QosFlowInfo_t::NumRxFilters

8.1178.2.5 uint8_t unpack_qos_QosFlowInfo_t::NumTxFilters

8.1178.2.6 unpack_qos_QosFlowInfoState_t unpack_qos_QosFlowInfo_t::QFlowState

8.1178.2.7 unpack_qos_swiQosFilter_t unpack_qos_QosFlowInfo_t::RxQFilter[25]

8.1178.2.8 unpack_qos_swiQosFlow_t unpack_qos_QosFlowInfo_t::RxQFlowGranted

8.1178.2.9 unpack_qos_swiQosFilter_t unpack_qos_QosFlowInfo_t::TxQFilter[25]

8.1178.2.10 unpack_qos_swiQosFlow_t unpack_qos_QosFlowInfo_t::TxQFlowGranted

8.1179 unpack_qos_QosFlowInfoState_t Struct Reference

Data Fields

- uint32_t id
- uint8_t isNewFlow
- uint8_t state

8.1179.1 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

8.1179.2 Field Documentation

8.1179.2.1 uint32_t unpack_qos_QosFlowInfoState_t::id

8.1179.2.2 uint8_t unpack_qos_QosFlowInfoState_t::isNewFlow

8.1179.2.3 uint8_t unpack_qos_QosFlowInfoState_t::state

8.1180 unpack_qos_SLQSQosGetNetworkStatus_t Struct Reference

Data Fields

- uint8_t NWQoSStatus

8.1180.1 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 • 0xff - Invalid Data.

8.1180.2 Field Documentation

8.1180.2.1 uint8_t unpack_qos_SLQSQosGetNetworkStatus_t::NWQoSStatus

8.1181 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

8.1181.1 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

8.1181.2 Field Documentation

8.1181.2.1 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_dl

8.1181.2.2 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_dl_ext

8.1181.2.3 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_dl_ext2

8.1181.2.4 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_ul

8.1181.2.5 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_ul_ext

8.1181.2.6 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_ul_ext2

8.1181.2.7 uint32_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::apnId

8.1182 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]

8.1182.1 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

8.1182.2 Field Documentation

8.1182.2.1 uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::apnId

8.1182.2.2 uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::numQosFlow

8.1182.2.3 unpack_QosFlowStat_t unpack_qos_SLQSQosSwiReadDataStats_t::qosFlow[10]

8.1182.2.4 uint64_t unpack_qos_SLQSQosSwiReadDataStats_t::total_rx_bytes

8.1182.2.5 uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::total_rx_pkt

8.1182.2.6 uint64_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_bytes

8.1182.2.7 uint64_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_bytes_drp

8.1182.2.8 uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_pkt

8.1182.2.9 uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_pkt_drp

8.1183 unpack_qos_SLQSSetQosEventCallback_ind_t Struct Reference

Data Fields

- uint8_t NumFlows
- unpack_qos_QosFlowInfo_t QosFlowInfo [8]

8.1183.1 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

8.1183.2 Field Documentation

8.1183.2.1 uint8_t unpack_qos_SLQSSetQosEventCallback_ind_t::NumFlows

8.1183.2.2 unpack_qos_QosFlowInfo_t unpack_qos_SLQSSetQosEventCallback_ind_t::QosFlowInfo[8]

8.1184 unpack_qos_SLQSSetQosNWStatusCallback_ind_t Struct Reference

Data Fields

- uint8_t status

8.1184.1 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

8.1184.2 Field Documentation

8.1184.2.1 uint8_t unpack_qos_SLQSSetQosNWStatusCallback_ind_t::status

8.1185 unpack_qos_SLQSSetQosPriEventCallback_ind_t Struct Reference

Data Fields

- uint16_t event

8.1185.1 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

8.1185.2 Field Documentation

8.1185.2.1 uint16_t unpack_qos_SLQSSetQosPriEventCallback_ind_t::event

8.1186 unpack_qos_SLQSSetQosStatusCallback_ind_t Struct Reference

Data Fields

- uint32_t id
- uint8_t status
- uint8_t event
- uint8_t reason

8.1186.1 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 • 0xff - Invalid Data.

8.1186.2 Field Documentation

8.1186.2.1 uint8_t unpack_qos_SLQSSetQosStatusCallback_ind_t::event

8.1186.2.2 uint32_t unpack_qos_SLQSSetQosStatusCallback_ind_t::id

8.1186.2.3 uint8_t unpack_qos_SLQSSetQosStatusCallback_ind_t::reason

8.1186.2.4 uint8_t unpack_qos_SLQSSetQosStatusCallback_ind_t::status

8.1187 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

8.1187.1 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

8.1187.2 Field Documentation

8.1187.2.1 uint32_t unpack_qos_swiQosFilter_t::EspSpi

8.1187.2.2 uint16_t unpack_qos_swiQosFilter_t::Id

8.1187.2.3 uint8_t unpack_qos_swiQosFilter_t::index

8.1187.2.4 unpack_qos_IPv4Addr_t unpack_qos_swiQosFilter_t::IPv4DstAddr

8.1187.2.5 unpack_qos_IPv4Addr_t unpack_qos_swiQosFilter_t::IPv4SrcAddr

8.1187.2.6 unpack_qos_Tos_t unpack_qos_swiQosFilter_t::IPv4Tos

8.1187.2.7 unpack_qos_IPv6Addr_t unpack_qos_swiQosFilter_t::IPv6DstAddr

8.1187.2.8 uint32_t unpack_qos_swiQosFilter_t::IPv6Label

8.1187.2.9 unpack_qos_IPv6Addr_t unpack_qos_swiQosFilter_t::IPv6SrcAddr

8.1187.2.10 unpack_qos_IPv6TrafCls_t unpack_qos_swiQosFilter_t::IPv6TrafCls

8.1187.2.11 uint8_t unpack_qos_swiQosFilter_t::is_EspSpi_Available

8.1187.2.12 uint8_t unpack_qos_swiQosFilter_t::is_Id_Available

8.1187.2.13 uint8_t unpack_qos_swiQosFilter_t::is_IPv4DstAddr_Available

8.1187.2.14 uint8_t unpack_qos_swiQosFilter_t::is_IPv4SrcAddr_Available

8.1187.2.15 uint8_t unpack_qos_swiQosFilter_t::is_IPv4Tos_Available

8.1187.2.16 uint8_t unpack_qos_swiQosFilter_t::is_IPv6DstAddr_Available

8.1187.2.17 uint8_t unpack_qos_swiQosFilter_t::is_IPv6Label_Available

8.1187.2.18 uint8_t unpack_qos_swiQosFilter_t::is_IPv6SrcAddr_Available

8.1187.2.19 uint8_t unpack_qos_swiQosFilter_t::is_IPv6TrafCls_Available

8.1187.2.20 uint8_t unpack_qos_swiQosFilter_t::is_NxtHdrProto_Available

8.1187.2.21 uint8_t unpack_qos_swiQosFilter_t::is_Precedence_Available

8.1187.2.22 uint8_t unpack_qos_swiQosFilter_t::is_TCPDstPort_Available

8.1187.2.23 uint8_t unpack_qos_swiQosFilter_t::is_TCPSrcPort_Available

8.1187.2.24 uint8_t unpack_qos_swiQosFilter_t::is_TranDstPort_Available

8.1187.2.25 uint8_t unpack_qos_swiQosFilter_t::is_TranSrcPort_Available

8.1187.2.26 uint8_t unpack_qos_swiQosFilter_t::is_UDPDstPort_Available

8.1187.2.27 uint8_t unpack_qos_swiQosFilter_t::is_UDPSrcPort_Available

8.1187.2.28 uint8_t unpack_qos_swiQosFilter_t::NxtHdrProto

8.1187.2.29 uint16_t unpack_qos_swiQosFilter_t::Precedence

8.1187.2.30 unpack_qos_Port_t unpack_qos_swiQosFilter_t::TCPDstPort

8.1187.2.31 unpack_qos_Port_t unpack_qos_swiQosFilter_t::TCPSrcPort

8.1187.2.32 unpack_qos_Port_t unpack_qos_swiQosFilter_t::TranDstPort

8.1187.2.33 unpack_qos_Port_t unpack_qos_swiQosFilter_t::TranSrcPort

8.1187.2.34 unpack_qos_Port_t unpack_qos_swiQosFilter_t::UDPDstPort

8.1187.2.35 unpack_qos_Port_t unpack_qos_swiQosFilter_t::UDPSrcPort

8.1187.2.36 uint8_t unpack_qos_swiQosFilter_t::version

8.1188 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

8.1188.1 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_3GPPRes-ResidualBER	• IP flow 3GPP residual bit error rate • residual_bit_error_rate • 0 = 5∗10-2 residual BER • 1 = 1∗10-2 residual BER • 2 = 5∗10-3 residual BER • 3 = 4∗10-3 residual BER • 4 = 1∗10-3 residual BER • 5 = 1∗10-4 residual BER • 6 = 1∗10-5 residual BER • 7 = 1∗10-6 residual BER • 8 = 6∗10-8 residual BER • Integer that indicates the undetected BER for each IP flow in the delivered packets; Tapplies only to 3GPP networks
val_3GPPTra-HdlPri	• 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

8.1188.2 Field Documentation

8.1188.2.1 unpack_qos_dataRate_t unpack_qos_swiQosFlow_t::DataRate

8.1188.2.2 uint8_t unpack_qos_swiQosFlow_t::index

8.1188.2.3 uint8_t unpack_qos_swiQosFlow_t::is_DataRate_Available

8.1188.2.4 uint8_t unpack_qos_swiQosFlow_t::is_Jitter_Available

8.1188.2.5 uint8_t unpack_qos_swiQosFlow_t::is_Latency_Available

8.1188.2.6 uint8_t unpack_qos_swiQosFlow_t::is_LteQci_Available

8.1188.2.7 uint8_t unpack_qos_swiQosFlow_t::is_MaxAllowedPktSz_Available

8.1188.2.8 uint8_t unpack_qos_swiQosFlow_t::is_MinPolicedPktSz_Available

8.1188.2.9 uint8_t unpack_qos_swiQosFlow_t::is_PktErrRate_Available

8.1188.2.10 uint8_t unpack_qos_swiQosFlow_t::is_ProfileId3GPP2_Available

8.1188.2.11 uint8_t unpack_qos_swiQosFlow_t::is_TokenBucket_Available

8.1188.2.12 uint8_t unpack_qos_swiQosFlow_t::is_TrafficClass_Available

8.1188.2.13 uint8_t unpack_qos_swiQosFlow_t::is_val_3GPP2Pri_Available

8.1188.2.14 uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPImCn_Available

8.1188.2.15 uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPResResidualBER_Available

8.1188.2.16 uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPSigInd_Available

8.1188.2.17 uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPTraHdlPri_Available

8.1188.2.18 uint32_t unpack_qos_swiQosFlow_t::Jitter

8.1188.2.19 uint32_t unpack_qos_swiQosFlow_t::Latency

8.1188.2.20 uint8_t unpack_qos_swiQosFlow_t::LteQci

8.1188.2.21 uint32_t unpack_qos_swiQosFlow_t::MaxAllowedPktSz

8.1188.2.22 uint32_t unpack_qos_swiQosFlow_t::MinPolicedPktSz

8.1188.2.23 unpack_qos_pktErrRate_t unpack_qos_swiQosFlow_t::PktErrRate

8.1188.2.24 uint16_t unpack_qos_swiQosFlow_t::ProfileId3GPP2

8.1188.2.25 unpack_qos_tokenBucket_t unpack_qos_swiQosFlow_t::TokenBucket

8.1188.2.26 uint8_t unpack_qos_swiQosFlow_t::TrafficClass

8.1188.2.27 uint8_t unpack_qos_swiQosFlow_t::val_3GPP2Pri

8.1188.2.28 uint8_t unpack_qos_swiQosFlow_t::val_3GPPImCn

8.1188.2.29 uint16_t unpack_qos_swiQosFlow_t::val_3GPPResResidualBER

8.1188.2.30 uint8_t unpack_qos_swiQosFlow_t::val_3GPPSigInd

8.1188.2.31 uint8_t unpack_qos_swiQosFlow_t::val_3GPPTraHdlPri

8.1189 unpack_qos_tokenBucket_t Struct Reference

Data Fields

- uint32_t peakRate
- uint32_t tokenRate
- uint32_t bucketSz

8.1189.1 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

8.1189.2 Field Documentation

8.1189.2.1 uint32_t unpack_qos_tokenBucket_t::bucketSz

8.1189.2.2 uint32_t unpack_qos_tokenBucket_t::peakRate

8.1189.2.3 uint32_t unpack_qos_tokenBucket_t::tokenRate

8.1190 unpack_qos_Tos_t Struct Reference

Data Fields

- uint8_t val
- uint8_t mask

8.1190.1 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.

8.1190.2 Field Documentation

8.1190.2.1 uint8_t unpack_qos_Tos_t::mask

8.1190.2.2 uint8_t unpack_qos_Tos_t::val

8.1191 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

8.1191.1 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

8.1191.2 Field Documentation

8.1191.2.1 uint32_t unpack_QosFlowStat_t::bearerId

8.1191.2.2 uint64_t unpack_QosFlowStat_t::tx_bytes

8.1191.2.3 uint64_t unpack_QosFlowStat_t::tx_bytes_drp

8.1191.2.4 uint32_t unpack_QosFlowStat_t::tx_pkt

8.1191.2.5 uint32_t unpack_QosFlowStat_t::tx_pkt_drp

8.1192 unpack_rms_GetSMSWake_t Struct Reference

Data Fields

- uint32_t enabled
- uint32_t wake_mask

8.1192.1 Detailed Description

Unpack get SMS wake parameters

Parameters

enabled	• SMS wake functionality enabled – 0 - Disabled – 1 - Enabled – 0xffffffff - invalid value.
wake_mask	• SMS wake mask to search for incoming messages (only relevant when enabled)

8.1192.2 Field Documentation

8.1192.2.1 uint32_t unpack_rms_GetSMSWake_t::enabled

8.1192.2.2 uint32_t unpack_rms_GetSMSWake_t::wake_mask

8.1193 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

8.1193.1 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.

8.1193.2 Field Documentation

8.1193.2.1 wds_DataULongTlv unpack_RMTransferStatistics_ind_t::RxDropConutTlv

8.1193.2.2 wds_DataULongLongTlv unpack_RMTransferStatistics_ind_t::RxOkByteCountTlv

8.1193.2.3 wds_DataULongTlv unpack_RMTransferStatistics_ind_t::RxOkConutTlv

8.1193.2.4 wds_DataULongTlv unpack_RMTransferStatistics_ind_t::TxDropConutTlv

8.1193.2.5 wds_DataULongLongTlv unpack_RMTransferStatistics_ind_t::TxOkByteCountTlv

8.1193.2.6 wds_DataULongTlv unpack_RMTransferStatistics_ind_t::TxOkConutTlv

8.1194 unpack_sar_SLQSGetRfSarState_t Struct Reference

Data Fields

- uint32_t * pSarRFState

8.1194.1 Detailed Description

This structure contains unpack get RF SAR state parameter.

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 • NULL pointer - Invalid data.

8.1194.2 Field Documentation

8.1194.2.1 uint32_t* unpack_sar_SLQSGetRfSarState_t::pSarRFState

8.1195 unpack_sms_GetSMSCAddress_t Struct Reference

Data Fields

- uint8_t addressSize
- uint8_t * pSMSCAddress
- uint8_t typeSize
- uint8_t * pSMSCType

8.1195.1 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.

8.1195.2 Field Documentation

8.1195.2.1 uint8_t unpack_sms_GetSMSCAddress_t::addressSize

8.1195.2.2 uint8_t* unpack_sms_GetSMSCAddress_t::pSMSCAddress

8.1195.2.3 uint8_t* unpack_sms_GetSMSCAddress_t::pSMSCType

8.1195.2.4 uint8_t unpack_sms_GetSMSCAddress_t::typeSize

8.1196 unpack_sms_SaveSMS_t Struct Reference

Data Fields

- uint32_t * pMessageIndex

8.1196.1 Detailed Description

This structure contains unpack save SMS parameter.

Parameters

pMessageIndex	• The message index assigned by the device

8.1196.2 Field Documentation

8.1196.2.1 uint32_t* unpack_sms_SaveSMS_t::pMessageIndex

8.1197 unpack_sms_SendSMS_t Struct Reference

Data Fields

- uint16_t messageID
- uint32_t messageFailureCode

8.1197.1 Detailed Description

This structure contains unpack send SMS parameters.

Parameters

messageID	• WMS message ID
messageFailure-Code	• pointer to message failure code. If cause code is not provided, then value will be 0xFFFFFFFF

8.1197.2 Field Documentation

8.1197.2.1 uint32_t unpack_sms_SendSMS_t::messageFailureCode

8.1197.2.2 uint16_t unpack_sms_SendSMS_t::messageID

8.1198 unpack_sms_SetNewSMSCallback_ind_t Struct Reference

Data Fields

- newMTMessageTlv NewMMTlv
- transferRouteMessageTlv TRMessageTlv
- messageModeTlv MMTlv
- sMSEtwsMessageTlv ETWSTlv
- eTWSPLMNInfoTlv ETWSPLMNTlv
- sMSCAddressTlv SMSCTlv
- sMSOnIMSTlv IMSTlv

8.1198.1 Detailed Description

This structure contains unpack new SMS callback indication.

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

8.1198.2 Field Documentation

8.1198.2.1 eTWSPLMNInfoTlv unpack_sms_SetNewSMSCallback_ind_t::ETWSPLMNTlv

8.1198.2.2 sMSEtwsMessageTlv unpack_sms_SetNewSMSCallback_ind_t::ETWSTlv

8.1198.2.3 sMSOnIMSTlv unpack_sms_SetNewSMSCallback_ind_t::IMSTlv

8.1198.2.4 messageModeTlv unpack_sms_SetNewSMSCallback_ind_t::MMTlv

8.1198.2.5 newMTMessageTlv unpack_sms_SetNewSMSCallback_ind_t::NewMMTlv

8.1198.2.6 sMSCAddressTlv unpack_sms_SetNewSMSCallback_ind_t::SMSCTlv

8.1198.2.7 transferRouteMessageTlv unpack_sms_SetNewSMSCallback_ind_t::TRMessageTlv

8.1199 unpack_sms_SetNewSMSCallback_t Struct Reference

8.1199.1 Detailed Description

This structure contains unpack set new SMS callback.

Parameters

NULL	dummy prototype.

8.1200 unpack_sms_SLQSDeleteSMS_t Struct Reference

8.1200.1 Detailed Description

This structure contains unpack delete SMS parameters.

Parameters

NULL	unpack prototype.

8.1201 unpack_sms_SLQSGetIndicationRegister_t Struct Reference

Data Fields

- sms_getIndicationReg * pGetIndicationRegInfo

8.1201.1 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

8.1201.2 Field Documentation

8.1201.2.1 sms_getIndicationReg* unpack_sms_SLQSGetIndicationRegister_t::pGetIndicationRegInfo

8.1202 unpack_sms_SLQSGetMessageWaiting_t Struct Reference

Data Fields

- sms_getMsgWaitingInfo * pGetMsgWaitingInfoResp

8.1202.1 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 • NULL pointer - Invalid data.

8.1202.2 Field Documentation

8.1202.2.1 sms_getMsgWaitingInfo* unpack_sms_SLQSGetMessageWaiting_t::pGetMsgWaitingInfoResp

8.1203 unpack_sms_SLQSGetSMS_t Struct Reference

Data Fields

- uint32_t messageTag
- uint32_t messageFormat
- uint32_t messageSize
- uint8_t message [2048]

8.1203.1 Detailed Description

This sturcture contains unpack get SMS parameters.

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

8.1203.2 Field Documentation

8.1203.2.1 uint8_t unpack_sms_SLQSGetSMS_t::message[2048]

8.1203.2.2 uint32_t unpack_sms_SLQSGetSMS_t::messageFormat

8.1203.2.3 uint32_t unpack_sms_SLQSGetSMS_t::messageSize

8.1203.2.4 uint32_t unpack_sms_SLQSGetSMS_t::messageTag

8.1204 unpack_sms_SLQSGetSmsBroadcastConfig_t Struct Reference

Data Fields

- sms_qaQmi3GPPBroadcastCfgInfo * pBroadcastConfig
- sms_qaQmi3GPP2BroadcastCfgInfo * pCDMABroadcastConfig

8.1204.1 Detailed Description

Structure contain parameters that Provides Information about the SMS BroadcastConfiguration

Parameters

pBroadcast-Config	• The data for 3GPP Broadcast Information(Optional). • NULL pointer - Invalid data.
pCDMA-BroadcastConfig	• The data for 3GPP2 Broadcast Information(Optional). • NULL pointer - Invalid data.

8.1204.2 Field Documentation

8.1204.2.1 sms_qaQmi3GPPBroadcastCfgInfo* unpack_sms_SLQSGetSmsBroadcastConfig_t::pBroadcastConfig

8.1204.2.2 sms_qaQmi3GPP2BroadcastCfgInfo* unpack_sms_SLQSGetSmsBroadcastConfig_t::pCDMABroadcast-Config

8.1205 unpack_sms_SLQSGetSMSList_t Struct Reference

Data Fields

- uint32_t messageListSize
- qmiSmsMessageList messageList [255]

8.1205.1 Detailed Description

This structure contains unpack get SMS list parameters.

Parameters

messageListSize	• Upon input the maximum number of elements that the message list array can contain. • Upon successful output the actual number of elements in the message list array.
messageList	• Message List • See qmiSmsMessageList for more information

8.1205.2 Field Documentation

8.1205.2.1 qmiSmsMessageList unpack_sms_SLQSGetSMSList_t::messageList[255]

8.1205.2.2 uint32_t unpack_sms_SLQSGetSMSList_t::messageListSize

8.1206 unpack_sms_SLQSGetTransLayerInfo_t Struct Reference

Data Fields

- sms_getTransLayerInfo * pGetTransLayerInfo

8.1206.1 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

8.1206.2 Field Documentation

8.1206.2.1 sms_getTransLayerInfo* unpack_sms_SLQSGetTransLayerInfo_t::pGetTransLayerInfo

8.1207 unpack_sms_SLQSGetTransNWRegInfo_t Struct Reference

Data Fields

- sms_getTransNWRegInfo * pGetTransNWRegInfo

8.1207.1 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

8.1207.2 Field Documentation

8.1207.2.1 sms_getTransNWRegInfo* unpack_sms_SLQSGetTransNWRegInfo_t::pGetTransNWRegInfo

8.1208 unpack_sms_SLQSModifySMSStatus_t Struct Reference

8.1208.1 Detailed Description

This structure contains unpack modify SMS status parameters.

Parameters

NULL	unpack prototype.

8.1209 unpack_sms_SLQSSmsGetMaxStorageSize_t Struct Reference

Data Fields

- sms_maxStorageSizeResp * pMaxStorageSizeResp

8.1209.1 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 • NULL pointer - Invalid data.

8.1209.2 Field Documentation

8.1209.2.1 sms_maxStorageSizeResp* unpack_sms_SLQSSmsGetMaxStorageSize_t::pMaxStorageSizeResp

8.1210 unpack_sms_SLQSSmsGetMessageProtocol_t Struct Reference

Data Fields

- sms_msgProtocolResp * pMessageProtocol

8.1210.1 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

8.1210.2 Field Documentation

8.1210.2.1 sms_msgProtocolResp* unpack_sms_SLQSSmsGetMessageProtocol_t::pMessageProtocol

8.1211 unpack_sms_SLQSSwiGetSMSStorage_t Struct Reference

Data Fields

- uint32_t * pSmsStorage

8.1211.1 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 • NULL pointer - Invalid data.

8.1211.2 Field Documentation

8.1211.2.1 uint32_t* unpack_sms_SLQSSwiGetSMSStorage_t::pSmsStorage

8.1212 unpack_sms_SLQSWmsMemoryFullCallBack_ind_t Struct Reference

Data Fields

- uint8_t storageType
- uint8_t messageMode

8.1212.1 Detailed Description

This structure contains unpack SMS memory full callback indication.

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)

8.1212.2 Field Documentation

8.1212.2.1 uint8_t unpack_sms_SLQSWmsMemoryFullCallBack_ind_t::messageMode

8.1212.2.2 uint8_t unpack_sms_SLQSWmsMemoryFullCallBack_ind_t::storageType

8.1213 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

8.1213.1 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 – 0xff - not reported by modem
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

8.1213.2 Field Documentation

8.1213.2.1 uint32_t unpack_swiloc_SwiLocGetAutoStart_t::fix_rate

8.1213.2.2 int unpack_swiloc_SwiLocGetAutoStart_t::fix_rate_reported

8.1213.2.3 uint8_t unpack_swiloc_SwiLocGetAutoStart_t::fix_type

8.1213.2.4 int unpack_swiloc_SwiLocGetAutoStart_t::fix_type_reported

8.1213.2.5 uint8_t unpack_swiloc_SwiLocGetAutoStart_t::function

8.1213.2.6 int unpack_swiloc_SwiLocGetAutoStart_t::function_reported

8.1213.2.7 uint32_t unpack_swiloc_SwiLocGetAutoStart_t::max_dist

8.1213.2.8 int unpack_swiloc_SwiLocGetAutoStart_t::max_dist_reported

8.1213.2.9 uint8_t unpack_swiloc_SwiLocGetAutoStart_t::max_time

8.1213.2.10 int unpack_swiloc_SwiLocGetAutoStart_t::max_time_reported

8.1214 unpack_swioma_SLQSOMADMAlertCallback_ind_t Struct Reference

Data Fields

- uint32_t eventType
- unpack_omaDmFotaTlv_t SessionInfoFota
- unpack_omaDmConfigTlv_t SessionInfoConfig
- unpack_omaDmNotificationsTlv_t SessionInfoNotification

8.1214.1 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 • 0xff - indication missing event information.
SessionInfo-Fota[OUT]	• See unpack_omaDmFotaTlv_t for more information
SessionInfo-Config[OUT]	• See unpack_omaDmConfigTlv_t for more information
SessionInfo-Notification[OUT]	• See unpack_omaDmNotificationsTlv_t for more information

8.1214.2 Field Documentation

8.1214.2.1 uint32_t unpack_swioma_SLQSOMADMAlertCallback_ind_t::eventType

8.1214.2.2 unpack_omaDmConfigTlv_t unpack_swioma_SLQSOMADMAlertCallback_ind_t::SessionInfoConfig

8.1214.2.3 unpack_omaDmFotaTlv_t unpack_swioma_SLQSOMADMAlertCallback_ind_t::SessionInfoFota

8.1214.2.4 unpack_omaDmNotificationsTlv_t unpack_swioma_SLQSOMADMAlertCallback_ind_t::SessionInfo-Notification

8.1215 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

8.1215.1 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 – 0xff - Invalid Session information.
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

8.1215.2 Field Documentation

8.1215.2.1 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::Date[255]

8.1215.2.2 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::DateLength

8.1215.2.3 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::PkgDescLength

8.1215.2.4 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::PkgDescription[255]

8.1215.2.5 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::PkgName[255]

8.1215.2.6 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::PkgNameLength

8.1215.2.7 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::RetryCount

8.1215.2.8 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::SessionState

8.1215.2.9 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::SessionType

8.1215.2.10 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::Severity

8.1215.2.11 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::Source[255]

8.1215.2.12 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::SourceLength

8.1215.2.13 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::Status

8.1215.2.14 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::Time[255]

8.1215.2.15 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::TimeLength

8.1215.2.16 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::UpdateCompleteStatus

8.1216 unpack_swioma_SLQSOMADMGetSettings_t Struct Reference

Data Fields

- uint32_t OMADMEnabled
- uint8_t FOTAdownload
- uint8_t FOTAUpdate
- uint8_t Autosdm
- uint8_t FwAutoCheck

8.1216.1 Detailed Description

Structure containing the OMA DM settings retrieved from the device

Parameters

OMADMEnabled	[out] • Optional 4 byte parameter indicating OMADM service enabled – 0x00000001 - Client-initiated device configuration – 0x00000002 - Network-initiated device configuration – 0x00000010 - Client-initiated FUMO – 0x00000020 - Network-initiated FUMO • function SLQSOMADMGetSettings2() returns a default value 0xFFFFFFFF in case this parameter is not returned by the modem.
FOTAdownload	[out] • Optional 1 Byte parameter indicating support for FOTA Automatic download – 0x00 - Host permission required before downloading – 0x01 - Automatically start downloading, no host permission required – 0x02 - Automatically start downloading, while not roaming – 0x03 - Automatically reject download – 0x04 - Automatically reject download with “Enterprise Reject Policy – 0xff - Invalid data.

- function SLQSOMADMGetSettings2() returns a default value 0xFF in case this parameter is not returned by the modem.

Parameters

FOTAUpdate	[out] • Optional 1 byte parameter indicating FOTA Automatic update – 0x00 - User permission required before updating firmware – 0x01 - No user permission required before updating firmware – 0x02 - User permission required, auto update on power up • function SLQSOMADMGetSettings2() returns a default value 0xFF in case this parameter is not returned by the modem.
Autosdm	[out] • Optional 1 byte parameter indicating OMA Automatic UI Alert Response – 0x00 - Disabled – 0x01 - Enabled Accept – 0x02 - Enabled Reject • function SLQSOMADMGetSettings2() returns a default value 0xFF in case this parameter is not returned by the modem.
FwAutoCheck	[out] • Optional 1 byte parameter indicating OMA Automatic Check for Firmware Update on Power-Up Response – 0x00 - Disabled – 0x01 - Enabled • function SLQSOMADMGetSettings2() returns a default value 0xFF in case this parameter is not returned by the modem.

8.1216.2 Field Documentation

8.1216.2.1 uint8_t unpack_swioma_SLQSOMADMGetSettings_t::Autosdm

8.1216.2.2 uint8_t unpack_swioma_SLQSOMADMGetSettings_t::FOTAdownload

8.1216.2.3 uint8_t unpack_swioma_SLQSOMADMGetSettings_t::FOTAUpdate

8.1216.2.4 uint8_t unpack_swioma_SLQSOMADMGetSettings_t::FwAutoCheck

8.1216.2.5 uint32_t unpack_swioma_SLQSOMADMGetSettings_t::OMADMEnabled

8.1217 unpack_swioma_SLQSOMADMStartSession_t Struct Reference

Data Fields

- uint32_t FwAvailability

8.1217.1 Detailed Description

Structure that contains the responses for OMA start session command

Parameters

FwAvailability	[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 – 0xFFFFFFFF - Invalid data.

8.1217.2 Field Documentation

8.1217.2.1 uint32_t unpack_swioma_SLQSOMADMStartSession_t::FwAvailability

8.1218 unpack_uim_ChangePin_t Struct Reference

Data Fields

- uim_remainingRetries * pRemainingRetries
- uim_encryptedPIN1 * pEncryptedPIN1
- uint32_t * pIndicationToken
- uint16_t TlvResult

8.1218.1 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. • NULL pointer - Invalid data.
pEncryptedPI-N1(optional)	• See uim_encryptedPIN1 for more information. • NULL pointer - Invalid data.
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 • NULL pointer - Invalid data.

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_encryptedPIN1* unpack_uim_ChangePin_t::pEncryptedPIN1

8.1218.2.2 uint32_t* unpack_uim_ChangePin_t::pIndicationToken

8.1218.2.3 uim_remainingRetries* unpack_uim_ChangePin_t::pRemainingRetries

8.1218.2.4 uint16_t unpack_uim_ChangePin_t::Tlvresult

8.1219 unpack_uim_GetCardStatus_t Struct Reference

Data Fields

- uim_cardStatus * pCardStatus
- uim_hotSwapStatus * pHotSwapStatus
- uint16_t Tlvresult

8.1219.1 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. • NULL pointer - Invalid data.
pHotSwap-Status(optional)	• See uim_hotSwapStatus for more information. • NULL pointer - Invalid data.

Note

Using NULL for the pointers would make sure that the parameter is not returned.

8.1219.2 Field Documentation

8.1219.2.1 uim_cardStatus* unpack_uim_GetCardStatus_t::pCardStatus

8.1219.2.2 uim_hotSwapStatus* unpack_uim_GetCardStatus_t::pHotSwapStatus

8.1219.2.3 uint16_t unpack_uim_GetCardStatus_t::Tlvresult

8.1220 unpack_uim_ReadTransparent_t Struct Reference

Data Fields

- uim_cardResult * pCardResult
- uim_readResult * pReadResult
- uint32_t * pIndicationToken
- uint8_t * pEncryptedData
- uint16_t Tlvresult

8.1220.1 Detailed Description

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

Parameters

pCardResult	• See uim_cardResult for more information. • NULL pointer - Invalid data.
pReadResult	• See uim_readResult for more information. • NULL pointer - Invalid data.
pIndication-Token(optional)	• Response in Indication. • When this TLV is present, it indicates that the result must be provided in a subsequent indication. • NULL pointer - Invalid data.
pEncrypted-Data(optional)	• Encrypted Data. • Indicates whether the data from the card passed in read_result is encrypted. • NULL pointer - Invalid data.

Note

Using NULL for the pointers would make sure that the parameter is not added to the request.

8.1220.2 Field Documentation

8.1220.2.1 uim_cardResult* unpack_uim_ReadTransparent_t::pCardResult

8.1220.2.2 uint8_t* unpack_uim_ReadTransparent_t::pEncryptedData

8.1220.2.3 uint32_t* unpack_uim_ReadTransparent_t::pIndicationToken

8.1220.2.4 uim_readResult* unpack_uim_ReadTransparent_t::pReadResult

8.1220.2.5 uint16_t unpack_uim_ReadTransparent_t::Tlvresult

8.1221 unpack_uim_SetPinProtection_t Struct Reference

Data Fields

- uim_remainingRetries * pRemainingRetries
- uim_encryptedPIN1 * pEncryptedPIN1
- uint32_t * pIndicationToken
- uint16_t Tlvresult

8.1221.1 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. • NULL pointer - Invalid data.
pEncryptedPI-N1(optional)	• See uim_encryptedPIN1 for more information. • NULL pointer - Invalid data.
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 • NULL pointer - Invalid data.

Note

Using NULL for the pointers would make sure that the parameter is not returned.

8.1221.2 Field Documentation

8.1221.2.1 uim_encryptedPIN1∗ unpack_uim_SetPinProtection_t::pEncryptedPIN1

8.1221.2.2 uint32_t∗ unpack_uim_SetPinProtection_t::pIndicationToken

8.1221.2.3 uim_remainingRetries∗ unpack_uim_SetPinProtection_t::pRemainingRetries

8.1221.2.4 uint16_t unpack_uim_SetPinProtection_t::TlvResult

8.1222 unpack_uim_SetUimSlotStatusChangeCallback_ind_t Struct Reference

Data Fields

- slots_t slotsstatusChange
- uint8_t bNumberOfPhySlots

8.1222.1 Detailed Description

Structure consist of card status params

Parameters

slotsstatus-Change	• See slot_t for more information
bNumberOfPhy-Slots	• Number of Physical Slot(s) • 0xff - Invalid card slot status.

8.1222.2 Field Documentation

8.1222.2.1 uint8_t unpack_uim_SetUimSlotStatusChangeCallback_ind_t::bNumberOfPhySlots

8.1222.2.2 slots_t unpack_uim_SetUimSlotStatusChangeCallback_ind_t::slotsstatusChange

8.1223 unpack_uim_SLQSUIMAuthenticate_t Struct Reference

Data Fields

- uim_cardResult * pCardResult
- uim_authenticateResult * pAuthenticateResult
- uint32_t * pIndicationToken

8.1223.1 Detailed Description

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

Parameters

pCard-Result(optional)	• See uim_cardResult for more information. • NULL pointer - Invalid data.
pAuthenticate-Result(optional)	• See uim_authenticateResult for more information. • NULL pointer - Invalid data.
pIndication-Token(optional)	• Response in Indication. • When this TLV is present, it indicates that the result must be provided in a subsequent indication. • NULL pointer - Invalid data.

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_authenticateResult* unpack_uim_SLQSUIMAuthenticate_t::pAuthenticateResult

8.1223.2.2 uim_cardResult* unpack_uim_SLQSUIMAuthenticate_t::pCardResult

8.1223.2.3 uint32_t* unpack_uim_SLQSUIMAuthenticate_t::pIndicationToken

8.1224 unpack_uim_SLQSUIMDepersonalization_t Struct Reference

Data Fields

- uim_remainingRetries * pRemainingRetries

8.1224.1 Detailed Description

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

Parameters

pRemaining-Retries(optional)	• See uim_remainingRetries for more information. • NULL pointer - Invalid data.

Note

Using NULL for the pointers would make sure that the parameter is not returned.

8.1224.2 Field Documentation

8.1224.2.1 uim_remainingRetries* unpack_uim_SLQSUIMDepersonalization_t::pRemainingRetries

8.1225 unpack_uim_SLQSUIMEventRegister_t Struct Reference

Data Fields

- uint32_t eventMask

8.1225.1 Detailed Description

This structure contains unpack event register parameter.

Parameters

eventMask	- bit 0 - card status • bit 1 - SAP connection • bit 4 - physical slot status • 0xffffffff - Invalid data.

8.1225.2 Field Documentation

8.1225.2.1 uint32_t unpack_uim_SLQSUIMEventRegister_t::eventMask

8.1226 unpack_uim_SLQSUIMGetConfiguration_t Struct Reference

Data Fields

- uint8_t * pAutoSelection
- uim_personalizationStatus * pPersonalizationStatus
- uint8_t * pHaltSubscription

8.1226.1 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 • NULL pointer - Invalid data.
p-Personalization-Status(optional)	• See uim_personalizationStatus for more information. • NULL pointer - Invalid data.
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 • NULL pointer - Invalid data.

8.1226.2 Field Documentation

8.1226.2.1 uint8_t* unpack_uim_SLQSUIMGetConfiguration_t::pAutoSelection

8.1226.2.2 uint8_t* unpack_uim_SLQSUIMGetConfiguration_t::pHaltSubscription

8.1226.2.3 uim_personalizationStatus* unpack_uim_SLQSUIMGetConfiguration_t::pPersonalizationStatus

8.1227 unpack_uim_SLQSUIMGetFileAttributes_t Struct Reference

Data Fields

- uim_cardResult * pCardResult
- uim_fileAttributes * pFileAttributes
- uint32_t * pIndicationToken

8.1227.1 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. • NULL pointer - Invalid data.
pFile-Attributes(optional)	• See uim_fileAttributes for more information. • NULL pointer - Invalid data.

pIndication-Token(optional)	• Response in Indication. • When this TLV is present, it indicates that the result must be provided in a subsequent indication. • NULL pointer - Invalid data.

Note

Using NULL for the pointers would make sure that the parameter is not returned.

8.1227.2 Field Documentation

8.1227.2.1 uim_cardResult* unpack_uim_SLQSUIMGetFileAttributes_t::pCardResult

8.1227.2.2 uim_fileAttributes* unpack_uim_SLQSUIMGetFileAttributes_t::pFileAttributes

8.1227.2.3 uint32_t* unpack_uim_SLQSUIMGetFileAttributes_t::pIndicationToken

8.1228 unpack_uim_SLQSUIMGetSlotsStatus_t Struct Reference

Data Fields

- uint8_t * pNumberOfPhySlot
- slots_t * pUimSlotsStatus

8.1228.1 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. • NULL pointer - Invalid data.
pUimSlotsStatus	• Slots Status See slots_t for more information..

8.1228.2 Field Documentation

8.1228.2.1 uint8_t* unpack_uim_SLQSUIMGetSlotsStatus_t::pNumberOfPhySlot

8.1228.2.2 slots_t* unpack_uim_SLQSUIMGetSlotsStatus_t::pUimSlotsStatus

8.1229 unpack_uim_SLQSUIMRefreshGetLastEvent_t Struct Reference

Data Fields

- uim_refreshevent * pRefreshEvent

8.1229.1 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. • NULL pointer - Invalid data.

8.1229.2 Field Documentation

8.1229.2.1 uim_refreshevent* unpack_uim_SLQSUIMRefreshGetLastEvent_t::pRefreshEvent

8.1230 unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t Struct Reference

Data Fields

- uim_cardStatus * pCardStatus

8.1230.1 Detailed Description

This structure contains information about Status change callback.

Parameters

pCardStatus	Card Status • See uim_cardStatus for more information. • NULL pointer - Invalid data.

8.1230.2 Field Documentation

8.1230.2.1 uim_cardStatus* unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t::pCardStatus

8.1231 unpack_uim_UnblockPin_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. • NULL pointer - Invalid data.
pEncryptedPI-N1(optional)	• See uim_encryptedPIN1 for more information. • NULL pointer - Invalid data.
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 • NULL pointer - Invalid data.

8.1231.2 Field Documentation

8.1231.2.1 uim_encryptedPIN1∗ unpack_uim_UnblockPin_t::pEncryptedPIN1

8.1231.2.2 uint32_t∗ unpack_uim_UnblockPin_t::pIndicationToken

8.1231.2.3 uim_remainingRetries∗ unpack_uim_UnblockPin_t::pRemainingRetries

8.1231.2.4 uint16_t unpack_uim_UnblockPin_t::Tlvresult

8.1232 unpack_uim_VerifyPin_t Struct Reference

Data Fields

- uim_remainingRetries ∗ pRemainingRetries
- uim_encryptedPIN1 ∗ pEncryptedPIN1
- uint32_t ∗ pIndicationToken
- uint16_t Tlvresult

8.1232.1 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. • NULL pointer - Invalid data.
pEncryptedPI-N1(optional)	• See uim_encryptedPIN1 for more information. • NULL pointer - Invalid data.

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 • NULL pointer - Invalid data.

Note

Using NULL for the pointers would make sure that the parameter is not returned.

8.1232.2 Field Documentation

8.1232.2.1 uim_encryptedPIN1∗ unpack_uim_VerifyPin_t::pEncryptedPIN1

8.1232.2.2 uint32_t∗ unpack_uim_VerifyPin_t::pIndicationToken

8.1232.2.3 uim_remainingRetries∗ unpack_uim_VerifyPin_t::pRemainingRetries

8.1232.2.4 uint16_t unpack_uim_VerifyPin_t::Tlvresult

8.1233 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

8.1233.1 Detailed Description

This structure contains the parameters of USS response.

Parameters

pfailureCause

- Failure Cause.
- 0 - QMI_FAILURE_CAUSE_OFFLINE - Phone is offline
- 20 - QMI_FAILURE_CAUSE_CDMA_LOCK - Phone is CDMA locked until a power cycle; CDMA only
- 21 - QMI_FAILURE_CAUSE_NO_SRV - Phone has no service
- 22 - QMI_FAILURE_CAUSE_FADE - Call has ended abnormally
- 23 - QMI_FAILURE_CAUSE_INTERCEPT - Received intercept from the base station; originating only; CDMA only
- 24 - QMI_FAILURE_CAUSE_REORDER - Received reorder from the base station; originating only; CDMA only
- 25 - QMI_FAILURE_CAUSE_REL_NORMAL - Received release from the base station; no reason was given
- 26 - QMI_FAILURE_CAUSE_REL_SO_REJ - Received release from the base station; SO reject; CDMA only
- 27 - QMI_FAILURE_CAUSE_INCOM_CALL - Received incoming call from the base station
- 28 - QMI_FAILURE_CAUSE_ALERT_STOP - Received alert stop from the base station; incoming only; CDMA only
- 29 - QMI_FAILURE_CAUSE_CLIENT_END - Client ended the call
- 30 - QMI_FAILURE_CAUSE_ACTIVATION - Received end activation; OTASP call only; CDMA only
- 31 - QMI_FAILURE_CAUSE_MC_ABORT - MC aborted the origination/conversation; CDMA only
- 32 - QMI_FAILURE_CAUSE_MAX_ACCESS_PROBE - Maximum access probes were transmitted; CDMA only
- 33 - QMI_FAILURE_CAUSE_PSIST_N - Persistence test failure; FEATURE_JCDMA only; CDMA only
- 34 - QMI_FAILURE_CAUSE_UIM_NOT_PRESENT - R-UIM is not present
- 35 - QMI_FAILURE_CAUSE_ACC_IN_PROG - Access attempt is already in progress
- 36 - QMI_FAILURE_CAUSE_ACC_FAIL - Access failure for a reason other than the above
- 37 - QMI_FAILURE_CAUSE_RETRY_ORDER - Received retry order; originating only; IS 2000; CDMA only
- 38 - QMI_FAILURE_CAUSE_CCS_NOT_SUPPORTED_BYBS - Concurrent service is not supported by the base station
- 39 - QMI_FAILURE_CAUSE_NO_RESPONSE_FROM_BS - No response was received from the base station
- 40 - QMI_FAILURE_CAUSE_REJECTED_BY_BS - Call was rejected by the base station; CDMA only
- 41 - QMI_FAILURE_CAUSE_INCOMPATIBLE - Concurrent services requested were not compatible; CDMA only
- 42 - QMI_FAILURE_CAUSE_ACCESS_BLOCK - Access is blocked by the base station; CDMA only
- 43 - QMI_FAILURE_CAUSE_ALREADY_IN_TC - Corresponds to CM_CALL_ORIGERR_ALREADY_IN_TC
- 44 - QMI_FAILURE_CAUSE_EMERGENCY_FLASHED - Call is ended because an emergency call was flashed over this call; CDMA only
- 45 - QMI_FAILURE_CAUSE_USER_CALL_ORIGDURING_GPS - Used if CM is ending a GPS call in preference of a user call
- 46 - QMI_FAILURE_CAUSE_USER_CALL_ORIGDURING_SMS - Used if CM is ending an SMS call in preference of a user call
- 47 - QMI_FAILURE_CAUSE_USER_CALL_ORIGDURING_DATA - Used if CM is ending a data call in preference of an emergency call
- 48 - QMI_FAILURE_CAUSE_REDIR_OR_HANDOFF - Call was rejected because of a redirection or handoff
- 49 - QMI_FAILURE_CAUSE_ACCESS_BLOCK_ALL - Access is blocked by the base station for all mobiles; KDDI-specific; CDMA only
- 50 - QMI_FAILURE_CAUSE_OTASP_SPC_ERR - To support OTASP SPC Error indication
- 51 - QMI_FAILURE_CAUSE_IS707B_MAX_ACC - Maximum access probes for an IS-707B

AlphaIdentifier	• see voice_alphaIDInfo definition • NULL pointer - Invalid data.
pUSSDInfo	• USS Data from Network (See structure voice_USSInfo)

- NULL pointer - Invalid data.

Parameters

pCcResultType	• CC result code • NULL pointer - Invalid data.
pCallId	• Caller ID information • NULL pointer - Invalid data.
pCCSuppsType	• See structure 'voice_ccSUPSType' definition • NULL pointer - Invalid data.

8.1233.2 Field Documentation

8.1233.2.1 voice_alphaIDInfo* unpack_voice_SLQSOriginateUSSD_t::pAlphaIDInfo

8.1233.2.2 uint8_t* unpack_voice_SLQSOriginateUSSD_t::pCallId

8.1233.2.3 uint8_t* unpack_voice_SLQSOriginateUSSD_t::pCcResultType

8.1233.2.4 voice_ccSUPSType* unpack_voice_SLQSOriginateUSSD_t::pCCSuppsType

8.1233.2.5 uint16_t* unpack_voice_SLQSOriginateUSSD_t::pfailureCause

8.1233.2.6 struct voice_USSInfo* unpack_voice_SLQSOriginateUSSD_t::pUSSDInfo

8.1234 unpack_voice_SLQSVoiceAnswerCall_t Struct Reference

Data Fields

- uint8_t * pCallId

8.1234.1 Detailed Description

Contains the parameters passed for pack voice Answer Call.

Parameters

pCallId	• Unique call identifier for the call that must be answered. • NULL pointer - Invalid data.

8.1234.2 Field Documentation

8.1234.2.1 uint8_t* unpack_voice_SLQSVoiceAnswerCall_t::pCallId

8.1235 unpack_voice_SLQSVoiceBurstDTMF_t Struct Reference

Data Fields

- uint8_t * pCallID

8.1235.1 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 • NULL pointer - Invalid data.

8.1235.2 Field Documentation

8.1235.2.1 uint8_t* unpack_voice_SLQSVoiceBurstDTMF_t::pCallID

8.1236 unpack_voice_SLQSVoiceDialCall_t Struct Reference

Data Fields

- uint8_t * pCallID
- voice_alphaIDInfo * pAlphaIDInfo
- uint8_t * pCCResultType
- voice_ccSUPSType * pCCSUPSType

8.1236.1 Detailed Description

This structure contains Voice Call Response Parameters

Parameters

pCallID(optional)	• Unique call identifier for the dialed call • NULL pointer - Invalid data.
pAlphaID-Info(optional)	• Pointer to structure of alphaIDInfo – See voice_alphaIDInfo for more information • NULL pointer - Invalid data.
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 – NULL pointer - Invalid data.
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 • NULL pointer - Invalid data.

8.1236.2 Field Documentation

8.1236.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceDialCall_t::pAlphaIDInfo

8.1236.2.2 uint8_t* unpack_voice_SLQSVoiceDialCall_t::pCallID

8.1236.2.3 uint8_t* unpack_voice_SLQSVoiceDialCall_t::pCCResultType

8.1236.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceDialCall_t::pCCSUPSType

8.1237 unpack_voice_SLQSVoiceEndCall_t Struct Reference

Data Fields

- uint8_t * pCallId

8.1237.1 Detailed Description

This structure contains unpack voice end call parameter.

Parameters

pCallId	• Unique call identifier for the call that must be ended • NULL pointer - Invalid data.

8.1237.2 Field Documentation

8.1237.2.1 uint8_t* unpack_voice_SLQSVoiceEndCall_t::pCallId

8.1238 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

8.1238.1 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. • NULL pointer - Invalid data.
pArrRemotePartyNum(optional)	• See voice_arrRemotePartyNum for more information. • NULL pointer - Invalid data.
pArrRemotePartyName(optional)	• See voice_arrRemotePartyName for more information. • NULL pointer - Invalid data.
pArrAlertingType(optional)	• See voice_arrAlertingType for more information. • NULL pointer - Invalid data.
pArrUUSInfo(optional)	• See voice_arrUUSInfo for more information. • NULL pointer - Invalid data.
pArrSvcOption(optional)	• See voice_arrSvcOption for more information. • NULL pointer - Invalid data.

pOTASP-Status(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 • NULL pointer - Invalid data.
pVoice-Privacy(optional)	• Voice Privacy. • Values. – 0x00 - VOICE_PRIVACY_STANDARD - Standard privacy – 0x01 - VOICE_PRIVACY_ENHANCED - Enhanced privacy – 0xFF - Not Available • NULL pointer - Invalid data.
pArrCallEnd-Reason(optional)	• See voice_arrCallEndReason for more information. • NULL pointer - Invalid data.
pArrAlphaI-D(optional)	• See voice_arrAlphaID for more information. • NULL pointer - Invalid data.
pArrConnect-Party-Num(optional)	• See voice_arrConnectPartyNum for more information. • NULL pointer - Invalid data.

pArrDiag-Info(optional)	• See voice_arrDiagInfo for more information. • NULL pointer - Invalid data.
pArrCalledParty-Num(optional)	• See voice_arrCalledPartyNum for more information. • NULL pointer - Invalid data.
pArrRedirParty-Num(optional)	• See voice_arrRedirPartyNum for more information. • NULL pointer - Invalid data.
pArrAlerting-Pattern(optional)	• See voice_arrAlertingPattern for more information. • NULL pointer - Invalid data.

8.1238.2 Field Documentation

8.1238.2.1 voice_arrAlertingPattern∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrAlertingPattern

8.1238.2.2 voice_arrAlertingType∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrAlertingType

8.1238.2.3 voice_arrAlphaID∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrAlphaID

8.1238.2.4 voice_arrCalledPartyNum∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrCalledPartyNum

8.1238.2.5 voice_arrCallEndReason∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrCallEndReason

8.1238.2.6 voice_arrCallInfo∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrCallInfo

8.1238.2.7 voice_arrConnectPartyNum∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrConnectPartyNum

8.1238.2.8 voice_arrDiagInfo∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrDiagInfo

8.1238.2.9 voice_arrRedirPartyNum∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrRedirPartyNum

8.1238.2.10 voice_arrRemotePartyName∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrRemotePartyName

8.1238.2.11 voice_arrRemotePartyNum∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrRemotePartyNum

8.1238.2.12 voice_arrSvcOption∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrSvcOption

8.1238.2.13 voice_arrUUSInfo∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrUUSInfo

8.1238.2.14 uint8_t∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pOTASPStatus

8.1238.2.15 uint8_t∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pVoicePrivacy

8.1239 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

8.1239.1 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 • NULL pointer - Invalid data.
pFailCause	• Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF,if Not Available • NULL pointer - Invalid data.
pAlphaIDInfo	• Pointer to structure of alphaIDInfo (optional) – See voice_alphaIDInfo for more information • NULL pointer - Invalid data.
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 • NULL pointer - Invalid data.
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. • NULL pointer - Invalid data.
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 • NULL pointer - Invalid data.

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_SLQSVoiceGetCallBarring_t::pAlphaIDInfo

8.1239.2.2 uint8_t* unpack_voice_SLQSVoiceGetCallBarring_t::pCallID

8.1239.2.3 uint8_t* unpack_voice_SLQSVoiceGetCallBarring_t::pCCResType

8.1239.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCallBarring_t::pCCSUPSType

8.1239.2.5 uint16_t* unpack_voice_SLQSVoiceGetCallBarring_t::pFailCause

8.1239.2.6 uint8_t* unpack_voice_SLQSVoiceGetCallBarring_t::pSvcClass

8.1240 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

8.1240.1 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 • NULL pointer - Invalid data.
pFailCause	• Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF,if Not Available • NULL pointer - Invalid data.
pAlphaIDInfo	• Pointer to structure of alphaIDInfo (optional) – See voice_alphaIDInfo for more information • NULL pointer - Invalid data.

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 • NULL pointer - Invalid data.
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. • NULL pointer - Invalid data.
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 • NULL pointer - Invalid data.
pGetCallFWExt-Info	• Pointer to structure of getCallFWExtInfo (optional) – See voice_getCallFWExtInfo for more information • NULL pointer - Invalid data.

Note

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

8.1240.2 Field Documentation

8.1240.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pAlphaIDInfo

8.1240.2.2 uint8_t* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pCallID

8.1240.2.3 uint8_t* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pCCResType

8.1240.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pCCSUPSType

8.1240.2.5 uint16_t* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pFailCause

8.1240.2.6 voice_getCallFWExtInfo* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pGetCallFWExtInfo

8.1240.2.7 voice_getCallFWInfo* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pGetCallFWInfo

8.1241 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

8.1241.1 Detailed Description

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

Parameters

pCall-Info(optional)	• See voice_callInfo for more information. • NULL pointer - Invalid data.
pRemoteParty-Num(optional)	• See voice_remotePartyNum for more information. • NULL pointer - Invalid data.
pSrvOpt	• Service option(optional) • Applicable only for 3GPP2 devices. • See Table8 qaGobiApiTableServiceOptions.h for standard service option number assignments. • NULL pointer - Invalid data.
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 • NULL pointer - Invalid data.

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 • NULL pointer - Invalid data.
pRemoteParty-Name(optional)	• Applicable only for 3GPP devices. • See voice_remotePartyName for more information. • NULL pointer - Invalid data.
pUUS-Info(optional)	• Applicable only for 3GPP devices. • See voice_UUSInfo for more information. • NULL pointer - Invalid data.
pAlert-Type(optional)	• Alerting type. • Applicable only for 3GPP devices. – 0x00 - ALERTING_LOCAL - Local – 0x01 - ALERTING_REMOTE - Remote – 0xFF - Not Available • NULL pointer - Invalid data.

pAlphaID-Info(optional)	• Applicable only for 3GPP devices. • See voice_alphaIDInfo for more information. • NULL pointer - Invalid data.
pConnectNum-Info(optional)	• See voice_connectNumInfo for more information. • NULL pointer - Invalid data.
pDiag-Info(optional)	• See voice_diagInfo for more information. • NULL pointer - Invalid data.
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 • NULL pointer - Invalid data.

8.1241.2 Field Documentation

8.1241.2.1 uint32_t* unpack_voice_SLQSVoiceGetCallInfo_t::pAlertingPattern

8.1241.2.2 uint8_t* unpack_voice_SLQSVoiceGetCallInfo_t::pAlertType

8.1241.2.3 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pAlphaIDInfo

8.1241.2.4 voice_callInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pCallInfo

8.1241.2.5 voice_connectNumInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pConnectNumInfo

8.1241.2.6 voice_diagInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pDiagInfo

8.1241.2.7 uint8_t* unpack_voice_SLQSVoiceGetCallInfo_t::pOTASPStatus

8.1241.2.8 voice_remotePartyName* unpack_voice_SLQSVoiceGetCallInfo_t::pRemotePartyName

8.1241.2.9 voice_remotePartyNum* unpack_voice_SLQSVoiceGetCallInfo_t::pRemotePartyNum

8.1241.2.10 uint16_t* unpack_voice_SLQSVoiceGetCallInfo_t::pSrvOpt

8.1241.2.11 voice_UUSInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pUUSInfo

8.1241.2.12 uint8_t* unpack_voice_SLQSVoiceGetCallInfo_t::pVoicePrivacy

8.1242 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

8.1242.1 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 • NULL pointer - Invalid data.
pFailCause	• Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF,if Not Available • NULL pointer - Invalid data.
pAlphaIDInfo	• Pointer to structure of alphaIDInfo (optional) – See voice_alphaIDInfo for more information • NULL pointer - Invalid data.
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 • NULL pointer - Invalid data.
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. • NULL pointer - Invalid data.

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 • NULL pointer - Invalid data.

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_SLQSVoiceGetCallWaiting_t::pAlphaIDInfo

8.1242.2.2 uint8_t* unpack_voice_SLQSVoiceGetCallWaiting_t::pCallID

8.1242.2.3 uint8_t* unpack_voice_SLQSVoiceGetCallWaiting_t::pCCResType

8.1242.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCallWaiting_t::pCCSUPSType

8.1242.2.5 uint16_t* unpack_voice_SLQSVoiceGetCallWaiting_t::pFailCause

8.1242.2.6 uint8_t* unpack_voice_SLQSVoiceGetCallWaiting_t::pSvcClass

8.1243 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

8.1243.1 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 • NULL pointer - Invalid data.
pFailCause	• Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF,if Not Available • NULL pointer - Invalid data.

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 • NULL pointer - Invalid data.
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. • NULL pointer - Invalid data.
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 • NULL pointer - Invalid data.

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_SLQSVoiceGetCLIP_t::pAlphaIDInfo

8.1243.2.2 uint8_t* unpack_voice_SLQSVoiceGetCLIP_t::pCallID

8.1243.2.3 uint8_t* unpack_voice_SLQSVoiceGetCLIP_t::pCCResType

8.1243.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCLIP_t::pCCSUPSType

8.1243.2.5 voice_CLIPResp* unpack_voice_SLQSVoiceGetCLIP_t::pCLIPResp

8.1243.2.6 uint16_t* unpack_voice_SLQSVoiceGetCLIP_t::pFailCause

8.1244 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

8.1244.1 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 • NULL pointer - Invalid data.
pFailCause	• Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF,if Not Available • NULL pointer - Invalid data.
pAlphaIDInfo	• Pointer to structure of alphaIDInfo (optional) – See voice_alphaIDInfo for more information • NULL pointer - Invalid data.
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 • NULL pointer - Invalid data.
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. • NULL pointer - Invalid data.
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 • NULL pointer - Invalid data.

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_SLQSVoiceGetCLIR_t::pAlphaIDInfo

8.1244.2.2 uint8_t* unpack_voice_SLQSVoiceGetCLIR_t::pCallID

8.1244.2.3 uint8_t* unpack_voice_SLQSVoiceGetCLIR_t::pCCResType

8.1244.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCLIR_t::pCCSUPSType

8.1244.2.5 voice_CLIRResp* unpack_voice_SLQSVoiceGetCLIR_t::pCLIRResp

8.1244.2.6 uint16_t* unpack_voice_SLQSVoiceGetCLIR_t::pFailCause

8.1245 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

8.1245.1 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 • NULL pointer - Invalid data.
pFailCause	• Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF,if Not Available • NULL pointer - Invalid data.
pAlphaIDInfo	• Pointer to structure of alphaIDInfo (optional) – See voice_alphaIDInfo for more information • NULL pointer - Invalid data.
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 • NULL pointer - Invalid data.

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. • NULL pointer - Invalid data.
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 • NULL pointer - Invalid data.

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_SLQSVoiceGetCNAP_t::pAlphaIDInfo

8.1245.2.2 uint8_t* unpack_voice_SLQSVoiceGetCNAP_t::pCallID

8.1245.2.3 uint8_t* unpack_voice_SLQSVoiceGetCNAP_t::pCCResType

8.1245.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCNAP_t::pCCSUPSType

8.1245.2.5 voice_CNAPResp* unpack_voice_SLQSVoiceGetCNAP_t::pCNAPResp

8.1245.2.6 uint16_t* unpack_voice_SLQSVoiceGetCNAP_t::pFailCause

8.1246 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

8.1246.1 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 • NULL pointer - Invalid data.

pFailCause	• Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF,if Not Available • NULL pointer - Invalid data.
pAlphaIDInfo	• Pointer to structure of alphaIDInfo (optional) – See voice_alphaIDInfo for more information • NULL pointer - Invalid data.
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 • NULL pointer - Invalid data.
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. • NULL pointer - Invalid data.
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 • NULL pointer - Invalid data.

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_SLQSVoiceGetCOLP_t::pAlphaIDInfo

8.1246.2.2 uint8_t* unpack_voice_SLQSVoiceGetCOLP_t::pCallID

8.1246.2.3 uint8_t* unpack_voice_SLQSVoiceGetCOLP_t::pCCResType

8.1246.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCOLP_t::pCCSUPSType

8.1246.2.5 voice_COLPResp* unpack_voice_SLQSVoiceGetCOLP_t::pCOLPResp

8.1246.2.6 uint16_t* unpack_voice_SLQSVoiceGetCOLP_t::pFailCause

8.1247 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

8.1247.1 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 • NULL pointer - Invalid data.
pFailCause	• Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF,if Not Available • NULL pointer - Invalid data.
pAlphaIDInfo	• Pointer to structure of alphaIDInfo (optional) – See voice_alphaIDInfo for more information • NULL pointer - Invalid data.
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 • NULL pointer - Invalid data.
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. • NULL pointer - Invalid data.
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 • NULL pointer - Invalid data.

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_alphaIDInfo* unpack_voice_SLQSVoiceGetCOLR_t::pAlphaIDInfo

8.1247.2.2 uint8_t* unpack_voice_SLQSVoiceGetCOLR_t::pCallID

8.1247.2.3 uint8_t* unpack_voice_SLQSVoiceGetCOLR_t::pCCResType

8.1247.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCOLR_t::pCCSUPSType

8.1247.2.5 voice_COLRResp* unpack_voice_SLQSVoiceGetCOLR_t::pCOLRResp

8.1247.2.6 uint16_t* unpack_voice_SLQSVoiceGetCOLR_t::pFailCause

8.1248 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

8.1248.1 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 • NULL pointer - Invalid data.
pAirTimer-Cnt(optional)	• Air Timer Count • Value returned is read from NV_AIR_CNT_I. • See voice_airTimer for more information • NULL pointer - Invalid data.

pRoamTimer-Cnt(optional)	• Roam Timer Count • Value returned is read from NV_ROAM_CNT_I. • See voice_roamTimer for more information • NULL pointer - Invalid data.
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 • NULL pointer - Invalid data.
pCurPrefVoiceS-O(optional)	• Current Preferred Voice SO • Value returned is read from NV_PREF_VOICE_SO_I. • See voice_prefVoiceSO for more information • NULL pointer - Invalid data.
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 voice_curAMRConfig for more information • NULL pointer - Invalid data.
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 • NULL pointer - Invalid data.
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 • NULL pointer - Invalid data.

Note

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

8.1248.2 Field Documentation

8.1248.2.1 voice_airTimer* unpack_voice_SLQSVoiceGetConfig_t::pAirTimerCnt

8.1248.2.2 uint8_t* unpack_voice_SLQSVoiceGetConfig_t::pAutoAnswerStat

8.1248.2.3 voice_curAMRConfig* unpack_voice_SLQSVoiceGetConfig_t::pCurAMRConfig

8.1248.2.4 voice_prefVoiceSO* unpack_voice_SLQSVoiceGetConfig_t::pCurPrefVoiceSO

8.1248.2.5 uint8_t* unpack_voice_SLQSVoiceGetConfig_t::pCurrTTYMode

8.1248.2.6 uint8_t* unpack_voice_SLQSVoiceGetConfig_t::pCurVoiceDomainPref

8.1248.2.7 uint8_t* unpack_voice_SLQSVoiceGetConfig_t::pCurVoicePrivacyPref

8.1248.2.8 voice_roamTimer* unpack_voice_SLQSVoiceGetConfig_t::pRoamTimerCnt

8.1249 unpack_voice_SLQSVoiceManageCalls_t Struct Reference

Data Fields

- uint16_t * pFailCause

8.1249.1 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 • NULL pointer - Invalid data.

8.1249.2 Field Documentation

8.1249.2.1 uint16_t* unpack_voice_SLQSVoiceManageCalls_t::pFailCause

8.1250 unpack_voice_SLQSVoiceSendFlash_t Struct Reference

Data Fields

- uint8_t * pCallID

8.1250.1 Detailed Description

This structure contains the flash information associated with a call.

Parameters

pCallID	• Unique call identifier associated with the current call. • NULL pointer - Invalid data.

8.1250.2 Field Documentation

8.1250.2.1 uint8_t* unpack_voice_SLQSVoiceSendFlash_t::pCallID

8.1251 unpack_voice_SLQSVoiceSetCallBarringPassword_t Struct Reference

Data Fields

- uint16_t * pFailCause
- voice_alphaIDInfo * pAlphaIDInfo
- uint8_t * pCCResType
- uint8_t * pCallID
- voice_ccSUPSType * pCCSUPSType

8.1251.1 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 • NULL pointer - Invalid data.
pAlphaIDInfo	• Pointer to structure of alphaIDInfo (optional) – See voice_alphaIDInfo for more information • NULL pointer - Invalid data.
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 • NULL pointer - Invalid data.
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. • NULL pointer - Invalid data.

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 • NULL pointer - Invalid data.

Note

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

8.1251.2 Field Documentation

8.1251.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceSetCallBarringPassword_t::pAlphaIDInfo

8.1251.2.2 uint8_t* unpack_voice_SLQSVoiceSetCallBarringPassword_t::pCallID

8.1251.2.3 uint8_t* unpack_voice_SLQSVoiceSetCallBarringPassword_t::pCCResType

8.1251.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceSetCallBarringPassword_t::pCCSUPSType

8.1251.2.5 uint16_t* unpack_voice_SLQSVoiceSetCallBarringPassword_t::pFailCause

8.1252 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

8.1252.1 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. – NULL pointer - Invalid data.

pAirTimerStatus	• Air Timer Status. (optional) • Values: – 0x00 - Information was written successfully – 0x01 - Information write failed – 0xFF - Not Available. – NULL pointer - Invalid data.
pRoamTimer-Status	• Roam Timer Status. (optional) • Values: – 0x00 - Information was written successfully – 0x01 - Information write failed – 0xFF - Not Available. – NULL pointer - Invalid data.
pTTYConfig-Status	• TTY Config Status. (optional) • Values: – 0x00 - Information was written successfully – 0x01 - Information write failed – 0xFF - Not Available. – NULL pointer - Invalid data.
pPrefVoiceSO-Status	• Preferred Voice SO Status. (optional) • Values: – 0x00 - Information was written successfully – 0x01 - Information write failed – 0xFF - Not Available. – NULL pointer - Invalid data.
pVoiceDomain-PrefStatus	• Voice-Domain Preference Status. (optional) • Values: – 0x00 - Information was written successfully – 0x01 - Information write failed – 0xFF - Not Available. – NULL pointer - Invalid data.

Note

Parameters which are mentioned as NULL will be ignored.

8.1252.2 Field Documentation

8.1252.2.1 uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pAirTimerStatus

8.1252.2.2 uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pAutoAnsStatus

8.1252.2.3 uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pPrefVoiceSOStatus

8.1252.2.4 uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pRoamTimerStatus

8.1252.2.5 uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pTTYConfigStatus

8.1252.2.6 uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pVoiceDomainPrefStatus

8.1253 unpack_voice_SLQSVoiceSetSUPSService_t Struct Reference

Data Fields

- uint16_t * pFailCause
- voice_alphaIDInfo * pAlphaIDInfo
- uint8_t * pCCResultType
- uint8_t * pCallID
- voice_ccSUPSType * pCCSUPSType

8.1253.1 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 – NULL pointer - Invalid data.
pAlphaIDInfo	• Pointer to structure of alphaIDInfo. The parameter used to pass the alpha (if any) given by the SIM/R-UIM after call control (optional, supply NULL if not required) – See voice_alphaIDInfo for more information – NULL pointer - Invalid data.
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 – NULL pointer - Invalid data.
pCallID	• Unique call identifier for the dialed call (optional, supply NULL if not required) – 0x00 - if the device does not provide this information – NULL pointer - Invalid data.

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 – NULL pointer - Invalid data.

8.1253.2 Field Documentation

8.1253.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceSetSUPSService_t::pAlphaIDInfo

8.1253.2.2 uint8_t* unpack_voice_SLQSVoiceSetSUPSService_t::pCallID

8.1253.2.3 uint8_t* unpack_voice_SLQSVoiceSetSUPSService_t::pCCResultType

8.1253.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceSetSUPSService_t::pCCSUPSType

8.1253.2.5 uint16_t* unpack_voice_SLQSVoiceSetSUPSService_t::pFailCause

8.1254 unpack_voice_SLQSVoiceStartContDTMF_t Struct Reference

Data Fields

- uint8_t * pCallID

8.1254.1 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 • NULL pointer - Invalid data.

8.1254.2 Field Documentation

8.1254.2.1 uint8_t* unpack_voice_SLQSVoiceStartContDTMF_t::pCallID

8.1255 unpack_voice_SLQSVoiceStopContDTMF_t Struct Reference

Data Fields

- uint8_t callID

8.1255.1 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

8.1255.2 Field Documentation

8.1255.2.1 uint8_t unpack_voice_SLQSVoiceStopContDTMF_t::callID

8.1256 unpack_wds_DHCPv4ClientLease_ind_t Struct Reference

Data Fields

- wds_DHCPProfileIdTlv ProfileIdTlv
- wds_DHCPLeaseStateTlv DHCPv4LeaseStateTlv
- wds_IPv4AdTlv IPv4AddrTlv
- wds_DHCPLeaseOptTlv DHCPv4LeaseOptTlv

8.1256.1 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

8.1256.2 Field Documentation

8.1256.2.1 wds_DHCPLeaseOptTlv unpack_wds_DHCPv4ClientLease_ind_t::DHCPv4LeaseOptTlv

8.1256.2.2 wds_DHCPLeaseStateTlv unpack_wds_DHCPv4ClientLease_ind_t::DHCPv4LeaseStateTlv

8.1256.2.3 wds_IPv4AdTlv unpack_wds_DHCPv4ClientLease_ind_t::IPv4AddrTlv

8.1256.2.4 wds_DHCPProfileIdTlv unpack_wds_DHCPv4ClientLease_ind_t::ProfileIdTlv

8.1257 unpack_wds_GetAutoconnect_t Struct Reference

Data Fields

- uint32_t * psetting

8.1257.1 Detailed Description

auto connect data session setting parameter.

Parameters

pSetting	• NDIS auto connect setting – 0 - Disabled – 1 - Enabled

8.1257.2 Field Documentation

8.1257.2.1 uint32_t* unpack_wds_GetAutoconnect_t::psetting

8.1258 unpack_wds_GetByteTotals_t Struct Reference

Data Fields

- uint64_t * pTXTotalBytes
- uint64_t * pRXTotalBytes

8.1258.1 Detailed Description

This structure contains unpack get byte totals information.

Parameters

pTXTotalBytes	• Bytes transmitted without error • NULL pointer - Invalid data.
pRXTotalBytes	• Bytes received without error • NULL pointer - Invalid data.

8.1258.2 Field Documentation

8.1258.2.1 uint64_t* unpack_wds_GetByteTotals_t::pRXTotalBytes

8.1258.2.2 uint64_t* unpack_wds_GetByteTotals_t::pTXTotalBytes

8.1259 unpack_wds_GetConnectionRate_t Struct Reference

Data Fields

- uint32_t currentChannelTXRate
- uint32_t currentChannelRXRate
- uint32_t maxChannelTXRate
- uint32_t maxChannelRXRate

8.1259.1 Detailed Description

This structure contains unpack get connection rate information.

Parameters

currentChannelTXRate	• Current channel Tx rate (in bps)
currentChannelRXRate	• Current channel Rx rate (in bps)
maxChannelTXRate	• Maximum Tx rate (bps) that may be assigned to device by serving system.
maxChannelRXRate	• Maximum Rx rate (bps) that may be assigned to device by serving system.

8.1259.2 Field Documentation

8.1259.2.1 uint32_t unpack_wds_GetConnectionRate_t::currentChannelRXRate

8.1259.2.2 uint32_t unpack_wds_GetConnectionRate_t::currentChannelTXRate

8.1259.2.3 uint32_t unpack_wds_GetConnectionRate_t::maxChannelRXRate

8.1259.2.4 uint32_t unpack_wds_GetConnectionRate_t::maxChannelTXRate

8.1260 unpack_wds_GetDataBearerTechnology_t Struct Reference

Data Fields

- uint32_t * pDataBearer

8.1260.1 Detailed Description

This structure contains unpack get data bearer technology information.

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

8.1260.2 Field Documentation

8.1260.2.1 uint32_t* unpack_wds_GetDataBearerTechnology_t::pDataBearer

8.1261 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]

8.1261.1 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 – 0xff - Invalid data.
ipaddr	• Preferred IPv4 address to be assigned to device • 0xffffffff - Invalid data.
pridns	• Primary DNS IPv4 address preference • 0xffffffff - Invalid data.
secdns	• Secondary DNS IPv4 address preference • 0xffffffff - Invalid data.
ipaddrv6	• Preferred IPv6 address to be assigned to device • 0xffff - Invalid data.
pridnsv6	• Primary DNS Pv6 address preference • 0xffff - Invalid data.
secdnsv6	• Secondary DNS IPv6 address preference • 0xffff - Invalid data.
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. – 0xffffffff - Invalid data.
namesize	• Maximum number of characters (including NULL terminator) that profile name array can contain. • 0xff - Invalid name data.

name	• Profile name
apnsize	• Maximum number of characters (including NULL terminator) that APN name array can contain • 0xff - Invalid apnname data.
apnname	• Access point name. NULL-terminated string parameter that is a logical name used to select GGSN and external packet data network. • 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. • 0xff - Invalid username data.
username	• Username used during network authentication

8.1261.2 Field Documentation

8.1261.2.1 int8_t unpack_wds_GetDefaultProfile_t::apnname[255]

8.1261.2.2 uint8_t unpack_wds_GetDefaultProfile_t::apnsize

8.1261.2.3 uint32_t unpack_wds_GetDefaultProfile_t::auth

8.1261.2.4 uint32_t unpack_wds_GetDefaultProfile_t::ipaddr

8.1261.2.5 uint16_t unpack_wds_GetDefaultProfile_t::ipaddrv6

8.1261.2.6 int8_t unpack_wds_GetDefaultProfile_t::name[255]

8.1261.2.7 uint8_t unpack_wds_GetDefaultProfile_t::namesize

8.1261.2.8 uint32_t unpack_wds_GetDefaultProfile_t::pdptype

8.1261.2.9 uint32_t unpack_wds_GetDefaultProfile_t::pridns

8.1261.2.10 uint16_t unpack_wds_GetDefaultProfile_t::pridnsv6

8.1261.2.11 uint32_t unpack_wds_GetDefaultProfile_t::secdns

8.1261.2.12 uint16_t unpack_wds_GetDefaultProfile_t::secdnsv6

8.1261.2.13 int8_t unpack_wds_GetDefaultProfile_t::username[255]

8.1261.2.14 uint8_t unpack_wds_GetDefaultProfile_t::usersize

8.1262 unpack_wds_GetDefaultProfileNum_t Struct Reference

Data Fields

- uint8_t index

8.1262.1 Detailed Description

This structure contains unpack Get Default Profile Number information.

Parameters

index	profile index

8.1262.2 Field Documentation

8.1262.2.1 uint8_t unpack_wds_GetDefaultProfileNum_t::index

8.1263 unpack_wds_GetDormancyState_t Struct Reference

Data Fields

- uint32_t dormancyState

8.1263.1 Detailed Description

This structure contains unpack get dormancy state information.

Parameters

dormancyState	• Dormancy state of current packet data session – 1 - Traffic channel dormant – 2 - Traffic channel active

8.1263.2 Field Documentation

8.1263.2.1 uint32_t unpack_wds_GetDormancyState_t::dormancyState

8.1264 unpack_wds_GetLastMobileIPError_t Struct Reference

Data Fields

- uint32_t error

8.1264.1 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

8.1264.2 Field Documentation

8.1264.2.1 uint32_t unpack_wds_GetLastMobileIPError_t::error

8.1265 unpack_wds_GetMobileIP_t Struct Reference

Data Fields

- uint32_t mipMode

8.1265.1 Detailed Description

This structure contains unpack get mobile IP information.

Parameters

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

8.1265.2 Field Documentation

8.1265.2.1 uint32_t unpack_wds_GetMobileIP_t::mipMode

8.1266 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

8.1266.1 Detailed Description

This structure contains unpack get mobile IP profile information.

Parameters

enabled	• Profile enabled: – 0 - Disabled – 1 - Enabled – 0xFF - Invalid data.
address	• Home IPv4 address: – 0xFFFFFFFF - Invalid data.
primaryHA	• Primary home agent IPv4 address – 0xFFFFFFFF - Invalid data.
secondaryHA	• Secondary home agent IPv4 address – 0xFFFFFFFF - Invalid data.
revTunneling	• Reverse tunneling enabled – 0 - Disabled – 1 - Enabled – 0xFF - Invalid data.
naiSize	• The maximum number of characters (including NULL terminator) that the NAI array can contain. • 0xFF - Invalid data.
NAI	• Network access identifier string • 0xFF - Invalid data.
HASPI	• Home agent security parameter index • 0xFF - Invalid data.
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

8.1266.2 Field Documentation

8.1266.2.1 uint32_t unpack_wds_GetMobileIPProfile_t::AAASPI

8.1266.2.2 uint32_t unpack_wds_GetMobileIPProfile_t::AAAState

8.1266.2.3 uint32_t unpack_wds_GetMobileIPProfile_t::address

8.1266.2.4 uint8_t unpack_wds_GetMobileIPProfile_t::enabled

8.1266.2.5 uint32_t unpack_wds_GetMobileIPProfile_t::HASPI

8.1266.2.6 uint32_t unpack_wds_GetMobileIPProfile_t::HAState

8.1266.2.7 int8_t unpack_wds_GetMobileIPProfile_t::NAI[255]

8.1266.2.8 uint8_t unpack_wds_GetMobileIPProfile_t::naiSize

8.1266.2.9 uint32_t unpack_wds_GetMobileIPProfile_t::primaryHA

8.1266.2.10 uint8_t unpack_wds_GetMobileIPProfile_t::revTunneling

8.1266.2.11 uint32_t unpack_wds_GetMobileIPProfile_t::secondaryHA

8.1267 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

8.1267.1 Detailed Description

This structure contains unpack get packet statistics information.

Parameters

pTXPacket-Successes	• No of transmitted Packets without error. • NULL pointer - Invalid data.
pRXPacket-Successes	• No of received Packets without error. • NULL pointer - Invalid data.
pTXPacketErrors	• Number of outgoing packets with framing errors. • NULL pointer - Invalid data.
pRXPacket-Errors	• Number of incoming packets with framing errors. • NULL pointer - Invalid data.
pTXPacket-Overflows	• Number of packets dropped because Tx buffer overflowed (out of memory). • NULL pointer - Invalid data.
pRXPacket-Overflows	• Number of packets dropped because Rx buffer overflowed (out of memory). • NULL pointer - Invalid data.
pTXOkBytes-Count	• Number of bytes transmitted without error. • NULL pointer - Invalid data.
pRXOkBytes-Count	• Number of bytes received without error. • NULL pointer - Invalid data.
pTXOKBytes-LastCall	• Number of bytes transmitted without error during the last data call (0 if no call was made earlier). Returned only if not in a call, and when the previous call was made using RmNet (for any devices that support • NULL pointer - Invalid data.
pRXOKBytes-LastCall	• Number of bytes received without error during the last data call (0 if no call was made earlier). Returned only if not in a call, and when the previous call was made using RmNet (for any devices that support • NULL pointer - Invalid data.
pTXDropped-Count	• Number of outgoing packets dropped. • NULL pointer - Invalid data.
pRXDropped-Count	• Number of incoming packets dropped. • NULL pointer - Invalid data.

8.1267.2 Field Documentation

8.1267.2.1 uint32_t* unpack_wds_GetPacketStatistics_t::pRXDroppedCount

8.1267.2.2 uint64_t* unpack_wds_GetPacketStatistics_t::pRXOkBytesCount

8.1267.2.3 uint64_t* unpack_wds_GetPacketStatistics_t::pRXOKBytesLastCall

8.1267.2.4 uint32_t* unpack_wds_GetPacketStatistics_t::pRXPacketErrors

8.1267.2.5 uint32_t* unpack_wds_GetPacketStatistics_t::pRXPacketOverflows

8.1267.2.6 uint32_t* unpack_wds_GetPacketStatistics_t::pRXPacketSuccesses

8.1267.2.7 uint32_t* unpack_wds_GetPacketStatistics_t::pTXDroppedCount

8.1267.2.8 uint64_t* unpack_wds_GetPacketStatistics_t::pTXOkBytesCount

8.1267.2.9 uint64_t* unpack_wds_GetPacketStatistics_t::pTXOKBytesLastCall

8.1267.2.10 uint32_t* unpack_wds_GetPacketStatistics_t::pTXPacketErrors

8.1267.2.11 uint32_t* unpack_wds_GetPacketStatistics_t::pTXPacketOverflows

8.1267.2.12 uint32_t* unpack_wds_GetPacketStatistics_t::pTXPacketSuccesses

8.1268 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

8.1268.1 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 • 0xffffffff - Invalid data.

rXPacket-Successes	• No. of packets received without error • 0xffffffff - Invalid data.
tXPacketErrors	• No. of outgoing packets with framing errors • 0xffffffff - Invalid data.
rXPacketErrors	• No. of incoming packets with framing errors • 0xffffffff - Invalid data.
txPacket-Overflows	• Number of packets dropped because Tx buffer overflowed • 0xffffffff - Invalid data.
rxPacket-Overflows	• Number of packets dropped because Rx buffer overflowed • 0xffffffff - Invalid data.
txOkBytesCount	• No of bytes transmitted without error. • 0xffffffffffffffff - Invalid data.
rxOkBytesCount	• No of bytes received without error. • 0xffffffffffffffff - Invalid data.
txOKBytesLast-Call	• No of bytes transmitted without error during the last data call (0 if no call was made earlier). Returned only if not in a call, and when the previous call was made using RmNet (for any devices that support • 0xffffffffffffffff - Invalid data.
rxOKBytesLast-Call	• Number of bytes received without error during the last data call (0 if no call was made earlier). Returned only if not in a call, and when the previous call was made using RmNet (for any devices that support • 0xffffffffffffffff - Invalid data.
txDroppedCount	• Number of outgoing packets dropped. • 0xffffffff - Invalid data.
rxDroppedCount	• Number of incoming packets dropped. • 0xffffffff - Invalid data.

8.1268.2 Field Documentation

8.1268.2.1 uint32_t unpack_wds_GetPacketStatus_t::rXDroppedCount

8.1268.2.2 uint64_t unpack_wds_GetPacketStatus_t::rXOkBytesCount

8.1268.2.3 uint64_t unpack_wds_GetPacketStatus_t::rXOKBytesLastCall

8.1268.2.4 uint32_t unpack_wds_GetPacketStatus_t::rXPacketErrors

8.1268.2.5 uint32_t unpack_wds_GetPacketStatus_t::rXPacketOverflows

8.1268.2.6 uint32_t unpack_wds_GetPacketStatus_t::rXPacketSuccesses

8.1268.2.7 uint32_t unpack_wds_GetPacketStatus_t::tXDroppedCount

8.1268.2.8 uint64_t unpack_wds_GetPacketStatus_t::tXOkBytesCount

8.1268.2.9 uint64_t unpack_wds_GetPacketStatus_t::tXOKBytesLastCall

8.1268.2.10 uint32_t unpack_wds_GetPacketStatus_t::tXPacketErrors

8.1268.2.11 uint32_t unpack_wds_GetPacketStatus_t::tXPacketOverflows

8.1268.2.12 uint32_t unpack_wds_GetPacketStatus_t::tXPacketSuccesses

8.1269 unpack_wds_GetSessionDuration_t Struct Reference

Data Fields

- uint64_t callDuration

8.1269.1 Detailed Description

This structure contains unpack get session duration information.

Parameters

callDuration	• Duration of the current packet session in milliseconds

8.1269.2 Field Documentation

8.1269.2.1 uint64_t unpack_wds_GetSessionDuration_t::callDuration

8.1270 unpack_wds_GetSessionState_t Struct Reference

Data Fields

- uint32_t connectionStatus

8.1270.1 Detailed Description

This structure contains unpack get session state information.

Parameters

connection-Status	- state of the current packet data session. • 1 - Disconnected. • 2 - Connected. • 3 - Suspended. • 4 - Authenticating.

8.1270.2 Field Documentation

8.1270.2.1 uint32_t unpack_wds_GetSessionState_t::connectionStatus

8.1271 unpack_wds_RMSetTransferStatistics_t Struct Reference

8.1271.1 Detailed Description

This structure contains unpack fetch current data system transfer statistics information.

Parameters

NULL	• this is a dummy structure

8.1272 unpack_wds_SetMobileIPProfile_t Struct Reference

8.1272.1 Detailed Description

This structure contains set mobile IP profile information.

Parameters

NULL	• this is a dummy structure

8.1273 unpack_wds_SLQSCreateProfile_t Struct Reference

Data Fields

- PackCreateProfileOut * pCreateProfileOut
- uint8_t * pProfileID
- uint16_t Tlvresult

8.1273.1 Detailed Description

This structure contains unpack create profile information.

Parameters

pCreateProfile-Out	• SLQS Create profile Information • See PackCreateProfileOut
pProfileID	• SLQS profile identifier information
Tlvresult	• TLV present flag, TURE if presented, otherwise FALSE

8.1273.2 Field Documentation

8.1273.2.1 PackCreateProfileOut* unpack_wds_SLQSCreateProfile_t::pCreateProfileOut

8.1273.2.2 uint8_t* unpack_wds_SLQSCreateProfile_t::pProfileID

8.1273.2.3 uint16_t unpack_wds_SLQSCreateProfile_t::Tlvresult

8.1274 unpack_wds_SLQSDeleteProfile_t Struct Reference

Data Fields

- uint16_t extendedErrorCode

8.1274.1 Detailed Description

This structure contains unpack delete profile information.

Parameters

extendedError-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. • 0xffff - Invalid data.

8.1274.2 Field Documentation

8.1274.2.1 uint16_t unpack_wds_SLQSDeleteProfile_t::extendedErrorCode

8.1275 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

8.1275.1 Detailed Description

This structure contains unpack get 3GPP configure item information.

Parameters

	LTEAttachProfile	• 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 • 0xffff - Invalid data.
	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 • 0xffff - Invalid data. • Note: the 5th entry is currently ignored, please set it to zero
out	*defaultPDN-Enabled*	• 0 - disabled • 1 - enabled • 0xff - Invalid data.
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) • 0xff - Invalid data.
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 • 0xffff - Invalid LTEAttachProfileList data.

8.1275.2 Field Documentation

8.1275.2.1 uint8_t unpack_wds_SLQSGet3GPPConfigItem_t::_3gppRelease

8.1275.2.2 uint8_t unpack_wds_SLQSGet3GPPConfigItem_t::defaultPDNEnabled

8.1275.2.3 uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::LTEAttachProfile

8.1275.2.4 uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::LTEAttachProfileList[24]

8.1275.2.5 uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::LTEAttachProfileListLen

8.1275.2.6 uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::profileList[5]

8.1276 unpack_wds_SLQSGetCurrDataSystemStat_t Struct Reference

Data Fields

- uint8_t prefNetwork
- uint8_t networkInfoLen
- currNetworkInfo currNetworkInfo [255]

8.1276.1 Detailed Description

Data System Status

Parameters

prefNetwork	• Preferred Network • Values: – 0 - 3GPP – 1 - 3GPP2 – 0xff - Invalid data system status.
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

8.1276.2 Field Documentation

8.1276.2.1 currNetworkInfo unpack_wds_SLQSGetCurrDataSystemStat_t::currNetworkInfo[255]

8.1276.2.2 uint8_t unpack_wds_SLQSGetCurrDataSystemStat_t::networkInfoLen

8.1276.2.3 uint8_t unpack_wds_SLQSGetCurrDataSystemStat_t::prefNetwork

8.1277 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

8.1277.1 Detailed Description

This structure contains unpack get current channel rate information.

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.

8.1277.2 Field Documentation

8.1277.2.1 uint32_t unpack_wds_SLQSGetCurrentChannelRate_t::current_channel_rx_rate

8.1277.2.2 uint32_t unpack_wds_SLQSGetCurrentChannelRate_t::current_channel_tx_rate

8.1277.2.3 uint32_t unpack_wds_SLQSGetCurrentChannelRate_t::max_channel_rx_rate

8.1277.2.4 uint32_t unpack_wds_SLQSGetCurrentChannelRate_t::max_channel_tx_rate

8.1278 unpack_wds_SLQSGetDataBearerTechnology_t Struct Reference

Data Fields

- uint8_t dataBearerMask
- qmiWDSDataBearerTechnology curDataBearerTechnology
- qmiWDSDataBearerTechnology lastCallDataBearerTechnology

8.1278.1 Detailed Description

Structure to hold the data bearer technology values

Parameters

dataBearerMask[OUT]	• This bit mask indicates if data bearer information for the current and/or last call has been received from the device. If a bit is set, then the information is available in the corresponding structure i.e. the one provided by the caller. Refer to liteQmiDataBearerMasks for bit-mask positions. • 0xff - Invalid data.
curDataBearerTechnology[OUT]	• current data bearer technology value. • See qmiWDSDataBearerTechnology – NULL if the parameter is not required
lastCallDataBearerTechnology[OUT]	• last call data bearer technology value. • See qmiWDSDataBearerTechnology – NULL if the parameter is not required

8.1278.2 Field Documentation

8.1278.2.1 qmiWDSDataBearerTechnology unpack_wds_SLQSGetDataBearerTechnology_t::curDataBearerTechnology

8.1278.2.2 uint8_t unpack_wds_SLQSGetDataBearerTechnology_t::dataBearerMask

8.1278.2.3 qmiWDSDataBearerTechnology unpack_wds_SLQSGetDataBearerTechnology_t::lastCallDataBearerTechnology

8.1279 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

8.1279.1 Detailed Description

This structure contains the DUN Call Info response parameters

Parameters

connection-Status	• 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 • 0xffff - Invalid data.
txOKBytesCount	• Number of bytes transmitted without error • Returned only if a data call is up • 0xffffffffffffffff - Invalid data.
rxOKBytesCount	• Number of bytes received without error • Returned only if a data call is up • 0xffffffffffffffff - Invalid data.
dormancyStatus	• Current traffic channel status • Returned if a data call is up – 0x01 - Traffic channel dormant – 0x02 - Traffic channel active – 0xff - Invalid data.

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. • 0xffffffffffffffff - Invalid data.
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. • 0xffffffffffffffff - Invalid data.
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) • 0xffffffffffffffff - Invalid data.

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

8.1279.2 Field Documentation

8.1279.2.1 uint16_t unpack_wds_SLQSGetDUNCallInfo_t::callEndReason

8.1279.2.2 dunchannelRate unpack_wds_SLQSGetDUNCallInfo_t::channelRate

8.1279.2.3 connectionStatus unpack_wds_SLQSGetDUNCallInfo_t::connectionStatus

8.1279.2.4 uint8_t unpack_wds_SLQSGetDUNCallInfo_t::dataBearerTech

8.1279.2.5 uint8_t unpack_wds_SLQSGetDUNCallInfo_t::dormancyStatus

8.1279.2.6 uint8_t unpack_wds_SLQSGetDUNCallInfo_t::lastCallDataBearerTech

8.1279.2.7 uint64_t unpack_wds_SLQSGetDUNCallInfo_t::lastCallRXOKBytesCnt

8.1279.2.8 uint64_t unpack_wds_SLQSGetDUNCallInfo_t::lastCallTXOKBytesCnt

8.1279.2.9 uint64_t unpack_wds_SLQSGetDUNCallInfo_t::mdmCallDurationActive

8.1279.2.10 uint64_t unpack_wds_SLQSGetDUNCallInfo_t::rxOKBytesCount

8.1279.2.11 uint64_t unpack_wds_SLQSGetDUNCallInfo_t::txOKBytesCount

8.1280 unpack_wds_SLQSGetProfileSettings_t Struct Reference

Data Fields

- UnPackGetProfileSettingOut * pProfileSettings
- uint8_t ProfileType
- uint16_t Tlvresult

8.1280.1 Detailed Description

This structure contains the profile setting information of parameter pOutput for API unpack_wds_SLQSGetProfile-Settings

Parameters

pProfileSettings	- Profile Settings
ProfileType	- Profile Type • 0 - Profile 3GPP • 1 - Profile 3GPP2
Tlvresult	- unpack Tlv Result.

8.1280.2 Field Documentation

8.1280.2.1 UnPackGetProfileSettingOut* unpack_wds_SLQSGetProfileSettings_t::pProfileSettings

8.1280.2.2 uint8_t unpack_wds_SLQSGetProfileSettings_t::ProfileType

8.1280.2.3 uint16_t unpack_wds_SLQSGetProfileSettings_t::Tlvresult

8.1281 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

8.1281.1 Detailed Description

This structure contains unpack get runtime settings information.

Parameters

IPv4	ipv4 address • 0xff - Invalid data.
ProfileName	profile name • Invalid data when index 0 of ProfileName is equal 0xff.
PDPType	PDP type • 0xffffffff - Invalid data.
APNName	APN name • Invalid data when index 0 of APNName is equal 0xff.
PrimaryDNSV4	primary dns IPV4 • 0xffffffff - Invalid data.
SecondaryDNS-V4	secondary dns IPV4 • 0xffffffff - Invalid data.
UMTSGranted-QoS	UMTS Granted Qos
GPRSGranted-QoS	GPRS Granted QoS
Username	username for authentication process • Invalid data when index 0 of APNName is equal 0xff.
Authentication	authentication for authentication process • 0xffffffff - Invalid data.
ProfielID	profile ID
GWAddressV4	Gateway IPv4 • 0xffffffff - Invalid data.
SubnetMaskV4	Subnet mask IPV4 • 0xffffffff - Invalid data.
PCSCFAddrPC-O	PCSCF address PCO • 0xff - Invalid data.
ServerAddrList	PCSCF server address list IPV4
PCSCFFQDN-AddrList	PCSCF FQDN address list IPV4
PrimaryDNSV6	Primary DNS IPV6 • invaild when all array data are equal 0xffff

SecondryDNS-V6	Secondry DNS IPV6 • invaild when all array data are equal 0xffff
Mtu	Maximum Transfer Unit • 0xffffffff - Invalid data.
DomainList	domain list
IPFamily-Preference	ip family preference, it could be IPV4 or IPV6 • 0xff - Invalid data.
IMCNflag	IM control flag, value: TRUE or FALSE • 0xff - Invalid data.
Technology	technology on which current packet data session is in progress • Values: – 32767 - CDMA – 32764 - UMTS – 30592 - EPC – 30584 - modem link local – 0xffff - Invalid data.
IPV6AddrInfo	address information IPV6
IPV6GWAddr-Info	gateway address information IPV6

8.1281.2 Field Documentation

8.1281.2.1 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::APNName[128]

8.1281.2.2 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::Authentication

8.1281.2.3 struct wds_DomainNameList unpack_wds_SLQSGetRuntimeSettings_t::DomainList

8.1281.2.4 struct wds_GPRSQoS unpack_wds_SLQSGetRuntimeSettings_t::GPRSGrantedQoS

8.1281.2.5 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::GWAddressV4

8.1281.2.6 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::IMCNflag

8.1281.2.7 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::IPFamilyPreference

8.1281.2.8 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::IPv4

8.1281.2.9 struct wds_IPV6AddressInfo unpack_wds_SLQSGetRuntimeSettings_t::IPV6AddrInfo

8.1281.2.10 struct wds_IPV6GWAddressInfo unpack_wds_SLQSGetRuntimeSettings_t::IPV6GWAddrInfo

8.1281.2.11 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::Mtu

8.1281.2.12 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::PCSCFAddrPCO

8.1281.2.13 struct wds_PCSCFFQDNAddressList unpack_wds_SLQSGetRuntimeSettings_t::PCSCFFQDNAddrList

8.1281.2.14 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::PDPType

8.1281.2.15 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::PrimaryDNSV4

8.1281.2.16 uint16_t unpack_wds_SLQSGetRuntimeSettings_t::PrimaryDNSV6[8]

8.1281.2.17 struct **wds_ProfileIdentifier** unpack_wds_SLQSGetRuntimeSettings_t::ProfileID

8.1281.2.18 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::ProfileName[128]

8.1281.2.19 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::SecondaryDNSV4

8.1281.2.20 uint16_t unpack_wds_SLQSGetRuntimeSettings_t::SecondaryDNSV6[8]

8.1281.2.21 struct **wds_PCSCFIPv4ServerAddressList** unpack_wds_SLQSGetRuntimeSettings_t::ServerAddrList

8.1281.2.22 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::SubnetMaskV4

8.1281.2.23 uint16_t unpack_wds_SLQSGetRuntimeSettings_t::Technology

8.1281.2.24 **LibPackUMTSQoS** unpack_wds_SLQSGetRuntimeSettings_t::UMTSGrantedQoS

8.1281.2.25 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::Username[128]

8.1282 unpack_wds_SLQSModifyProfile_t Struct Reference

Data Fields

- uint16_t * pExtErrorCode

8.1282.1 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. • NULL pointer - Invalid data.

8.1282.2 Field Documentation

8.1282.2.1 uint16_t* unpack_wds_SLQSModifyProfile_t::pExtErrorCode

8.1283 unpack_wds_SLQSSetIPFamilyPreference_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1283.1 Detailed Description

This structure contains unpack Set IP family preference information

Parameters

Tlvresult	unpack result

8.1283.2 Field Documentation

8.1283.2.1 uint16_t unpack_wds_SLQSSetIPFamilyPreference_t::Tlvresult

8.1284 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

8.1284.1 Detailed Description

This structure contains unpack set packet service status callback information.

Parameters

conn_status	connection status. • Current link status. Values: – 1 - DISCONNECTED – 2 - CONNECTED – 3 - SUSPENDED – 4 - AUTHENTICATING – 0xff Invalid data.
reconfigReqd	Indicates whether the network interface on the host needs to be reconfigured. • Values: – 0 - No need to reconfigure. – 1 - Reconfiguration required. – 0xff - Invalid data.
sessionEnd-Reason	Call End Reason • See qaGobiApiTableCallEndReasons.h for Call End Reason • 0xffff - invalid data.

verboseSessn-EndReasonType	Verbose call end reason type • Call end reason type. Values: – 0 - Unspecified – 1 - Mobile IP – 2 - Internal – 3 - Call Manager defined – 6 - 3GPP Specification defined – 7 - PPP – 8 - EHRPD – 9 - IPv6 – 0xffff - invalid data.
verboseSessn-EndReason	Reason the call ended (verbose) • See qaGobiApiTableCallEndReasons.h for Call End Reason • 0xffff - invalid data.
ipFamily	IP family of the packet data connection. • Values – 4 - IPv4 – 6 - IPv6 – 0xff - Invalid data.
techName	Technology name of the packet data connection. • Values – 32767 - CDMA – 32764 - UMTS – 30592 - EPC – 30590 - EMBMS – 30584 - Modem Link Local EPC is a logical interface to support LTE/eHRPD handoff. It is returned if the device supports IP session continuity. Modem Link Local is an interface for transferring data between entities on the AP and modem. – 0xffff - Invalid data.
bearerID	• bearer ID (3GPP) or RLP ID (3GPP2) of the packet data connection. • Valid Values - 0 to 16 • 0xFF - Invalid value.

8.1284.2 Field Documentation

8.1284.2.1 uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::bearerID

8.1284.2.2 uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::conn_status

8.1284.2.3 uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::ipFamily

8.1284.2.4 uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::reconfigReqd

8.1284.2.5 uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::sessionEndReason

8.1284.2.6 uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::techName

8.1284.2.7 uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::verboseSessnEndReason

8.1284.2.8 uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::verboseSessnEndReasonType

8.1285 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

8.1285.1 Detailed Description

This structure contains unpack set WDS event callback information.

Parameters

xferStatAvail	transfer statistic available
tx_bytes	transmit bytes
rx_bytes	received bytes
tx_pkts	transmit packets
rx_pkts	received packets
mipstatAvail	Mobile IP status available
mipStatus	Mobile IP status
dBTechAvail	Data Bearer technology available
dBTechnology	Data Bearer technology
dormancyStat-Avail	Dormancy status available
dormancyStatus	Dormancy status
currDBTechAvail	Current Data Bearer technology available
ratMask	RAT mask to indicate type of technology
soMask	SO mask to indicate the service type
dataSysStatAvail	Data System Status available

prefNetwork	preferred network
currNWInfo	Current Network Info
dBtechExtAvail	Data bearer technology extended available
dBtechnologyExt	Data bearer technology Extended.
dBTechExtRatValue	Data bearer technology Extended RAT Value
dBTechExtSoMask	Data bearer technology Extended SO Mask

8.1285.2 Field Documentation

8.1285.2.1 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::currDBTechAvail

8.1285.2.2 wds_currNetworkInfo unpack_wds_SLQSSetWdsEventCallback_ind_t::currNWInfo[255]

8.1285.2.3 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dataSysStatAvail

8.1285.2.4 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechAvail

8.1285.2.5 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBtechExtAvail

8.1285.2.6 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechExtRatValue

8.1285.2.7 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechExtSoMask

8.1285.2.8 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechnology

8.1285.2.9 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBtechnologyExt

8.1285.2.10 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dormancyStatAvail

8.1285.2.11 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dormancyStatus

8.1285.2.12 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::mipstatAvail

8.1285.2.13 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::mipStatus

8.1285.2.14 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::netInfoLen

8.1285.2.15 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::prefNetwork

8.1285.2.16 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::ratMask

8.1285.2.17 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::rx_bytes

8.1285.2.18 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::rx_pkts

8.1285.2.19 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::soMask

8.1285.2.20 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::tx_bytes

8.1285.2.21 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::tx_pkts

8.1285.2.22 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::xferStatAvail

8.1286 unpack_wds_SLQSSGetDHCPv4ClientConfig_t Struct Reference

Data Fields

- wdsDhcpv4HwConfig * pHwConfig
- wdsDhcpv4OptionList * pRequestOptionList

8.1286.1 Detailed Description

WDS SWI DHCPv4 Config Structure

Parameters

pHWConfig	• pointer to HW Config structure • See wdsDhcpv4HwConfig • NULL pointer - Invalid data.
pRequestOption-List	• pointer to Option List structure to be sent in DHCP request • See wdsDhcpv4OptionList • NULL pointer - Invalid data.

8.1286.2 Field Documentation

8.1286.2.1 wdsDhcpv4HwConfig* unpack_wds_SLQSSGetDHCPv4ClientConfig_t::pHwConfig

8.1286.2.2 wdsDhcpv4OptionList* unpack_wds_SLQSSGetDHCPv4ClientConfig_t::pRequestOptionList

8.1287 unpack_wds_SLQSSGetLoopback_t Struct Reference

Data Fields

- uint8_t ByteLoopbackMode
- uint8_t ByteLoopbackMultiplier

8.1287.1 Detailed Description

This structure contains unpack Get loopback information.

Parameters

ByteLoopback-Mode	• Loopback Mode. – 0 - Disable – 1 - Enable – 0xff - Invalid data.
ByteLoopback-Multiplier	• Loopback multiplier. Number of downlink bytes to send for each uplink byte. • 0xff - Invalid data.

8.1287.2 Field Documentation

8.1287.2.1 uint8_t unpack_wds_SLQSSGetLoopback_t::ByteLoopbackMode

8.1287.2.2 uint8_t unpack_wds_SLQSSGetLoopback_t::ByteLoopbackMultiplier

8.1288 unpack_wds_SLQSStartDataSession_t Struct Reference

Data Fields

- uint32_t * psid
- uint32_t * pFailureReason
- uint32_t * pVerboseFailReasonType
- uint32_t * pVerboseFailureReason

8.1288.1 Detailed Description

This structure contains unpack Start Data Session Information.

Parameters

psid	• Assigned session ID when starting a data session
pFailureReason	• Reason data session failed to be established • See qaGobiApiTableCallEndReasons.h for Call End Reason • NULL pointer - Invalid data.
pVerboseFail-ReasonType	• Parameter describing type of verbose failure reason • See qaGobiApiTableCallEndReasons.h for Call End Reason Type • NULL pointer - Invalid data.
pVerboseFailure-Reason	• Verbose reason explaining why call failed. Depends on verbFailReasonType parameter • See qaGobiApiTableCallEndReasons.h for Call End Reason • NULL pointer - Invalid data.

8.1288.2 Field Documentation

8.1288.2.1 uint32_t* unpack_wds_SLQSStartDataSession_t::pFailureReason

8.1288.2.2 uint32_t* unpack_wds_SLQSStartDataSession_t::psid

8.1288.2.3 uint32_t* unpack_wds_SLQSStartDataSession_t::pVerboseFailReasonType

8.1288.2.4 uint32_t* unpack_wds_SLQSStartDataSession_t::pVerboseFailureReason

8.1289 unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference

Data Fields

- uint8_t contextId
- uint8_t bearerId
- int8_t apnName [100]
- uint32_t ipv4Address
- uint32_t ipv4GWAddress
- uint32_t prDNSIPv4Address
- uint32_t seDNSIPv4Address
- ipv6AddressInfo ipv6Address
- ipv6AddressInfo ipv6GWAddress
- uint16_t prDNSIPv6Address [8]
- uint16_t seDNSIPv6Address [8]
- uint32_t prPCSCFIPv4Address
- uint32_t sePCSCFIPv4Address
- uint16_t prPCSCFIPv6Address [8]
- uint16_t sePCSCFIPv6Address [8]

8.1289.1 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 • 0xffffffff - Invalid data.
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 • 0xffffffff - Invalid data.
prPCSCFIPv4-Address	(optional) • Primary PCSCF IPv4 Address • 0xffffffff - Invalid data.
sePCSCFIPv4-Address	(optional) • Secondary PCSCF IPv4 Address • 0xffffffff - Invalid data.
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 • 0xffff - Invalid data.
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 • 0xffff - Invalid data.

8.1289.2 Field Documentation

8.1289.2.1 int8_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::apnName[100]

8.1289.2.2 uint8_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::bearerId

8.1289.2.3 uint8_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextId

8.1289.2.4 uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv4Address

8.1289.2.5 uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv4GWAddress

8.1289.2.6 ipv6AddressInfo unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv6Address

8.1289.2.7 ipv6AddressInfo unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv6GWAddress

8.1289.2.8 uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prDNSIPv4Address

8.1289.2.9 uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prDNSIPv6Address[8]

8.1289.2.10 uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prPCSCFIPv4Address

8.1289.2.11 uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prPCSCFIPv6Address[8]

8.1289.2.12 uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::seDNSIPv4Address

8.1289.2.13 uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::seDNSIPv6Address[8]

8.1289.2.14 uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::sePCSCFIPv4Address

8.1289.2.15 uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::sePCSCFIPv6Address[8]

8.1290 UnPackGetProfileSettingOut Struct Reference

Data Fields

- UnpackQmiProfileInfo curProfile
- uint16_t * pExtErrCode

8.1290.1 Detailed Description

This structure contains the profile settings output

Parameters

curProfile	• Structure containing details of the current profile • See UnpackQmiProfileInfo
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.

8.1290.2 Field Documentation

8.1290.2.1 UnpackQmiProfileInfo UnPackGetProfileSettingOut::curProfile

8.1290.2.2 uint16_t* UnPackGetProfileSettingOut::pExtErrCode

8.1291 unpackWdsProfileParam Union Reference

Data Fields

- LibpackProfile3GPP SlqsProfile3GPP
- LibpackProfile3GPP2 SlqsProfile3GPP2

8.1291.1 Detailed Description

This union WdsProfileParam consists of Profile3GPP and Profile3GPP2 out of which one will be used to create profile.

Parameters

SlqsProfile3GPP	3GPP profile • See LibpackProfile3GPP
SlqsProfile3GP-P2	3GPP2 profile • See LibpackProfile3GPP2

8.1291.2 Field Documentation

8.1291.2.1 LibpackProfile3GPP unpackWdsProfileParam::SlqsProfile3GPP

8.1291.2.2 LibpackProfile3GPP2 unpackWdsProfileParam::SlqsProfile3GPP2

8.1292 urlAddr Struct Reference

Data Fields

- CHAR address [256]
- BYTE TlvPresent

8.1292.1 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

8.1292.2 Field Documentation

8.1292.2.1 CHAR urlAddr::address[256]

8.1292.2.2 BYTE urlAddr::TlvPresent

8.1293 USBCompConfig Struct Reference

Data Fields

- BYTE * pUSBComp

8.1293.1 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

8.1293.2 Field Documentation

8.1293.2.1 BYTE* USBCompConfig::pUSBComp

8.1294 USBCompParams Struct Reference

Data Fields

- BYTE * pUSBComp
- BYTE * pNumSupUSBComps
- BYTE * pSupUSBComps

8.1294.1 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

8.1294.2 Field Documentation

8.1294.2.1 BYTE* USBCompParams::pNumSupUSBComps

8.1294.2.2 BYTE* USBCompParams::pSupUSBComps

8.1294.2.3 BYTE* USBCompParams::pUSBComp

8.1295 USSDNoWaitIndicationInfo Struct Reference

Data Fields

- BYTE * pError
- BYTE * pFailureCause
- struct USSInfo * pUSSDData
- alphaIDInfo * pAlphaIdentifier

8.1295.1 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.

8.1295.2 Field Documentation

8.1295.2.1 alphaIDInfo* USSDNoWaitIndicationInfo::pAlphaIdentifier

8.1295.2.2 BYTE* USSDNoWaitIndicationInfo::pError

8.1295.2.3 BYTE* USSDNoWaitIndicationInfo::pFailureCause

8.1295.2.4 struct USSInfo* USSDNoWaitIndicationInfo::pUSSDData

8.1296 USSDRespFNetwork Struct Reference

Data Fields

- char * pTypeCode
- char * pRespData

8.1296.1 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

8.1296.2 Field Documentation

8.1296.2.1 char* USSDRespFNetwork::pRespData

8.1296.2.2 char* USSDRespFNetwork::pTypeCode

8.1297 USSInfo Struct Reference

Data Fields

- BYTE ussDCS
- BYTE ussLen
- BYTE ussData [182]

8.1297.1 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

8.1297.2 Field Documentation

8.1297.2.1 BYTE USSInfo::ussData[182]

8.1297.2.2 BYTE USSInfo::ussDCS

8.1297.2.3 BYTE USSInfo::ussLen

8.1298 USSResp Struct Reference

Data Fields

- WORD * pfailureCause
- alphaIDInfo * pAlphaIDInfo
- struct USSInfo * pUSSDInfo
- BYTE * pCcResultType
- BYTE * pCallId
- ccSUPSType * pCCSuppsType

8.1298.1 Field Documentation

8.1298.1.1 alphaIDInfo* USSResp::pAlphaIDInfo

8.1298.1.2 BYTE* USSResp::pCallId

8.1298.1.3 BYTE* USSResp::pCcResultType

8.1298.1.4 ccSUPSType* USSResp::pCCSuppsType

8.1298.1.5 WORD* USSResp::pfailureCause

8.1298.1.6 struct USSInfo* USSResp::pUSSDInfo

8.1299 UUSInfo Struct Reference

Data Fields

- BYTE UUSType
- BYTE UUSDcs
- BYTE UUSDatalen
- BYTE UUSData [255]

8.1299.1 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

8.1299.2 Field Documentation

8.1299.2.1 BYTE UUSInfo::UUSData[255]

8.1299.2.2 BYTE UUSInfo::UUSDatalen

8.1299.2.3 BYTE UUSInfo::UUSDcs

8.1299.2.4 BYTE UUSInfo::UUSType

8.1300 verifyUIMPIN Struct Reference

Data Fields

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

8.1300.1 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.

8.1300.2 Field Documentation

8.1300.2.1 BYTE verifyUIMPIN::pinID

8.1300.2.2 BYTE verifyUIMPIN::pinLen

8.1300.2.3 BYTE verifyUIMPIN::pinVal[255]

8.1301 voice_airTimer Struct Reference

Data Fields

- uint8_t namID
- uint32_t airTimerValue

8.1301.1 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.

8.1301.2 Field Documentation

8.1301.2.1 uint32_t voice_airTimer::airTimerValue

8.1301.2.2 uint8_t voice_airTimer::namID

8.1302 voice_allCallsAlphaIDInfo Struct Reference

Data Fields

- uint8_t callID
- voice_alphaIDInfo AlphaIDInfo

8.1302.1 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.

8.1302.2 Field Documentation

8.1302.2.1 voice_alphaIDInfo voice_allCallsAlphaIDInfo::AlphaIDInfo

8.1302.2.2 uint8_t voice_allCallsAlphaIDInfo::callID

8.1303 voice_allCallsDiagInfo Struct Reference

Data Fields

- uint8_t callID
- voice_diagInfo DiagInfo

8.1303.1 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.

8.1303.2 Field Documentation

8.1303.2.1 uint8_t voice_allCallsDiagInfo::callID

8.1303.2.2 voice_diagInfo voice_allCallsDiagInfo::DiagInfo

8.1304 voice_allCallsUUSInfo Struct Reference

Data Fields

- uint8_t callID
- voice_UUSInfo uusInfo

8.1304.1 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.

8.1304.2 Field Documentation

8.1304.2.1 uint8_t voice_allCallsUUSInfo::callID

8.1304.2.2 voice_UUSInfo voice_allCallsUUSInfo::uusInfo

8.1305 voice_alphaIDInfo Struct Reference

Data Fields

- uint8_t alphaDcs
- uint8_t alphaLen
- uint8_t alphaText [255]

8.1305.1 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

8.1305.2 Field Documentation

8.1305.2.1 uint8_t voice_alphaIDInfo::alphaDcs

8.1305.2.2 uint8_t voice_alphaIDInfo::alphaLen

8.1305.2.3 uint8_t voice_alphaIDInfo::alphaText[255]

8.1306 voice_arrAlertingPattern Struct Reference

Data Fields

- uint8_t numInstances
- uint8_t callID [20]
- uint32_t alertingPattern [20]

8.1306.1 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

8.1306.2 Field Documentation

8.1306.2.1 uint32_t voice_arrAlertingPattern::alertingPattern[20]

8.1306.2.2 uint8_t voice_arrAlertingPattern::callID[20]

8.1306.2.3 uint8_t voice_arrAlertingPattern::numInstances

8.1307 voice_arrAlertingType Struct Reference

Data Fields

- uint8_t numInstances
- uint8_t callID [20]
- uint8_t AlertingType [20]

8.1307.1 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

8.1307.2 Field Documentation

8.1307.2.1 uint8_t voice_arrAlertingType::AlertingType[20]

8.1307.2.2 uint8_t voice_arrAlertingType::callID[20]

8.1307.2.3 uint8_t voice_arrAlertingType::numInstances

8.1308 voice_arrAlphaID Struct Reference

Data Fields

- uint8_t numInstances
- voice_allCallsAlphaIDInfo allCallsAlphaIDInfoArr [20]

8.1308.1 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.

8.1308.2 Field Documentation

8.1308.2.1 voice_allCallsAlphaIDInfo voice_arrAlphaID::allCallsAlphaIDInfoArr[20]

8.1308.2.2 uint8_t voice_arrAlphaID::numInstances

8.1309 voice_arrCalledPartyNum Struct Reference

Data Fields

- uint8_t numInstances
- voice_peerNumberInfo CalledPartyNum [20]

8.1309.1 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.

8.1309.2 Field Documentation

8.1309.2.1 voice_peerNumberInfo voice_arrCalledPartyNum::CalledPartyNum[20]

8.1309.2.2 uint8_t voice_arrCalledPartyNum::numInstances

8.1310 voice_arrCallEndReason Struct Reference

Data Fields

- uint8_t numInstances
- uint8_t callID [20]
- uint16_t callEndReason [20]

8.1310.1 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

8.1310.2 Field Documentation

8.1310.2.1 uint16_t voice_arrCallEndReason::callEndReason[20]

8.1310.2.2 uint8_t voice_arrCallEndReason::callID[20]

8.1310.2.3 uint8_t voice_arrCallEndReason::numInstances

8.1311 voice_arrCallInfo Struct Reference

Data Fields

- uint8_t numInstances

- voice_getAllCallInformation getAllCallInfo [20]

8.1311.1 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.

8.1311.2 Field Documentation

8.1311.2.1 voice_getAllCallInformation voice_arrCallInfo::getAllCallInfo[20]

8.1311.2.2 uint8_t voice_arrCallInfo::numInstances

8.1312 voice_arrConnectPartyNum Struct Reference

Data Fields

- uint8_t numInstances
- voice_peerNumberInfo ConnectedPartyNum [20]

8.1312.1 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.

8.1312.2 Field Documentation

8.1312.2.1 voice_peerNumberInfo voice_arrConnectPartyNum::ConnectedPartyNum[20]

8.1312.2.2 uint8_t voice_arrConnectPartyNum::numInstances

8.1313 voice_arrDiagInfo Struct Reference

Data Fields

- uint8_t numInstances
- voice_allCallsDiagInfo DiagInfo [20]

8.1313.1 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.

8.1313.2 Field Documentation

8.1313.2.1 voice_allCallsDiagInfo voice_arrDiagInfo::DiagInfo[20]

8.1313.2.2 uint8_t voice_arrDiagInfo::numInstances

8.1314 voice_arrRedirPartyNum Struct Reference

Data Fields

- uint8_t numInstances
- voice_peerNumberInfo RedirPartyNum [20]

8.1314.1 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.

8.1314.2 Field Documentation

8.1314.2.1 uint8_t voice_arrRedirPartyNum::numInstances

8.1314.2.2 voice_peerNumberInfo voice_arrRedirPartyNum::RedirPartyNum[20]

8.1315 voice_arrRemotePartyName Struct Reference

Data Fields

- uint8_t numInstances
- voice_getAllCallRmtPtyName GetAllCallRmtPtyName [20]

8.1315.1 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.

8.1315.2 Field Documentation

8.1315.2.1 voice_getAllCallRmtPtyName voice_arrRemotePartyName::GetAllCallRmtPtyName[20]

8.1315.2.2 uint8_t voice_arrRemotePartyName::numInstances

8.1316 voice_arrRemotePartyNum Struct Reference

Data Fields

- uint8_t numInstances
- voice_getAllCallRmtPtyNum RmtPtyNum [20]

8.1316.1 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.

8.1316.2 Field Documentation

8.1316.2.1 uint8_t voice_arrRemotePartyNum::numInstances

8.1316.2.2 voice_getAllCallRmtPtyNum voice_arrRemotePartyNum::RmtPtyNum[20]

8.1317 voice_arrSvcOption Struct Reference

Data Fields

- uint8_t numInstances
- uint8_t callID [20]
- uint16_t srvOption [20]

8.1317.1 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.

8.1317.2 Field Documentation

8.1317.2.1 uint8_t voice_arrSvcOption::callID[20]

8.1317.2.2 uint8_t voice_arrSvcOption::numInstances

8.1317.2.3 uint16_t voice_arrSvcOption::srvOption[20]

8.1318 voice_arrUUSInfo Struct Reference

Data Fields

- uint8_t numInstances
- voice_allCallsUUSInfo AllCallsUUSInfo [20]

8.1318.1 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.

8.1318.2 Field Documentation

8.1318.2.1 voice_allCallsUUSInfo voice_arrUUSInfo::AllCallsUUSInfo[20]

8.1318.2.2 uint8_t voice_arrUUSInfo::numInstances

8.1319 voice_burstDTMFInfo Struct Reference

Data Fields

- uint8_t * pCallID
- uint8_t digitCnt
- uint8_t pDigitBuff [255]

8.1319.1 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 • NULL pointer - Invalid data.
digitCnt	• Length of DTMF digit buffer which follows
pDigitBuff[MAX_VOICE_DESCRIPTION_LENGTH]	• DTMF digit buffer in ASCII, NULL terminated

8.1319.2 Field Documentation

8.1319.2.1 uint8_t voice_burstDTMFInfo::digitCnt

8.1319.2.2 uint8_t* voice_burstDTMFInfo::pCallID

8.1319.2.3 uint8_t voice_burstDTMFInfo::pDigitBuff[255]

8.1320 voice_calledPartySubAdd Struct Reference

Data Fields

- uint8_t extBit
- uint8_t subAddrType
- uint8_t oddEvenInd
- uint8_t subAddrLen
- uint8_t subAddr [255]

8.1320.1 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.

8.1320.2 Field Documentation

8.1320.2.1 uint8_t voice_calledPartySubAdd::extBit

8.1320.2.2 uint8_t voice_calledPartySubAdd::oddEvenInd

8.1320.2.3 uint8_t voice_calledPartySubAdd::subAddr[255]

8.1320.2.4 uint8_t voice_calledPartySubAdd::subAddrLen

8.1320.2.5 uint8_t voice_calledPartySubAdd::subAddrType

8.1321 voice_callFwdTypeAndPlan Struct Reference

Data Fields

- uint8_t numberType
- uint8_t numberPlan

8.1321.1 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

8.1321.2 Field Documentation

8.1321.2.1 uint8_t voice_callFwdTypeAndPlan::numberPlan

8.1321.2.2 uint8_t voice_callFwdTypeAndPlan::numberType

8.1322 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]

8.1322.1 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.

8.1322.2 Field Documentation

8.1322.2.1 uint8_t voice_callFWExtInfo::noReplyTimer

8.1322.2.2 uint8_t voice_callFWExtInfo::number[255]

8.1322.2.3 uint8_t voice_callFWExtInfo::numLen

8.1322.2.4 uint8_t voice_callFWExtInfo::numPlan

8.1322.2.5 uint8_t voice_callFWExtInfo::numType

8.1322.2.6 uint8_t voice_callFWExtInfo::PI

8.1322.2.7 uint8_t voice_callFWExtInfo::SI

8.1322.2.8 uint8_t voice_callFWExtInfo::SvcClass

8.1322.2.9 uint8_t voice_callFWExtInfo::SvcStatus

8.1323 voice_callFWInfo Struct Reference

Data Fields

- uint8_t SvcStatus
- uint8_t SvcClass
- uint8_t numLen
- uint8_t number [255]
- uint8_t noReplyTimer

8.1323.1 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.

8.1323.2 Field Documentation

8.1323.2.1 uint8_t voice_callFWInfo::noReplyTimer

8.1323.2.2 uint8_t voice_callFWInfo::number[255]

8.1323.2.3 uint8_t voice_callFWInfo::numLen

8.1323.2.4 uint8_t voice_callFWInfo::SvcClass

8.1323.2.5 uint8_t voice_callFWInfo::SvcStatus

8.1324 voice_callInfo Struct Reference

Data Fields

- uint8_t callID
- uint8_t callState
- uint8_t callType
- uint8_t direction
- uint8_t mode

8.1324.1 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

8.1324.2 Field Documentation

8.1324.2.1 uint8_t voice_callInfo::callID

8.1324.2.2 uint8_t voice_callInfo::callState

8.1324.2.3 uint8_t voice_callInfo::callType

8.1324.2.4 uint8_t voice_callInfo::direction

8.1324.2.5 uint8_t voice_callInfo::mode

8.1325 voice_ccSUPSType Struct Reference

Data Fields

- uint8_t svcType
- uint8_t reason

8.1325.1 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.

8.1325.2 Field Documentation

8.1325.2.1 uint8_t voice_ccSUPSType::reason

8.1325.2.2 uint8_t voice_ccSUPSType::svcType

8.1326 voice_CLIPResp Struct Reference

Data Fields

- uint8_t ActiveStatus
- uint8_t ProvisionStatus

8.1326.1 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

8.1326.2 Field Documentation

8.1326.2.1 uint8_t voice_CLIPResp::ActiveStatus

8.1326.2.2 uint8_t voice_CLIPResp::ProvisionStatus

8.1327 voice_CLIRResp Struct Reference

Data Fields

- uint8_t ActiveStatus
- uint8_t ProvisionStatus

8.1327.1 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

8.1327.2 Field Documentation

8.1327.2.1 uint8_t voice_CLIRResp::ActiveStatus

8.1327.2.2 uint8_t voice_CLIRResp::ProvisionStatus

8.1328 voice_CNAPResp Struct Reference

Data Fields

- uint8_t ActiveStatus
- uint8_t ProvisionStatus

8.1328.1 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

8.1328.2 Field Documentation

8.1328.2.1 uint8_t voice_CNAPResp::ActiveStatus

8.1328.2.2 uint8_t voice_CNAPResp::ProvisionStatus

8.1329 voice_COLPResp Struct Reference

Data Fields

- uint8_t ActiveStatus
- uint8_t ProvisionStatus

8.1329.1 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

8.1329.2 Field Documentation

8.1329.2.1 uint8_t voice_COLPResp::ActiveStatus

8.1329.2.2 uint8_t voice_COLPResp::ProvisionStatus

8.1330 voice_COLRResp Struct Reference

Data Fields

- uint8_t ActiveStatus
- uint8_t ProvisionStatus

8.1330.1 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

8.1330.2 Field Documentation

8.1330.2.1 uint8_t voice_COLRResp::ActiveStatus

8.1330.2.2 uint8_t voice_COLRResp::ProvisionStatus

8.1331 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]

8.1331.1 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.

8.1331.2 Field Documentation

8.1331.2.1 uint8_t voice_connectNumInfo::callerID[81]

8.1331.2.2 uint8_t voice_connectNumInfo::callerIDLen

8.1331.2.3 uint8_t voice_connectNumInfo::numPlan

8.1331.2.4 uint8_t voice_connectNumInfo::numPresInd

8.1331.2.5 uint8_t voice_connectNumInfo::numType

8.1331.2.6 uint8_t voice_connectNumInfo::screeningInd

8.1332 voice_CUGInfo Struct Reference

Data Fields

- uint16_t CUGIndex
- uint8_t SuppPrefCUG
- uint8_t SuppOA

8.1332.1 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

8.1332.2 Field Documentation

8.1332.2.1 uint16_t voice_CUGInfo::CUGIndex

8.1332.2.2 uint8_t voice_CUGInfo::SuppOA

8.1332.2.3 uint8_t voice_CUGInfo::SuppPrefCUG

8.1333 voice_curAMRConfig Struct Reference

Data Fields

- uint8_t gsmAmrStat
- uint8_t wcdmaAmrStat

8.1333.1 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

8.1333.2 Field Documentation

8.1333.2.1 uint8_t voice_curAMRConfig::gsmAmrStat

8.1333.2.2 uint8_t voice_curAMRConfig::wcdmaAmrStat

8.1334 voice_diagInfo Struct Reference

Data Fields

- uint8_t diagInfoLen
- uint8_t diagnosticInfo [255]

8.1334.1 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.

8.1334.2 Field Documentation

8.1334.2.1 uint8_t voice_diagInfo::diagInfoLen

8.1334.2.2 uint8_t voice_diagInfo::diagnosticInfo[255]

8.1335 voice_DTMFLengths Struct Reference

Data Fields

- uint8_t DTMFPulseWidth
- uint8_t DTMFInterdigitInterval

8.1335.1 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

8.1335.2 Field Documentation

8.1335.2.1 uint8_t voice_DTMFLengths::DTMFInterdigitInterval

8.1335.2.2 uint8_t voice_DTMFLengths::DTMFPulseWidth

8.1336 voice_getAllCallInformation Struct Reference

Data Fields

- voice_callInfo CallInfo
- uint8_t isEmpty
- uint8_t ALS

8.1336.1 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

8.1336.2 Field Documentation

8.1336.2.1 uint8_t voice_getAllCallInformation::ALS

8.1336.2.2 voice_callInfo voice_getAllCallInformation::CallInfo

8.1336.2.3 uint8_t voice_getAllCallInformation::isEmpty

8.1337 voice_getAllCallRmtPtyName Struct Reference

Data Fields

- uint8_t callID
- voice_remotePartyName RemotePartyName

8.1337.1 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.

8.1337.2 Field Documentation

8.1337.2.1 uint8_t voice_getAllCallRmtPtyName::callID

8.1337.2.2 voice_remotePartyName voice_getAllCallRmtPtyName::RemotePartyName

8.1338 voice_getAllCallRmtPtyNum Struct Reference

Data Fields

- uint8_t callID
- voice_remotePartyNum RemotePartyNum

8.1338.1 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.

8.1338.2 Field Documentation

8.1338.2.1 uint8_t voice_getAllCallRmtPtyNum::callID

8.1338.2.2 voice_remotePartyNum voice_getAllCallRmtPtyNum::RemotePartyNum

8.1339 voice_getCallFWExtInfo Struct Reference

Data Fields

- uint8_t numInstances
- voice_callFWExtInfo CallFWExtInfo [20]

8.1339.1 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.

8.1339.2 Field Documentation

8.1339.2.1 voice_callFWExtInfo voice_getCallFWExtInfo::CallFWExtInfo[20]

8.1339.2.2 uint8_t voice_getCallFWExtInfo::numInstances

8.1340 voice_getCallFWInfo Struct Reference

Data Fields

- uint8_t numInstances
- voice_callFWInfo CallFWInfo [20]

8.1340.1 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.

8.1340.2 Field Documentation

8.1340.2.1 voice_callFWInfo voice_getCallFWInfo::CallFWInfo[20]

8.1340.2.2 uint8_t voice_getCallFWInfo::numInstances

8.1341 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]

8.1341.1 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.

8.1341.2 Field Documentation

8.1341.2.1 uint8_t voice_peerNumberInfo::callID

8.1341.2.2 uint8_t voice_peerNumberInfo::number[81]

8.1341.2.3 uint8_t voice_peerNumberInfo::numLen

8.1341.2.4 uint8_t voice_peerNumberInfo::numPI

8.1341.2.5 uint8_t voice_peerNumberInfo::numPlan

8.1341.2.6 uint8_t voice_peerNumberInfo::numSI

8.1341.2.7 uint8_t voice_peerNumberInfo::numType

8.1342 voice_prefVoiceSO Struct Reference

Data Fields

- uint8_t namID
- uint8_t evrcCapability
- uint16_t homePageVoiceSO
- uint16_t homeOrigVoiceSO
- uint16_t roamOrigVoiceSO

8.1342.1 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

8.1342.2 Field Documentation

8.1342.2.1 uint8_t voice_prefVoiceSO::evrcCapability

8.1342.2.2 uint16_t voice_prefVoiceSO::homeOrigVoiceSO

8.1342.2.3 uint16_t voice_prefVoiceSO::homePageVoiceSO

8.1342.2.4 uint8_t voice_prefVoiceSO::namID

8.1342.2.5 uint16_t voice_prefVoiceSO::roamOrigVoiceSO

8.1343 voice_remotePartyName Struct Reference

Data Fields

- uint8_t namePI
- uint8_t codingScheme
- uint8_t nameLen
- uint8_t callerName [255]

8.1343.1 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.

8.1343.2 Field Documentation

8.1343.2.1 uint8_t voice_remotePartyName::callerName[255]

8.1343.2.2 uint8_t voice_remotePartyName::codingScheme

8.1343.2.3 uint8_t voice_remotePartyName::nameLen

8.1343.2.4 uint8_t voice_remotePartyName::namePI

8.1344 voice_remotePartyNum Struct Reference

Data Fields

- uint8_t presentationInd
- uint8_t numLen
- uint8_t remPartyNumber [81]

8.1344.1 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.

8.1344.2 Field Documentation

8.1344.2.1 uint8_t voice_remotePartyNum::numLen

8.1344.2.2 uint8_t voice_remotePartyNum::presentationInd

8.1344.2.3 uint8_t voice_remotePartyNum::remPartyNumber[81]

8.1345 voice_roamTimer Struct Reference

Data Fields

- uint8_t namID
- uint32_t roamTimerValue

8.1345.1 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.

8.1345.2 Field Documentation

8.1345.2.1 uint8_t voice_roamTimer::namID

8.1345.2.2 uint32_t voice_roamTimer::roamTimerValue

8.1346 voice_USSInfo Struct Reference

Data Fields

- uint8_t ussDCS
- uint8_t ussLen
- uint8_t ussData [182]

8.1346.1 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

8.1346.2 Field Documentation

8.1346.2.1 uint8_t voice_USSInfo::ussData[182]

8.1346.2.2 uint8_t voice_USSInfo::ussDCS

8.1346.2.3 uint8_t voice_USSInfo::ussLen

8.1347 voice_UUSInfo Struct Reference

Data Fields

- uint8_t UUSType
- uint8_t UUSDcs
- uint8_t UUSDatalen
- uint8_t UUSData [255]

8.1347.1 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

8.1347.2 Field Documentation

8.1347.2.1 uint8_t voice_UUSInfo::UUSData[255]

8.1347.2.2 uint8_t voice_UUSInfo::UUSDatalen

8.1347.2.3 uint8_t voice_UUSInfo::UUSDcs

8.1347.2.4 uint8_t voice_UUSInfo::UUSType

8.1348 voiceALSSelectLineInfo Struct Reference

Data Fields

- BYTE lineValue

8.1348.1 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

8.1348.2 Field Documentation

8.1348.2.1 BYTE voiceALSSelectLineInfo::lineValue

8.1349 voiceALSSetLineSwitchInfo Struct Reference

Data Fields

- BYTE switchOption

8.1349.1 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

8.1349.2 Field Documentation

8.1349.2.1 BYTE voiceALSSetLineSwitchInfo::switchOption

8.1350 voiceAnswerCall Struct Reference

Data Fields

- BYTE * pCallId

8.1350.1 Detailed Description

Contains the parameters passed for SLQSVoiceAnswerCall.

Parameters

pCallId[IN/OUT]	• Unique call identifier for the call that must be answered.

8.1350.2 Field Documentation

8.1350.2.1 BYTE* voiceAnswerCall::pCallId

8.1351 voiceBindSubscriptionInfo Struct Reference

Data Fields

- BYTE subsType

8.1351.1 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

8.1351.2 Field Documentation

8.1351.2.1 BYTE voiceBindSubscriptionInfo::subsType

8.1352 voiceBurstDTMFInfo Struct Reference

Data Fields

- burstDTMFInfo BurstDTMFInfo
- DTMFLengths * pBurstDTMFLengths

8.1352.1 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

8.1352.2 Field Documentation

8.1352.2.1 burstDTMFInfo voiceBurstDTMFInfo::BurstDTMFInfo

8.1352.2.2 DTMFLengths* voiceBurstDTMFInfo::pBurstDTMFLengths

8.1353 voiceCallInfoReq Struct Reference

Data Fields

- BYTE callID

8.1353.1 Detailed Description

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

Parameters

callID	• Call identifier for the call queried for information.

8.1353.2 Field Documentation

8.1353.2.1 BYTE voiceCallInfoReq::callID

8.1354 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

8.1354.1 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

8.1354.2 Field Documentation

8.1354.2.1 ULONG* voiceCallInfoResp::pAlertingPattern

8.1354.2.2 BYTE* voiceCallInfoResp::pAlertType

8.1354.2.3 alphaIDInfo* voiceCallInfoResp::pAlphaIDInfo

8.1354.2.4 callInfo* voiceCallInfoResp::pCallInfo

8.1354.2.5 connectNumInfo* voiceCallInfoResp::pConnectNumInfo

8.1354.2.6 diagInfo* voiceCallInfoResp::pDiagInfo

8.1354.2.7 BYTE* voiceCallInfoResp::pOTASPStatus

8.1354.2.8 remotePartyName* voiceCallInfoResp::pRemotePartyName

8.1354.2.9 remotePartyNum* voiceCallInfoResp::pRemotePartyNum

8.1354.2.10 WORD* voiceCallInfoResp::pSrvOpt

8.1354.2.11 UUSInfo* voiceCallInfoResp::pUUSInfo

8.1354.2.12 BYTE* voiceCallInfoResp::pVoicePrivacy

8.1355 voiceCallRequestParams Struct Reference

Data Fields

- BYTE callNumber [81]
- BYTE * pCallType
- BYTE * pCLIRType
- UUSInfo * pUUSInFo
- CUGInfo * pCUGInfo
- BYTE * pEmergencyCategory

- calledPartySubAdd * pCallPartySubAdd
- ULONG * pSvcType

8.1355.1 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

8.1355.2 Field Documentation

8.1355.2.1 BYTE voiceCallRequestParams::callNumber[81]

8.1355.2.2 calledPartySubAdd* voiceCallRequestParams::pCallPartySubAdd

8.1355.2.3 BYTE* voiceCallRequestParams::pCallType

8.1355.2.4 BYTE* voiceCallRequestParams::pCLIRType

8.1355.2.5 CUGInfo* voiceCallRequestParams::pCUGInfo

8.1355.2.6 BYTE* voiceCallRequestParams::pEmergencyCategory

8.1355.2.7 ULONG* voiceCallRequestParams::pSvcType

8.1355.2.8 UUSInfo* voiceCallRequestParams::pUUSInFo

8.1356 voiceCallResponseParams Struct Reference

Data Fields

- BYTE * pCallID
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResultType
- ccSUPSType * pCCSUPSType

8.1356.1 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

8.1356.2 Field Documentation

8.1356.2.1 alphaIDInfo∗ voiceCallResponseParams::pAlphaIDInfo

8.1356.2.2 BYTE∗ voiceCallResponseParams::pCallID

8.1356.2.3 BYTE∗ voiceCallResponseParams::pCCResultType

8.1356.2.4 ccSUPSType∗ voiceCallResponseParams::pCCSUPSType

8.1357 voiceContDTMFinfo Struct Reference

Data Fields

- BYTE ∗ pCallID
- BYTE DTMFdigit

8.1357.1 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.

8.1357.2 Field Documentation

8.1357.2.1 BYTE voiceContDTMFinfo::DTMFdigit

8.1357.2.2 BYTE* voiceContDTMFinfo::pCallID

8.1358 voiceDTMFEventInfo Struct Reference

Data Fields

- DTMFInfo DTMFInformation
- BYTE * pOnLength
- BYTE * pOffLength

8.1358.1 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

8.1358.2 Field Documentation

8.1358.2.1 DTMFInfo voiceDTMFEventInfo::DTMFInformation

8.1358.2.2 BYTE* voiceDTMFEventInfo::pOffLength

8.1358.2.3 BYTE* voiceDTMFEventInfo::pOnLength

8.1359 voiceFlashInfo Struct Reference

Data Fields

- BYTE * pCallID
- BYTE * pFlashPayLd
- BYTE * pFlashType

8.1359.1 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

8.1359.2 Field Documentation

8.1359.2.1 BYTE* voiceFlashInfo::pCallID

8.1359.2.2 BYTE* voiceFlashInfo::pFlashPayLd

8.1359.2.3 BYTE* voiceFlashInfo::pFlashType

8.1360 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

8.1360.1 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.

8.1360.2 Field Documentation

8.1360.2.1 arrAlertingPattern∗ voiceGetAllCallInfo::pArrAlertingPattern

8.1360.2.2 arrAlertingType∗ voiceGetAllCallInfo::pArrAlertingType

8.1360.2.3 arrAlphaID∗ voiceGetAllCallInfo::pArrAlphaID

8.1360.2.4 arrCalledPartyNum∗ voiceGetAllCallInfo::pArrCalledPartyNum

8.1360.2.5 arrCallEndReason∗ voiceGetAllCallInfo::pArrCallEndReason

8.1360.2.6 arrCallInfo∗ voiceGetAllCallInfo::pArrCallInfo

8.1360.2.7 arrConnectPartyNum∗ voiceGetAllCallInfo::pArrConnectPartyNum

8.1360.2.8 arrDiagInfo∗ voiceGetAllCallInfo::pArrDiagInfo

8.1360.2.9 arrRedirPartyNum∗ voiceGetAllCallInfo::pArrRedirPartyNum

8.1360.2.10 arrRemotePartyName∗ voiceGetAllCallInfo::pArrRemotePartyName

8.1360.2.11 arrRemotePartyNum∗ voiceGetAllCallInfo::pArrRemotePartyNum

8.1360.2.12 arrSvcOption* voiceGetAllCallInfo::pArrSvcOption

8.1360.2.13 arrUUSInfo* voiceGetAllCallInfo::pArrUUSInfo

8.1360.2.14 BYTE* voiceGetAllCallInfo::pOTASPStatus

8.1360.2.15 BYTE* voiceGetAllCallInfo::pVoicePrivacy

8.1361 voiceGetCallBarringReq Struct Reference

Data Fields

- BYTE reason
- BYTE * pSvcClass

8.1361.1 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

8.1361.2 Field Documentation

8.1361.2.1 BYTE* voiceGetCallBarringReq::pSvcClass

8.1361.2.2 BYTE voiceGetCallBarringReq::reason

8.1362 voiceGetCallBarringResp Struct Reference

Data Fields

- BYTE * pSvcClass
- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType

8.1362.1 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.

8.1362.2 Field Documentation

8.1362.2.1 alphaIDInfo* voiceGetCallBarringResp::pAlphaIDInfo

8.1362.2.2 BYTE* voiceGetCallBarringResp::pCallID

8.1362.2.3 BYTE* voiceGetCallBarringResp::pCCResType

8.1362.2.4 ccSUPSType* voiceGetCallBarringResp::pCCSUPSType

8.1362.2.5 WORD* voiceGetCallBarringResp::pFailCause

8.1362.2.6 BYTE* voiceGetCallBarringResp::pSvcClass

8.1363 voiceGetCallFWReq Struct Reference

Data Fields

- BYTE Reason
- BYTE * pSvcClass

8.1363.1 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.

8.1363.2 Field Documentation

8.1363.2.1 BYTE* voiceGetCallFWReq::pSvcClass

8.1363.2.2 BYTE voiceGetCallFWReq::Reason

8.1364 voiceGetCallFWResp Struct Reference

Data Fields

- getCallFWInfo * pGetCallFWInfo
- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType
- getCallFWExtInfo * pGetCallFWExtInfo

8.1364.1 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.

8.1364.2 Field Documentation

8.1364.2.1 alphaIDInfo* voiceGetCallFWResp::pAlphaIDInfo

8.1364.2.2 BYTE* voiceGetCallFWResp::pCallID

8.1364.2.3 BYTE* voiceGetCallFWResp::pCCResType

8.1364.2.4 ccSUPSType* voiceGetCallFWResp::pCCSUPSType

8.1364.2.5 WORD* voiceGetCallFWResp::pFailCause

8.1364.2.6 getCallFWExtInfo* voiceGetCallFWResp::pGetCallFWExtInfo

8.1364.2.7 getCallFWInfo* voiceGetCallFWResp::pGetCallFWInfo

8.1365 voiceGetCallWaitInfo Struct Reference

Data Fields

- BYTE * pSvcClass
- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType

8.1365.1 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.

8.1365.2 Field Documentation

8.1365.2.1 alphaIDInfo* voiceGetCallWaitInfo::pAlphaIDInfo

8.1365.2.2 BYTE* voiceGetCallWaitInfo::pCallID

8.1365.2.3 BYTE* voiceGetCallWaitInfo::pCCResType

8.1365.2.4 ccSUPSType* voiceGetCallWaitInfo::pCCSUPSType

8.1365.2.5 WORD* voiceGetCallWaitInfo::pFailCause

8.1365.2.6 BYTE* voiceGetCallWaitInfo::pSvcClass

8.1366 voiceGetCLIPResp Struct Reference

Data Fields

- CLIPResp * pCLIPResp
- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType

8.1366.1 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.

8.1366.2 Field Documentation

8.1366.2.1 alphaIDInfo* voiceGetCLIPResp::pAlphaIDInfo

8.1366.2.2 BYTE* voiceGetCLIPResp::pCallID

8.1366.2.3 BYTE* voiceGetCLIPResp::pCCResType

8.1366.2.4 ccSUPSType* voiceGetCLIPResp::pCCSUPSType

8.1366.2.5 CLIPResp* voiceGetCLIPResp::pCLIPResp

8.1366.2.6 WORD* voiceGetCLIPResp::pFailCause

8.1367 voiceGetCLIRResp Struct Reference

Data Fields

- CLIRResp * pCLIRResp
- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType

8.1367.1 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.

8.1367.2 Field Documentation

8.1367.2.1 alphaIDInfo* voiceGetCLIRResp::pAlphaIDInfo

8.1367.2.2 BYTE* voiceGetCLIRResp::pCallID

8.1367.2.3 BYTE* voiceGetCLIRResp::pCCResType

8.1367.2.4 ccSUPSType* voiceGetCLIRResp::pCCSUPSType

8.1367.2.5 CLIRResp* voiceGetCLIRResp::pCLIRResp

8.1367.2.6 WORD* voiceGetCLIRResp::pFailCause

8.1368 voiceGetCNAPResp Struct Reference

Data Fields

- CNAPResp * pCNAPResp
- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType

8.1368.1 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.

8.1368.2 Field Documentation

8.1368.2.1 alphaIDInfo* voiceGetCNAPResp::pAlphaIDInfo

8.1368.2.2 BYTE* voiceGetCNAPResp::pCallID

8.1368.2.3 BYTE* voiceGetCNAPResp::pCCResType

8.1368.2.4 ccSUPSType* voiceGetCNAPResp::pCCSUPSType

8.1368.2.5 CNAPResp* voiceGetCNAPResp::pCNAPResp

8.1368.2.6 WORD* voiceGetCNAPResp::pFailCause

8.1369 voiceGetCOLPResp Struct Reference

Data Fields

- COLPResp * pCOLPResp
- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType

8.1369.1 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.

8.1369.2 Field Documentation

8.1369.2.1 alphaIDInfo∗ voiceGetCOLPResp::pAlphaIDInfo

8.1369.2.2 BYTE∗ voiceGetCOLPResp::pCallID

8.1369.2.3 BYTE∗ voiceGetCOLPResp::pCCResType

8.1369.2.4 ccSUPSType∗ voiceGetCOLPResp::pCCSUPSType

8.1369.2.5 COLPResp∗ voiceGetCOLPResp::pCOLPResp

8.1369.2.6 WORD∗ voiceGetCOLPResp::pFailCause

8.1370 voiceGetCOLRResp Struct Reference

Data Fields

- COLRResp ∗ pCOLRResp
- WORD ∗ pFailCause
- alphaIDInfo ∗ pAlphaIDInfo
- BYTE ∗ pCCResType
- BYTE ∗ pCallID
- ccSUPSType ∗ pCCSUPSType

8.1370.1 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.

8.1370.2 Field Documentation

8.1370.2.1 alphaIDInfo* voiceGetCOLRResp::pAlphaIDInfo

8.1370.2.2 BYTE* voiceGetCOLRResp::pCallID

8.1370.2.3 BYTE* voiceGetCOLRResp::pCCResType

8.1370.2.4 ccSUPSType* voiceGetCOLRResp::pCCSUPSType

8.1370.2.5 COLRResp* voiceGetCOLRResp::pCOLRResp

8.1370.2.6 WORD* voiceGetCOLRResp::pFailCause

8.1371 voiceGetConfigReq Struct Reference

Data Fields

- BYTE * pAutoAnswer
- BYTE * pAirTimer
- BYTE * pRoamTimer
- BYTE * pTTYMode
- BYTE * pPrefVoiceSO
- BYTE * pAMRStatus
- BYTE * pPrefVoicePrivacy
- BYTE * pNamID
- BYTE * pVoiceDomainPref

8.1371.1 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.

8.1371.2 Field Documentation

8.1371.2.1 BYTE* voiceGetConfigReq::pAirTimer

8.1371.2.2 BYTE* voiceGetConfigReq::pAMRStatus

8.1371.2.3 BYTE* voiceGetConfigReq::pAutoAnswer

8.1371.2.4 BYTE* voiceGetConfigReq::pNamID

8.1371.2.5 BYTE* voiceGetConfigReq::pPrefVoicePrivacy

8.1371.2.6 BYTE* voiceGetConfigReq::pPrefVoiceSO

8.1371.2.7 BYTE* voiceGetConfigReq::pRoamTimer

8.1371.2.8 BYTE* voiceGetConfigReq::pTTYMode

8.1371.2.9 BYTE* voiceGetConfigReq::pVoiceDomainPref

8.1372 voiceGetConfigResp Struct Reference

Data Fields

- BYTE * pAutoAnswerStat
- airTimer * pAirTimerCnt
- roamTimer * pRoamTimerCnt
- BYTE * pCurrTTYMode
- prefVoiceSO * pCurPrefVoiceSO
- curAMRConfig * pCurAMRConfig
- BYTE * pCurVoicePrivacyPref
- BYTE * pCurVoiceDomainPref

8.1372.1 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.

8.1372.2 Field Documentation

8.1372.2.1 airTimer∗ voiceGetConfigResp::pAirTimerCnt

8.1372.2.2 BYTE∗ voiceGetConfigResp::pAutoAnswerStat

8.1372.2.3 curAMRConfig∗ voiceGetConfigResp::pCurAMRConfig

8.1372.2.4 prefVoiceSO∗ voiceGetConfigResp::pCurPrefVoiceSO

8.1372.2.5 BYTE∗ voiceGetConfigResp::pCurrTTYMode

8.1372.2.6 BYTE∗ voiceGetConfigResp::pCurVoiceDomainPref

8.1372.2.7 BYTE∗ voiceGetConfigResp::pCurVoicePrivacyPref

8.1372.2.8 roamTimer∗ voiceGetConfigResp::pRoamTimerCnt

8.1373 voiceIndicationRegisterInfo Struct Reference

Data Fields

- BYTE ∗ pRegDTMFEvents
- BYTE ∗ pRegVoicePrivacyEvents
- BYTE ∗ pSuppsNotifEvents

8.1373.1 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.

8.1373.2 Field Documentation

8.1373.2.1 BYTE* voiceIndicationRegisterInfo::pRegDTMFEvents

8.1373.2.2 BYTE* voiceIndicationRegisterInfo::pRegVoicePrivacyEvents

8.1373.2.3 BYTE* voiceIndicationRegisterInfo::pSuppsNotifEvents

8.1374 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

8.1374.1 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

8.1374.2 Field Documentation

8.1374.2.1 BYTE voiceInfoRec::callID

8.1374.2.2 calledPartyInfo* voiceInfoRec::pCalledPartyInfo

8.1374.2.3 callerIDInfo* voiceInfoRec::pCallerIDInfo

8.1374.2.4 BYTE* voiceInfoRec::pCallerNameInfo

8.1374.2.5 connectNumInfo* voiceInfoRec::pCallingPartyInfo

8.1374.2.6 BYTE* voiceInfoRec::pCallWaitInd

8.1374.2.7 BYTE* voiceInfoRec::pCLIRCause

8.1374.2.8 connectNumInfo* voiceInfoRec::pConnectNumInfo

8.1374.2.9 BYTE* voiceInfoRec::pDispInfo

8.1374.2.10 BYTE* voiceInfoRec::pExtDispInfo

8.1374.2.11 extDispRecInfo* voiceInfoRec::pExtDispRecInfo

8.1374.2.12 lineCtrlInfo* voiceInfoRec::pLineCtrlInfo

8.1374.2.13 NSSAudioCtrl* voiceInfoRec::pNSSAudioCtrl

8.1374.2.14 BYTE* voiceInfoRec::pNSSRelease

8.1374.2.15 redirNumInfo* voiceInfoRec::pRedirNumInfo

8.1374.2.16 signalInfo* voiceInfoRec::pSignalInfo

8.1375 voiceManageCallsReq Struct Reference

Data Fields

- BYTE SUPSType
- BYTE * pCallID

8.1375.1 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

8.1375.2 Field Documentation

8.1375.2.1 BYTE* voiceManageCallsReq::pCallID

8.1375.2.2 BYTE voiceManageCallsReq::SUPSType

8.1376 voiceManageCallsResp Struct Reference

Data Fields

- WORD * pFailCause

8.1376.1 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

8.1376.2 Field Documentation

8.1376.2.1 WORD* voiceManageCallsResp::pFailCause

8.1377 voiceOrigUSSDNoWaitInfo Struct Reference

Data Fields

- struct USSInfo USSInformation

8.1377.1 Detailed Description

This structure contains Orig USSD No Wait Information Parameters.

Parameters

USSInformation	• See USSInfo for more information.

8.1377.2 Field Documentation

8.1377.2.1 struct USSInfo voiceOrigUSSDNoWaitInfo::USSInformation

8.1378 voiceOTASPStatusInfo Struct Reference

Data Fields

- BYTE callID
- BYTE OTASPStatus

8.1378.1 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)

8.1378.2 Field Documentation

8.1378.2.1 BYTE voiceOTASPStatusInfo::callID

8.1378.2.2 BYTE voiceOTASPStatusInfo::OTASPStatus

8.1379 voicePrivacyInfo Struct Reference

Data Fields

- BYTE callID
- BYTE voicePrivacy

8.1379.1 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

8.1379.2 Field Documentation

8.1379.2.1 BYTE voicePrivacyInfo::callID

8.1379.2.2 BYTE voicePrivacyInfo::voicePrivacy

8.1380 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

8.1380.1 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.

8.1380.2 Field Documentation

8.1380.2.1 arrCallInfo voiceSetAllCallStatusCbkInfo::arrCallInfomation

8.1380.2.2 arrAlertingPattern* voiceSetAllCallStatusCbkInfo::pArrAlertingPattern

8.1380.2.3 arrAlertingType* voiceSetAllCallStatusCbkInfo::pArrAlertingType

8.1380.2.4 arrAlphaID* voiceSetAllCallStatusCbkInfo::pArrAlphaID

8.1380.2.5 arrCalledPartyNum* voiceSetAllCallStatusCbkInfo::pArrCalledPartyNum

8.1380.2.6 arrCallEndReason* voiceSetAllCallStatusCbkInfo::pArrCallEndReason

8.1380.2.7 arrConnectPartyNum* voiceSetAllCallStatusCbkInfo::pArrConnectPartyNum

8.1380.2.8 arrDiagInfo* voiceSetAllCallStatusCbkInfo::pArrDiagInfo

8.1380.2.9 arrRedirPartyNum* voiceSetAllCallStatusCbkInfo::pArrRedirPartyNum

8.1380.2.10 arrRemotePartyName* voiceSetAllCallStatusCbkInfo::pArrRemotePartyName

8.1380.2.11 arrRemotePartyNum* voiceSetAllCallStatusCbkInfo::pArrRemotePartyNum

8.1380.2.12 arrSvcOption* voiceSetAllCallStatusCbkInfo::pArrSvcOption

8.1381 voiceSetCallBarringPwdInfo Struct Reference

Data Fields

- BYTE Reason
- BYTE oldPasswd [4]
- BYTE newPasswd [4]
- BYTE newPasswdAgain [4]

8.1381.1 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.

8.1381.2 Field Documentation

8.1381.2.1 BYTE voiceSetCallBarringPwdInfo::newPasswd[4]

8.1381.2.2 BYTE voiceSetCallBarringPwdInfo::newPasswdAgain[4]

8.1381.2.3 BYTE voiceSetCallBarringPwdInfo::oldPasswd[4]

8.1381.2.4 BYTE voiceSetCallBarringPwdInfo::Reason

8.1382 voiceSetCallBarringPwdResp Struct Reference

Data Fields

- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType

8.1382.1 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.

8.1382.2 Field Documentation

8.1382.2.1 alphaIDInfo* voiceSetCallBarringPwdResp::pAlphaIDInfo

8.1382.2.2 BYTE* voiceSetCallBarringPwdResp::pCallID

8.1382.2.3 BYTE* voiceSetCallBarringPwdResp::pCCResType

8.1382.2.4 ccSUPSType* voiceSetCallBarringPwdResp::pCCSUPSType

8.1382.2.5 WORD* voiceSetCallBarringPwdResp::pFailCause

8.1383 voiceSetConfigReq Struct Reference

Data Fields

- BYTE * pAutoAnswer

- airTimer * pAirTimerConfig
- roamTimer * pRoamTimerConfig
- BYTE * pTTYMode
- prefVoiceSO * pPrefVoiceSO
- BYTE * pPrefVoiceDomain

8.1383.1 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.

8.1383.2 Field Documentation

8.1383.2.1 airTimer* voiceSetConfigReq::pAirTimerConfig

8.1383.2.2 BYTE* voiceSetConfigReq::pAutoAnswer

8.1383.2.3 BYTE* voiceSetConfigReq::pPrefVoiceDomain

8.1383.2.4 prefVoiceSO* voiceSetConfigReq::pPrefVoiceSO

8.1383.2.5 roamTimer* voiceSetConfigReq::pRoamTimerConfig

8.1383.2.6 BYTE* voiceSetConfigReq::pTTYMode

8.1384 voiceSetConfigResp Struct Reference

Data Fields

- BYTE * pAutoAnsStatus
- BYTE * pAirTimerStatus
- BYTE * pRoamTimerStatus
- BYTE * pTTYConfigStatus
- BYTE * pPrefVoiceSOStatus
- BYTE * pVoiceDomainPrefStatus

8.1384.1 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.

8.1384.2 Field Documentation

8.1384.2.1 BYTE* voiceSetConfigResp::pAirTimerStatus

8.1384.2.2 BYTE* voiceSetConfigResp::pAutoAnsStatus

8.1384.2.3 BYTE* voiceSetConfigResp::pPrefVoiceSOStatus

8.1384.2.4 BYTE* voiceSetConfigResp::pRoamTimerStatus

8.1384.2.5 BYTE* voiceSetConfigResp::pTTYConfigStatus

8.1384.2.6 BYTE* voiceSetConfigResp::pVoiceDomainPrefStatus

8.1385 voiceSetPrefPrivacy Struct Reference

Data Fields

- BYTE privacyPref

8.1385.1 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

8.1385.2 Field Documentation

8.1385.2.1 BYTE voiceSetPrefPrivacy::privacyPref

8.1386 voiceSetSUPSServiceReq Struct Reference

Data Fields

- BYTE voiceSvc
- BYTE reason
- BYTE * pServiceClass
- BYTE * pCallBarringPasswd
- BYTE * pCallForwardingNumber
- BYTE * pTimerVal
- callFwdTypeAndPlan * pCallFwdTypeAndPlan

8.1386.1 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

8.1386.2 Field Documentation

8.1386.2.1 BYTE* voiceSetSUPSServiceReq::pCallBarringPasswd

8.1386.2.2 BYTE* voiceSetSUPSServiceReq::pCallForwardingNumber

8.1386.2.3 callFwdTypeAndPlan* voiceSetSUPSServiceReq::pCallFwdTypeAndPlan

8.1386.2.4 BYTE* voiceSetSUPSServiceReq::pServiceClass

8.1386.2.5 BYTE* voiceSetSUPSServiceReq::pTimerVal

8.1386.2.6 BYTE voiceSetSUPSServiceReq::reason

8.1386.2.7 BYTE voiceSetSUPSServiceReq::voiceSvc

8.1387 voiceSetSUPSServiceResp Struct Reference

Data Fields

- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResultType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType

8.1387.1 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

8.1387.2 Field Documentation

8.1387.2.1 alphaIDInfo* voiceSetSUPSServiceResp::pAlphaIDInfo

8.1387.2.2 BYTE* voiceSetSUPSServiceResp::pCallID

8.1387.2.3 BYTE* voiceSetSUPSServiceResp::pCCResultType

8.1387.2.4 ccSUPSType* voiceSetSUPSServiceResp::pCCSUPSType

8.1387.2.5 WORD* voiceSetSUPSServiceResp::pFailCause

8.1388 voiceStopContDTMFinfo Struct Reference

Data Fields

- BYTE callID

8.1388.1 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

8.1388.2 Field Documentation

8.1388.2.1 BYTE voiceStopContDTMFinfo::callID

8.1389 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

8.1389.1 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

8.1389.2 Field Documentation

8.1389.2.1 alphaIDInfo* voiceSUPSInfo::pAlphaIDInfo

8.1389.2.2 BYTE* voiceSUPSInfo::pCallBarPasswd

8.1389.2.3 getCallFWInfo* voiceSUPSInfo::pCallFwdInfo

8.1389.2.4 BYTE* voiceSUPSInfo::pCallFWNum

8.1389.2.5 BYTE* voiceSUPSInfo::pCallFWTimerVal

8.1389.2.6 BYTE* voiceSUPSInfo::pCallID

8.1389.2.7 CLIPResp* voiceSUPSInfo::pCLIPstatus

8.1389.2.8 CLIRResp* voiceSUPSInfo::pCLIRstatus

8.1389.2.9 CNAPResp* voiceSUPSInfo::pCNAPstatus

8.1389.2.10 COLPResp* voiceSUPSInfo::pCOLPstatus

8.1389.2.11 COLRResp* voiceSUPSInfo::pCOLRstatus

8.1389.2.12 BYTE* voiceSUPSInfo::pDataSrc

8.1389.2.13 WORD* voiceSUPSInfo::pFailCause

8.1389.2.14 newPwdData* voiceSUPSInfo::pNewPwdData

8.1389.2.15 BYTE* voiceSUPSInfo::pReason

8.1389.2.16 BYTE* voiceSUPSInfo::pSvcClass

8.1389.2.17 struct USSInfo* voiceSUPSInfo::pUSSInfo

8.1389.2.18 SUPSInfo voiceSUPSInfo::SUPSInformation

8.1390 voiceSUPSNotification Struct Reference

Data Fields

- BYTE callID
- BYTE notifType
- WORD * pCUGIndex
- ECTNum * pECTNum

8.1390.1 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

8.1390.2 Field Documentation

8.1390.2.1 BYTE voiceSUPSNotification::callID

8.1390.2.2 BYTE voiceSUPSNotification::notifType

8.1390.2.3 WORD* voiceSUPSNotification::pCUGIndex

8.1390.2.4 ECTNum* voiceSUPSNotification::pECTNum

8.1391 wcdmaCellInfo Struct Reference

Data Fields

- WORD psc
- SHORT cpich_rscp
- SHORT cpich_ecno
- SHORT srxlev

8.1391.1 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.

8.1391.2 Field Documentation

8.1391.2.1 SHORT wcdmaCellInfo::cpich_ecno

8.1391.2.2 SHORT wcdmaCellInfo::cpich_rscp

8.1391.2.3 WORD wcdmaCellInfo::psc

8.1391.2.4 SHORT wcdmaCellInfo::srxlev

8.1392 WCDMAECIOThresh Struct Reference

Data Fields

- BYTE WCDMAECIOThreshListLen
- WORD * pWCDMAECIOThreshList

8.1392.1 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)

8.1392.2 Field Documentation

8.1392.2.1 WORD* WCDMAECIOThresh::pWCDMAECIOThreshList

8.1392.2.2 BYTE WCDMAECIOThresh::WCDMAECIOThreshListLen

8.1393 WCDMAInfoLTENeighborCell Struct Reference

Data Fields

- ULONG wcdmaRRCState
- BYTE umtsLTENbrCellLen
- umtsLTENbrCell UMTSLTENbrCell [255]

8.1393.1 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.

8.1393.2 Field Documentation

8.1393.2.1 umtsLTENbrCell WCDMAInfoLTENeighborCell::UMTSLTENbrCell[255]

8.1393.2.2 BYTE WCDMAInfoLTENeighborCell::umtsLTENbrCellLen

8.1393.2.3 ULONG WCDMAInfoLTENeighborCell::wcdmaRRCState

8.1394 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

8.1394.1 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
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.

8.1394.2 Field Documentation

8.1394.2.1 BYTE wcdmaLongMsgDecodingParams::Date[0x09]

8.1394.2.2 BOOL* wcdmaLongMsgDecodingParams::pIsUDHPresent

8.1394.2.3 BYTE* wcdmaLongMsgDecodingParams::pMessage

8.1394.2.4 BYTE* wcdmaLongMsgDecodingParams::pPartNum

8.1394.2.5 BYTE* wcdmaLongMsgDecodingParams::pReferenceNum

8.1394.2.6 CHAR* wcdmaLongMsgDecodingParams::pScAddr

8.1394.2.7 BYTE* wcdmaLongMsgDecodingParams::pScAddrLength

8.1394.2.8 CHAR* wcdmaLongMsgDecodingParams::pSenderAddr

8.1394.2.9 BYTE* wcdmaLongMsgDecodingParams::pSenderAddrLength

8.1394.2.10 CHAR* wcdmaLongMsgDecodingParams::pTextMsg

8.1394.2.11 BYTE* wcdmaLongMsgDecodingParams::pTextMsgLength

8.1394.2.12 BYTE* wcdmaLongMsgDecodingParams::pTotalNum

8.1394.2.13 BYTE wcdmaLongMsgDecodingParams::Time[0x09]

8.1395 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]

8.1395.1 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

8.1395.2 Field Documentation

8.1395.2.1 BYTE wcdmaMsgDecodingParams::Date[0x09]

8.1395.2.2 BYTE* wcdmaMsgDecodingParams::pMessage

8.1395.2.3 CHAR* wcdmaMsgDecodingParams::pScAddr

8.1395.2.4 BYTE* wcdmaMsgDecodingParams::pScAddrLength

8.1395.2.5 CHAR* wcdmaMsgDecodingParams::pSenderAddr

8.1395.2.6 BYTE* wcdmaMsgDecodingParams::pSenderAddrLength

8.1395.2.7 CHAR* wcdmaMsgDecodingParams::pTextMsg

8.1395.2.8 BYTE* wcdmaMsgDecodingParams::pTextMsgLength

8.1395.2.9 BYTE wcdmaMsgDecodingParams::Time[0x09]

8.1396 wcdmaMsgEncodingParams Struct Reference

Data Fields

- ULONG messageSize
- CHAR * pDestAddr
- CHAR * pTextMsg
- CHAR * pPDUMessage
- BYTE alphabet

8.1396.1 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

8.1396.2 Field Documentation

8.1396.2.1 BYTE wcdmaMsgEncodingParams::alphabet

8.1396.2.2 ULONG wcdmaMsgEncodingParams::messageSize

8.1396.2.3 CHAR* wcdmaMsgEncodingParams::pDestAddr

8.1396.2.4 CHAR* wcdmaMsgEncodingParams::pPDUMessage

8.1396.2.5 CHAR* wcdmaMsgEncodingParams::pTextMsg

8.1397 WCDMARSSIThresh Struct Reference

Data Fields

- BYTE WCDMARSSIThreshListLen
- WORD * pWCDMARSSIThreshList

8.1397.1 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)

8.1397.2 Field Documentation

8.1397.2.1 WORD* WCDMARSSIThresh::pWCDMARSSIThreshList

8.1397.2.2 BYTE WCDMARSSIThresh::WCDMARSSIThreshListLen

8.1398 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

8.1398.1 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

8.1398.2 Field Documentation

8.1398.2.1 ULONG WCDMASysInfo::cellId

8.1398.2.2 BYTE WCDMASysInfo::cellIdValid

8.1398.2.3 BYTE WCDMASysInfo::hsCallStatus

8.1398.2.4 BYTE WCDMASysInfo::hsCallStatusValid

8.1398.2.5 BYTE WCDMASysInfo::hsInd

8.1398.2.6 BYTE WCDMASysInfo::hsIndValid

8.1398.2.7 WORD WCDMASysInfo::lac

8.1398.2.8 BYTE WCDMASysInfo::lacValid

8.1398.2.9 BYTE WCDMASysInfo::MCC[3]

8.1398.2.10 BYTE WCDMASysInfo::MNC[3]

8.1398.2.11 BYTE WCDMASysInfo::networkIdValid

8.1398.2.12 WORD WCDMASysInfo::psc

8.1398.2.13 BYTE WCDMASysInfo::pscValid

8.1398.2.14 BYTE WCDMASysInfo::regRejectInfoValid

8.1398.2.15 BYTE WCDMASysInfo::rejCause

8.1398.2.16 BYTE WCDMASysInfo::rejectSrvDomain

8.1398.2.17 sysInfoCommon WCDMASysInfo::sysInfoWCDMA

8.1399 wcdmaUARFCN Struct Reference

Data Fields

- BYTE status
- ULONG uarfcn

8.1399.1 Detailed Description

This structure contains the parameters for WCDMA UARFCN.

Parameters

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

8.1399.2 Field Documentation

8.1399.2.1 BYTE wcdmaUARFCN::status

8.1399.2.2 ULONG wcdmaUARFCN::uarfcn

8.1400 wds_currNetworkInfo Struct Reference

Data Fields

- uint8_t NetworkType
- uint32_t RATMask
- uint32_t SOMask

8.1400.1 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

8.1400.2 Field Documentation

8.1400.2.1 uint8_t wds_currNetworkInfo::NetworkType

8.1400.2.2 uint32_t wds_currNetworkInfo::RATMask

8.1400.2.3 uint32_t wds_currNetworkInfo::SOMask

8.1401 wds_DataULongLongTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint64_t ullData

8.1401.1 Detailed Description

This structure contains Data unsigned long long TLV data.

Parameters

TlvPresent	a flag indicates whether TLV presented or not
ullData	unsigend long long data in 64bit

8.1401.2 Field Documentation

8.1401.2.1 uint8_t wds_DataULongLongTlv::TlvPresent

8.1401.2.2 uint64_t wds_DataULongLongTlv::ullData

8.1402 wds_DataULongTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint32_t ulData

8.1402.1 Detailed Description

This structure contains Data unsigned long TLV data.

Parameters

TlvPresent	a flag indicates whether TLV presented or not
ulData	unsigend long data in 32bit

8.1402.2 Field Documentation

8.1402.2.1 uint8_t wds_DataULongTlv::TlvPresent

8.1402.2.2 uint32_t wds_DataULongTlv::ulData

8.1403 wds_DHCPLeaseOptTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint8_t numOpt
- wds_DHCPOpt optList [30]
- uint8_t optListData [2048]

8.1403.1 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

8.1403.2 Field Documentation

8.1403.2.1 uint8_t wds_DHCPLeaseOptTlv::numOpt

8.1403.2.2 wds_DHCPOpt wds_DHCPLeaseOptTlv::optList[30]

8.1403.2.3 uint8_t wds_DHCPLeaseOptTlv::optListData[2048]

8.1403.2.4 uint8_t wds_DHCPLeaseOptTlv::TlvPresent

8.1404 wds_DHCPLeaseStateTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint8_t leaseState

8.1404.1 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

8.1404.2 Field Documentation

8.1404.2.1 uint8_t wds_DHCPLeaseStateTlv::leaseState

8.1404.2.2 uint8_t wds_DHCPLeaseStateTlv::TlvPresent

8.1405 wds_DHCPOpt Struct Reference

Data Fields

- uint8_t optCode
- uint8_t optValLen
- uint8_t * pOptVal

8.1405.1 Detailed Description

DHCP option code information

Parameters

optCode	option code • values: – 0 - 255

optValLen	length of option code • values: – 0 - 255
pOptVal	option value

8.1405.2 Field Documentation

8.1405.2.1 uint8_t wds_DHCPOpt::optCode

8.1405.2.2 uint8_t wds_DHCPOpt::optValLen

8.1405.2.3 uint8_t* wds_DHCPOpt::pOptVal

8.1406 wds_DHCPProfileIdTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint8_t profileType
- uint8_t profileId

8.1406.1 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)

8.1406.2 Field Documentation

8.1406.2.1 uint8_t wds_DHCPProfileIdTlv::profileId

8.1406.2.2 uint8_t wds_DHCPProfileIdTlv::profileType

8.1406.2.3 uint8_t wds_DHCPProfileIdTlv::TlvPresent

8.1407 wds_DHCPv4HWConfig Struct Reference

Data Fields

- uint8_t hwType
- uint8_t chaddrLen
- uint8_t chaddr [16]

8.1407.1 Detailed Description

WDS SWI DHCPv4 HW Config Structure.

Parameters

hwType	• HW Type 1 - Ethernet 20 - Serial
chaddrlen	• chaddrlen
chaddr	• chaddr. Max size 16 bytes

8.1407.2 Field Documentation

8.1407.2.1 uint8_t wds_DHCPv4HWConfig::chaddr[16]

8.1407.2.2 uint8_t wds_DHCPv4HWConfig::chaddrLen

8.1407.2.3 uint8_t wds_DHCPv4HWConfig::hwType

8.1408 wds_DHCPv4Option Struct Reference

Data Fields

- uint8_t optCode
- uint8_t optValLen
- uint8_t optVal [255]

8.1408.1 Detailed Description

WDS SWI DHCPv4 Option Structure

Parameters

optCode	• Option code – 0 - 255
optValLen	• Option value length – 0 - 255
optVal	• Option value

8.1408.2 Field Documentation

8.1408.2.1 uint8_t wds_DHCPv4Option::optCode

8.1408.2.2 uint8_t wds_DHCPv4Option::optVal[255]

8.1408.2.3 uint8_t wds_DHCPv4Option::optValLen

8.1409 wds_DHCPv4OptionList Struct Reference

Data Fields

- uint8_t numOpt
- wds_DHCPv4Option * pOptList

8.1409.1 Detailed Description

WDS SWI DHCPv4 Option List Structure

Parameters

numOpt	• number of options – 0 - 255
pOptList	• pointer to list of DHCP Options

8.1409.2 Field Documentation

8.1409.2.1 uint8_t wds_DHCPv4OptionList::numOpt

8.1409.2.2 wds_DHCPv4Option* wds_DHCPv4OptionList::pOptList

8.1410 wds_DHCPv4ProfileId Struct Reference

Data Fields

- uint8_t profileType
- uint8_t profileId

8.1410.1 Detailed Description

WDS SWI DHCPv4 Profile Identifier Structure

Parameters

profileType	• 0 for 3GPP
profileId	• 1 to 24 for 3GPP profile

8.1410.2 Field Documentation

8.1410.2.1 uint8_t wds_DHCPv4ProfileId::profileId

8.1410.2.2 uint8_t wds_DHCPv4ProfileId::profileType

8.1411 wds_Domain Struct Reference

Data Fields

- uint16_t domainLen
- uint8_t domainName [256]

8.1411.1 Detailed Description

This structure contains the DomainName Information

Parameters

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

8.1411.2 Field Documentation

8.1411.2.1 uint16_t wds_Domain::domainLen

8.1411.2.2 uint8_t wds_Domain::domainName[256]

8.1412 wds_DomainNameList Struct Reference

Data Fields

- uint8_t numInstances
- struct wds_Domain domain [10]

8.1412.1 Detailed Description

This structure contains the DomainNameList Information

Parameters

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

Note

Data invalid when numInstances equal 0xff.

8.1412.2 Field Documentation

8.1412.2.1 struct wds_Domain wds_DomainNameList::domain[10]

8.1412.2.2 uint8_t wds_DomainNameList::numInstances

8.1413 wds_GPRSQoS Struct Reference

Data Fields

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

8.1413.1 Detailed Description

This structure contains the GPRS Quality Of Service Information

Parameters

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

Note

Data invalid when all parameters are equal to 0xffffffff.

8.1413.2 Field Documentation

8.1413.2.1 uint32_t wds_GPRSQoS::delayClass

8.1413.2.2 uint32_t wds_GPRSQoS::meanThroughputClass

8.1413.2.3 uint32_t wds_GPRSQoS::peakThroughputClass

8.1413.2.4 uint32_t wds_GPRSQoS::precedenceClass

8.1413.2.5 uint32_t wds_GPRSQoS::reliabilityClass

8.1414 wds_IPv4AdTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint32_t IPv4Addr

8.1414.1 Detailed Description

IPv4 address TLV information

Parameters

TlvPresent	a flag indicates whether TLV presented or not
IPv4Addr	IPv4 address

8.1414.2 Field Documentation

8.1414.2.1 uint32_t wds_IPv4AdTlv::IPv4Addr

8.1414.2.2 uint8_t wds_IPv4AdTlv::TlvPresent

8.1415 wds_IPV6AddressInfo Struct Reference

Data Fields

- uint8_t IPV6PrefixLen
- uint16_t IPAddressV6 [8]

8.1415.1 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.

8.1415.2 Field Documentation

8.1415.2.1 uint16_t wds_IPV6AddressInfo::IPAddressV6[8]

8.1415.2.2 uint8_t wds_IPV6AddressInfo::IPV6PrefixLen

8.1416 wds_IPV6GWAddressInfo Struct Reference

Data Fields

- uint8_t gwV6PrefixLen
- uint16_t gwAddressV6 [8]

8.1416.1 Detailed Description

This structure contains the IPV6 Gateway Address Information

Parameters

gwV6PrefixLen	• Length of the received IPv6 Gateway address in no. of bits; can take value between 0 and 128
IPAddressV6	• IPv6 Gateway address(in network byte order); This is an 8-element array of 16 bit numbers, each of which is in big endian format.

Note

Data invalid when IPV6PrefixLen equal 0xff.

8.1416.2 Field Documentation

8.1416.2.1 uint16_t wds_IPV6GWAddressInfo::gwAddressV6[8]

8.1416.2.2 uint8_t wds_IPV6GWAddressInfo::gwV6PrefixLen

8.1417 wds_PCSCFFQDNAddress Struct Reference

Data Fields

- uint16_t fqdnLen
- uint8_t fqdnAddr [256]

8.1417.1 Detailed Description

This structure contains the PCSCFFQDNAddress Information

Parameters

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

8.1417.2 Field Documentation

8.1417.2.1 uint8_t wds_PCSCFFQDNAddress::fqdnAddr[256]

8.1417.2.2 uint16_t wds_PCSCFFQDNAddress::fqdnLen

8.1418 wds_PCSCFFQDNAddressList Struct Reference

Data Fields

- uint8_t numInstances
- struct wds_PCSCFFQDNAddress pcsfFQDNAddress [10]

8.1418.1 Detailed Description

This structure contains the PCSCFFQDNAddressList Information

Parameters

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

Note

Data invalid when numInstances equal 0xff.

8.1418.2 Field Documentation

8.1418.2.1 uint8_t wds_PCSCFFQDNAddressList::numInstances

8.1418.2.2 struct wds_PCSCFFQDNAddress wds_PCSCFFQDNAddressList::pcsfFQDNAddress[10]

8.1419 wds_PCSCFIPv4ServerAddressList Struct Reference

Data Fields

- uint8_t numInstances
- uint32_t pscsfIPv4Addr [64]

8.1419.1 Detailed Description

This structure contains the PCSCFIPv4ServerAddressList Information

Parameters

numInstances	• number of address following
pscsfIPv4Addr	• P-CSCF IPv4 server addresses(Max 16 address, 4 bytes each)

Note

Data invalid when numInstances equal 0xff and pscsfIPv4Addr equal to 0xffffffff.

8.1419.2 Field Documentation

8.1419.2.1 uint8_t wds_PCSCFIPv4ServerAddressList::numInstances

8.1419.2.2 uint32_t wds_PCSCFIPv4ServerAddressList::pscsfIPv4Addr[64]

8.1420 wds_ProfileIdentifier Struct Reference

Data Fields

- uint8_t profileType
- uint8_t profileIndex

8.1420.1 Detailed Description

This structure contains the Profile Identifier Information

Parameters

profileType	• Identifies the type of profile 0x00 = 3GPP
profileIndex	• Index of profile whose settings were loaded prior to session parameter negotiation for the current call. If this TLV is not present, data call parameters are based on device default settings for each parameter

Note

Data invalid when all parameters are equal to 0xff.

8.1420.2 Field Documentation

8.1420.2.1 uint8_t wds_ProfileIdentifier::profileIndex

8.1420.2.2 uint8_t wds_ProfileIdentifier::profileType

8.1421 wds_profileInfo Union Reference

Data Fields

- LibPackprofile_3GPP SlqsProfile3GPP
- LibPackprofile_3GPP2 SlqsProfile3GPP2

8.1421.1 Detailed Description

This union consist of profile_3GPP and profile_3GPP2 out of which one will be used to create profile.

Parameters

SlqsProfile3GPP	3GPP profile See LibPackprofile_3GPP
SlqsProfile3GP-P2	3GPP2 profile See LibPackprofile_3GPP2

8.1421.2 Field Documentation

8.1421.2.1 LibPackprofile_3GPP wds_profileInfo::SlqsProfile3GPP

8.1421.2.2 LibPackprofile_3GPP2 wds_profileInfo::SlqsProfile3GPP2

8.1422 wds_transferStatInd Struct Reference

Data Fields

- uint8_t StatsPeriod
- uint32_t StatsMask

8.1422.1 Detailed Description

This structure contains the transfer statistic indication information

Parameters

StatsPeriod	Field Period between transfer statistic reports. • 0 - Do not report. • 1 - Other - Period between reports (in seconds).
StatsMask	requested statistic bit mask. • 0x00000040 - Tx bytes OK • 0x00000080 - Rx bytes OK

8.1422.2 Field Documentation

8.1422.2.1 uint32_t wds_transferStatInd::StatsMask

8.1422.2.2 uint8_t wds_transferStatInd::StatsPeriod

8.1423 wds_TrStatInd Struct Reference

Data Fields

- uint8_t statsPeriod
- uint32_t statsMask

8.1423.1 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.

8.1423.2 Field Documentation

8.1423.2.1 uint32_t wds_TrStatInd::statsMask

8.1423.2.2 uint8_t wds_TrStatInd::statsPeriod

8.1424 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

8.1424.1 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.

8.1424.2 Field Documentation

8.1424.2.1 uint8_t wds_UMTSMinQoS::deliveryErrSDU

8.1424.2.2 uint32_t wds_UMTSMinQoS::grntDownlinkBitrate

8.1424.2.3 uint32_t wds_UMTSMinQoS::grntUplinkBitrate

8.1424.2.4 uint32_t wds_UMTSMinQoS::maxDownlinkBitrate

8.1424.2.5 uint32_t wds_UMTSMinQoS::maxSDUSize

8.1424.2.6 uint32_t wds_UMTSMinQoS::maxUplinkBitrate

8.1424.2.7 uint8_t wds_UMTSMinQoS::qosDeliveryOrder

8.1424.2.8 uint8_t wds_UMTSMinQoS::resBerRatio

8.1424.2.9 uint8_t wds_UMTSMinQoS::sduErrorRatio

8.1424.2.10 uint8_t wds_UMTSMinQoS::trafficClass

8.1424.2.11 uint32_t wds_UMTSMinQoS::trafficPriority

8.1424.2.12 uint32_t wds_UMTSMinQoS::transferDelay

8.1425 WdsByteTotals Struct Reference

Data Fields

- ULONG * pV4sessionId
- ULONG * pV6sessionId
- struct WdsByteTotalsElmnts ByteTotalsElmntsV4
- struct WdsByteTotalsElmnts ByteTotalsElmntsV6

8.1425.1 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.

8.1425.2 Field Documentation

8.1425.2.1 struct WdsByteTotalsElmnts WdsByteTotals::ByteTotalsElmntsV4

8.1425.2.2 struct WdsByteTotalsElmnts WdsByteTotals::ByteTotalsElmntsV6

8.1425.2.3 ULONG* WdsByteTotals::pV4sessionId

8.1425.2.4 ULONG* WdsByteTotals::pV6sessionId

8.1426 WdsByteTotalsElmnts Struct Reference

Data Fields

- ULONGLONG * pTXTotalBytes
- ULONGLONG * pRXTotalBytes

8.1426.1 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.

8.1426.2 Field Documentation

8.1426.2.1 ULONGLONG* WdsByteTotalsElmnts::pRXTotalBytes

8.1426.2.2 ULONGLONG* WdsByteTotalsElmnts::pTXTotalBytes

8.1427 WdsClientLeaseChange Struct Reference

Data Fields

- BYTE * pEnableNotification

8.1427.1 Detailed Description

WDS SWI DHCPv4 Client Lease Change Structure

Parameters

pEnable-Notification	[IN] • Enable Notification or not

8.1427.2 Field Documentation

8.1427.2.1 BYTE* WdsClientLeaseChange::pEnableNotification

8.1428 WdsConnectionRate Struct Reference

Data Fields

- ULONG * pV4sessionId
- ULONG * pV6sessionId
- struct WdsConnectionRateElmnts ConnRateElmntsV4
- struct WdsConnectionRateElmnts ConnRateElmntsV6

8.1428.1 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.

8.1428.2 Field Documentation

8.1428.2.1 struct WdsConnectionRateElmnts WdsConnectionRate::ConnRateElmntsV4

8.1428.2.2 struct WdsConnectionRateElmnts WdsConnectionRate::ConnRateElmntsV6

8.1428.2.3 ULONG* WdsConnectionRate::pV4sessionId

8.1428.2.4 ULONG* WdsConnectionRate::pV6sessionId

8.1429 WdsConnectionRateElmnts Struct Reference

Data Fields

- ULONG * pCurrentChannelTXRate
- ULONG * pCurrentChannelRXRate
- ULONG * pMaxChannelTXRate
- ULONG * pMaxChannelRXRate

8.1429.1 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

8.1429.2 Field Documentation

8.1429.2.1 ULONG* WdsConnectionRateElmnts::pCurrentChannelRXRate

8.1429.2.2 ULONG* WdsConnectionRateElmnts::pCurrentChannelTXRate

8.1429.2.3 ULONG* WdsConnectionRateElmnts::pMaxChannelRXRate

8.1429.2.4 ULONG* WdsConnectionRateElmnts::pMaxChannelTXRate

8.1430 WdsDHCPv4ClientLeaseInd Struct Reference

Data Fields

- WdsDHCPv4ProfileId * pProfileId
- BYTE * pLeaseState
- ULONG * pIPv4Addr
- DHCPOptionList * pOptList

8.1430.1 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

8.1430.2 Field Documentation

8.1430.2.1 ULONG* WdsDHCPv4ClientLeaseInd::pIPv4Addr

8.1430.2.2 BYTE* WdsDHCPv4ClientLeaseInd::pLeaseState

8.1430.2.3 DHCPOptionList* WdsDHCPv4ClientLeaseInd::pOptList

8.1430.2.4 WdsDHCPv4ProfileId* WdsDHCPv4ClientLeaseInd::pProfileId

8.1431 WdsDHCPv4Config Struct Reference

Data Fields

- WdsDHCPv4ProfileId * pProfileId
- WdsDHCPv4HWConfig * pHwConfig
- WdsDHCPv4OptionList * pRequestOptionList

8.1431.1 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

8.1431.2 Field Documentation

8.1431.2.1 WdsDHCPv4HWConfig* WdsDHCPv4Config::pHwConfig

8.1431.2.2 WdsDHCPv4ProfileId* WdsDHCPv4Config::pProfileId

8.1431.2.3 WdsDHCPv4OptionList* WdsDHCPv4Config::pRequestOptionList

8.1432 WdsDHCPv4HWConfig Struct Reference

Data Fields

- BYTE hwType
- BYTE chaddrLen
- BYTE chaddr [16]

8.1432.1 Detailed Description

WDS SWI DHCPv4 HW Config Structure.

Parameters

hwType	• HW Type 1 - Ethernet 20 - Serial

chaddrlen	• chaddrlen
chaddr	• chaddr. Max size 16 bytes

8.1432.2 Field Documentation

8.1432.2.1 BYTE WdsDHCPv4HWConfig::chaddr[16]

8.1432.2.2 BYTE WdsDHCPv4HWConfig::chaddrLen

8.1432.2.3 BYTE WdsDHCPv4HWConfig::hwType

8.1433 wdsDhcpv4HwConfig Struct Reference

Data Fields

- uint8_t hwType
- uint8_t chaddrLen
- uint8_t chaddr [16]

8.1433.1 Detailed Description

Structure contain DHCP V4 Hardware Configuration.

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

8.1433.2 Field Documentation

8.1433.2.1 uint8_t wdsDhcpv4HwConfig::chaddr[16]

8.1433.2.2 uint8_t wdsDhcpv4HwConfig::chaddrLen

8.1433.2.3 uint8_t wdsDhcpv4HwConfig::hwType

8.1434 wdsDhcpv4Option Struct Reference

Data Fields

- uint8_t optCode

- uint8_t optValLen
- uint8_t optVal [255]

8.1434.1 Detailed Description

This structure contains DHCP V4 Option information.

Parameters

optCode	Option code • 0 - 255
optValLen	Option value length • 0 - 255
optVal	Option Value

8.1434.2 Field Documentation

8.1434.2.1 uint8_t wdsDhcpv4Option::optCode

8.1434.2.2 uint8_t wdsDhcpv4Option::optVal[255]

8.1434.2.3 uint8_t wdsDhcpv4Option::optValLen

8.1435 WdsDHCPv4Option Struct Reference

Data Fields

- BYTE optCode
- BYTE optValLen
- BYTE optVal [255]

8.1435.1 Detailed Description

WDS SWI DHCPv4 Option Structure

Parameters

optCode	• Option code – 0 - 255
optValLen	• Option value length – 0 - 255
optVal	• Option value

8.1435.2 Field Documentation

8.1435.2.1 BYTE WdsDHCPv4Option::optCode

8.1435.2.2 BYTE WdsDHCPv4Option::optVal[255]

8.1435.2.3 BYTE WdsDHCPv4Option::optValLen

8.1436 wdsDhcpv4OptionList Struct Reference

Data Fields

- uint8_t numOpt
- wdsDhcpv4Option * pOptList

8.1436.1 Detailed Description

Structure contain DHCP V4 Option List

Parameters

numOpt	number of options • 0 - 255
pOptList	pointer to list of DHCP Options • See wdsDhcpv4Option

8.1436.2 Field Documentation

8.1436.2.1 uint8_t wdsDhcpv4OptionList::numOpt

8.1436.2.2 wdsDhcpv4Option* wdsDhcpv4OptionList::pOptList

8.1437 WdsDHCPv4OptionList Struct Reference

Data Fields

- BYTE numOpt
- WdsDHCPv4Option * pOptList

8.1437.1 Detailed Description

WDS SWI DHCPv4 Option List Structure

Parameters

numOpt	• number of options – 0 - 255
pOptList	• pointer to list of DHCP Options

8.1437.2 Field Documentation

8.1437.2.1 BYTE WdsDHCPv4OptionList::numOpt

8.1437.2.2 WdsDHCPv4Option* WdsDHCPv4OptionList::pOptList

8.1438 WdsDHCPv4ProfileId Struct Reference

Data Fields

- BYTE profileType
- BYTE profileId

8.1438.1 Detailed Description

WDS SWI DHCPv4 Profile Identifier Structure

Parameters

profileType	• 0 for 3GPP
profileId	• 1 to 24 for 3GPP profile

8.1438.2 Field Documentation

8.1438.2.1 BYTE WdsDHCPv4ProfileId::profileId

8.1438.2.2 BYTE WdsDHCPv4ProfileId::profileType

8.1439 wdsDhcpv4ProfileId Struct Reference

Data Fields

- uint8_t profileType
- uint8_t profileId

8.1439.1 Detailed Description

This structure contains DHCP v4 ProfieIID Information.

Parameters

profileType	profile type • 0 - 3GPP
profileId	profile index • index identifiying the profile 1-24 valid for 3GPP profile type (EM74xx and onwards)

8.1439.2 Field Documentation

8.1439.2.1 uint8_t wdsDhcpv4ProfileId::profileId

8.1439.2.2 uint8_t wdsDhcpv4ProfileId::profileType

8.1440 WDSGetLoopbackData Struct Reference

Data Fields

- BYTE ByteLoopbackMode
- BYTE ByteLoopbackMultiplier

8.1440.1 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.

8.1440.2 Field Documentation

8.1440.2.1 BYTE WDSGetLoopbackData::ByteLoopbackMode

8.1440.2.2 BYTE WDSGetLoopbackData::ByteLoopbackMultiplier

8.1441 WdsIpAddressInfoReq Struct Reference

Data Fields

- ULONG * pv4sessionId
- ULONG * pv6sessionId
- QmiWdsIpAddressInfo ip

8.1441.1 Detailed Description

Parameters

pv4sessionId[OUT]	• IPv4 session id
pv6sessionId[OUT]	• IPv6 session id
ip[OUT]	• IP address information see QmiWdsIpAddressInfo for more details

8.1441.2 Field Documentation

8.1441.2.1 QmiWdsIpAddressInfo WdsIpAddressInfoReq::ip

8.1441.2.2 ULONG* WdsIpAddressInfoReq::pv4sessionId

8.1441.2.3 ULONG* WdsIpAddressInfoReq::pv6sessionId

8.1442 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

8.1442.1 Detailed Description

WDS Pkt Statistics request data structure for individual session

Parameters

pTXPacket-Successes	• Number of transmitted Packets without error.
pRXPacket-Successes	• Number of received Packets without error.
pTXPacketErrors	• Number of outgoing packets with framing errors.
pRXPacket-Errors	• Number of incoming packets with framing errors.
pTXPacket-Overflows	• Number of packets dropped because Tx buffer overflowed (out of memory).
pRXPacket-Overflows	• Number of packets dropped because Rx buffer overflowed (out of memory).
pTXOkBytes-Count	• Number of bytes transmitted without error.
pRXOkBytes-Count	• Number of bytes received without error.
pTXOKBytes-LastCall	• Number of bytes transmitted without error during the last data call (0 if no call was made earlier). Returned only if not in a call, and when the previous call was made using RmNet (for any devices that support
pRXOKBytes-LastCall	• Number of bytes received without error during the last data call (0 if no call was made earlier). Returned only if not in a call, and when the previous call was made using RmNet (for any devices that support
pTXDropped-Count	• Number of outgoing packets dropped.
pRXDropped-Count	• Number of incoming packets dropped.

8.1442.2 Field Documentation

8.1442.2.1 ULONG* WdsPktStatisticsElmnts::pRXDroppedCount

8.1442.2.2 ULONGLONG* WdsPktStatisticsElmnts::pRXOkBytesCount

8.1442.2.3 ULONGLONG* WdsPktStatisticsElmnts::pRXOKBytesLastCall

8.1442.2.4 ULONG* WdsPktStatisticsElmnts::pRXPacketErrors

8.1442.2.5 ULONG* WdsPktStatisticsElmnts::pRXPacketOverflows

8.1442.2.6 ULONG* WdsPktStatisticsElmnts::pRXPacketSuccesses

8.1442.2.7 ULONG* WdsPktStatisticsElmnts::pTXDroppedCount

8.1442.2.8 ULONGLONG* WdsPktStatisticsElmnts::pTXOkBytesCount

8.1442.2.9 ULONGLONG* WdsPktStatisticsElmnts::pTXOKBytesLastCall

8.1442.2.10 ULONG* WdsPktStatisticsElmnts::pTXPacketErrors

8.1442.2.11 ULONG* WdsPktStatisticsElmnts::pTXPacketOverflows

8.1442.2.12 ULONG* WdsPktStatisticsElmnts::pTXPacketSuccesses

8.1443 WdsPktStatisticsReq Struct Reference

Data Fields

- ULONG * pStatMask

8.1443.1 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

8.1443.2 Field Documentation

8.1443.2.1 ULONG* WdsPktStatisticsReq::pStatMask

8.1444 WdsPktStatisticsResp Struct Reference

Data Fields

- ULONG * pV4sessionId
- ULONG * pV6sessionId
- struct WdsPktStatisticsElmnts PktStatElmntsV4
- struct WdsPktStatisticsElmnts PktStatElmntsV6

8.1444.1 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.

8.1444.2 Field Documentation

8.1444.2.1 struct WdsPktStatisticsElmnts WdsPktStatisticsResp::PktStatElmntsV4

8.1444.2.2 struct WdsPktStatisticsElmnts WdsPktStatisticsResp::PktStatElmntsV6

8.1444.2.3 ULONG* WdsPktStatisticsResp::pV4sessionId

8.1444.2.4 ULONG* WdsPktStatisticsResp::pV6sessionId

8.1445 WdsProfileParam Union Reference

Data Fields

- struct Profile3GPP SlqsProfile3GPP
- struct Profile3GPP2 SlqsProfile3GPP2

8.1445.1 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

8.1445.2 Field Documentation

8.1445.2.1 struct Profile3GPP WdsProfileParam::SlqsProfile3GPP

8.1445.2.2 struct Profile3GPP2 WdsProfileParam::SlqsProfile3GPP2

8.1446 WdsRunTimeSettings Struct Reference

Data Fields

- ULONG * v4sessionId
- ULONG * v6sessionId
- struct qmiWdsRunTimeSettings rts

8.1446.1 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.

8.1446.2 Field Documentation

8.1446.2.1 struct qmiWdsRunTimeSettings WdsRunTimeSettings::rts

8.1446.2.2 ULONG* WdsRunTimeSettings::v4sessionId

8.1446.2.3 ULONG* WdsRunTimeSettings::v6sessionId

8.1447 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

8.1447.1 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

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.

8.1447.2 Field Documentation

8.1447.2.1 BYTE* wdsSetEventReportReq::pCurrChannelRateInd

8.1447.2.2 BYTE* wdsSetEventReportReq::pCurrDataBearerTechInd

8.1447.2.3 BYTE* wdsSetEventReportReq::pCurrPrefDataSysInd

8.1447.2.4 BYTE* wdsSetEventReportReq::pDataBearerTechInd

8.1447.2.5 BYTE* wdsSetEventReportReq::pDataCallStatusChangeInd

8.1447.2.6 BYTE* wdsSetEventReportReq::pDataSystemStatusChangeInd

8.1447.2.7 BYTE* wdsSetEventReportReq::pDormancyStatusInd

8.1447.2.8 BYTE* wdsSetEventReportReq::pEVDOPageMonPerChangeInd

8.1447.2.9 BYTE* wdsSetEventReportReq::pMIPStatusInd

8.1447.2.10 TrStatInd* wdsSetEventReportReq::pTransferStatInd

8.1448 WDSSetLoopbackData Struct Reference

Data Fields

- BYTE * pLoopbackMode
- BYTE * pLoopbackMultiplier

8.1448.1 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.

8.1448.2 Field Documentation

8.1448.2.1 BYTE* WDSSetLoopbackData::pLoopbackMode

8.1448.2.2 BYTE* WDSSetLoopbackData::pLoopbackMultiplier

8.1449 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

8.1449.1 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.

8.1449.2 Field Documentation

8.1449.2.1 unsigned long WDSSWICurrentChannelRates::current_channel_rx_rate

8.1449.2.2 unsigned long WDSSWICurrentChannelRates::current_channel_tx_rate

8.1449.2.3 unsigned long WDSSWICurrentChannelRates::max_channel_rx_rate

8.1449.2.4 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,
  eIMS =0x12,
  eTMD =0x18,
  eIMSA =0x21,
  eRMS =225,
  eSWIOMA =240,
  eAUDIO =241,
  eSWIM2MCMD =243,
  eSWIOMAEXT =244,
  eSWIAUDIO = 245,
  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
eIMS
eTMD
eIMSA
eRMS
eSWIOMA
eAUDIO
eSWIM2MCMD
eSWIOMAEXT
eSWIAUDIO
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_ResetToFactory-Defaults_t * *pOutput*)

Requests the device to perform automatic service activation unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.64 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_UIMSetPINProtection_t * *pOutput*)

To get UIM Set PIN protection unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.123 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*)

Pack get server.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

Note

The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through unpack_loc_GetServer_Ind callback If successful, the callback also contains the A-GPS server address.

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.7 int pack_loc_SLQSLOCInjectPosition (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_SLQSLOCInjectPosition_t * *reqArg*)

Pack inject position.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.8 int pack_loc_SLQSLOCInjectSensorData (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_SLQSLOCInjectSensorData_t * *reqArg*)

Pack inject sensor data.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.9 int pack_loc_SLQSLOCInjectUTCTime (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_SLQSLOCInjectUTCTime_t * *reqArg*)

Pack inject UTC time.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.10 int pack_loc_SLQSLOCSetCradleMountConfig (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_SLQSLOCSetCradleMountConfig_t * *reqArg*)

Pack set cradle mount configure.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.11 int pack_loc_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*)

Unpack get server.

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.28 int unpack_loc_SLQSLOCInjectPosition (uint8_t * *pResp*, uint16_t *respLen*)

Unpack inject position.

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.29 int unpack_loc_SLQSLOCInjectSensorData (uint8_t * *pResp*, uint16_t *respLen*)

Unpack inject sensor data.

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.30 int unpack_loc_SLQSLOCInjectUTCTime (uint8_t * *pResp*, uint16_t *respLen*)

Unpack inject UTC time.

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.31 int unpack_loc_SLQSLOCSetCradleMountConfig (uint8_t * *pResp*, uint16_t *respLen*)

Unpack set cradle mount configure.

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.32 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_valid_nas_SLQSNasGetSigInfo (uint8_t *pResp, uint8_t u8Info)
- int unpack_nas_SLQSNasSigInfoCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasSigInfoCallback_ind_t *pOutput)
- int unpack_nas_GetHomeNetwork (uint8_t *pResp, uint16_t respLen, unpack_nas_GetHomeNetwork_t *pOutput)
- int pack_nas_GetHomeNetwork (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int pack_nas_SLQSGetSysInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSGetSysInfo (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSGetSysInfo_t *pOutput)

- int unpack_nas_SLQSNasSysInfoCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSSysInfoCallback_ind_t *pOutput)
- int pack_nas_GetServingNetwork (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_GetServingNetwork (uint8_t *pResp, uint16_t respLen, unpack_nas_GetServingNetwork_t *pOutput)
- int pack_nas_GetServingNetworkCapabilities (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_GetServingNetworkCapabilities (uint8_t *pResp, uint16_t respLen, unpack_nas_GetServingNetworkCapabilities_t *pOutput)
- int pack_nas_PerformNetworkScan (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_PerformNetworkScan (uint8_t *pResp, uint16_t respLen, unpack_nas_PerformNetworkScan_t *pOutput)
- int pack_nas_SLQSSwiGetLteCQI (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSSwiGetLteCQI (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSSwiGetLteCQI_t *pOutput)
- int pack_nas_SLQSNasSwiModemStatus (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSNasSwiModemStatus (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasSwiModemStatus_t *pOutput)
- int pack_nas_SLQSGetServingSystem (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSGetServingSystem (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSGetServingSystem_t *pOutput)
- int unpack_valid_nas_SLQSGetServingSystem (uint8_t *pResp, uint8_t u8Info)
- int pack_nas_SLQSGetSignalStrength (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, uint16_t reqMask)
- int unpack_nas_SLQSGetSignalStrength (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSGetSignalStrength_t *pOutput)
- int unpack_valid_nas_SLQSGetSignalStrength (uint8_t *pResp, uint8_t u8Info)
- int pack_nas_SLQSSetSignalStrengthsCallback (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSSetSignalStrengthsCallback_t *pReqParam)
- int unpack_nas_SLQSSetSignalStrengthsCallback (uint8_t *pResp, uint16_t respLen)
- int pack_nas_SetRFInfoCallback (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, uint8_t *pBenable)
- int unpack_nas_SetRFInfoCallback (uint8_t *pResp, uint16_t respLen)
- int pack_nas_SetLURejectCallback (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, uint8_t *pBenable)
- int unpack_nas_SetLURejectCallback (uint8_t *pResp, uint16_t respLen)
- int unpack_nas_SetEventReportInd (uint8_t *pResp, uint16_t respLen, unpack_nas_SetEventReportInd_t *pOutput)
- int pack_nas_GetCDMANetworkParameters (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_GetCDMANetworkParameters (uint8_t *pResp, uint16_t respLen, unpack_nas_GetCDMANetworkParameters_t *pOutput)
- int unpack_valid_nas_GetCDMANetworkParameters (uint8_t *pResp, uint8_t u8Info)
- int pack_nas_GetANAAAAuthenticationStatus (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_GetANAAAAuthenticationStatus (uint8_t *pResp, uint16_t respLen, uint32_t *pAuthStatus)
- int pack_nas_GetACCOLC (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_GetACCOLC (uint8_t *pResp, uint16_t respLen, uint8_t *pAccolc)
- int pack_nas_SetACCOLC (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SetACCOLC_t reqParam)
- int unpack_nas_SetACCOLC (uint8_t *pResp, uint16_t respLen)
- int pack_nas_SLQSNasConfigSigInfo2 (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSNasConfigSigInfo2_t *pReqParam)
- int unpack_nas_SLQSNasConfigSigInfo2 (uint8_t *pResp, uint16_t respLen)
- int unpack_nas_SetDataCapabilitiesCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SetDataCapabilitiesCallback_ind_t *pOutput)
- int pack_nas_GetNetworkPreference (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_GetNetworkPreference (uint8_t *pResp, uint16_t respLen, unpack_nas_GetNetworkPreference_t *pOutput)

- int pack_nas_SetNetworkPreference (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SetNetworkPreference_t *reqArg)
- int unpack_nas_SetNetworkPreference (uint8_t *pResp, uint16_t respLen, unpack_nas_SetNetworkPreference_t *pOutput)
- int unpack_nas_SetRoamingIndicatorCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SetRoamingIndicatorCallback_ind_t *pOutput)
- int unpack_nas_SetServingSystemCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SetServingSystemCallback_ind_t *pOutput)
- int pack_nas_SlqsGetLTECphyCAInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SlqsGetLTECphyCAInfo (uint8_t *pResp, uint16_t respLen, unpack_nas_SlqsGetLTECphyCAInfo_t *pOutput)
- int unpack_nas_SLQSSetSysSelectionPrefCallBack_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t *pOutput)
- int unpack_nas_SLQSNasSwiOTAMessageCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t *pOutput)
- int pack_nas_SLQSInitiateNetworkRegistration (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSInitiateNetworkRegistration_t *pReqParam)
- int unpack_nas_SLQSInitiateNetworkRegistration (uint8_t *pResp, uint16_t respLen)
- int pack_nas_SLQSNasSwiIndicationRegister (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSNasSwiIndicationRegister_t *pReqParam)
- int unpack_nas_SLQSNasSwiIndicationRegister (uint8_t *pResp, uint16_t respLen)
- int pack_nas_SLQSGetPLMNName (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSGetPLMNName_t *reqArg)
- int unpack_nas_SLQSGetPLMNName (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSGetPLMNName_t *pOutput)
- int pack_nas_SLQSNasGetCellLocationInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSNasGetCellLocationInfo (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasGetCellLocationInfo_t *pOutput)
- int pack_nas_SLQSGetNetworkTime (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSGetNetworkTime (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSGetNetworkTime_t *pOutput)
- int unpack_nas_SLQSNasNetworkTimeCallBack_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasNetworkTimeCallBack_ind_t *pOutput)
- int unpack_nas_SetNasLTECphyCaIndCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SetNasLTECphyCaIndCallback_ind_t *pOutput)
- int pack_nas_SLQSSwiGetLteSccRxInfo (pack_qmi_t *pCtx, uint8_t *pReq, uint16_t *pLen)
- int unpack_nas_SLQSSwiGetLteSccRxInfo (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSSwiGetLteSccRxInfo_t *pOutput)
- int unpack_nas_SLQSNasTimerCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasTimerCallback_ind_t *pOutput)
- int pack_nas_InitiateDomainAttach (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_InitiateDomainAttach_t *pReqParam)
- int unpack_nas_InitiateDomainAttach (uint8_t *pResp, uint16_t respLen)
- int pack_nas_SetCDMANetworkParameters (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SetCDMANetworkParameters_t *pReqParam)
- int unpack_nas_SetCDMANetworkParameters (uint8_t *pResp, uint16_t respLen)
- int pack_nas_SLQSNasGetHDRColorCode (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSNasGetHDRColorCode (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasGetHDRColorCode_t *pOutput)
- int pack_nas_SLQSNasGetTxRxInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSNasGetTxRxInfo_t *reqArg)
- int unpack_nas_SLQSNasGetTxRxInfo (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasGetTxRxInfo_t *pOutput)
- int pack_nas_SLQSGetOperatorNameData (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSGetOperatorNameData (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSGetOperatorNameData_t *pOutput)

- int pack_nas_SLQSNasGet3GPP2Subscription (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSNasGet3GPP2Subscription_t *reqArg)
- int unpack_nas_SLQSNasGet3GPP2Subscription (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasGet3GPP2Subscription_t *pOutput)
- int pack_nas_SLQSSwiGetHDRPersonality (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSSwiGetHDRPersonality (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSSwiGetHDRPersonality_t *pOutput)
- int pack_nas_SLQSSwiGetHDRProtSubtype (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSSwiGetHDRProtSubtype (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSSwiGetHDRProtSubtype_t *pOutput)
- int pack_nas_SLQSSwiPSDetach (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSSwiPSDetach_t *pReqParam)
- int unpack_nas_SLQSSwiPSDetach (uint8_t *pResp, uint16_t respLen)
- int pack_nas_SLQSGetErrorRate (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSGetErrorRate (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSGetErrorRate_t *pOutput)
- int pack_nas_SLQSSwiGetHRPDStats (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSSwiGetHRPDStats (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSSwiGetHRPDStats_t *pOutput)
- int pack_nas_SLQSSwiNetworkDebug (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSSwiNetworkDebug (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSSwiNetworkDebug_t *pOutput)
- int pack_nas_SLQSConfigSigInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSConfigSigInfo_t *pReqParam)
- int unpack_nas_SLQSConfigSigInfo (uint8_t *pResp, uint16_t respLen)
- int pack_nas_GetHomeNetwork3GPP2 (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_GetHomeNetwork3GPP2 (uint8_t *pResp, uint16_t respLen, unpack_nas_GetHomeNetwork3GPP2_t *pOutput)
- int pack_nas_SLQSNASSwiSetChannelLock (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSNASSwiSetChannelLock_t *pReqParam)
- int unpack_nas_SLQSNASSwiSetChannelLock (uint8_t *pResp, uint16_t respLen)
- int pack_nas_SLQSNASSwiGetChannelLock (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSNASSwiGetChannelLock (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNASSwiGetChannelLock_t *pOutput)

9.6.1 Macro Definition Documentation

9.6.1.1 #define NAS_IMSI_M_S1_LENGTH 7

9.6.1.2 #define NAS_IMSI_M_S2_LENGTH 3

9.6.1.3 #define NAS_MAX_DESCRIPTION_LENGTH 255

9.6.1.4 #define NAS_MAX_NUM_NETWORKS 30

9.6.1.5 #define NAS_MAX_SCC_RX_INFO_INSTANCES 255

9.6.1.6 #define NAS_NAM_NAME_LENGTH 12

9.6.1.7 #define NAS_OTA_MESSAGE_MAX_BUF_SIZE 2048

9.6.1.8 #define NAS_PLMN_LENGTH 3

9.6.1.9 #define NAS_SERVING_SYSTEM_INFO_MAX_RADIO_INTERFACE_LIST 255

9.6.2 Enumeration Type Documentation

9.6.2.1 enum LITEQMI_NAS_LTE_CPHY_CA_BW_NRB

Enumerator

eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_6

eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_15

eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_25

eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_50

eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_75

eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_100

9.6.2.2 enum LITEQMI_NAS_LTE_CPHY_SCELL_STATE

Enumerator

eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_DECONFIGURED

eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED

eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED

9.6.2.3 enum NAS_LTE_CPHY_CA_BW_NRB_LITE

Enumerator

eNAS_LTE_CPHY_CA_BW_NRB_LITE_6

eNAS_LTE_CPHY_CA_BW_NRB_LITE_15

eNAS_LTE_CPHY_CA_BW_NRB_LITE_25

eNAS_LTE_CPHY_CA_BW_NRB_LITE_50

eNAS_LTE_CPHY_CA_BW_NRB_LITE_75

eNAS_LTE_CPHY_CA_BW_NRB_LITE_100

9.6.2.4 enum NAS_LTE_CPHY_SCELL_STATE_LITE

Enumerator

eNAS_LTE_CPHY_SCELL_STATE_DECONFIGURED_LITE

eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED_LITE

eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED_LITE

9.6.3 Function Documentation

9.6.3.1 int pack_nas_GetACCOLC (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Retrieves information about the access overload class (ACCOLC) pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.2 int pack_nas_GetANAAAAuthenticationStatus (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

AN-AAA authentication status of the device pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.3 int pack_nas_GetCDMANetworkParameters (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Get CDMA Network Parameters pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.4 int pack_nas_GetHomeNetwork (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

get home network pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.5 int pack_nas_GetHomeNetwork3GPP2 (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Get Home Network 3GPP2 Parameters pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.6 int pack_nas_GetNetworkPreference (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Gets the network registration preference pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

9.6.3.7 int pack_nas_GetRFInfo (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

get rf info pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.8 int pack_nas_GetServingNetwork (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)

Pack provides information about the system that provides service to the device.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.9 int pack_nas_GetServingNetworkCapabilities (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)

Pack returns information regarding the data capabilities of the system that currently provides service to the device.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.10 int pack_nas_GetSignalStrengths (pack_qmi_t * pCtx, uint8_t * pReq, uint16_t * pLen)

get signal strengths pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.11 int pack_nas_InitiateDomainAttach (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_InitiateDomainAttach_t * *pReqParam*)

Initiates a domain attach/detach of the device pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request Parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.12 int pack_nas_PerformNetworkScan (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Pack performs scan for available networks and scans for RAT info as well.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.13 int pack_nas_SetACCOLC (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SetACCOLC_t *reqParam*)

Pack Sets the access overload class (ACCOLC)

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqParam*	request Parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.14 int pack_nas_SetCDMANetworkParameters (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SetCDMANetworkParameters_t * *pReqParam*)

CDMA Network Parameters pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request Parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.15 int pack_nas_SetLURejectCallback (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, uint8_t * *pBenable*)

Pack set registration reject Callback

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pBenable*	0/1 value to disable/enable indication respectively

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.16 int pack_nas_SetNetworkPreference (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SetNetworkPreference_t * *reqArg*)

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

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

9.6.3.17 int pack_nas_SetRFInfoCallback (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, uint8_t * pBenable)

Pack Set RF Band Information callback.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pBenable*	0/1 value to disable/enable indication respectively

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.18 int pack_nas_SLQSConfigSigInfo (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSConfigSigInfo_t * pReqParam)

Config Sig Info Parameters pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request prarmeters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.19 int pack_nas_SLQSGetErrorRate (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)

Get Error Rate Parameters pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.20 int pack_nas_SlqsGetLTECphyCAInfo (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Pack get carrier aggregation event information.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

9.6.3.21 int pack_nas_SLQSGetNetworkTime (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Pack Get Network Time.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.22 int pack_nas_SLQSGetOperatorNameData (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Operator Name Data Parameters pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.23 int pack_nas_SLQSGetPLMNName (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SLQSGetPLMNName_t * *reqArg*)

Pack get operator name for specificed network.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request prarmeters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.24 int pack_nas_SLQSGetServingSystem (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Pack queries information regarding the system that currently provides service.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.25 int pack_nas_SLQSGetSignalStrength (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, uint16_t *reqMask*)

Queries the current signal strength as measured by the device. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use 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)

Pack get system information.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.27 int pack_nas_SLQSGetSysSelectionPref (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)

Queries the different system selection preferences of the device pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.28 int pack_nas_SLQSInitiateNetworkRegistration (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSInitiateNetworkRegistration_t * pReqParam)

Pack initiates a network registration.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request Parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.29 int pack_nas_SLQSNasConfigSigInfo2 (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSNasConfigSigInfo2_t * pReqParam)

Pack sets the signal strength reporting thresholds

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request Parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.30 int pack_nas_SLQSNasGet3GPP2Subscription (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSNasGet3GPP2Subscription_t * reqArg)

3GPP2 Subscription Parameters pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request prarmeters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.31 int pack_nas_SLQSNasGetCellLocationInfo (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)

Pack retrieves cell location-related information.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.32 int pack_nas_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*)

Pack used to Registers/De-registers for different NAS (Network access service) indications.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request Parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

Note

This function is used by a control point to register/deregister for different QMI_NAS indications. The control point's registration state variables, controlling registration for indications, are modified to reflect the settings indicated in the parameters that are present in the request message. At least one optional parameter must be present in the request.

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.36 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)

Pack sets the registration state for different QMI_NAS SWI indications.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request prarmeters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.38 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)

Pack sets the different system selection preferences of the device.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request Parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.41 int pack_nas_SLQSSetSignalStrengthsCallback (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSSetSignalStrengthsCallback_t * pReqParam)

Pack set signal strength thresholds callback.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request prarmeters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.42 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*)

Pack fetch CQI parameters for LTE data session.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.47 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*	An 8-bit integer representation of the ACCOLC. • Range: 0 to 15 (0x00 to 0x0F). • 0xFF invalid data.

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.51 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*	AN-AAA Authentication Status. Status of the last AN-AAA authentication request, if any, for the current 1xEV-DO session. • 0 - AAA_STATUS_FAILED - Authentication failed • 1 - AAA_STATUS_SUCCESS - Authentication success • 2 - AAA_STATUS_NO_REQUEST - No authentication requested • 0xffffffff - invalid data.

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.52 int unpack_nas_GetCDMANetworkParameters (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_GetCDMANetworkParameters_t * *pOutput*)

Get CDMA Network Parameters unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	qmi output parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.53 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_GetNetworkPreference_t * *pOutput*)

Gets the network registration preference unpack

Parameters

in	*pResp*	qmi resp from modem
in	*respLen*	qmi resp length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.56 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*)

Unpack provides information about the system that provides service to the device. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use API unpack_nas_SLQSGetSysInfo() for new firmware versions and new modules

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.58 int unpack_nas_GetServingNetworkCapabilities (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_GetServingNetworkCapabilities_t * *pOutput*)

Unpack returns information regarding the data capabilities of the system that currently provides service to the device.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.59 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*)

Unpack performs scan for available networks and scans for RAT info as well.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.62 int unpack_nas_SetACCOLC (uint8_t * *pResp*, uint16_t *respLen*)

Unpack Sets the access overload class (ACCOLC)

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.63 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*)

Unpack set event report indication.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.66 int unpack_nas_SetLURejectCallback (uint8_t * *pResp*, uint16_t *respLen*)

Unpack set registration reject Callback

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.67 int unpack_nas_SetNasLTECphyCaIndCallback_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SetNasLTECphyCaIndCallback_ind_t * *pOutput*)

Unpack set LTE PHY CA indication callback.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	sig info indication unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.68 int unpack_nas_SetNetworkPreference (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SetNetwork-Preference_t * *pOutput*)

Sets the network registration preference unpack

Parameters

in	*pResp*	qmi resp from modem
in	*respLen*	qmi resp length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.69 int unpack_nas_SetRFInfoCallback (uint8_t * *pResp*, uint16_t *respLen*)

Unack Set RF Band Information callback.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.70 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*)

Unack get carrier aggregation event information.

Parameters

in	*pResp*	qmi resp from modem
in	*respLen*	qmi resp length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.75 int unpack_nas_SLQSGetNetworkTime (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSGetNetwork-Time_t * *pOutput*)

Unpack get network time.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.76 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*)

unpack get operator name for specificed network.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.78 int unpack_nas_SLQSGetServingSystem (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSGetServing-System_t * *pOutput*)

Unack queries information regarding the system that currently provides service.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.79 int unpack_nas_SLQSGetSignalStrength (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSGetSignal-Strength_t * *pOutput*)

Unpack get signal strength.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	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*)

Unpack initiates a network registration.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.83 int unpack_nas_SLQSNasConfigSigInfo2 (uint8_t * *pResp*, uint16_t *respLen*)

Unpack sets the signal strength reporting thresholds

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.84 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*)

Unpack retrieves cell location-related information.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.86 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*)

Unpack used to Registers/De-registers for different NAS (Network access service) indications.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.90 int unpack_nas_SLQSNasNetworkTimeCallBack_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSNasNetworkTimeCallBack_ind_t * *pOutput*)

Unpack Network time callback indication.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	sig info indication unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.91 int unpack_nas_SLQSNasSigInfoCallback_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSNasSigInfoCallback_ind_t * *pOutput*)

Unpack Signal Information callback indication.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	sig info indication unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.92 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*)

unpack sets the registration state for different QMI_NAS SWI indications.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.94 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*)

Unpack system information callback indication.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.98 int unpack_nas_SLQSNasTimerCallback_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSNasTimerCallback_ind_t * *pOutput*)

Unpack timer callback indication.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	network timer indication unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.99 int unpack_nas_SLQSSetBandPreference (uint8_t * *pResp*, uint16_t *respLen*)

Unpack sets the different system selection preferences of the device.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

9.6.3.100 int unpack_nas_SLQSSetSignalStrengthsCallback (uint8_t * *pResp*, uint16_t *respLen*)

Unpack set signal strength thresholds callback.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.101 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*)

Unack fetch CQI parameters for LTE data session.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.107 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

9.6.3.110 int unpack_valid_nas_GetCDMANetworkParameters (uint8_t * *pResp*, uint8_t *u8Info*)

valid CDMA Network Parameters

Parameters

in	*pResp*	qmi response from modem
in	*u8Info*	value to check unpack_nas_GetCDMANetworkParameters_t param valid. • 0 - SCI • 1 - SCM • 2 - RegHomeSID, RegForeignSID and RegForeignNID • 3 - ForceRev0 • 4 - CustomSCP, Protocol ,Broadcast and Application • 5 - Roaming

Returns

eQCWWAN_ERR_NONE on signal info valid, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.111 int unpack_valid_nas_SLQSGetServingSystem (uint8_t * *pResp*, uint8_t *u8Info*)

valid queries information regarding the system that currently provides service

Parameters

in	*pResp*	qmi response from modem
in	*u8Info*	value to check unpack_nas_SLQSGetServingSystem_t param valid. • 0 - RoamIndicatorVal • 1 - DataSrvCapabilities • 2 - CurrentPLMN • 3 - SystemID and NetworkID • 4 - BaseStationID, BaseStationLatitude and BasestationLongitude • 5 - RoamingIndicatorList • 6 - DefaultRoamInd • 7 - Gpp2TimeZone • 8 - CDMA_P_Rev • 9 - GppTimeZone • 10 - GppNetworkDSTAdjustment • 11 - Lac • 12 - CellID • 13 - ConcSvcInfo • 14 - PRLInd • 15 - DTMInd • 16 - DetailedSvcInfo • 17 - CDMASystemInfoExt • 18 - HdrPersonality • 19 - TrackAreaCode • 20 - CallBarStatus

Returns

eQCWWAN_ERR_NONE on signal info valid, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.112 int unpack_valid_nas_SLQSGetSignalStrength (uint8_t * *pResp*, uint8_t *u8Info*)

valid queries information regarding the system that currently provides service

Parameters

in	*pResp*	qmi response from modem
in	*u8Info*	value to check unpack_nas_SLQSGetSignalStrength_t param valid. • 0 - rxSignalStrengthList and rxSignalStrengthListLen • 1 - ecioList and ecioListLen • 2 - Io • 3 - sinr • 4 - errorRateListLen, errorRateList • 5 - rsrqInfo • 6 - ltesnr • 7 - ltersrp

Returns

eQCWWAN_ERR_NONE on signal info valid, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.113 int unpack_valid_nas_SLQSNasGetSigInfo (uint8_t * *pResp*, uint8_t *u8Info*)

valid get sig info

Parameters

in	*pResp*	qmi response from modem
in	*u8Info*	value to check unpack_nas_SLQSNasGetSigInfo_t param valid. • 0 - CDMASSInfo • 1 - HDRSSInfo. • 2 - GSMSSInfo. • 3 - WCDMASSInfo. • 4 - LTESSInfo.

Returns

eQCWWAN_ERR_NONE on signal info valid, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7 pds.h File Reference

Data Structures

- struct unpack_pds_GetPDSState_t
- struct pack_pds_SetPDSState_t
- struct unpack_pds_GetPDSDefaults_t
- struct pack_pds_SetPDSDefaults_t
- struct unpack_pds_GetPortAutomaticTracking_t
- struct pack_pds_SetPortAutomaticTracking_t
- struct pack_pds_StartPDSTrackingSessionExt_t
- struct pack_pds_PDSInjectTimeReference_t
- struct unpack_pds_GetXTRAAutomaticDownload_t
- struct pack_pds_SetXTRAAutomaticDownload_t
- struct unpack_pds_GetXTRANetwork_t
- struct pack_pds_SetXTRANetwork_t
- struct unpack_pds_GetXTRAValidity_t
- struct unpack_pds_GetServiceAutomaticTracking_t
- struct pack_pds_SetServiceAutomaticTracking_t
- struct pack_pds_ResetPDSData_t
- struct pack_pds_SLQSSetAGPSConfig_t
- struct pack_pds_SLQSPDSInjectAbsoluteTimeReference_t

- struct pack_pds_SLQSGetAGPSConfig_t
- struct unpack_pds_SLQSGetAGPSConfig_t
- struct pack_pds_SLQSPDSInjectPositionData_t
- struct unpack_pds_SLQSGetGPSStateInfo_t
- struct pack_pds_SLQSSetPositionMethodState_t
- struct pack_pds_SetEventReportCallback_t
- struct unpack_pds_SetEventReport_Ind_t
- struct unpack_pds_SetPdsState_Ind_t

Functions

- int pack_pds_GetPDSState (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_pds_GetPDSState (uint8_t *pResp, uint16_t respLen, unpack_pds_GetPDSState_t *pOutput)
- int pack_pds_SetPDSState (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_SetPDSState_t *pReqParam)
- int unpack_pds_SetPDSState (uint8_t *pResp, uint16_t respLen)
- int pack_pds_GetPDSDefaults (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_pds_GetPDSDefaults (uint8_t *pResp, uint16_t respLen, unpack_pds_GetPDSDefaults_t *pOutput)
- int pack_pds_SetPDSDefaults (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_SetPDSDefaults_t *pReqParam)
- int unpack_pds_SetPDSDefaults (uint8_t *pResp, uint16_t respLen)
- int pack_pds_GetPortAutomaticTracking (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_pds_GetPortAutomaticTracking (uint8_t *pResp, uint16_t respLen, unpack_pds_GetPortAutomaticTracking_t *pOutput)
- int pack_pds_SetPortAutomaticTracking (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_SetPortAutomaticTracking_t *pReqParam)
- int unpack_pds_SetPortAutomaticTracking (uint8_t *pResp, uint16_t respLen)
- int pack_pds_StartPDSTrackingSessionExt (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_StartPDSTrackingSessionExt_t *pReqParam)
- int unpack_pds_StartPDSTrackingSessionExt (uint8_t *pResp, uint16_t respLen)
- int pack_pds_StopPDSTrackingSession (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_pds_StopPDSTrackingSession (uint8_t *pResp, uint16_t respLen)
- int pack_pds_PDSInjectTimeReference (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_PDSInjectTimeReference_t *pReqParam)
- int unpack_pds_PDSInjectTimeReference (uint8_t *pResp, uint16_t respLen)
- int pack_pds_GetXTRAAutomaticDownload (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_pds_GetXTRAAutomaticDownload (uint8_t *pResp, uint16_t respLen, unpack_pds_GetXTRAAutomaticDownload_t *pOutput)
- int pack_pds_SetXTRAAutomaticDownload (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_SetXTRAAutomaticDownload_t *pReqParam)
- int unpack_pds_SetXTRAAutomaticDownload (uint8_t *pResp, uint16_t respLen)
- int pack_pds_GetXTRANetwork (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_pds_GetXTRANetwork (uint8_t *pResp, uint16_t respLen, unpack_pds_GetXTRANetwork_t *pOutput)
- int pack_pds_SetXTRANetwork (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_SetXTRANetwork_t *pReqParam)
- int unpack_pds_SetXTRANetwork (uint8_t *pResp, uint16_t respLen)
- int pack_pds_GetXTRAValidity (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_pds_GetXTRAValidity (uint8_t *pResp, uint16_t respLen, unpack_pds_GetXTRAValidity_t *pOutput)
- int pack_pds_ForceXTRADownload (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_pds_ForceXTRADownload (uint8_t *pResp, uint16_t respLen)
- int pack_pds_GetServiceAutomaticTracking (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)

- int unpack_pds_GetServiceAutomaticTracking (uint8_t *pResp, uint16_t respLen, unpack_pds_GetServiceAutomaticTracking_t *pOutput)
- int pack_pds_SetServiceAutomaticTracking (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_SetServiceAutomaticTracking_t *pReqParam)
- int unpack_pds_SetServiceAutomaticTracking (uint8_t *pResp, uint16_t respLen)
- int pack_pds_ResetPDSData (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_ResetPDSData_t *pReqParam)
- int unpack_pds_ResetPDSData (uint8_t *pResp, uint16_t respLen)
- int pack_pds_SLQSSetAGPSConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_SLQSSetAGPSConfig_t *pReqParam)
- int unpack_pds_SLQSSetAGPSConfig (uint8_t *pResp, uint16_t respLen)
- int pack_pds_SLQSPDSInjectAbsoluteTimeReference (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_SLQSPDSInjectAbsoluteTimeReference_t *pReqParam)
- int unpack_pds_SLQSPDSInjectAbsoluteTimeReference (uint8_t *pResp, uint16_t respLen)
- int pack_pds_SLQSGetAGPSConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_SLQSGetAGPSConfig_t *pReqParam)
- int unpack_pds_SLQSGetAGPSConfig (uint8_t *pResp, uint16_t respLen, unpack_pds_SLQSGetAGPSConfig_t *pOutput)
- int pack_pds_SLQSPDSInjectPositionData (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_SLQSPDSInjectPositionData_t *pReqParam)
- int unpack_pds_SLQSPDSInjectPositionData (uint8_t *pResp, uint16_t respLen)
- int pack_pds_SLQSPDSDeterminePosition (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_pds_SLQSPDSDeterminePosition (uint8_t *pResp, uint16_t respLen)
- int pack_pds_SLQSGetGPSStateInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_pds_SLQSGetGPSStateInfo (uint8_t *pResp, uint16_t respLen, unpack_pds_SLQSGetGPSStateInfo_t *pOutput)
- int pack_pds_SLQSSetPositionMethodState (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_SLQSSetPositionMethodState_t *pReqParam)
- int unpack_pds_SLQSSetPositionMethodState (uint8_t *pResp, uint16_t respLen)
- int pack_pds_SetEventReportCallback (pack_qmi_t *pCtx, uint8_t *pReq, uint16_t *pLen, pack_pds_SetEventReportCallback_t *pReqParam)
- int unpack_pds_SetEventReportCallback (uint8_t *pResp, uint16_t respLen)
- int unpack_pds_SetEventReport_Ind (uint8_t *pResp, uint16_t respLen, unpack_pds_SetEventReport_Ind_t *pOutput)
- int unpack_pds_SetPdsState_Ind (uint8_t *pResp, uint16_t respLen, unpack_pds_SetPdsState_Ind_t *pOutput)

9.7.1 Function Documentation

9.7.1.1 int pack_pds_ForceXTRADownload (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Forces the XTRA database to be downloaded to the device pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.2 int pack_pds_GetPDSDefaults (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Get the default tracking session configuration pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.3 int pack_pds_GetPDSState (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Get current PDS state pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.4 int pack_pds_GetPortAutomaticTracking (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Returns the automatic tracking configuration for the NMEA COM port pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.5 int pack_pds_GetServiceAutomaticTracking (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

automatic tracking state for the service pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.6 int pack_pds_GetXTRAAutomaticDownload (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

XTRA automatic database download configuration pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.7 int pack_pds_GetXTRANetwork (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

XTRA WWAN network preference parameter pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.8 int pack_pds_GetXTRAValidity (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

XTRA database validity period parameter pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.9 int pack_pds_PDSInjectTimeReference (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_PDSInjectTimeReference_t * *pReqParam*)

Inject system time parameters pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.10 int pack_pds_ResetPDSData (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_ResetPDSData_t * *pReqParam*)

Resets the specified PDS data pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.11 int pack_pds_SetEventReportCallback (pack_qmi_t * *pCtx*, uint8_t * *pReq*, uint16_t * *pLen*, pack_pds_SetEventReportCallback_t * *pReqParam*)

Enables/disables the PDS Event report callback function pack.

Parameters

in, out	*pCtx*	qmi request context
out	*pReq*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.12 int pack_pds_SetPDSDefaults (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_SetPDSDefaults_t * *pReqParam*)

Sets the default tracking session configuration pack.

Parameters

in, out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.13 int pack_pds_SetPDSState (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_SetPDSState_t * *pReqParam*)

Sets the PDS state pack.

Parameters

in, out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.14 int pack_pds_SetPortAutomaticTracking (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_SetPortAutomaticTracking_t * *pReqParam*)

Sets the automatic tracking configuration for the NMEA COM port pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.15 int pack_pds_SetServiceAutomaticTracking (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_SetServiceAutomaticTracking_t * *pReqParam*)

Sets the automatic tracking state for the service pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.16 int pack_pds_SetXTRAAutomaticDownload (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_SetXTRAAutomaticDownload_t * *pReqParam*)

Sets the XTRA automatic database download configuration pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.17 int pack_pds_SetXTRANetwork (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_SetXTRANetwork_t * *pReqParam*)

Sets the XTRA WWAN network preference pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.18 int pack_pds_SLQSGetAGPSConfig (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_SLQSGetAGPSConfig_t * *pReqParam*)

Gets the PDS AGPS (MS-based) configuration pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.19 int pack_pds_SLQSGetGPSStateInfo (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Queries the MSM GPS server for receiver state information pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.20 int pack_pds_SLQSPDSDeterminePosition (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Requests the MSM GPS service to obtain the current position for manually controlled tracking sessions pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.21 int pack_pds_SLQSPDSInjectAbsoluteTimeReference (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_SLQSPDSInjectAbsoluteTimeReference_t * *pReqParam*)

Injects a absolute time reference into the PDS engine pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.22 int pack_pds_SLQSPDSInjectPositionData (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_SLQSPDSInjectPositionData_t * *pReqParam*)

Injects position data into the PDS engine pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.23 int pack_pds_SLQSSetAGPSConfig (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_SLQSSetAGPSConfig_t * *pReqParam*)

Sets the PDS AGPS (MS-based) configuration pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.24 int pack_pds_SLQSSetPositionMethodState (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_SLQSSetPositionMethodState_t * *pReqParam*)

Sets the state of positioning methods for the device pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.25 int pack_pds_StartPDSTrackingSessionExt (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_StartPDSTrackingSessionExt_t * *pReqParam*)

Starts a PDS tracking session pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.26 int pack_pds_StopPDSTrackingSession (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Stops a PDS tracking session pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.27 int unpack_pds_ForceXTRADownload (uint8_t * *pResp*, uint16_t *respLen*)

Forces the XTRA database to be downloaded to the device unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.28 int unpack_pds_GetPDSDefaults (uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_GetPDSDefaults_t * *pOutput*)

Get the default tracking session configuration unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.29 int unpack_pds_GetPDSState (uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_GetPDSState_t * *pOutput*)

Get current PDS state unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.30 int unpack_pds_GetPortAutomaticTracking (uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_GetPortAutomaticTracking_t * *pOutput*)

Returns the automatic tracking configuration for the NMEA COM port unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.31 int unpack_pds_GetServiceAutomaticTracking (uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_GetServiceAutomaticTracking_t * *pOutput*)

automatic tracking state for the service unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.32 int unpack_pds_GetXTRAAutomaticDownload (uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_GetXTRAAutomaticDownload_t * *pOutput*)

XTRA automatic database download configuration unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.33 int unpack_pds_GetXTRANetwork (uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_GetXTRANetwork_t * *pOutput*)

XTRA WWAN network preference parameter unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.34 int unpack_pds_GetXTRAValidity (uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_GetXTRAValidity_t * *pOutput*)

XTRA database validity period parameter unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.35 int unpack_pds_PDSInjectTimeReference (uint8_t * *pResp*, uint16_t *respLen*)

Inject system time parameters unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.36 int unpack_pds_ResetPDSData (uint8_t * *pResp*, uint16_t *respLen*)

Resets the specified PDS data unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.37 int unpack_pds_SetEventReport_Ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_SetEventReport_Ind_t * *pOutput*)

PDS Event Report Indication unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.38 int unpack_pds_SetEventReportCallback (uint8_t * *pResp*, uint16_t *respLen*)

Enables/disables the PDS Event report callback function unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.39 int unpack_pds_SetPDSDefaults (uint8_t * *pResp*, uint16_t *respLen*)

Sets the default tracking session configuration unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.40 int unpack_pds_SetPDSState (uint8_t * *pResp*, uint16_t *respLen*)

Sets the PDS state unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.41 int unpack_pds_SetPdsState_Ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_SetPdsState_Ind_t * *pOutput*)

PDS State Indication unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.42 int unpack_pds_SetPortAutomaticTracking (uint8_t * *pResp*, uint16_t *respLen*)

Sets the automatic tracking configuration for the NMEA COM port unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.43 int unpack_pds_SetServiceAutomaticTracking (uint8_t * *pResp*, uint16_t *respLen*)

Sets the automatic tracking state for the service unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.44 int unpack_pds_SetXTRAAutomaticDownload (uint8_t * *pResp*, uint16_t *respLen*)

Sets the XTRA automatic database download configuration unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.45 int unpack_pds_SetXTRANetwork (uint8_t * *pResp*, uint16_t *respLen*)

Sets the XTRA WWAN network preference unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.46 int unpack_pds_SLQSGetAGPSConfig (uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_SLQSGetAGPSConfig_t * *pOutput*)

Gets the PDS AGPS (MS-based) configuration unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.47 int unpack_pds_SLQSGetGPSStateInfo (uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_SLQSGetGPSStateInfo_t * *pOutput*)

Queries the MSM GPS server for receiver state information unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.48 int unpack_pds_SLQSPDSDeterminePosition (uint8_t * *pResp*, uint16_t *respLen*)

Requests the MSM GPS service to obtain the current position for manually controlled tracking sessions unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.49 int unpack_pds_SLQSPDSInjectAbsoluteTimeReference (uint8_t * *pResp*, uint16_t *respLen*)

Injects a absolute time reference into the PDS engine unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.50 int unpack_pds_SLQSPDSInjectPositionData (uint8_t * *pResp*, uint16_t *respLen*)

Injects position data into the PDS engine unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.51 int unpack_pds_SLQSSetAGPSConfig (uint8_t * *pResp*, uint16_t *respLen*)

Sets the PDS AGPS (MS-based) configuration unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.52 int unpack_pds_SLQSSetPositionMethodState (uint8_t * *pResp*, uint16_t *respLen*)

Sets the state of positioning methods for the device unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.53 int unpack_pds_StartPDSTrackingSessionExt (uint8_t * *pResp*, uint16_t *respLen*)

Starts a PDS tracking session unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.54 int unpack_pds_StopPDSTrackingSession (uint8_t * *pResp*, uint16_t *respLen*)

Stops a PDS tracking session unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.8 qaCbkCatEventReportInd.h File Reference

Data Structures

- struct CatEventIDDataTlv
- struct CatAlPhaIdentifierTlv
- struct CatEventListTlv
- struct CatRefreshTlv
- struct CatEndProactiveSessionTlv
- union currentCatEvent
- struct CatCommonEventTlv
- struct QmiCbkCatEventStatusReportInd

Macros

- #define QMI_MAX_CAT_EVENT_DATA_LENGTH 255
- #define QMI_CAN_COMMON_EVENT_TLV_NUMBER 11

Enumerations

- enum eQMI_CAT_EVENT_REPORT_IND_TLV {
  eTLV_CBK_DISPLAY_TEXT = 0x10,
  eTLV_CBK_GET_IN_KEY = 0x11,
  eTLV_CBK_GET_INPUT = 0x12,
  eTLV_CBK_SETUP_MENU = 0x13,
  eTLV_CBK_SELECT_ITEM = 0x14,
  eTLV_CBK_ALPHA_IDENTIFIER = 0x15,
  eTLV_CBK_SETUP_EVENT_LIST = 0x16,
  eTLV_CBK_SETUP_IDLE_MODE_TEXT = 0x17,
  eTLV_CBK_LANGUAGE_NOTIFICATION = 0x18,
  eTLV_CBK_REFRESH = 0x19,
  eTLV_CBK_END_PROACTIVE_SESSION = 0x1A }
- enum eQMI_CAT_EVENT_REPORT_IND_TLV_LENGTH {
  eTLV_SETUP_EVENT_LIST_LENGTH = 0x04,
  eTLV_REFRESH_LENGTH = 0x03,
  eTLV_END_PROACTIVE_SESSION_LENGTH = 0x01 }

Functions

- enum eQCWWANError UpkQmiCbkCatEventReportInd (BYTE *pMdmResp, struct QmiCbkCatEventStatus-ReportInd *pAipResp)

9.8.1 Macro Definition Documentation

9.8.1.1 #define QMI_CAN_COMMON_EVENT_TLV_NUMBER 11

9.8.1.2 #define QMI_MAX_CAT_EVENT_DATA_LENGTH 255

9.8.2 Enumeration Type Documentation

9.8.2.1 enum eQMI_CAT_EVENT_REPORT_IND_TLV

Enumerator

eTLV_CBK_DISPLAY_TEXT

eTLV_CBK_GET_IN_KEY

eTLV_CBK_GET_INPUT

eTLV_CBK_SETUP_MENU

eTLV_CBK_SELECT_ITEM

eTLV_CBK_ALPHA_IDENTIFIER

eTLV_CBK_SETUP_EVENT_LIST

eTLV_CBK_SETUP_IDLE_MODE_TEXT

eTLV_CBK_LANGUAGE_NOTIFICATION

eTLV_CBK_REFRESH

eTLV_CBK_END_PROACTIVE_SESSION

9.8.2.2 enum eQMI_CAT_EVENT_REPORT_IND_TLV_LENGTH

Enumerator

eTLV_SETUP_EVENT_LIST_LENGTH

eTLV_REFRESH_LENGTH

eTLV_END_PROACTIVE_SESSION_LENGTH

9.8.3 Function Documentation

9.8.3.1 enum eQCWWANError UpkQmiCbkCatEventReportInd (BYTE * *pMdmResp*, struct QmiCbkCatEventStatusReportInd * *pAipResp*)

9.9 qaCbkSwiOmaDmEventReportInd.h File Reference

Data Structures

- struct sessionInfoTlv
- struct sessionInfoTlvExt
- struct QmiCbkSwiOmaDmEventStatusReportInd
- struct QmiCbkSwiOmaDmEventStatusReportIndExt

Macros

- #define QMI_SWIOMA_DM_FOTA 0x00
- #define QMI_SWIOMA_DM_CONFIG 0x01
- #define QMI_SWIOMA_DM_NOT 0x02

Enumerations

- enum eQMI_SWIOMA_DM_EVENT_REPORT_IND {
  eTLV_IND_OMA_DM_FOTA = 0x10,
  eTLV_IND_OMA_DM_CONFIG = 0x11,
  eTLV_IND_OMA_DM_NOT = 0x12 }

Functions

- enum eQCWWANError UpkQmiCbkSwiOmaDmEventReportInd (BYTE *pMdmResp, struct QmiCbkSwiOmaDmEventStatusReportInd *pApiResp)
- package enum eQCWWANError UpkQmiCbkSwiOmaDmEventReportIndExt (BYTE *pMdmResp, struct QmiCbkSwiOmaDmEventStatusReportInd *pApiResp)

9.9.1 Macro Definition Documentation

9.9.1.1 #define QMI_SWIOMA_DM_CONFIG 0x01

9.9.1.2 #define QMI_SWIOMA_DM_FOTA 0x00

9.9.1.3 #define QMI_SWIOMA_DM_NOT 0x02

9.9.2 Enumeration Type Documentation

9.9.2.1 enum eQMI_SWIOMA_DM_EVENT_REPORT_IND

Enumerator

eTLV_IND_OMA_DM_FOTA

eTLV_IND_OMA_DM_CONFIG

eTLV_IND_OMA_DM_NOT

9.9.3 Function Documentation

9.9.3.1 enum eQCWWANError UpkQmiCbkSwiOmaDmEventReportInd (BYTE * *pMdmResp,* struct QmiCbkSwiOmaDmEventStatusReportInd * *pApiResp*)

9.9.3.2 package enum eQCWWANError UpkQmiCbkSwiOmaDmEventReportIndExt (BYTE * *pMdmResp,* struct QmiCbkSwiOmaDmEventStatusReportInd * *pApiResp*)

9.10 qaGobiApiAudio.h File Reference

Audio Service API function prototypes.

Data Structures

- struct GetAudioProfileReq
- struct GetAudioProfileResp
- struct SetAudioProfileReq
- struct GetAudioPathConfigReq
- struct TXPCMIIRFltr
- struct RXPCMIIRFltr
- struct RXAGCList

- struct RXAVCList
- struct TXAGCList
- struct GetAudioPathConfigResp
- struct SetAudioPathConfigReq
- struct GetAudioVolTLBConfigReq
- struct GetAudioVolTLBConfigResp
- struct SetAudioVolTLBConfigReq
- struct SetAudioVolTLBConfigResp

Functions

- ULONG SLQSGetAudioProfile (GetAudioProfileReq *pGetAudioProfileReq, GetAudioProfileResp *pGetAudioProfileResp)
- ULONG SLQSSetAudioProfile (SetAudioProfileReq *pSetAudioProfileReq)
- ULONG SLQSGetAudioPathConfig (GetAudioPathConfigReq *pGetAudioPathConfigReq, GetAudioPathConfigResp *pGetAudioPathConfigResp)
- ULONG SLQSSetAudioPathConfig (SetAudioPathConfigReq *pSetAudioPathConfigReq)
- ULONG SLQSGetAudioVolTLBConfig (GetAudioVolTLBConfigReq *pGetAudioVolTLBCfgReq, GetAudioVolTLBConfigResp *pGetAudioVolTLBCfgResp)
- ULONG SLQSSetAudioVolTLBConfig (SetAudioVolTLBConfigReq *pSetAudioVolTLBCfgReq, SetAudioVolTLBConfigResp *pSetAudioVolTLBCfgResp)

9.10.1 Detailed Description

Audio Service API function prototypes.

9.10.2 Function Documentation

9.10.2.1 ULONG SLQSGetAudioPathConfig (GetAudioPathConfigReq * *pGetAudioPathConfigReq*, GetAudioPathConfigResp * *pGetAudioPathConfigResp*)

This API gets the audio path configuration parameters.

Parameters

pGetAudioPathConfigReq[IN]	• See GetAudioPathConfigReq for more information
pGetAudioPathConfigResp[OUT]	• See GetAudioPathConfigResp for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Device Supported: SL9090
Timeout: 5 seconds

9.10.2.2 ULONG SLQSGetAudioProfile (GetAudioProfileReq * *pGetAudioProfileReq*, GetAudioProfileResp * *pGetAudioProfileResp*)

This API get the profile content of the requested audio generator.

Parameters

pGetAudioProfileReq[IN]	• See GetAudioProfileReq for more information
pGetAudioProfileResp[OUT]	• See GetAudioProfileResp for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Device Supported: SL9090
Timeout: 5 seconds

9.10.2.3 ULONG SLQSGetAudioVolTLBConfig (GetAudioVolTLBConfigReq * *pGetAudioVolTLBCfgReq*, GetAudioVolTLBConfigResp * *pGetAudioVolTLBCfgResp*)

This API gets the audio path configuration parameters.

Parameters

pGetAudioVolTLBCfgReq[IN]	• See GetAudioVolTLBConfigReq for more information
pGetAudioVolTLBCfgResp[OUT]	• See GetAudioVolTLBConfigResp for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Device Supported: SL9090
Timeout: 5 seconds

9.10.2.4 ULONG SLQSSetAudioPathConfig (SetAudioPathConfigReq * *pSetAudioPathConfigReq*)

This API sets the audio path configuration parameters.

Parameters

pSetAudioPath-ConfigReq[IN]	• See SetAudioPathConfigReq for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Device Supported: SL9090
Timeout: 5 seconds

9.10.2.5 ULONG SLQSSetAudioProfile (SetAudioProfileReq * *pSetAudioProfileReq*)

This API sets an audio profile.

Parameters

pSetAudio-ProfileReq[IN]	• See SetAudioProfileReq for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Device Supported: SL9090
Timeout: 5 seconds

9.10.2.6 ULONG 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

samplesPerBatch	• 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

ModemTempState	• provides the temperature state of the modem • Values: – 0 - unknown – 1 - normal – 2 - high(warning) – 3 - high(critical) – 4 - low(critical)
ModemTemperature	• 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

mobileCountry-Code	• 16 bit representation of MCC value range : 0 -999
mobileNetwork-Code	• 16 bit representation of MNC value range : 0 -999

9.12.3.20 typedef struct SMSEventInfo_s SMSEventInfo

This structure will hold the information related to received SMS events

Parameters

smsEventType	• Type of the SMS events that are received. This is a bit map of SMSEventType. Only the parameters (which follows) related to the events received would be filled, and the rest of the parameters would be NULL
pMTMessage-Info	• pointer to the SMSMTMessageInfo structure NULL, if this event is not present in the smsEventType parameter
pTransferRoute-MTMessageInfo	• pointer to the SMSTransferRouteMTMessageInfo structure . NULL, if this event is not present in the smsEventType parameter
pMessageMode-Info	• pointer to the SMSMessageModeInfo structure NULL, if this event is not present in the smsEventType parameter
pEtwsMessage-Info	• pointer to the SMSEtwsMessageInfo structure NULL, if this event is not present in the smsEventType parameter
pEtwsPlmnInfo	• pointer to the SMSEtwsPlmnInfo structure NULL, if this event is not present in the smsEventType parameter
pSMSCAddress-Info	• pointer to the SMSCAddressInfo structure NULL, if this event is not present in the smsEventType parameter
pSMSOnIMSInfo	• pointer to the SMSOnIMSInfo structure NULL, if this event is not present in the smsEventType parameter Note: None

9.12.3.21 typedef struct SMSMessageMode SMSMessageModeInfo

This structure holds information related to message mode

Parameters

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

9.12.3.22 typedef struct SMSMTMessage SMSMTMessageInfo

This structure holds information related to MT SMS

Parameters

storageType	• SMS message storage type for the new message 0 - UIM 1 - NV

messageIndex	• Index of the new message

9.12.3.23 typedef struct SMSOnIMS SMSOnIMSInfo

This structure holds information related to message mode

Parameters

smsOnIMS	• Indicates whether the message is received from IMS 0x00 - Message is not received from IMS 0x01 - Message is received from IMS 0x02-0xFF - Reserved Note: In multiple modem solutions, this TLV may be used to help the client determine with which modem to communicate. This TLV may not be supported on all implementations.

9.12.3.24 typedef struct SMSTransferRouteMTMessage SMSTransferRouteMTMessageInfo

This structure holds information related to transfer route MT SMS

Parameters

ackIndicator	• Parameter to indicate if ACK must be sent by the control point 0x00 - Send ACK 0x01 - Do not send ACK
transactionID	• Transaction ID of the message
format	• Message format 0x00 - CDMA 0x02 - 0x05 - Reserved 0x06 - GW_PP 0x07 - GW_BC
length	• Length of the raw message. This length should not exceed the maximum WMS payload length of 256 bytes
data	• Raw message data

9.12.3.25 typedef struct svUsedforFix_s svUsedforFix

This structure contains SVs Used to Calculate the Fix.

Parameters

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

9.12.3.26 typedef struct _SwiAvmsEventInd SwiAvmsEventInd

9.12.3.27 typedef struct SwiOTAMsg_s SwiOTAMsg

This structure contains OTA message

Parameters

type	• message type – 0 - LTE ESM uplink – 1 - LTE ESM downlink – 2 - LTE EMM uplink – 3 - LTE EMM downlink – 4 - GSM/UMTS uplink – 5 - GSM/UMTS downlink
data_len	• OTA Message Content Length
data	• OTA Message Content
pLteNasRelInfo	• LTE NAS Release Info • see LteNasReleaseInfo for details
pTime	• Seconds in local time since Jan. 6th 1980 00:00:00 UTC

9.12.3.28 typedef void(∗ tFNActivationStatus)(ULONG activationStatus)

Activation status callback function.

Parameters

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

9.12.3.29 typedef void(∗ tFNAllCallStatus)(voiceSetAllCallStatusCbkInfo ∗pVoiceSetAllCallStatusCbkInfo)

Voice Call Status Callback function. This function pointer will be executed to process received Indication.

Parameters

pVoiceSetAllCallStatusCbkInfo	• 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

pSMSMemory-FullInfo[OUT]	• pointer to SMSMemoryInfo. • see SMSMemoryInfo for details.

9.12.3.66 typedef void(∗ tFNMessageWaiting)(msgWaitingInfo ∗pSMSMessageWaitingInfo)

SMS Memory related callback function.

Parameters

pSMSMessage-WaitingInfo[OUT]	• pointer to msgWaitingInfo. • see msgWaitingInfo for details.

9.12.3.67 typedef void(∗ tFNMitiLvlRpt)(QmiCbkTmdMitiLvlRptInd ∗pSetLocCradleMount)

Set Loc cradle mount Indication callback.

Parameters

pSetLocCradle-Mount	• 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

pTransNWRegInfoNotification	• See transNWRegInfoNotification for more information.

9.12.3.114 typedef void(∗ tFNUIMRefresh)(UIMRefreshEvent ∗pUIMRefreshEvent)

UIM Refresh Callback function

Parameters

pUIMRefreshEvent	• 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

pUIMStatusChangeInfo	• 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)
- ULONG GetQmapIsSupported (BYTE *pQmapsupport)

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 GetQmapIsSupported (BYTE * *pQmapsupport*)

Returns if the module supports QMAP feature.

Parameters

pQmapsupport[-OUT]	• value – 0 means module does not support QMAP, otherwise QMAP is supported

See Also

see qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 5 seconds

9.13.3.2 ULONG QCWWAN2kConnect (CHAR * *pDeviceID*, CHAR * *pDeviceKey*)

Connects the Connection Manager API to the first detected QC WWAN device. This function MUST be called after QCWWAN2kEnumerateDevices has been called.

Parameters

pDeviceID[IN]	• Device path pertaining to the device for which the API is being invoked e.g. /dev/qcqmi0.
pDeviceKey[IN]	• Device key pertaining to the device for which the API is being invoked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_ERR_NO_DEVICE otherwise

Note

Timeout: 2 seconds

9.13.3.3 ULONG QCWWAN2kEnumerateDevices (BYTE * *pDevicesSize,* BYTE * *pDevices*)

Enumerates the QC WWAN devices currently attached to the host. This API MUST be called before any other API.

Parameters

pDeviceSize[IN/-OUT]	• Upon input, maximum number of elements that the device array can contain. • 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

9.13.3.4 ULONG QCWWAN2kGetConnectedDeviceID (ULONG *deviceIDSize,* CHAR * *pDeviceID,* ULONG *deviceKeySize,* CHAR * *pDeviceKey*)

Returns the device ID and device key of the currently connected QC WWAN device.

Parameters

deviceIDSize	• Maximum number of characters (including NULL terminator) that the device ID array can contain.
pDeviceID[OUT]	• Device path string

deviceKeySize	• Maximum number of characters (including NULL terminator) that the device key array can contain.
pDeviceKey[OUT]	• Device key string

Returns

eQCWWAN_ERR_NONE if device found, eQCWWAN_ERR_NO_DEVICE otherwise

Note

Timeout: 2 seconds

9.13.3.5 ULONG QCWWANConnect (CHAR * *pDeviceID,* CHAR * *pDeviceKey*)

Enumerates the QC WWAN devices currently attached to the host. This API MUST be called before any other API.

Parameters

pDeviceID[IN]	• Device path pertaining to the device for which the API is being invoked e.g. /dev/qcqmi0.
pDeviceKey[IN]	• Device key pertaining to the device for which the API is being invoked

Returns

eQCWWAN_ERR_NONE if device found, eQCWWAN_ERR_NO_DEVICE otherwise

Note

Timeout: 2 seconds
This API is deprecated; use QCWWAN2kConnect() instead

9.13.3.6 ULONG QCWWANDisconnect ( )

Disconnects the Connection Manager API from a previously connected QC device. This function de-registers all the callback functions that have been registered.

Parameters

none	

Returns

eQCWWAN_ERR_NONE

Note

Timeout: 2 seconds

9.13.3.7 ULONG QCWWANEnumerateDevices (BYTE * *pDevicesSize,* BYTE * *pDevices*)

Enumerates the QC WWAN devices currently attached to the host. This API is deprecated; use QCWWAN2kEnumerateDevices() instead.

Parameters

pDeviceSize[IN/OUT]	• Upon input, maximum number of elements that the device array can contain. • 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.

9.13.3.8 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

9.13.3.9 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

9.13.3.10 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

9.13.3.11 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

9.13.3.12 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.

9.13.3.13 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

9.13.3.14 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

9.13.3.15 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

9.13.3.16 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 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 GetVoiceNumber (BYTE voiceNumberSize, CHAR *pVoiceNumber, BYTE minSize, CHAR *pMIN)
- ULONG SetPower (ULONG powerMode)
- ULONG GetPower (ULONG *pPowerMode)
- ULONG GetOfflineReason (ULONG *pReasonMask, ULONG *pbPlatform)
- 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 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 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

9.14.4.8 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

9.14.4.9 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.

9.14.4.10 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

9.14.4.11 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

9.14.4.12 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

9.14.4.13 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

9.14.4.14 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

9.14.4.15 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

9.14.4.16 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

9.14.4.17 ULONG SLQSDmsSwiGetResetInfo (dmsSwiGetResetInfo * *pGetResetInfoResp*)

This function is used to get reset info

Parameters

pGetResetInfo-Resp	• See dmsSwiGetResetInfo for more information of the input structure

9.14.4.18 ULONG SLQSDmsSwiIndicationRegister (dmsIndicationRegisterReq * *pIndicationRegisterReq*)

This function used to set Swi Indication Register

Parameters

pConfig	• See dmsIndicationRegisterReq for more information of the inpurt structure

9.14.4.19 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

9.14.4.20 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

9.14.4.21 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

9.14.4.22 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

9.14.4.23 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

9.14.4.24 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

9.14.4.25 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

9.14.4.26 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

9.14.4.27 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

9.14.4.28 ULONG SLQSSwiClearDyingGaspStatistics ( )

This function Clear Dying GASP Statistics.

9.14.4.29 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

9.14.4.30 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

9.14.4.31 ULONG SLQSSwiGetDyingGaspCfg (getDyingGaspCfg * *pConfig*)

This function queries Dying GASP Config.

Parameters

pGetCustom-FeatureV2	• See getDyingGaspCfg for more information of the inpurt structure *

9.14.4.32 ULONG SLQSSwiGetDyingGaspStatistics (getDyingGaspStatistics * *pStatistics*)

This function queries Dying GASP Statistics.

Parameters

pStatistics	• See getDyingGaspStatistics for more information of the inpurt structure *

9.14.4.33 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

9.14.4.34 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

9.14.4.35 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

9.14.4.36 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

9.14.4.37 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

9.14.4.38 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

9.14.4.39 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

9.14.4.40 ULONG SLQSSwiSetDyingGaspCfg (setDyingGaspCfg * *pConfig*)

This function set Dying GASP Config.

Parameters

pConfig	• See setDyingGaspCfg for more information of the inpurt structure

9.14.4.41 ULONG SLQSSwiSetHostDevInfo (SLQSSwiSetHostDevInfoParams * *pSetHostDevInfoParams*)

This API Sets the host device info configured on the modem for OMA-DM reporting

Parameters

pSetHostDev-InfoParams	• See SLQSSwiSetHostDevInfoParams for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: NA
Timeout: 2 Secs

9.14.4.42 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

9.14.4.43 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

9.14.4.44 ULONG ValidateSPC (CHAR * *pSPC*)

This function Validates Service Programming code of the device

Parameters

pSPC[IN]	• NULL-terminated string representing a six-digit service programming code

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: CDMA
Device Supported: MC83x5, MC7750
Timeout: 2 seconds

9.15 qaGobiApiFms.h File Reference

Firmware Management Service API function prototypes.

Data Structures

- struct fwinfo_s
- struct slqsfwinfo_s
- struct qmifwinfo_s

- struct ImageElement
- struct PrefImageList
- struct ImageIdElement
- struct ImageIDEntries
- struct ImageList
- struct sGetDeviceSeriesResult
- struct SWI_STRUCT_CarrierImage
- struct CurrImageInfo
- struct CurrentImgList

Macros

- #define SLQSFWINFO_MODELID_SZ 20
- #define SLQSFWINFO_BOOTVERSION_SZ 85
- #define SLQSFWINFO_APPVERSION_SZ 85
- #define SLQSFWINFO_SKU_SZ 15
- #define SLQSFWINFO_PACKAGEID_SZ 85
- #define SLQSFWINFO_CARRIER_SZ 20
- #define SLQSFWINFO_PRIVERSION_SZ 16
- #define SLQSFWINFO_CUR_CARR_NAME 17
- #define SLQSFWINFO_CUR_CARR_REV 13
- #define GOBI_MBN_IMG_ID_STR_LEN 16
- #define GOBI_MBN_BUILD_ID_STR_LEN 100
- #define GOBI_LISTENTRIES_MAX 2
- #define GOBI_SET_IMG_PREF_RSPLEN 40
- #define DEVICE_SHUTDOWN 5
- #define DEVICE_RESET 4
- #define DEVICE_OFFLINE 3
- #define FIRMWARE_UPDATE_SUCCESS 0x01
- #define FIRMWARE_UPDATE_FAIL 0x01
- #define PRI_UPDATE_FAIL 0x02
- #define FIRMWARE_UPGRADE_SUCCESS 0x00
- #define IMG_ID_LEN 16
- #define BUILD_ID_LEN 100
- #define G3K_FIRMWARE_DOWNLOAD 1
- #define SPKG_FIRMWARE_DOWNLOAD 2
- #define UNIQUE_ID_LEN 16
- #define BUILD_ID_MAX_LEN 255
- #define IMGDETAILS_LEN 16
- #define MAX_IMAGE_IDE_ELEMENTS 50

Enumerations

- enum eGobiImageTech {
  eGOBI_IMG_TECH_CDMA = 0,
  eGOBI_IMG_TECH_UMTS }

- enum eGobiImageCarrier {
  eGOBI_IMG_CAR_GENERIC = 1,
  eGOBI_IMG_CAR_FACTORY,
  eGOBI_IMG_CAR_NORF,
  eGOBI_IMG_CAR_VERIZON = 101,
  eGOBI_IMG_CAR_SPRINT,
  eGOBI_IMG_CAR_ALLTEL,
  eGOBI_IMG_CAR_BELL,
  eGOBI_IMG_CAR_TELUS,
  eGOBI_IMG_CAR_US,
  eGOBI_IMG_CAR_TELSTRA1,
  eGOBI_IMG_CAR_CHINA_UNICOM,
  eGOBI_IMG_CAR_TELCOM_NZ,
  eGOBI_IMG_CAR_SK_TELCOM1,
  eGOBI_IMG_CAR_RELIANCE1,
  eGOBI_IMG_CAR_TATA,
  eGOBI_IMG_CAR_METROPCS,
  eGOBI_IMG_CAR_LEAP,
  eGOBI_IMG_CAR_KDDI,
  eGOBI_IMG_CAR_IUSACELL,
  eGOBI_IMG_CAR_CHINA_TELECOM,
  eGOBI_IMG_CAR_OMH,
  eGOBI_IMG_CAR_GENERIC_CDMA,
  eGOBI_IMG_CAR_ATT = 201,
  eGOBI_IMG_CAR_VODAFONE,
  eGOBI_IMG_CAR_TMOBILE,
  eGOBI_IMG_CAR_ORANGE,
  eGOBI_IMG_CAR_TELEFONICA,
  eGOBI_IMG_CAR_TELCOM_ITALIA,
  eGOBI_IMG_CAR_3,
  eGOBI_IMG_CAR_O2,
  eGOBI_IMG_CAR_SFR,
  eGOBI_IMG_CAR_SWISSCOM,
  eGOBI_IMG_CAR_CHINA_MOBILE,
  eGOBI_IMG_CAR_TELSTRA2,
  eGOBI_IMG_CAR_SINGTEL_OPTUS,
  eGOBI_IMG_CAR_RELIANCE2,
  eGOBI_IMG_CAR_BHARTI,
  eGOBI_IMG_CAR_NTT_DOCOMO,
  eGOBI_IMG_CAR_EMOBILE,
  eGOBI_IMG_CAR_SOFTBANK,
  eGOBI_IMG_CAR_KT_FREETEL,
  eGOBI_IMG_CAR_SK_TELCOM2,
  eGOBI_IMG_CAR_TELENOR,
  eGOBI_IMG_CAR_NETCOM,
  eGOBI_IMG_CAR_TELIASONERA,
  eGOBI_IMG_CAR_AMX_TELCEL,
  eGOBI_IMG_CAR_BRASIL_VIVO,
  eGOBI_IMG_CAR_AERIS,
  eGOBI_IMG_CAR_ROGERS }
- enum eGobiImageRegion {
  eGOBI_IMG_REG_NA = 0,
  eGOBI_IMG_REG_LA,
  eGOBI_IMG_REG_EU,
  eGOBI_IMG_REG_ASIA,
  eGOBI_IMG_REG_AUS,
  eGOBI_IMG_REG_GLOBAL }
- enum eGobiImageGPS {

eGOBI_IMG_GPS_NONE = 0,
eGOBI_IMG_GPS_STAND_ALONE,
eGOBI_IMG_GPS_ASSISTED,
eGOBI_IMG_GPS_NO_XTRA }

- enum eGobiDeviceSeries {
eGOBI_DEV_SERIES_UNKNOWN = -1,
eGOBI_DEV_SERIES_NON_GOBI = 0,
eGOBI_DEV_SERIES_G3K,
eGOBI_DEV_SERIES_SIERRA_GOBI,
eGOBI_DEV_SERIES_9X15,
eGOBI_DEV_SERIES_9X30,
eGOBI_DEV_SERIES_9X07_9X50,
eGobi_DEV_SERIES_MC83 }

Functions

- ULONG GetImageStore (WORD imageStorePathSize, CHAR *pImageStorePath)
- ULONG SLQSGetFirmwareInfo (struct qmifwinfo_s *pinfo)
- ULONG SLQSGetImageInfoMC77xx (LPCSTR path, struct qmifwinfo_s *pinfo)
- ULONG SLQSGetImageInfoMC83xx (LPCSTR path, struct qmifwinfo_s *pinfo)
- ULONG SLQSGetImageInfo (LPCSTR path, struct qmifwinfo_s *pinfo)
- ULONG UpgradeFirmware2k (CHAR *pDestinationPath)
- ULONG GetImagesPreference (ULONG *pImageListSize, struct PrefImageList *pImageList)
- ULONG SetImagesPreference (ULONG imageListSize, BYTE *pImageList, ULONG bForceDownload, BYTE modemIndex, ULONG *pImageTypesSize, BYTE *pImageTypes)
- ULONG GetStoredImages (ULONG *pImageListSize, BYTE *pImageList)
- ULONG DeleteStoredImage (ULONG imageInfoSize, BYTE *pImageInfo)
- ULONG SLQSGetImageInfo_9x15 (LPCSTR path, BYTE imgType, struct slqsfwinfo_s *pinfo)
- ULONG SLQSUpgradeFirmware9x07 (CHAR *pDestinationPath)
- ULONG SLQSUpgradeFirmware9x15 (CHAR *pDestinationPath)
- ULONG SLQSGetBootVersionNumber (ULONG *bootversion)
- BOOL SLQSIsSpkgFormatRequired (void)
- void SLQSSetSpkgFormatRequired (BYTE isneeded)
- ULONG upgrade_mc77xx_fw (LPCSTR path)
- void eGetDeviceSeries (struct sGetDeviceSeriesResult *)
- ULONG SLQSSwiGetAllCarrierImages (ULONG *pNumOfItems, struct SWI_STRUCT_CarrierImage *pCarrierImages, char *pFolderPath)
- ULONG DownloadToSlot (CHAR *path, struct slqsfwinfo_s fwImgInfo, struct slqsfwinfo_s priImgInfo, BYTE slot, BYTE forceDownload)
- ULONG SLQSDownloadFirmwareToSlot (CHAR *pPath, BYTE slot_index, BYTE force_download)
- ULONG SLQSGetValidFwPriCombinations (struct ImageList *pStoredImageList, ULONG *pValidCombinationSize, struct SWI_STRUCT_CarrierImage *pValidCombinations)
- ULONG SLQSSetSIMBasedImageSwitching (void)
- void SLQSSetCrashStateCheckIgnore (BOOL ignore)
- ULONG SLQSSwiGetFirmwareCurr (CurrentImgList *pCurrentImgList)

9.15.1 Detailed Description

Firmware Management Service API function prototypes.

9.15.2 Macro Definition Documentation

9.15.2.1 #define BUILD_ID_LEN 100

9.15.2.2 #define BUILD_ID_MAX_LEN 255

9.15.2.3 #define DEVICE_OFFLINE 3

9.15.2.4 #define DEVICE_RESET 4

9.15.2.5 #define DEVICE_SHUTDOWN 5

9.15.2.6 #define FIRMWARE_UPDATE_FAIL 0x01

9.15.2.7 #define FIRMWARE_UPDATE_SUCCESS 0x01

9.15.2.8 #define FIRMWARE_UPGRADE_SUCCESS 0x00

9.15.2.9 #define G3K_FIRMWARE_DOWNLOAD 1

9.15.2.10 #define GOBI_LISTENTRIES_MAX 2

9.15.2.11 #define GOBI_MBN_BUILD_ID_STR_LEN 100

9.15.2.12 #define GOBI_MBN_IMG_ID_STR_LEN 16

9.15.2.13 #define GOBI_SET_IMG_PREF_RSPLEN 40

9.15.2.14 #define IMG_ID_LEN 16

9.15.2.15 #define IMGDETAILS_LEN 16

9.15.2.16 #define MAX_IMAGE_IDE_ELEMENTS 50

9.15.2.17 #define PRI_UPDATE_FAIL 0x02

9.15.2.18 #define SLQSFWINFO_APPVERSION_SZ 85

9.15.2.19 #define SLQSFWINFO_BOOTVERSION_SZ 85

9.15.2.20 #define SLQSFWINFO_CARRIER_SZ 20

9.15.2.21 #define SLQSFWINFO_CUR_CARR_NAME 17

9.15.2.22 #define SLQSFWINFO_CUR_CARR_REV 13

9.15.2.23 #define SLQSFWINFO_MODELID_SZ 20

9.15.2.24 #define SLQSFWINFO_PACKAGEID_SZ 85

9.15.2.25 #define SLQSFWINFO_PRIVERSION_SZ 16

9.15.2.26 #define SLQSFWINFO_SKU_SZ 15

9.15.2.27 #define SPKG_FIRMWARE_DOWNLOAD 2

9.15.2.28 #define UNIQUE_ID_LEN 16

9.15.3 Enumeration Type Documentation

9.15.3.1 enum eGobiDeviceSeries

enumeration which lists the Device Series

Enumerator

eGOBI_DEV_SERIES_UNKNOWN
eGOBI_DEV_SERIES_NON_GOBI
eGOBI_DEV_SERIES_G3K
eGOBI_DEV_SERIES_SIERRA_GOBI
eGOBI_DEV_SERIES_9X15
eGOBI_DEV_SERIES_9X30
eGOBI_DEV_SERIES_9X07_9X50
eGobi_DEV_SERIES_MC83

9.15.3.2 enum eGobiImageCarrier

enumeration which lists the carrier supported by the image

Enumerator

eGOBI_IMG_CAR_GENERIC
eGOBI_IMG_CAR_FACTORY
eGOBI_IMG_CAR_NORF
eGOBI_IMG_CAR_VERIZON
eGOBI_IMG_CAR_SPRINT
eGOBI_IMG_CAR_ALLTEL
eGOBI_IMG_CAR_BELL
eGOBI_IMG_CAR_TELUS
eGOBI_IMG_CAR_US
eGOBI_IMG_CAR_TELSTRA1
eGOBI_IMG_CAR_CHINA_UNICOM
eGOBI_IMG_CAR_TELCOM_NZ
eGOBI_IMG_CAR_SK_TELCOM1
eGOBI_IMG_CAR_RELIANCE1
eGOBI_IMG_CAR_TATA
eGOBI_IMG_CAR_METROPCS
eGOBI_IMG_CAR_LEAP
eGOBI_IMG_CAR_KDDI
eGOBI_IMG_CAR_IUSACELL
eGOBI_IMG_CAR_CHINA_TELECOM
eGOBI_IMG_CAR_OMH
eGOBI_IMG_CAR_GENERIC_CDMA
eGOBI_IMG_CAR_ATT
eGOBI_IMG_CAR_VODAFONE

eGOBI_IMG_CAR_TMOBILE
eGOBI_IMG_CAR_ORANGE
eGOBI_IMG_CAR_TELEFONICA
eGOBI_IMG_CAR_TELCOM_ITALIA
eGOBI_IMG_CAR_3
eGOBI_IMG_CAR_O2
eGOBI_IMG_CAR_SFR
eGOBI_IMG_CAR_SWISSCOM
eGOBI_IMG_CAR_CHINA_MOBILE
eGOBI_IMG_CAR_TELSTRA2
eGOBI_IMG_CAR_SINGTEL_OPTUS
eGOBI_IMG_CAR_RELIANCE2
eGOBI_IMG_CAR_BHARTI
eGOBI_IMG_CAR_NTT_DOCOMO
eGOBI_IMG_CAR_EMOBILE
eGOBI_IMG_CAR_SOFTBANK
eGOBI_IMG_CAR_KT_FREETEL
eGOBI_IMG_CAR_SK_TELCOM2
eGOBI_IMG_CAR_TELENOR
eGOBI_IMG_CAR_NETCOM
eGOBI_IMG_CAR_TELIASONERA
eGOBI_IMG_CAR_AMX_TELCEL
eGOBI_IMG_CAR_BRASIL_VIVO
eGOBI_IMG_CAR_AERIS
eGOBI_IMG_CAR_ROGERS

9.15.3.3 enum eGobiImageGPS

enumeration which lists the GPS type supported by the image

Enumerator

eGOBI_IMG_GPS_NONE
eGOBI_IMG_GPS_STAND_ALONE
eGOBI_IMG_GPS_ASSISTED
eGOBI_IMG_GPS_NO_XTRA

9.15.3.4 enum eGobiImageRegion

enumeration which lists the region supported by the image

Enumerator

eGOBI_IMG_REG_NA
eGOBI_IMG_REG_LA
eGOBI_IMG_REG_EU
eGOBI_IMG_REG_ASIA
eGOBI_IMG_REG_AUS
eGOBI_IMG_REG_GLOBAL

9.15.3.5 enum eGobiImageTech

enumeration which lists the technology supported by the image

Enumerator

eGOBI_IMG_TECH_CDMA

eGOBI_IMG_TECH_UMTS

9.15.4 Function Documentation

9.15.4.1 ULONG DeleteStoredImage (ULONG *imageInfoSize,* BYTE * *pImageInfo*)

Used to delete the specified image from the device. This API function is only relevant to devices with the ability to store multiple firmware images(see Device Supported section below).

Parameters

imageInfoSize[IN]	• The size in BYTEs of the image info array
pImageInfo[IN]	• The image info list array containing information about the image to be deleted. • See ImageElement

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Device Supported: MC83x5/SL9090
Timeout: 2 Secs

9.15.4.2 ULONG DownloadToSlot (CHAR * *path,* struct slqsfwinfo_s *fwImgInfo,* struct slqsfwinfo_s *priImgInfo,* BYTE *slot,* BYTE *forceDownload*)

This API is dedicated for downloading a firmware image into a specific slot. it is used for EM74xx/MC74xx only

Parameters

path	• fully qualified path to firmware image to download
fwImgInfo	• See struct slqsfwinfo_s
priImgInfo	• See struct slqsfwinfo_s

slot	• slot id in the modem to store the firmware
forceDownload	• a flag to force download take place.

Returns

See qmerrno.h for eQCWWAN_xxx error values

Note

only EM74xx/MC74xx is supported

9.15.4.3 void eGetDeviceSeries (struct sGetDeviceSeriesResult *)

Name : eGetDeviceSeries

Parameters

none	

Returns

sGetDeviceSeriesResult

Note

Get Devie Series

9.15.4.4 ULONG GetImagesPreference (ULONG * *pImageListSize,* struct PrefImageList * *pImageList*)

restore original alignment from stack Gets the current images preference from the device.

Parameters

pImageListSize[-IN/OUT]	• Upon input, the size of structure ImageList ImageList • Upon successful output, the number of BYTEs copied to the image list array
pImageList[OUT]	• The caller must supply a pointer to a ImageList structure typecast as a BYTE pointer

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 2 seconds

9.15.4.5 ULONG GetImageStore (WORD *imageStorePathSize,* CHAR * *pImageStorePath*)

Returns the image store folder, i.e., the folder containing one or more carrier-specific image subfolders compatible with the currently connected QC WWAN device.

Parameters

imageStorePathSize	• Maximum number of characters (including NULL terminator) that can be copied to the image store path array.
pImageStorePath[OUT]	• The path to the image store

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.15.4.6 ULONG GetStoredImages (ULONG * *pImageListSize,* BYTE * *pImageList*)

restore original alignment from stack Gets the list of images stored on the device.

Parameters

pImageListSize[IN/OUT]	• Upon input, the size of structure ImageList ImageList • Upon successful output, the number of BYTEs copied to the image list array
pImageList[OUT]	• The caller must supply a pointer to a ImageList structure typecast as a BYTE pointer

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Device Supported: MC83x5/SL9090
Timeout: 2 seconds

9.15.4.7 ULONG SetImagesPreference (ULONG *imageListSize,* BYTE * *pImageList,* ULONG *bForceDownload,* BYTE *modemIndex,* ULONG * *pImageTypesSize,* BYTE * *pImageTypes*)

Sets the current images preference on the device. After this function successfully completes, the device must be reset for the selected image preference to be realized. Additionally, when the returned list of image types that require downloading is not empty, the device opens in QDL mode after the reset. At that point, the QDL portion of this API must be used to download the selected image preference to the device.

Parameters

imageListSize	• The size in BYTEs of the image list array
pImageList[IN]	• The image info list array containing Image Elements – See PrefImageList
bForce-Download[IN]	• Force device to download images from host? 0 - No Nonzero - Yes
modemIndex	• Desired storage index for downloaded modem image (optional, a value of 0xFF indicates unspecified)
pImageTypes-Size[IN/OUT]	• Upon input, maximum number of elements that download image types array can contain • Upon successful output, number of elements in download image types array
pImageTypes[OUT]	-The download image types array.

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 2 seconds

9.15.4.8 ULONG SLQSDownloadFirmwareToSlot (CHAR * *pPath,* BYTE *slot_index,* BYTE *force_download*)

This API is used to download firmware to a specific slot id of the modem. It is only applicable for EM74xx variant. This API encapsulates all steps involved in the firmware download process.Hence it is a blocking API call.

This API will not return until the entire process has been completed. This API will takes significant amount of time (in order of minutes, normally should be less than 10 minutes).

This API Performs the following steps:

1. Verifies arguments.
2. Retrieve and store the details of the firmware and the PRI file

3. Enable device state change callback.
4. Enable firmware download callback.
5. Set Image preference on the device and reset the device.
6. Wait for the firmware to download and device to become ready.
7. Check the firmware update status. If fail, return an error.
8. If update status is OK, check if current image preference and preferred image preference(from step 2) match
9. If match, firmware download is successful. otherwise, report FW_PREFERENCE_MISMATCH

   10.Disable callbacks and exit.

   The call to this API blocks until step 7 or 10. This could be a significant amount of time (in order of minutes). Also note that the device state change callback and firmware download callback are used internally

   within this API. Hence the user application's instance of these callbacks (if any) are cleared. The user must re-enable these callbacks after a call to this API in order to use them.

Parameters

pPath[IN]	• fully qualified path to firmware image to download.
slot_index[IN]	• slot id in the modem to store the firmware
force_download[IN]	• a flag to force download take place.

Returns

- eQCWWAN_ERR_NONE - Firmware download/Switch success.
- eQCWWAN_ERR_INVALID_ARG - The path input does not contain any image
- eQCWWAN_ERR_SWIIM_FW_UPDATE_FAIL - Firmware download/switch failed
- eQCWWAN_ERR_SWIIM_FW_PREFERENCE_MISMATCH - Download success but device offline due to image preference mismatch (ref. syslogs for cause)

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: NA

9.15.4.9 ULONG SLQSGetBootVersionNumber (ULONG * *bootversion*)

Gets the boot loader version number

Parameters

bootversion[OUT]	• boot loader version presented by a 4 byte integer

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Device Supported: MC9090/SL9090
Timeout: 2 seconds

9.15.4.10 ULONG SLQSGetFirmwareInfo (struct qmifwinfo_s * *pinfo*)

Returns firmware image information from the connected device

Parameters

pinfo[OUT]	• firmware image information record

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values
struct qmifwinfo_s

Note

Timeout: 2 Seconds.

9.15.4.11 ULONG SLQSGetImageInfo (LPCSTR *path,* struct qmifwinfo_s * *pinfo*)

Returns firmware image information from a CWE file or mbn files stored on the host. For CWE, information is returned for the first CWE image found at the specified path. For MBN, the provided path must be located under the image store for the currently connected QC WWAN device. Note that as this API supports multiple firmware image types, it relies on the presence of a supported device. Otherwise, refer to SLQSGetImageInfoMC83xx and SLQSGetImageInfoMC77xx for APIs which do not rely on the presence of a supported device.

Parameters

path[IN]	• fully qualified path to folder containing CWE image or MBN images • should use a "/" at the end of the path.
pinfo[OUT]	• firmware image information record

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values
struct qmifwinfo_s

Note

Timeout: N/A

9.15.4.12 ULONG SLQSGetImageInfo_9x15 (LPCSTR *path*, BYTE *imgType*, struct slqsfwinfo_s * *pinfo*)

Returns firmware image information from a CWE file(s) stored on the host. It does not rely on the presence of a supported device.

Parameters

path[IN]	• fully qualified path to folder containing the image(s) • should use a "/" at the end of the path.
imgType[IN]	• 2 - Firmware Image(.cwe extension) • 3 - PRI Image (.nvu extension)
pinfo[OUT]	• firmware image information record

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values
struct qmifwinfo_s

Note

Device Supported: MC73xx
Timeout: N/A

9.15.4.13 ULONG SLQSGetImageInfoMC77xx (LPCSTR *path*, struct qmifwinfo_s * *pinfo*)

Returns firmware image information from a SPKGS CWE file stored on the host. The information is returned for the first SPKGS CWE image found at the specified path. This API executes independent of a MC77xx being connected to the target.

Parameters

path[IN]	• fully qualified path to folder containing SPKG CWE image • should use a "/" at the end of the path.
pinfo[OUT]	• firmware image information record

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values
struct qmifwinfo_s

Note

Device Supported: MC77xx
Timeout: N/A

9.15.4.14 ULONG SLQSGetImageInfoMC83xx (LPCSTR *path,* struct qmifwinfo_s * *pinfo*)

Returns firmware image information from an MBN file located on the host. This API executes independent of a MC83xx being connected to the target.

Parameters

path[IN]	• fully qualified path to folder containing MBN file • should use a "/" at the end of the path.
pinfo[OUT]	• firmware image information record

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values
struct qmifwinfo_s

Note

Device Supported: MC83xx/SL9090
Timeout: N/A

9.15.4.15 ULONG SLQSGetValidFwPriCombinations (struct ImageList * *pStoredImageList*, ULONG * *pValidCombinationSize*, struct SWI_STRUCT_CarrierImage * *pValidCombinations*)

This API distills valid Firmware/PRI combinations from GetStoredImages result

Parameters

in	*pStoredImage-List*	• image list returned from GetStoredImages
in,out	*pValid-CombinationSize*	• number of combination passed in and returned
out	*pValid-Combinations*	• valid combinations returned

Returns

- eQCWWAN_ERR_INVALID_ARG - Invalid parameters
- eQCWWAN_ERR_BUFFER_SZ - No enough element to store combinatons returned

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.15.4.16 BOOL SLQSIsSpkgFormatRequired (void)

Check if SPKG format download is required for SL9090/MC9090, it returns the value whcih was set by API SLQS-SetSpkgFormatRequired()

Parameters

none	

Returns

return TRUE if required, otherwise, return FALSE

Note

Device Supported: MC9090/SL9090

9.15.4.17 void SLQSSetCrashStateCheckIgnore (BOOL *ignore*)

This API is used to set whether ignore crash state checking before proceed firmware download using the API UpgradeFirmware2k().

Parameters

in	*ignore*	• 0 - crash state checking applied (default value) • 1 - ignore crash state checking

Note

Device Supported: EM73xx/MC73xx, EM74xx/MC74xx

9.15.4.18 ULONG SLQSSetSIMBasedImageSwitching (void)

This API is used to enable the SIM-based Image Switching. The modem will reboot automatically to take effect of the enabing

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Device Supported: EM74xx/MC74xx

9.15.4.19 void SLQSSetSpkgFormatRequired (BYTE *isneeded*)

Set if SPKG format download is required for SL9090/MC9090

Parameters

isneeded	[INPUT] user inputs the firmware download method preference • 1 - Gobi3K download method, use mbn files. This is default value • 2 - SPKG download method, use cwe file

Returns

None

Note

Device Supported: MC9090/SL9090

9.15.4.20 ULONG SLQSSwiGetAllCarrierImages (ULONG * *pNumOfItems,* struct SWI_STRUCT_CarrierImage * *pCarrierImages,* char * *pFolderPath*)

This API gets a list of all images stored on both the host and the device

Parameters

pNumOfItems	• Number of Images{IN/OUT]
pCarrierImages[-OUT]	• See SWI_STRUCT_CarrierImage
pFolderPath	• Path of Input folder [IN]

Returns

TRUE/FALSE

Note

In case pFolderPath is invalid, API does not return invalid path error as SLQSSwiGetAllCarrierImages get carrier images from device also.

9.15.4.21 ULONG SLQSSwiGetFirmwareCurr (CurrentImgList * *pCurrentImgList*)

This API gets the currently active images on the device.

Parameters

pCurrentImgList	• Pointer to structure CurrentImgList • See CurrentImgList for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: NA
Device Supported: MC73xx, MC74xx/EM74xx
Timeout: 5 Secs

9.15.4.22 ULONG SLQSUpgradeFirmware9x07 (CHAR * *pDestinationPath*)

This API is used to download firmware to a WP760x device.

This API Performs the following steps:

1. Verifies arguments.
2. Informs the SDK of the firmware upgrade path
3. Send the DM command to requests the device reset (device will reset after all open handles are released).

   Upon successful completion, the above steps will have been completed, however, the actual upgrade of the firmware will necessarily then follow.

Parameters

pDestination-Path[IN]	• fully qualified path to firmware image to download.

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 5 seconds

9.15.4.23 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

9.15.4.24 ULONG upgrade_mc77xx_fw (LPCSTR *path*)

9.15.4.25 ULONG UpgradeFirmware2k (CHAR * *pDestinationPath*)

This API is used to download firmware to a MC77xx or Gobi 3000 device. For SL909/MC9090, SLQSSetSpkgFormatRequired() needs to be called in advance to specify the download method, Gobi3K or SPKG download, please refer to the API SLQSSetSpkgFormatRequired() for more details of the input values. If SLQSSetSpkgFormatRequired() is not called in advance, it will us Gobi3K firmware download method (MBN files) as the default download method.

This API Performs the following steps:

1. Verifies arguments.
2. Verify that device Crash State should be 1 (RESET State).
3. Informs the SDK of the firmware upgrade path
4. Updates the images preference on the currently connected device.
5. Requests the device reset (device will reset after all open handles are released).

   Upon successful completion, the above steps will have been completed, however, the actual upgrade of the firmware will necessarily then follow.

Parameters

pDestination-Path[IN]	• fully qualified path to firmware image to download. The path must end with a forward slash. For a Gobi 3000 device the path should specify the carrier image folder index i.e. "<path\to\carrier\image>/<carrier index>/" where <carrier index>="">" is a valid sub-directory entry. For 9x30 devices if pDestinationPath is not valid on host, it will use pseudo path for image switching.

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 12 seconds

9.16 qaGobiApiIms.h File Reference

IMS Service API function prototypes.

Data Structures

- struct SetSIPConfigReq
- struct SetSIPConfigResp
- struct SetRegMgrConfigReq
- struct SetRegMgrConfigResp
- struct SetIMSSMSConfigReq
- struct SetIMSSMSConfigResp
- struct SetIMSUserConfigReq
- struct SetIMSUserConfigResp
- struct SetIMSVoIPConfigReq
- struct SetIMSVoIPConfigResp
- struct GetSIPConfigResp
- struct GetRegMgrConfigParams
- struct GetIMSSMSConfigParams
- struct GetIMSUserConfigParams
- struct GetIMSVoIPConfigResp
- struct imsCfgIndRegisterInfo

Functions

- ULONG SLQSSetSIPConfig (SetSIPConfigReq *pSetSIPConfigReq, SetSIPConfigResp *pSetSIPConfigResp)
- ULONG SLQSSetRegMgrConfig (SetRegMgrConfigReq *pSetRegMgrConfigReq, SetRegMgrConfigResp *pSetRegMgrConfigResp)
- ULONG SLQSSetIMSSMSConfig (SetIMSSMSConfigReq *pSetIMSSMSConfigReq, SetIMSSMSConfigResp *pSetIMSSMSConfigResp)
- ULONG SLQSSetIMSUserConfig (SetIMSUserConfigReq *pSetIMSUserConfigReq, SetIMSUserConfigResp *pSetIMSUserConfigResp)
- ULONG SLQSSetIMSVoIPConfig (SetIMSVoIPConfigReq *pSetIMSVoIPConfigReq, SetIMSVoIPConfigResp *pSetIMSVoIPConfigResp)
- ULONG SLQSGetSIPConfig (GetSIPConfigResp *pGetSIPConfigResp)
- ULONG SLQSGetRegMgrConfig (GetRegMgrConfigParams *pGetRegMgrConfigParams)
- ULONG SLQSGetIMSSMSConfig (GetIMSSMSConfigParams *pGetIMSSMSConfigParams)
- ULONG SLQSGetIMSUserConfig (GetIMSUserConfigParams *pGetIMSUserConfigParams)
- ULONG SLQSGetIMSVoIPConfig (GetIMSVoIPConfigResp *pGetIMSVoIPConfigResp)
- ULONG SLQSImsConfigIndicationRegister (imsCfgIndRegisterInfo *pImsCfgIndRegisterInfo)

9.16.1 Detailed Description

IMS Service API function prototypes.

9.16.2 Function Documentation

9.16.2.1 ULONG SLQSGetIMSSMSConfig (GetIMSSMSConfigParams * *pGetIMSSMSConfigParams*)

This API retrieves the SMS configuration parameters.

Parameters

pGetIMSSMSConfigParams[IN/OUT]	• See GetIMSSMSConfigParams for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: NA
Device Supported: MC73xx, MC74xx and EM74xx
Timeout: 5 seconds

9.16.2.2 ULONG SLQSGetIMSUserConfig (GetIMSUserConfigParams * *pGetIMSUserConfigParams*)

This API retrieves the IMS User configuration parameters.

Parameters

pGetIMSUserConfigParams[IN/OUT]	• See GetIMSUserConfigParams for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: NA
Device Supported: MC73xx, MC74xx and EM74xx
Timeout: 5 seconds

9.16.2.3 ULONG SLQSGetIMSVoIPConfig (GetIMSVoIPConfigResp * *pGetIMSVoIPConfigResp*)

This API retrieves the IMS VoIP configuration parameters.

Parameters

GetIMSVoIPConfigResp[OUT]	• See GetIMSVoIPConfigResp for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: NA
Device Supported: MC73xx, MC74xx and EM74xx
Timeout: 5 seconds

9.16.2.4 ULONG SLQSGetRegMgrConfig (GetRegMgrConfigParams * *pGetRegMgrConfigParams*)

This API retrieves the registration manager configuration parameters.

Parameters

pGetRegMgrConfigParams[IN/OUT]	• See GetRegMgrConfigResp for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: NA
Device Supported: MC73xx, MC74xx and EM74xx
Timeout: 5 seconds

9.16.2.5 ULONG SLQSGetSIPConfig (GetSIPConfigResp * *pGetSIPConfigResp*)

This API retrieves the Session Initiation Protocol(SIP) configuration parameters.

Parameters

pGetSIPConfigResp[OUT]	• See GetSIPConfigResp for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: NA
Device Supported: MC73xx, MC74xx and EM74xx
Timeout: 5 seconds

9.16.2.6 ULONG SLQSImsConfigIndicationRegister (imsCfgIndRegisterInfo * *pImsCfgIndRegisterInfo*)

Sets the registration state for different QMI_IMS indications for the requesting control point

Parameters

pImsCfgInd-RegisterInfo[IN]	• Structure containing Indication Register Information. – See imsCfgIndRegisterInfo for more informtaion.

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: UMTS
Device Supported: MC73xx, MC74xx and EM74xx
Timeout: 10 Secs

This API is used by a device to register/deregister for different QMI IMS indications. The device's registration state variables that control registration for indications will be modified to reflect the settings indicated in the request message. At least one optional parameter must be present in the request.

9.16.2.7 ULONG SLQSSetIMSSMSConfig (SetIMSSMSConfigReq * *pSetIMSSMSConfigReq*, SetIMSSMSConfigResp * *pSetIMSSMSConfigResp*)

This API sets the IMS SMS configuration parameters for the requesting control point.

Parameters

pSetIMSSMS-ConfigReq[IN]	• See SetIMSSMSConfigReq for more information
pSetIMSSMS-ConfigResp[OU-T]	• See SetIMSSMSConfigResp for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: NA
Device Supported: MC73xx, MC74xx and EM74xx
Timeout: 5 seconds

9.16.2.8 ULONG SLQSSetIMSUserConfig (SetIMSUserConfigReq * *pSetIMSUserConfigReq*, SetIMSUserConfigResp * *pSetIMSUserConfigResp*)

This API sets the IMS user configuration parameters for the requesting control point.

Parameters

pSetIMSUserConfigReq[IN]	• See SetIMSUserConfigReq for more information
pSetIMSUserConfigResp[OUT]	• See SetIMSUserConfigResp for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: NA
Device Supported: MC73xx, MC74xx and EM74xx
Timeout: 5 seconds

9.16.2.9 ULONG SLQSSetIMSVoIPConfig (SetIMSVoIPConfigReq * *pSetIMSVoIPConfigReq*, SetIMSVoIPConfigResp * *pSetIMSVoIPConfigResp*)

This API sets the IMS Voice over Internet Protocol (VoIP) configuration parameters for the requesting control point.

Parameters

pSetIMSVoIPConfigReq[IN]	• See SetIMSVoIPConfigReq for more information
pSetIMSVoIPConfigResp[OUT]	• See SetIMSVoIPConfigResp for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: NA
Device Supported: MC73xx, MC74xx and EM74xx
Timeout: 5 seconds

9.16.2.10 ULONG SLQSSetRegMgrConfig (SetRegMgrConfigReq * *pSetRegMgrConfigReq*, SetRegMgrConfigResp * *pSetRegMgrConfigResp*)

This API sets the IMS registration manager configuration parameters for the requesting control point.

Parameters

pSetRegMgrConfigReq[IN]	• See SetRegMgrConfigReq for more information
pSetRegMgrConfigResp[OUT]	• See SetRegMgrConfigResp for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: NA
Device Supported: MC73xx, MC74xx and EM74xx
Timeout: 5 seconds

9.16.2.11 ULONG SLQSSetSIPConfig (SetSIPConfigReq * *pSetSIPConfigReq*, SetSIPConfigResp * *pSetSIPConfigResp*)

This API sets the IMS Session Initiation Protocol(SIP) configuration parameters for the requesting control point.

Parameters

pSetSIPConfigReq[IN]	• See SetSIPConfigReq for more information
pSetSIPConfigResp[OUT]	• See SetSIPConfigResp for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: NA
Device Supported: MC73xx, MC74xx and EM74xx
Timeout: 5 seconds

9.17 qaGobiApiImsa.h File Reference

IMSA Service API function prototypes.

Data Structures

- struct IMSAIndRegisterInfo
- struct SupportedMsgList
- struct IMSASupportedMsgInfo
- struct ReqFieldsList
- struct RespFieldsList
- struct IndFieldsList
- struct IMSASupportedFieldsResp
- struct IMSARegistrationStatus
- struct IMSAServiceStatus

Functions

- ULONG SLQSRegisterIMSAIndication (IMSAIndRegisterInfo *pImsaIndRegisterInfo)
- ULONG SLQSGetIMSASupportedMsg (IMSASupportedMsgInfo *pIMSASupportedMsgInfo)
- ULONG SLQSGetIMSASupportedFields (WORD messageID, IMSASupportedFieldsResp *pIMSASupportedFieldsResp)
- ULONG SLQSGetIMSARegStatus (IMSARegistrationStatus *pIMSARegistrationStatus)
- ULONG SLQSGetIMSAServiceStatus (IMSAServiceStatus *pIMSAServiceStatus)

9.17.1 Detailed Description

IMSA Service API function prototypes.

9.17.2 Function Documentation

9.17.2.1 ULONG SLQSGetIMSARegStatus (IMSARegistrationStatus * *pIMSARegistrationStatus*)

Queries the set of messages implemented by the currently running software.

Parameters

pIMSA-Registration-Status[OUT]	• Structure containing response parameters for registration status. – See IMSARegistrationStatus for more information.

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 5 Secs

```
This API is used by a device to get the registration status for
various IMS services for the requesting control point.
```

9.17.2.2 ULONG SLQSGetIMSAServiceStatus (IMSAServiceStatus * *pIMSAServiceStatus*)

Gets the service status for various IMS services for the requesting control point.

Parameters

pIMSAService-Status[OUT]	• Structure containing response parameters for service status. – See IMSAServiceStatus for more information.

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 5 Secs

```
This API is used by a device to Gets the service status for various
IMS services for the requesting control point.
```

9.17.2.3 ULONG SLQSGetIMSASupportedFields (WORD *messageID,* IMSASupportedFieldsResp * *pIMSASupportedFieldsResp*)

Queries the set of supported fields implemented by the currently running software.

Parameters

messageID[IN]	• Service Message ID.
pIMSA-Supported-FieldsResp[OU-T]	• Structure containing Supported Fields Response. – 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

pIMSASupportedMsgInfo[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

pImsaIndRegisterInfo[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

pLocGetServer-Req[IN]	• See LocGetServerReq for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 5 seconds

9.18.3.5 ULONG SLQSLOCInjectPosition (LocInjectPositionReq * *pLocInjectPositionReq*)

Injects a position to the location engine.

Parameters

pLocInject-PositionReq[IN]	• See LocInjectPositionReq for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 5 seconds

9.18.3.6 ULONG SLQSLOCInjectSensorData (LocInjectSensorDataReq * *pLocInjectSensorDataReq*)

Control point to to inject sensor data into the GNSS location engine.

Parameters

pLocInject-SensorData-Req[IN]	• See LocInjectSensorDataReq for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 5 seconds

9.18.3.7 ULONG SLQSLOCInjectUTCTime (ULONGLONG *timeMsec*, ULONG *timeUncMsec*)

Injects UTC time in the location engine.

Parameters

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

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 5 seconds

9.18.3.8 ULONG SLQSLOCSetCradleMountConfig (LocSetCradleMountReq * *pLocSetCradleMountReq*)

Control point to set the current cradle mount configuration.

Parameters

pLocSetCradleMountReq[IN]	• See LocSetCradleMountReq for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 5 seconds

9.18.3.9 ULONG SLQSLOCSetExtPowerState (LOCExtPowerStateReqResp * *pLOCExtPowerStateReqResp*)

Used by the control point to set the current external power configuration.

Parameters

pLOCExtPowerStateReqResp[IN]	• See LOCExtPowerStateReqResp for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 5 seconds

9.18.3.10 ULONG SLQSLOCSetOpMode (ULONG *mode*)

Used by the control point to tells the engine to use the specified operation mode while making the position fixes

Parameters

mode[IN]	• Valid values: – 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[OUT]	• status of network scan • 0x00 - scan successful • 0x01 - scan was aborted • 0x02 - scan did not complete due to a radio link failure recovery in progress

9.19.3.3 typedef struct _sysSelectPrefInfo sysSelectPrefInfo

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

Parameters

pEmerMode	• Optional parameter specifying the emergency Mode • Values: – 0 - OFF (normal) – 1 - ON (Emergency) • function SLQSGetSysSelectionPref() returns a default value FF if no value is returned by the device.
pModePref	• Optional parameter • Bit Mask indicating the radio technology mode preference • Bit values: – Bit 0 - cdma2000 1x – Bit 1 - cdma2000 HRPD(1xEV-DO) – Bit 2 - GSM – Bit 3 - UMTS – Bit 4 - LTE • function SLQSGetSysSelectionPref() returns a default value FF if no value is returned by the device.

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

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

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

pNetSelPref	• Optional parameter indicating network selection preference • Values: – 0x00 - Automatic network selection – 0x01 - Manual network selection • function SLQSGetSysSelectionPref() returns a default value FF if no value is returned by the device.
pSrvDomainPref	• Optional parameter indicating Service domain preference • Values: – 0x00 - Circuit switched only – 0x01 - Packet switched only – 0x02 - Circuit switched and packet switched – 0x03 - Packet switched attach – 0x04 - Packet switched detach • function SLQSGetSysSelectionPref() returns a default value FFFFFFFF if no value is returned by the device.
pGWAcqOrder-Pref	• Optional parameter indicating GSM/WCDMA Acquisition order Preference • Values: – 0x00 - Automatic – 0x01 - GSM then WCDMA – 0x02 - WCDMA then GSM • function SLQSGetSysSelectionPref() returns a default value FFFFFFFF if no value is returned by the device.

Note

None

9.19.3.4 typedef struct _sysSelectPrefParams sysSelectPrefParams

Contain the system selection preferences.

Parameters

pEmerMode	• Optional parameter specifying the emergency Mode • Values: – 0 - OFF (normal) – 1 - ON (Emergency)
pModePref	• Optional parameter • Bit Mask indicating the radio technology mode preference • Bit values: – Bit 0 - cdma2000 1x – Bit 1 - cdma2000 HRPD(1xEV-DO) – Bit 2 - GSM – Bit 3 - UMTS – Bit 4 - LTE

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

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

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

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

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

9.19.4 Enumeration Type Documentation

9.19.4.1 enum _NAMS_RADIO_IF_TECHNOLOGY_

Enumerator

eNAS_RADIO_IF_GSM

eNAS_RADIO_IF_UMTS

eNAS_RADIO_IF_LTE

eNAS_RADIO_IF_TDSCDMA

9.19.4.2 enum eSYS_SRV_DOMAIN

Enumerator

eSYS_SRV_DOMAIN_NO_SRV

eSYS_SRV_DOMAIN_CS_ONLY

eSYS_SRV_DOMAIN_PS_ONLY

eSYS_SRV_DOMAIN_CS_PS

eSYS_SRV_DOMAIN_CAMPED

eSYS_SRV_DOMAIN_UNKNOWN

9.19.4.3 enum NAS_LTE_CPHY_CA_BW_NRB

Enumerator

eNAS_LTE_CPHY_CA_BW_NRB_6

eNAS_LTE_CPHY_CA_BW_NRB_15

eNAS_LTE_CPHY_CA_BW_NRB_25

eNAS_LTE_CPHY_CA_BW_NRB_50

eNAS_LTE_CPHY_CA_BW_NRB_75

eNAS_LTE_CPHY_CA_BW_NRB_100

9.19.4.4 enum NAS_LTE_CPHY_SCELL_STATE

Enumerator

eNAS_LTE_CPHY_SCELL_STATE_DECONFIGURED

eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED

eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED

9.19.5 Function Documentation

9.19.5.1 ULONG GetACCOLC (BYTE * *pACCOLC*)

Retrieves information about the access overload class (ACCOLC)

Parameters

pACCOLC[OUT]	• ACCOLC : Valid range is 0 to 15

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: CDMA
Timeout: 2 seconds

9.19.5.2 ULONG GetANAAAAuthenticationStatus (ULONG * *pStatus*)

AN-AAA authentication status of the device.

Parameters

pStatus[OUT]	• Status of last AN-AAA authentication attempt – 0 - Failure – 1 - Success – 2 - Not Requested

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: CDMA
Timeout: 2 seconds

9.19.5.3 ULONG GetCDMANetworkParameters (BYTE * *pSCI*, BYTE * *pSCM*, BYTE * *pRegHomeSID*, BYTE * *pRegForeignSID*, BYTE * *pRegForeignNID*, BYTE * *pForceRev0*, BYTE * *pCustomSCP*, ULONG * *pProtocol*, ULONG * *pBroadcast*, ULONG * *pApplication*, ULONG * *pRoaming*)

Gets the current CDMA network parameters

Parameters

pSCI[OUT]	• Slot cycle index – 0xFF-Unknown
pSCM[OUT]	• Station class mark – 0xFF-Unknown
pRegHomeSID[OUT]	• Register on home SID – 0 - Disabled – 1 - Enabled – 0xFF - Unknown
pRegForeignSID[OUT]	• Register on foreign SID – 0 - Disabled – 1 - Enabled – 0xFF - Unknown
pRegForeignNID[OUT]	• Register on foreign NID – 0 - Disabled – 1 - Enabled – 0xFF - Unknown
pForceRev0[OUT]	• Force CDMA 1x-EV-DO Rev. 0 mode – 0 - Disabled – 1 - Enabled – 0xFF - Unknown

pCustomSCP[OUT]	• Use a custom config for CDMA 1x-EV-DO SCP – 0 - Disabled – 1 - Enabled – 0xFF - Unknown
pProtocol[OUT]	• Protocol mask for custom SCP config – 0x00000001 - Subtype 2 Physical Layer – 0x00000002 - Enhanced CCMAC – 0x00000004 - Enhanced ACMAC – 0x00000008 - Enhanced FTCMAC – 0x00000010 - Subtype 3 RTCMAC – 0x00000020 - Subsystem 1 RTCMAC – 0x00000040 - Enhanced Idle – 0x00000080 - Generic Multimode Capable Disc Port – 0xFFFFFFFF - Unknown
pBroadcast[OUT]	• Broadcast mask for custom SCP config – 0x00000001 - Generic broadcast enabled – 0xFFFFFFFF - Unknown
pApplication[OUT]	• Application mask for custom SCP config – 0x00000001 - SN Multiflow Packet Application – 0x00000002 - SN Enhanced Multiflow Packet Application – 0xFFFFFFFF - Unknown
pRoaming[OUT]	• Roaming preference – 0 - Automatic – 1 - Home Only – 2 - Affiliated Roaming Only – 3 - Home and Affiliated Roaming – 0xFFFFFFFF - Unknown

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: CDMA
Timeout: 5 seconds

9.19.5.4 ULONG GetHomeNetwork (WORD * *pMCC*, WORD * *pMNC*, BYTE *nameSize*, CHAR * *pName*, WORD * *pSID*, WORD * *pNID*)

Retrieves information about the home network of the device. For 3GPP2 home network information use GetHomeNetwork3GPP2.

Parameters

pMCC[OUT]	• Mobile country code (UMTS only).
pMNC[OUT]	• Mobile network code (UMTS only).
nameSize	• Maximum number of characters (including NULL terminator) that 8 network name array can contain (UMTS only).
pName[OUT]	• Network name or description represented as a NULL terminated string (empty string returned when unknown) (UMTS only).
pSID[OUT]	• Home network system ID – 0xFFFF - Unknown. – Only applies to cdma2000
pNID[OUT]	• Home network ID – 0xFFFF - Unknown. – Only applies to cdma2000

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 2 seconds

9.19.5.5 ULONG GetHomeNetwork3GPP2 (WORD * *pMCC*, WORD * *pMNC*, BYTE *nameSize*, CHAR * *pName*, WORD * *pSID*, WORD * *pNID*, WORD * *pNw2MCC*, WORD * *pNw2MNC*, BYTE * *pNw2DescDisp*, BYTE * *pNw2DescEnc*, BYTE * *pNw2DescLen*, BYTE * *pNw2Name*)

Retrieves information about the home network of the device. It will extract 3GPP2 Network Information also.

Parameters

pMCC[OUT]	• Mobile country code (UMTS only).
pMNC[OUT]	• Mobile network code (UMTS only).
nameSize	• Maximum number of characters (including NULL terminator) that 8 network name array can contain (UMTS only).
pName[OUT]	• Network name or description represented as a NULL terminated string (empty string returned when unknown) (UMTS only).
pSID[OUT]	• Home network system ID – 0xFFFF - Unknown. – Only applies to cdma2000
pNID[OUT]	• Home network ID – 0xFFFF - Unknown. – Only applies to cdma2000
pNw2MCC[OUT]	• Mobile country code (3GPP2 only). • Range : 0 to 999
pNw2MNC[OUT]	• Mobile network code (3GPP2 only). • Range : 0 to 999
pNw2DescDisp[OUT]	• Network Name Display (3GPP2 only). -Valid Value – 0x00 - Do not display – 0x01 - Display – 0xFF - Unknown
pNw2DescDisp[OUT]	• Encoding of the network description (3GPP2 only). • Valid Value – 0x00 - Octet, unspecified – 0x02 - 7-bit ASCII – 0x04 - Unicode – 0x09 - GSM 7-bit default
nw2DescLen[OUT]	• Network Description Length (3GPP2 only).
pNw2Name[OUT]	• Network Name (3GPP2 only).

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 2 seconds

9.19.5.6 ULONG GetNetworkPreference (ULONG * *pTechnologyPref,* ULONG * *pDuration,* ULONG * *pPersistentTechnologyPref*)

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

Parameters

pTechnology-Pref[OUT]	• Bitmask representing the radio technology preference set. • No bits set indicates to the device to automatically determine the technology to use • Values: – Bit 0 - Technology is 3GPP2 – Bit 1 - Technology is 3GPP • Any combination of the following may be returned: – Bit 2 - Analog - AMPS if 3GPP2, GSM if 3GPP – Bit 3 - Digital - CDMA if 3GPP2, WCDMA if 3GPP – Bit 4 - HDR – Bit 5 - LTE – Bits 6 to 15 - Reserved
pDuration[OUT]	• Duration of active preference – 0 - Permanent – 1 - Power cycle – 2 - Until the end of the next call or a power cycle – 3 - Until the end of the next call, a specified time, or a power cycle – 4 to 6 - Until the end of the next call
pPersistent-TechnologyPref[-OUT]	• Bit field representing persistent radio technology preference – Same representation as the pTechnologyPref parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 2 seconds

9.19.5.7 ULONG GetRFInfo (BYTE * *pInstanceSize,* struct RFBandInfoElements * *pRFBandInfo*)

Sets the RFInfoList

Parameters

pInstanceSize[IN/OUT]	• Upon input, maximum number of elements that the RF info instances array can contain. • Upon successful output, actual number of elements in RF info instances array.
pInstances[OUT]	• RF info instances array – See RFBandInfoElements for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 2 seconds

9.19.5.8 ULONG GetServingNetwork (ULONG * *pRegistrationState,* ULONG * *pCSDomain,* ULONG * *pPSDomain,* ULONG * *pRAN,* BYTE * *pRadioIfacesSize,* BYTE * *pRadioIfaces,* ULONG * *pRoaming,* WORD * *pMCC,* WORD * *pMNC,* BYTE *nameSize,* CHAR * *pName*)

Provides information about the system that provides service to the device. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use API SLQSNasGetSysInfo() for new firmware versions and new modules

Parameters

pRegistrationState[OUT]	• Registration state: – 0 - Not registered – 1 - Registered – 2 - Searching/Not Registered – 3 - Registration Denied – 4 - Unknown
pCSDomain[OUT]	• Circuit switch domain status: – 0 - Unknown/Not Applicable – 1 - Attached – 2 - Detached

pPSDomain[OUT]	• Packet switch domain status – 0 - Unknown/Not Applicable – 1 - Attached – 2 - Detached
pRAN[OUT]	• Type of radio access network on which mobile is registered: – 0 - Unknown – 1 - cdma2000 network – 2 - UMTS network
pRadioIfacesSize[IN/OUT]	• Upon input, maximum number of elements that the radio interface array contain. • Upon successful output, actual number of elements in the radio interface array.
pRadioIfaces[OUT]	• An array of Radio Interface Technology – See qaGobiApiTableRadioInterfaces.h for the Radio Interface Technologies
pRoaming[OUT]	• Roaming indicator
pMCC[OUT]	• Mobile country code
pMNC[OUT]	• Mobile network code
nameSize	• Maximum number of characters (including NULL terminator) that network name array can contain; applicable only for UMTS networks
pName[OUT]	• Network name or description represented as a NULL terminated string; empty string is returned when unknown; applicable only for UMTS networks

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 2 seconds

9.19.5.9 ULONG GetServingNetworkCapabilities (BYTE * *pDataCapsSize*, BYTE * *pDataCaps*)

Returns information regarding the data capabilities of the system that currently provides service to the device.

Parameters

pDataCapsSize[IN/OUT]	• Upon input, the maximum number of elements the data capabilities array can contain. • Upon output, the actual number of elements in the data capabilities array.
pDataCaps[OUT]	• Data capabilities array of unsigned long type – 1 - GPRS – 2 - EDGE – 3 - HSDPA – 4 - HSUPA – 5 - WCDMA – 6 - CDMA 1xRTT – 7 - CDMA 1xEV-DO Rev 0 – 8 - CDMA 1xEV-DO Rev. A – 9 - GSM – 10 - EVDO Rev. B – 11 - LTE – 12 - HSDPA Plus – 13 - Dual Carrier HSDPA Plus

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 2 seconds

9.19.5.10 ULONG GetSignalStrengths (ULONG * *pArraySizes*, INT8 * *pSignalStrength*, ULONG * *pRadioInterface*)

Returns the available signal strengths (in dBm) as measured by the device in an array. The API also provides the corresponding radio radio interfaces in an array.

Parameters

pArraySizes[IN/OUT]	• Upon input maximum number of elements that each array can contain. • Upon successful output actual number of elements in the array.
pSignalStrength[OUT]	• Received signal strength array (in dBm)
pRadioInterface[OUT]	• Radio interface technology array of the signal being measured – See qaGobiApiTableRadioInterfaces.h for Radio Interface info

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 2 seconds

9.19.5.11 ULONG InitiateDomainAttach (ULONG *action*)

Initiates a domain attach/detach of the device. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use API SLQSSetSysSelectionPref() for new firmware versions and new modules

Parameters

action[IN]	• Domain action to attempt 1 - Attach 2 - Detach

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 30 seconds

9.19.5.12 ULONG InitiateNetworkRegistration (ULONG *regType,* WORD *mcc,* WORD *mnc,* ULONG *rat*)

Initiates the network registration process. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use API SLQSSetSysSelectionPref() and SLQSSetBandPreference() for new firmware versions and new modules

Parameters

regType	• Registration type – 1 - Automatic – 2 - Manual
mcc	• Mobile country code

mnc	• Mobile network code
rat	• Radio access technology – 4 - GSM – 5 - UMTS

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: UMTS
Timeout: 30 seconds

9.19.5.13 ULONG PerformNetworkScan (BYTE * *pInstanceSize,* BYTE * *pInstances*)

Performs scan for available networks.

Parameters

pInstanceSize[IN/OUT]	• Upon input, maximum number of elements that the network info instance array can contain. • Upon successful output, the actual number of elements in the network info instance array.
pInstances[OUT]	• Network info instance array – See QmiNas3GppNetworkInfo

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: UMTS
Timeout: 5 minutes

9.19.5.14 ULONG SetACCOLC (CHAR * *spc,* BYTE *accolc*)

Sets the access overload class (ACCOLC)

Parameters

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

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: CDMA
Timeout: 5 seconds

9.19.5.15 ULONG SetCDMANetworkParameters (CHAR * *pSPC,* BYTE * *pForceRev0,* BYTE * *pCustomSCP,* ULONG * *pProtocol,* ULONG * *pBroadcast,* ULONG * *pApplication,* ULONG * *pRoaming*)

Sets the CDMA network parameters. Currently the modified settings will not be utilized until the device has been reset. For this reason, the recommended approach when using SetCDMANetworkParameters is for the application to perform the following steps:

1 - Call SetCDMANetworkParameters() 2 - Call SetPower(5) 3 - Call QCWWANDisconnect() 4 - Reconnect after the device power cycles

Parameters

pSPC[IN]	• Six digit service programming code (not necessary when only the roaming field is being set)
pForceRev0[IN]	• (Optional)Force CDMA 1x-EV-DO Rev. 0 mode – 0 - Disabled – 1 - Enabled Note: Enabled can only be specified if pCustomSCP state is set to Disabled
pCustomSCP[IN]	• (Optional)Use a custom config for CDMA 1x-EV-DO SCP – 0 - Disabled – 1 - Enabled Note: Enabled can only be specified if pForceRev0 is set to Disabled

pProtocol[IN]	• Protocol mask for custom SCP config – 0x00000001 - Subtype 2 Physical Layer – 0x00000002 - Enhanced CCMAC – 0x00000004 - Enhanced ACMAC – 0x00000008 - Enhanced FTCMAC – 0x00000010 - Subtype 3 RTCMAC – 0x00000020 - Subsystem 1 RTCMAC – 0x00000040 - Enhanced Idle – 0x00000080 - Generic Multimode Capable Disc Port – 0xFFFFFFFF - Unknown
pBroadcast[IN]	• Broadcast mask for custom SCP config – 0x00000001 - Generic broadcast enabled – 0xFFFFFFFF - Unknown
pApplication[IN]	• Application mask for custom SCP config – 0x00000001 - SN Multiflow Packet Application – 0x00000002 - SN Enhanced Multiflow Packet Application – 0xFFFFFFFF - Unknown
pRoaming[IN]	• Roaming preference – 0 - Automatic – 1 - Home Only – 2 - Affiliated Roaming Only – 3 - Home and Affiliated Roaming – 0xFFFFFFFF - Unknown

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: CDMA
Timeout: 5 seconds

9.19.5.16 ULONG SetNetworkPreference (ULONG *technologyPref,* ULONG *duration*)

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

Parameters

technologyPref[IN]	• 2 Byte Bitmask representing radio technology preference – No bits set indicates device to automatically determine the technology to use. – Type of technology * Bit 0 - Technology is 3GPP2 * Bit 1 - Technology is 3GPP – Technology-specific protocol bitmask * Bit 2 - Analog · AMPS if 3GPP2, GSM if 3GPP * Bit 3 - Digital · CDMA if 3GPP2, WCDMA if 3GPP * Bit 4 - HDR * Bit 5 - LTE * All other bits are reserved.
duration[IN]	• Duration of active preference – 0 - Persistent – 1 - Power cycle

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 2 seconds

9.19.5.17 ULONG SLQSConfigSigInfo (sigInfo * *pSigInfo*)

Sets the signal reporting thresholds. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use SLQSNasConfigSigInfo2() instead for new firmware versions and new modules.

Parameters

pSigInfo[IN]	• See sigInfo for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 5 seconds

9.19.5.18 ULONG SLQSGetErrorRate (GetErrRateResp * *pGetErrRateResp*)

This API retrieves current error rate information

Parameters

pGetErrRate-Resp[OUT]	• See GetErrRateResp for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 5 seconds

9.19.5.19 ULONG SLQSGetNetworkTime (GetNetworkTimeResp * *pGetNetworkTimeResp*)

This API retrieves the last known network time information from the UE.

Parameters

pGetNetwork-TimeResp[OUT]	• See GetNetworkTimeResp for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 5 seconds

9.19.5.20 ULONG SLQSGetOperatorNameData (nasOperatorNameResp * *pOperatorNameData*)

Get the operator name data from the network. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use API SLQSGetPLMNName() for new firmware versions and new modules

Parameters

pOperatorNameData[OUT]	• See nasOperatorNameResp for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 10 seconds

9.19.5.21 ULONG SLQSGetPLMNName (nasPLMNNameReq * *pPLMNNameReq*, nasPLMNNameResp * *pPLMNNameResp*)

Get the operator name data from the network.

Parameters

pPLMNNameReq[IN]	• See nasPLMNNameReq for more information
pPLMNNameResp[OUT]	• See nasPLMNNameResp for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 10 seconds

9.19.5.22 ULONG SLQSGetServingSystem (qaQmiServingSystemParam * *pServingSystem*)

Provides information about the system that provides service to the device. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use API SLQSNasGetSysInfo() for new firmware versions and new modules. Also report available radio interface technology. If there are more than one radio interface, please choose the right interface(usually the first pair).

- See SLQSSetSysSelectionPref

Parameters

pServing-System[OUT]	• serving system parameters obtained from the system

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise.

See Also

See qmerrno.h for eQCWWAN_xxx error values.

Note

Timeout: 2 seconds

9.19.5.23 ULONG SLQSGetSignalStrength (struct slqsSignalStrengthInfo * *pSignalInfo*)

Queries the current signal strength as measured by the device. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use API SLQSNasGetSigInfo() for new firmware versions and new modules

Parameters

pSignalInfo[IN/-OUT]	• See slqsSignalStrengthInfo for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 2 seconds

9.19.5.24 ULONG SLQSGetSysSelectionPref (sysSelectPrefInfo * *pSysSelectPrefInfo*)

Queries the different system selection preferences of the device.

Parameters

pSysSelectPref-Info[OUT]	• See sysSelectPrefInfo for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 2 seconds

9.19.5.25 ULONG SLQSInitiateNetworkRegistration (nasInitNetworkReg * *pNasInitNetRegistrationReg*)

Initiates the network registration process.

Parameters

pNasInitNet-Registration-Req[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

pSetSignal-StrengthInfo[IN]	• See setSignalStrengthInfo for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 5 seconds
Mixture of threshold and delta values can be provided in the request. But for each type and RAT,only one of threshold list or delta value is to be provided.

9.19.5.27 ULONG SLQSNasGet3GPP2Subscription (nasGet3GPP2SubscriptionInfoReq * *pGet3GPP2SubsInfoReq*, nasGet3GPP2SubscriptionInfoResp * *pGet3GPP2SubsInfoResp*)

This API retrieves 3GPP2 subscription-related information.

Parameters

pGet3GPP2-SubsInfoReq[IN]	• See nasGet3GPP2SubscriptionInfoReq for more information
pGet3GPP2-SubsInfoResp[-OUT]	• See nasGet3GPP2SubscriptionInfoResp for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: CDMA
Timeout: 2 seconds
This command retrieves 3GPP2 subscription-related information. The QMI_ERR_INTERNAL error is returned when no information can be retrieved from the modem.

9.19.5.28 ULONG SLQSNasGetCellLocationInfo (nasCellLocationInfoResp * *pNasCellLocationInfoResp*)

This API retrieves cell location-related information

Parameters

pNasCell-LocationInfo-Resp[OUT]	• See nasCellLocationInfoResp for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 2 seconds
This API retrieves cell location-related information, depending on current serving system.

9.19.5.29 ULONG 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 cdmaMsgDecodingParamsExt
- struct wcdmaMsgEncodingParams
- struct wcdmaMsgDecodingParams
- struct wcdmaLongMsgDecodingParams
- struct _transLayerinfo
- struct _getTransLayerInfoResp
- struct _getTransNWRegInfoResp
- struct _getIndicationRegResp
- struct _setIndicationRegReq
- struct smsRouteEntry
- struct smsSetRoutesReq
- struct smsMsgprotocolResp
- struct smsMaxStorageSizeReq
- struct smsMaxStorageSizeResp
- struct messageWaitingInfoContent
- struct getMsgWaitingInfo
- struct slqssendasyncsmsparams_s

Macros

- #define CONFIG_LEN 0x05
- #define TIME_STAMP_BUF 0x08
- #define ABSOLUTE_VALIDITY 0x08
- #define TIME_DATE_BUF 0x09
- #define MAX_SMS_ROUTES 0x0A
- #define NUM_OF_SET 0xFF

Typedefs

- typedef struct _qaQmi3GPPBroadcastCfgInfo qaQmi3GPPBroadcastCfgInfo
- typedef struct _qaQmi3GPP2BroadcastCfgInfo qaQmi3GPP2BroadcastCfgInfo
- typedef struct _transLayerinfo transLayerInfo
- typedef struct _getTransLayerInfoResp getTransLayerInfoResp
- typedef struct _getTransNWRegInfoResp getTransNWRegInfoResp
- typedef struct _getIndicationRegResp getIndicationRegResp
- typedef struct _setIndicationRegReq setIndicationRegReq

Functions

- ULONG SLQSDeleteSMS (ULONG storageType, ULONG *pMessageIndex, ULONG *pMessageTag, BYTE *pMessageMode)
- ULONG SLQSGetSMS (ULONG storageType, ULONG messageIndex, ULONG *pMessageTag, ULONG *pMessageFormat, ULONG *pMessageSize, BYTE *pMessage, BYTE *pMessageMode)
- ULONG SendSMS (ULONG messageFormat, ULONG messageSize, BYTE *pMessage, ULONG *pMessageFailureCode, BYTE *pSmsOnIms)
- ULONG SLQSSendSMS (slqssendsmsparams_s *pSendSmsParams)
- ULONG GetSMSCAddress (BYTE addressSize, CHAR *pSMSCAddress, BYTE typeSize, CHAR *pSMSCType)
- ULONG SetSMSCAddress (CHAR *pSMSCAddress, CHAR *pSMSCType)

- ULONG SaveSMS (ULONG storageType, ULONG messageFormat, ULONG messageSize, BYTE *pMessage, ULONG *pMessageIndex)
- ULONG SLQSGetSMSList (ULONG storageType, ULONG *pRequestedTag, ULONG *pMessageListSize, BYTE *pMessageList, BYTE *pMessageMode)
- ULONG SLQSModifySMSStatus (ULONG storageType, ULONG messageIndex, ULONG messageTag, BYTE *pMessageMode)
- ULONG SLQSGetSmsBroadcastConfig (BYTE mode, qaQmi3GPPBroadcastCfgInfo *pBroadcastConfig, qaQmi3GPP2BroadcastCfgInfo *pCDMABroadcastConfig)
- ULONG SLQSSetSmsBroadcastConfig (BYTE mode, qaQmi3GPPBroadcastCfgInfo *pBroadcastConfig, qaQmi3GPP2BroadcastCfgInfo *pCDMABroadcastConfig)
- ULONG SLQSSetSmsBroadcastActivation (BYTE mode, BYTE broadcastActivate)
- ULONG SLQSCDMAEncodeMOTextMsg (struct cdmaMsgEncodingParams *pCdmaMsgEncodingParams)
- ULONG SLQSCDMADecodeMTTextMsg (struct cdmaMsgDecodingParams *pCdmaMsgDecodingParams)
- ULONG SLQSCDMADecodeMTTextMsgExt (struct cdmaMsgDecodingParamsExt *pCdmaMsgDecodingParamsExt)
- ULONG SLQSWCDMAEncodeMOTextMsg (struct wcdmaMsgEncodingParams *pWcdmaMsgEncodingParams)
- ULONG SLQSWCDMADecodeMTTextMsg (struct wcdmaMsgDecodingParams *pWcdmaMsgDecodingParams)
- ULONG SLQSWCDMADecodeLongTextMsg (struct wcdmaLongMsgDecodingParams *pWcdmaLongMsgDecodingParams)
- ULONG SLQSGetTransLayerInfo (getTransLayerInfoResp *pGetTransLayerInfoResp)
- ULONG SLQSGetTransNWRegInfo (getTransNWRegInfoResp *pGetTransNWRegInfoResp)
- ULONG SLQSGetIndicationRegister (getIndicationRegResp *pGetIndicationRegInfo)
- ULONG SLQSSetIndicationRegister (setIndicationRegReq *pSetIndicationRegReq)
- ULONG SLQSSmsSetRoutes (smsSetRoutesReq *pSetRoutesReq)
- ULONG SLQSSmsGetMessageProtocol (smsMsgprotocolResp *pMessageProtocol)
- ULONG SLQSSmsGetMaxStorageSize (smsMaxStorageSizeReq *pMaxStorageSizeReq, smsMaxStorageSizeResp *pMaxStorageSizeResp)
- ULONG SLQSGetMessageWaiting (getMsgWaitingInfo *pGetMsgWaitingInfoResp)
- ULONG SLQSSendAsyncSMS (slqssendasyncsmsparams_s *pSendSmsParams)
- ULONG SLQSSetSmsStorage (BYTE smsStorage)
- ULONG SLQSSwiGetSMSStorage (ULONG *pSmsStorage)
- ULONG SLQSSendLongSMS (ULONG messageFormat, ULONG messageSize, CHAR *pMessage, BYTE encodingScheme, ULONG *pMessageFailureCode, CHAR *pMobileNum, BYTE *pSmsOnIMS)

9.25.1 Detailed Description

Short Message Service API function prototypes.

9.25.2 Macro Definition Documentation

9.25.2.1 #define ABSOLUTE_VALIDITY 0x08

9.25.2.2 #define CONFIG_LEN 0x05

9.25.2.3 #define MAX_SMS_ROUTES 0x0A

9.25.2.4 #define NUM_OF_SET 0xFF

9.25.2.5 #define TIME_DATE_BUF 0x09

9.25.2.6 #define TIME_STAMP_BUF 0x08

9.25.3 Typedef Documentation

9.25.3.1 typedef struct _getIndicationRegResp getIndicationRegResp

This structure contains Get Indication Register Response parameters

Parameters

pRegTrans-LayerInfoEvt	- • 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.
pMessageFailureCode[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

pSMSCAddress[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

pMsgToBeEncodedCDMA[IN/OUT]	• Pointer to structure containing parameters needed for decoding

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

see qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: CDMA
Timeout: None

9.25.4.6 ULONG SLQSCDMADecodeMTTextMsgExt (struct cdmaMsgDecodingParamsExt * *pCdmaMsgDecodingParamsExt*)

Decodes text message to CDMA PDU message Ext

Parameters

pCdmaMsgDecodingParamsExt[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

9.25.4.7 ULONG SLQSCDMAEncodeMOTextMsg (struct cdmaMsgEncodingParams * *pCdmaMsgEncodingParams*)

Encodes text message to CDMA PDU message.

Parameters

pMsgToBeEncodedCDMA[IN/OUT]	• SLQS Runtime Settings Information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

see qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: CDMA
Timeout: None

9.25.4.8 ULONG SLQSDeleteSMS (ULONG *storageType,* ULONG * *pMessageIndex,* ULONG * *pMessageTag,* BYTE * *pMessageMode*)

Deletes one or more SMSs from device memory. If both of the optional parameters, messageIndex and messageTag, are NULL, all messages are deleted from the storage location specified in the mandatory storageType parameter. The optional index and tag parameters narrow the range of messages being deleted. If an index is specified, the single message with the index from the specified memory store will be deleted. If a tag is specified, all messages in the specified memory store whose tag matches that specified will be deleted.

There are three ways to use this message:

- Specify storageType only
  - Deletes all messages from memory storage
- Specify storageType and a tag
  - Deletes all messages from memory storage that match the given message tag
- Specify storageType and an index
  - Deletes only the message with the given index from memory storage

Parameters

storageType	• SMS message storage type – 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

9.25.4.9 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

9.25.4.10 ULONG SLQSGetMessageWaiting (getMsgWaitingInfo * *pGetMsgWaitingInfoResp*)

This API provides information about the message waiting information.

Parameters

pGetMsgWaitingInfoResp	[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

9.25.4.11 ULONG SLQSGetSMS (ULONG *storageType*, ULONG *messageIndex*, ULONG * *pMessageTag*, ULONG * *pMessageFormat*, ULONG * *pMessageSize*, BYTE * *pMessage*, BYTE * *pMessageMode*)

Returns an SMS from device memory.

Parameters

storageType	• SMS message storage type – 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
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

9.25.4.12 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)
pBroadcastConfig[OUT]	• The data for 3GPP Broadcast Information(Optional).
pCDMABroadcastConfig[OUT]	• The data for 3GPP2 Broadcast Information(Optional).

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: ALL
Timeout: 5 seconds

9.25.4.13 ULONG SLQSGetSMSList (ULONG *storageType,* ULONG * *pRequestedTag,* ULONG * *pMessageListSize,* BYTE * *pMessageList,* BYTE * *pMessageMode*)

Returns the list of SMS messages stored on the device.

Parameters

storageType[IN]	• SMS message storage type – 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

9.25.4.14 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

9.25.4.15 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

9.25.4.16 ULONG SLQSModifySMSStatus (ULONG *storageType,* ULONG *messageIndex,* ULONG *messageTag,* BYTE * *pMessageMode*)

Modifies the status of an SMS message saved in storage on the device.

Parameters

storageType[IN]	• SMS message storage type – 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

9.25.4.17 ULONG SLQSSendAsyncSMS (slqssendasyncsmsparams_s * *pSendSmsParams*)

Sends an SMS message for immediate over-the-air transmission

Parameters

pSendSmsParams	• structure containing the SMS parameters. Refer slqssendasyncsmsparams_s

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: ALL
Timeout: 5 minutes

9.25.4.18 ULONG SLQSSendLongSMS (ULONG *messageFormat*, ULONG *messageSize*, CHAR * *pMessage*, BYTE *encodingScheme*, ULONG * *pMessageFailureCode*, CHAR * *pMobileNum*, BYTE * *pSmsOnIMS*)

Sends a long SMS message for immediate over-the-air transmission, a short SMS can be sent by this API as well, the input message is text string without any encoding

Parameters

messageFormat[IN]	• Message format – 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

9.25.4.19 ULONG SLQSSendSMS (slqssendsmsparams_s * *pSendSmsParams*)

Sends an SMS message for immediate over-the-air transmission

Parameters

pSendSms-Params	• structure containing the SMS parameters. Refer slqssendsmsparams_s

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: ALL
Timeout: 5 minutes

9.25.4.20 ULONG SLQSSetIndicationRegister (setIndicationRegReq * *pSetIndicationRegReq*)

This API sets the registration state of different WMS indications.

Parameters

pSetIndication-RegReq	[IN] • 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

9.25.4.21 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

9.25.4.22 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

9.25.4.23 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

9.25.4.24 ULONG SLQSSmsGetMaxStorageSize (smsMaxStorageSizeReq * *pMaxStorageSizeReq*, smsMaxStorageSizeResp * *pMaxStorageSizeResp*)

This API provides the maximum number of messages that can be stored in the specified memory storage. Also it provides the number of slots currently available

Parameters

pMaxStorage-SizeReq[IN]	• Request parameters for SmsSLQSGetMaxStorageSize – 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

9.25.4.25 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

9.25.4.26 ULONG SLQSSmsSetRoutes (smsSetRoutesReq * *pSetRoutesReq*)

This API sets the action performed on SMS message receipt for specified message routes. It also specifies the action performed on SMS receipt of status reports.

Parameters

pSetRoutesReq	[IN] • 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

9.25.4.27 ULONG SLQSSwiGetSMSStorage (ULONG * *pSmsStorage*)

This API queries the device to return current SMS configuration that is applied to all incoming and outgoing messages.

Parameters

pSmsStorage[OUT]	• Values: – 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

9.25.4.28 ULONG SLQSWCDMADecodeLongTextMsg (struct wcdmaLongMsgDecodingParams * *pWcdmaLongMsgDecodingParams*)

Decodes WCDMA Long SMS PDU message, returns structure filled with decoded parameters

Parameters

pwcdmaMsg-Decoding-Params[IN/OUT]	• Pointer to parameters required for decoding

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

see qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: UMTS
Timeout: none

9.25.4.29 ULONG SLQSWCDMADecodeMTTextMsg (struct wcdmaMsgDecodingParams * *pWcdmaMsgDecodingParams*)

Decodes WCDMA PDU message, returns structure filled with decoded parameters

Parameters

pwcdmaMsg-Decoding-Params[IN/OUT]	• Pointer to parameters required for decoding

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

see qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: UMTS
Timeout: none

9.25.4.30 ULONG SLQSWCDMAEncodeMOTextMsg (struct wcdmaMsgEncodingParams * *pWcdmaMsgEncodingParams*)

Returns the encoded WCDMA PDU message.

Parameters

pwcdmaMsg-Encoding-Params[IN/OUT]	• Pointer to parameters Required for encoding

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

see qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: UMTS
Timeout: None

9.26 qaGobiApiSwi.h File Reference

SWI API function prototypes.

Functions

- ULONG SLQSGetSdkVersion (CHAR **sdkversionpp)
- ULONG SLQSSendRawQMI (BYTE *pReqBuf, USHORT service, USHORT length, ULONG timeout, BYTE **ppInParm, USHORT *pParamLength)
- int SLQSGetPidof (CHAR *pProcName)

9.26.1 Detailed Description

SWI API function prototypes.

9.26.2 Function Documentation

9.26.2.1 int SLQSGetPidof (CHAR * *pProcName*)

Internal Wrapper function for enabling invocation of SLQS implementation pidof() function

Parameters

pProcName[IN]	• Process name whose PID is to be retrieved

Returns

pid if process exists else 0

See Also

NA

Note

NA

9.26.2.2 ULONG SLQSGetSdkVersion (CHAR ** *sdkversionpp*)

Returns the SDK version string

Parameters

ppString[OUT]	• pointer to pointer of NULL terminated string

Returns

eQCWWAN_ERR_NONE success eQCWWAN_ERR_INVALID_ARG provided pointer is NULL

Note

Technology Supported: N/A Timeout: 2 seconds

9.26.2.3 ULONG SLQSSendRawQMI (BYTE * *pReqBuf,* USHORT *service,* USHORT *length,* ULONG *timeout,* BYTE ** *ppInParm,* USHORT * *pParamLength*)

9.27 qaGobiApiSwiAudio.h File Reference

M2M Audio Service API function prototypes.

Data Structures

- struct GetM2MAudioProfileReq
- struct GetM2MAudioProfileResp
- struct SetM2MAudioProfileReq
- struct GetM2MAudioVolumeReq
- struct GetM2MAudioVolumeResp
- struct SetM2MAudioVolumeReq
- struct PCMparams
- struct SetM2MAudioAVCFGReq
- struct SetM2MAudioLPBKReq
- struct GetM2MSpkrGainReq
- struct GetM2MSpkrGainResp
- struct SetM2MSpkrGainReq
- struct GetM2MAVMuteReq
- struct GetM2MAVMuteResp
- struct SetM2MAVMuteReq

Macros

- #define MAX_LEN_IFACE_TABLE 255

Functions

- ULONG SLQSGetM2MAudioProfile (GetM2MAudioProfileReq *pGetM2MAudioProfileReq, GetM2MAudioProfileResp *pGetM2MAudioProfileResp)
- ULONG SLQSSetM2MAudioProfile (SetM2MAudioProfileReq *pSetM2MAudioProfileReq)
- ULONG SLQSGetM2MAudioVolume (GetM2MAudioVolumeReq *pGetM2MAudioVolumeReq, GetM2MAudioVolumeResp *pGetM2MAudioVolumeResp)
- ULONG SLQSSetM2MAudioVolume (SetM2MAudioVolumeReq *pSetM2MAudioVolumeReq)
- ULONG SLQSSetM2MAudioAVCFG (SetM2MAudioAVCFGReq *pSetM2MAudioAVCFGReq)
- ULONG SLQSSetM2MAudioLPBK (SetM2MAudioLPBKReq *pSetM2MAudioLPBKReq)
- ULONG SLQSSetM2MAudioNVDef ()
- ULONG SLQSGetM2MSpkrGain (GetM2MSpkrGainReq *pSpkrGainReq, GetM2MSpkrGainResp *pSpkrGainResp)
- ULONG SLQSSetM2MSpkrGain (SetM2MSpkrGainReq *pSpkrGainReq)
- ULONG SLQSGetM2MAVMute (GetM2MAVMuteReq *pGetM2MAVMuteReq, GetM2MAVMuteResp *pGetM2MAVMuteResp)
- ULONG SLQSSetM2MAVMute (SetM2MAVMuteReq *pSetM2MAVMuteReq)

9.27.1 Detailed Description

M2M Audio Service API function prototypes.

9.27.2 Macro Definition Documentation

9.27.2.1 #define MAX_LEN_IFACE_TABLE 255

9.27.3 Function Documentation

9.27.3.1 ULONG SLQSGetM2MAudioProfile (GetM2MAudioProfileReq * *pGetM2MAudioProfileReq*, GetM2MAudioProfileResp * *pGetM2MAudioProfileResp*)

This API gets the profile content.

Parameters

pGetM2MAudioProfileReq[OUT]	• See GetM2MAudioProfileReq for more information
pGetM2MAudioProfileResp[OUT]	• See GetM2MAudioProfileResp for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 5 seconds

9.27.3.2 ULONG SLQSGetM2MAudioVolume (GetM2MAudioVolumeReq * *pGetM2MAudioVolumeReq*, GetM2MAudioVolumeResp * *pGetM2MAudioVolumeResp*)

This API gets the Volume content.

Parameters

pGetM2MAudioVolumeReq[IN]	• See GetM2MAudioVolumeReq for more information
pGetM2MAudioVolumeResp[OUT]	• See GetM2MAudioVolumeResp for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 5 seconds

9.27.3.3 ULONG SLQSGetM2MAVMute (GetM2MAVMuteReq * *pGetM2MAVMuteReq*, GetM2MAVMuteResp * *pGetM2MAVMuteResp*)

This API Gets the AV Mute content.

Parameters

pGetM2MAVMuteReq[IN]	• See GetM2MAVMuteReq for more information
pGetM2MAVMuteResp[OUT]	• See GetM2MAVMuteResp for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 5 seconds

9.27.3.4 ULONG SLQSGetM2MSpkrGain (GetM2MSpkrGainReq * *pSpkrGainReq*, GetM2MSpkrGainResp * *pSpkrGainResp*)

This API Gets the SPKRGAIN content.

Parameters

pSpkrGainReq[IN]	• See GetM2MSpkrGainReq for more information
pSpkrGainResp[OUT]	• See GetM2MSpkrGainResp for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 5 seconds

9.27.3.5 ULONG SLQSSetM2MAudioAVCFG (SetM2MAudioAVCFGReq * *pSetM2MAudioAVCFGReq*)

This API sets the AVCFG content.

Parameters

pSetM2MAudioAVCFGReq[IN]	• See SetM2MAudioAVCFGReq for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 5 seconds

9.27.3.6 ULONG SLQSSetM2MAudioLPBK (SetM2MAudioLPBKReq * *pSetM2MAudioLPBKReq*)

This API sets the LPBK content.

Parameters

pSetM2MAudioLPBKReq[IN]	• See SetM2MAudioLPBKReq for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 5 seconds

9.27.3.7 ULONG SLQSSetM2MAudioNVDef ( )

This API sets the NVDef content.

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 5 seconds

9.27.3.8 ULONG SLQSSetM2MAudioProfile (SetM2MAudioProfileReq * *pSetM2MAudioProfileReq*)

This API sets the profile content.

Parameters

pSetM2MAudio-ProfileReq[IN]	• See SetM2MAudioProfileReq for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 5 seconds

9.27.3.9 ULONG SLQSSetM2MAudioVolume (SetM2MAudioVolumeReq * *pSetM2MAudioVolumeReq*)

This API sets the Volume content.

Parameters

pSetM2MAudio-VolumeReq[IN]	• See SetM2MAudioVolumeReq for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 5 seconds

9.27.3.10 ULONG SLQSSetM2MAVMute (SetM2MAVMuteReq * *pSetM2MAVMuteReq*)

This API Sets the AV Mute content.

Parameters

pSetM2MAV-MuteReq[IN]	• See SetM2MAVMuteReq for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 5 seconds

9.27.3.11 ULONG SLQSSetM2MSpkrGain (SetM2MSpkrGainReq * *pSpkrGainReq*)

This API Sets the SPKRGAIN content.

Parameters

pSpkrGainReq[I-N]	• See GetM2MSpkrGainReq for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 5 seconds

9.28 qaGobiApiSwiAvms.h File Reference

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

Data Structures

- struct _QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp
- struct _QmiSwiAvmsEventReportConfigResp
- struct _QmiSwiAvmsEventReportNotificationResp
- struct _QmiSwiAvmsEventReportConnectionRequestResp
- struct _QmiSwiAvmsEventReportWAMSParamChangeResp
- struct _QmiSwiAvmsEventReportPackageIDResp
- struct _QmiSwiAvmsEventReportRegStatusResp
- struct _QmiSwiAvmsEventReportDataSessionStatusResp
- struct _QmiSwiAvmsEventReportSessionTypeResp
- struct _QmiSwiAvmsEventReportHTTPStatusResp
- struct _SLQSAVMSSettingsAPNInfo
- struct _SLQSAVMSSettingsConnectionRetryTimers
- struct _SLQSAVMSSettingsPeriodsInfo
- struct _SLQSAVMSSettings
- struct _QmiSwiAvmsSetSettingsPeriodInfo
- struct _QmiSwiAvmsSetSettingsAPNInfo
- struct _QmiSwiAvmsSetSettingsConnectionRetryTimers
- struct _SLQSAVMSSetSettingsReqPara

Macros

- #define MAX_GET_SETTINGS_AVMS_APN_STRING_LENGTH 49
- #define MAX_GET_SETTINGS_AVMS_UNAME_STRING_LENGTH 29
- #define MAX_GET_SETTINGS_AVMS_PWD_STRING_LENGTH 29
- #define QMISWIAVMSSESSIONGETINFO_BINARYUPDATESESSIONINFO_VERSION_LENGTH 128
- #define QMISWIAVMSSESSIONGETINFO_BINARYUPDATESESSIONINFO_NAME_LENGTH 128
- #define QMISWIAVMSSESSIONGETINFO_BINARYUPDATESESSIONINFO_DESC_LENGTH 1024
- #define QMISWIAVMSSESSIONGETINFO_CONFIG_ALERT_MSG_LENGTH 200
- #define MAX_AVMS_RETRY_TIMER_NUMBER 8
- #define MAX_SWI_AVMS_SET_SETTING_APN_LENGTH 49
- #define MAX_SWI_AVMS_SET_SETTING_UNAME_LENGTH 29
- #define MAX_SWI_AVMS_SET_SETTING_PWD_LENGTH 29
- #define MAX_SWI_AVMS_SET_SETTING_CONNECTION_RETRY_TIMMERS 8

Typedefs

- typedef struct _QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp
- typedef struct _QmiSwiAvmsEventReportConfigResp QmiSwiAvmsEventReportConfigResp
- typedef struct _QmiSwiAvmsEventReportNotificationResp QmiSwiAvmsEventReportNotificationResp
- typedef struct _QmiSwiAvmsEventReportConnectionRequestResp QmiSwiAvmsEventReportConnectionRequestResp
- typedef struct _QmiSwiAvmsEventReportWAMSParamChangeResp QmiSwiAvmsEventReportWAMSParamChangeResp
- typedef struct _QmiSwiAvmsEventReportPackageIDResp QmiSwiAvmsEventReportPackageIDResp
- typedef struct _QmiSwiAvmsEventReportRegStatusResp QmiSwiAvmsEventReportRegStatusResp
- typedef struct _QmiSwiAvmsEventReportDataSessionStatusResp QmiSwiAvmsEventReportDataSessionStatusResp
- typedef struct _QmiSwiAvmsEventReportSessionTypeResp QmiSwiAvmsEventReportSessionTypeResp
- typedef struct _QmiSwiAvmsEventReportHTTPStatusResp QmiSwiAvmsEventReportHTTPStatusResp
- typedef struct _SLQSAVMSSettingsAPNInfo SLQSAVMSSettingsAPNInfo
- typedef struct _SLQSAVMSSettingsConnectionRetryTimers SLQSAVMSSettingsConnectionRetryTimers
- typedef struct _SLQSAVMSSettingsPeriodsInfo SLQSAVMSSettingsPeriodsInfo
- typedef struct _SLQSAVMSSettings SLQSAVMSSettings
- typedef struct _QmiSwiAvmsSetSettingsPeriodInfo QmiSwiAvmsSetSettingsPeriodInfo
- typedef struct _QmiSwiAvmsSetSettingsAPNInfo QmiSwiAvmsSetSettingsAPNInfo
- typedef struct _QmiSwiAvmsSetSettingsConnectionRetryTimers QmiSwiAvmsSetSettingsConnectionRetryTimers
- typedef struct _SLQSAVMSSetSettingsReqPara SLQSAVMSSetSettingsReq

Functions

- ULONG SLQSAVMSStartSession (BYTE sessionType, ULONG *pSessionResponse)
- ULONG SLQSAVMSStopSession (BYTE sessiontype, BYTE *pSessionRep)
- ULONG SLQSAVMSGetSessionInfo (QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp *pBinaryUpdteSessionInfo, QmiSwiAvmsEventReportConfigResp *pAVMSConfig, QmiSwiAvmsEventReportNotificationResp *pNotif, QmiSwiAvmsEventReportPackageIDResp *pPackageID)
- ULONG SLQSAVMSSelection (BYTE selection, ULONG *pDeferTime, BYTE *pClientPerfOperationFlag, BYTE *pPackageID, WORD *pRejectReason)
- ULONG SLQSAVMSGetSettings (SLQSAVMSSettings *settings)
- ULONG SLQSAVMSSetSettings (SLQSAVMSSetSettingsReq Request)
- ULONG SLQSAVMSSetSettingsIgnoreAutoReboot (SLQSAVMSSetSettingsReq Request)
- ULONG SLQSAVMSSetEventReport ()

9.28.1 Detailed Description

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

9.28.2 Macro Definition Documentation

9.28.2.1 #define MAX_AVMS_RETRY_TIMER_NUMBER 8

9.28.2.2 #define MAX_GET_SETTINGS_AVMS_APN_STRING_LENGTH 49

9.28.2.3 #define MAX_GET_SETTINGS_AVMS_PWD_STRING_LENGTH 29

9.28.2.4 #define MAX_GET_SETTINGS_AVMS_UNAME_STRING_LENGTH 29

9.28.2.5 #define MAX_SWI_AVMS_SET_SETTING_APN_LENGTH 49

9.28.2.6 #define MAX_SWI_AVMS_SET_SETTING_CONNECTION_RETRY_TIMMERS 8

9.28.2.7 #define MAX_SWI_AVMS_SET_SETTING_PWD_LENGTH 29

9.28.2.8 #define MAX_SWI_AVMS_SET_SETTING_UNAME_LENGTH 29

9.28.2.9 #define QMISWIAVMSSESSIONGETINFO_BINARYUPDATESESSIONINFO_DESC_LENGTH 1024

9.28.2.10 #define QMISWIAVMSSESSIONGETINFO_BINARYUPDATESESSIONINFO_NAME_LENGTH 128

9.28.2.11 #define QMISWIAVMSSESSIONGETINFO_BINARYUPDATESESSIONINFO_VERSION_LENGTH 128

9.28.2.12 #define QMISWIAVMSSESSIONGETINFO_CONFIG_ALERT_MSG_LENGTH 200

9.28.3 Typedef Documentation

9.28.3.1 typedef struct _QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp

Parameters

bBinaryType	: Type • 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 to AirVantage server – 0x00 - FALSE – 0x01 - TRUE
bAutoReboot[IN]	• Automatic device reboot when the request is received from AirVantage server – 0x00 - FALSE – 0x01 - TRUE

bFwPromptdownload[IN]	• Start Firmware Download promptly/automatically – 0x00 - FALSE – 0x01 - TRUE
bFwPromptUpdate[IN]	• Start Firmware Download promptly/automatically – 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

9.28.3.15 typedef struct _SLQSAVMSSettings SLQSAVMSSettings

9.28.3.16 typedef struct _SLQSAVMSSettingsAPNInfo SLQSAVMSSettingsAPNInfo

9.28.3.17 typedef struct _SLQSAVMSSettingsConnectionRetryTimers SLQSAVMSSettingsConnectionRetryTimers

9.28.3.18 typedef struct _SLQSAVMSSettingsPeriodsInfo SLQSAVMSSettingsPeriodsInfo

9.28.4 Function Documentation

9.28.4.1 ULONG SLQSAVMSGetSessionInfo (QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp * *pBinaryUpdteSessionInfo*, QmiSwiAvmsEventReportConfigResp * *pAVMSConfig*, QmiSwiAvmsEventReportNotificationResp * *pNotif*, QmiSwiAvmsEventReportPackageIDResp * *pPackageID*)

Returns information related to the current AVMS session.

Parameters

pBinaryUpdte-SessionInfo[OU-T]	• See QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp for more information.
pAVMSConfig[-OUT]	• See QmiSwiAvmsEventReportConfigResp for more information.
pNotif[OUT]	• See QmiSwiAvmsEventReportNotificationResp for more information.
pPackageID[OU-T]	• See QmiSwiAvmsEventReportPackageIDResp for more information.

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 2 seconds

9.28.4.2 ULONG SLQSAVMSGetSettings (SLQSAVMSSettings * *settings*)

9.28.4.3 ULONG SLQSAVMSSelection (BYTE *selection*, ULONG * *pDeferTime*, BYTE * *pClientPerfOperationFlag*, BYTE * *pPackageID*, WORD * *pRejectReason*)

Sends the specified AVMS selection for the current network initiated session.

Parameters

selection[IN]	• AVMS User Selection. – 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 S1

- 0x01 - QMI_VOICE_REASON_FWD_UNCONDITIONAL - Unconditional call forwarding
- 0x02 - QMI_VOICE_REASON_FWD_MOBILEBUSY - Forward when the mobile is busy
- 0x03 - QMI_VOICE_REASON_FWD_NOREPLY - Forward when there is no reply
- 0x04 - QMI_VOICE_REASON_FWD_UNREACHABLE - Forward when the call is unreachable
- 0x05 - QMI_VOICE_REASON_FWD_ALLFORWARDING - All forwarding
- 0x06 - QMI_VOICE_REASON_FWD_ALLCONDITIONAL - All conditional forwarding
- 0x07 - QMI_VOICE_REASON_BARR_ALLOUTGOING - All outgoing
- 0x08 - QMI_VOICE_REASON_BARR_OUTGOINGINT - Outgoing internal

- 0x09 - QMI_VOICE_REASON_BARR_OUTGOINGINTEXTOHOME - Outgoing external to home
- 0x0A - QMI_VOICE_REASON_BARR_ALLINCOMING - All incoming
- 0x0B - QMI_VOICE_REASON_BARR_INCOMINGROAMING - Roaming incoming
- 0x0C - QMI_VOICE_REASON_BARR_ALLBARRING - All calls are barred
- 0x0D - QMI_VOICE_REASON_BARR_ALLOUTGOINGBARRING - All outgoing calls are barred
- 0x0E - QMI_VOICE_REASON_BARR_ALLINCOMINGBARRING - All incoming calls are barred
- 0x0F - QMI_VOICE_REASON_CALLWAITING - Call waiting
- 0x10 - VOICE_CC_SUPS_RESULT_REASON_CLIP - CLIP
- 0x11 - VOICE_CC_SUPS_RESULT_REASON_CLIR - CLIR
- 0x12 - VOICE_CC_SUPS_RESULT_REASON_COLP - COLP
- 0x13 - VOICE_CC_SUPS_RESULT_REASON_COLR - COLR
- 0x14 - VOICE_CC_SUPS_RESULT_REASON_CNAP - CNAP
- 0xFF - Not Available

Copyright: © 2012 Sierra Wireless, Inc. all rights reserved

9.32 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 S1

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

pVoiceGetCall-WaitInfo[IN/OUT]	• Pointer to structure of voiceGetCallWaitInfo – See voiceGetCallWaitInfo for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: UMTS
Timeout: 30 Secs
This API queries the status of the call waiting supplementary service, i.e., to find whether the call waiting supplementary service is active. The request is a blocking request, i.e., the response is sent only after confirmation is received from the network. The maximum time it takes for the response to be sent is approximately 30 sec. This API is applicable only in 3GPP devices.

9.47.4.17 ULONG SLQSVoiceGetCLIP (voiceGetCLIPResp * *pVoiceGetCLIPResp*)

Queries the status of the Calling Line Identification Presentation (CLIP) supplementary service (applicable only for 3GPP).

Parameters

pVoiceGetCLIP-Resp[OUT]	• Pointer to structure of voiceGetCLIPResp – See voiceGetCLIPResp for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: UMTS
Timeout: 30 Secs
This API queries the status of the CLIP supplementary service. The active_status field is only applicable when provision_status is PROVISIONED, i.e., there is not any case where provision_status is NOT_PROVISIONED and active_status is ACTIVE. The request is a blocking request, i.e., the response is sent only after confirmation is received from the network. The maximum time it takes for the response to be sent is approximately 30 sec. This API is applicable only in 3GPP devices.

9.47.4.18 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[OU-T]	• current data bearer technology value. – NULL if the parameter is not required
pLastCallData-Bearer-Technology[OU-T]	• last call data bearer technology value. – NULL if the parameter is not required

9.48.3.5 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 SLQSWdsSetEventReport-CallBack. 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 sMSMTMessageInfo
- struct newMTMessageTlv
- struct sMSTransferRouteMTMessageInfo
- struct transferRouteMessageTlv
- struct sMSMessageModeInfo
- struct messageModeTlv
- struct sMSEtwsMessageInfo
- struct sMSEtwsMessageTlv
- struct sMSEtwsPlmnInfo
- struct eTWSPLMNInfoTlv
- struct sMSCAddressInfo
- struct sMSCAddressTlv
- struct sMSOnIMSInfo
- struct sMSOnIMSTlv
- struct unpack_sms_SetNewSMSCallback_ind_t
- struct unpack_sms_SLQSWmsMemoryFullCallBack_ind_t
- struct unpack_sms_GetSMSCAddress_t
- struct pack_sms_SetSMSCAddress_t
- struct pack_sms_SaveSMS_t
- struct unpack_sms_SaveSMS_t

- struct sms_BroadcastConfig
- struct sms_qaQmi3GPPBroadcastCfgInfo
- struct sms_CDMABroadcastConfig
- struct sms_qaQmi3GPP2BroadcastCfgInfo
- struct unpack_sms_SLQSGetSmsBroadcastConfig_t
- struct pack_sms_SLQSGetSmsBroadcastConfig_t
- struct pack_sms_SLQSSetSmsBroadcastConfig_t
- struct pack_sms_SLQSSetSmsBroadcastActivation_t
- struct sms_transLayerInfo
- struct sms_getTransLayerInfo
- struct unpack_sms_SLQSGetTransLayerInfo_t
- struct sms_getTransNWRegInfo
- struct unpack_sms_SLQSGetTransNWRegInfo_t
- struct sms_getIndicationReg
- struct unpack_sms_SLQSGetIndicationRegister_t
- struct sms_setIndicationReg
- struct pack_sms_SLQSSetIndicationRegister_t
- struct sms_routeEntry
- struct sms_setRoutesReq
- struct pack_sms_SLQSSmsSetRoutes_t
- struct sms_msgProtocolResp
- struct unpack_sms_SLQSSmsGetMessageProtocol_t
- struct sms_maxStorageSizeReq
- struct sms_maxStorageSizeResp
- struct pack_sms_SLQSSmsGetMaxStorageSize_t
- struct unpack_sms_SLQSSmsGetMaxStorageSize_t
- struct sms_messageWaitingInfoContent
- struct sms_getMsgWaitingInfo
- struct unpack_sms_SLQSGetMessageWaiting_t
- struct sms_sendAsyncsmsParams
- struct pack_sms_SLQSSendAsyncSMS_t
- struct pack_sms_SLQSSetSmsStorage_t
- struct unpack_sms_SLQSSwiGetSMSStorage_t

Macros

- #define MAX_SMS_MESSAGE_SIZE 2048
- #define MAX_SMS_LIST_SIZE 255
- #define MAX_MS_TRANSFER_ROUTE_MSG 256
- #define MAX_MSE_TWS_MSG 1254
- #define MAX_MSC_ADDRESS_SIZE 256
- #define MAX_CDMA_ENC_MO_TXT_MSG_SIZE 255
- #define SMSC_TYPE_LEN 0x03
- #define SMS_CONFIG_LEN 0x05
- #define SMS_MAX_SMS_ROUTES 0x0A
- #define SMS_NUM_OF_SET 0xFF

Enumerations

- enum eqmiCbkSetStatus {
  LITEQMI_QMI_CBK_PARAM_RESET = 0,
  LITEQMI_QMI_CBK_PARAM_SET = 1,
  LITEQMI_QMI_CBK_PARAM_NOCHANGE }

Functions

- int pack_sms_SLQSGetSMS (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SLQSGetSMS_t *reqParam)
- int unpack_sms_SLQSGetSMS (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSGetSMS_t *pOutput)
- int pack_sms_SLQSGetSMSList (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SLQSGetSMSList_t *reqParam)
- int unpack_sms_SLQSGetSMSList (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSGetSMSList_t *pOutput)
- int pack_sms_SLQSModifySMSStatus (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SLQSModifySMSStatus_t *reqParam)
- int unpack_sms_SLQSModifySMSStatus (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSModifySMSStatus_t *pOutput)
- int pack_sms_SLQSDeleteSMS (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SLQSDeleteSMS_t *reqParam)
- int unpack_sms_SLQSDeleteSMS (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSDeleteSMS_t *pOutput)
- int pack_sms_SendSMS (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SendSMS_t *reqParam)
- int unpack_sms_SendSMS (uint8_t *pResp, uint16_t respLen, unpack_sms_SendSMS_t *pOutput)
- int pack_sms_SetNewSMSCallback (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SetNewSMSCallback_t reqParam)
- int unpack_sms_SetNewSMSCallback (uint8_t *pResp, uint16_t respLen, unpack_sms_SetNewSMSCallback_t *Output)
- int unpack_sms_SetNewSMSCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_sms_SetNewSMSCallback_ind_t *pOutput)
- int unpack_sms_SLQSWmsMemoryFullCallBack_ind (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSWmsMemoryFullCallBack_ind_t *pOutput)
- int pack_sms_GetSMSCAddress (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_sms_GetSMSCAddress (uint8_t *pResp, uint16_t respLen, unpack_sms_GetSMSCAddress_t *pOutput)
- int pack_sms_SetSMSCAddress (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SetSMSCAddress_t *reqParam)
- int unpack_sms_SetSMSCAddress (uint8_t *pResp, uint16_t respLen)
- int pack_sms_SaveSMS (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SaveSMS_t *reqParam)
- int unpack_sms_SaveSMS (uint8_t *pResp, uint16_t respLen, unpack_sms_SaveSMS_t *pOutput)
- int pack_sms_SLQSGetSmsBroadcastConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SLQSGetSmsBroadcastConfig_t *reqParam)
- int unpack_sms_SLQSGetSmsBroadcastConfig (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSGetSmsBroadcastConfig_t *pOutput)
- int pack_sms_SLQSSetSmsBroadcastConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SLQSSetSmsBroadcastConfig_t *reqParam)
- int unpack_sms_SLQSSetSmsBroadcastConfig (uint8_t *pResp, uint16_t respLen)
- int pack_sms_SLQSSetSmsBroadcastActivation (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SLQSSetSmsBroadcastActivation_t *reqParam)
- int unpack_sms_SLQSSetSmsBroadcastActivation (uint8_t *pResp, uint16_t respLen)
- int pack_sms_SLQSGetTransLayerInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_sms_SLQSGetTransLayerInfo (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSGetTransLayerInfo_t *pOutput)
- int pack_sms_SLQSGetTransNWRegInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_sms_SLQSGetTransNWRegInfo (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSGetTransNWRegInfo_t *pOutput)
- int pack_sms_SLQSGetIndicationRegister (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_sms_SLQSGetIndicationRegister (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSGetIndicationRegister_t *pOutput)

- int pack_sms_SLQSSetIndicationRegister (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SLQSSetIndicationRegister_t *reqParam)
- int unpack_sms_SLQSSetIndicationRegister (uint8_t *pResp, uint16_t respLen)
- int pack_sms_SLQSSmsSetRoutes (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SLQSSmsSetRoutes_t *reqParam)
- int unpack_sms_SLQSSmsSetRoutes (uint8_t *pResp, uint16_t respLen)
- int pack_sms_SLQSSmsGetMessageProtocol (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_sms_SLQSSmsGetMessageProtocol (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSSmsGetMessageProtocol_t *pOutput)
- int pack_sms_SLQSSmsGetMaxStorageSize (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SLQSSmsGetMaxStorageSize_t *reqParam)
- int unpack_sms_SLQSSmsGetMaxStorageSize (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSSmsGetMaxStorageSize_t *pOutput)
- int pack_sms_SLQSGetMessageWaiting (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_sms_SLQSGetMessageWaiting (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSGetMessageWaiting_t *pOutput)
- int pack_sms_SLQSSendAsyncSMS (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SLQSSendAsyncSMS_t *reqParam)
- int unpack_sms_SLQSSendAsyncSMS (uint8_t *pResp, uint16_t respLen)
- int pack_sms_SLQSSetSmsStorage (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SLQSSetSmsStorage_t *reqParam)
- int unpack_sms_SLQSSetSmsStorage (uint8_t *pResp, uint16_t respLen)
- int pack_sms_SLQSSwiGetSMSStorage (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_sms_SLQSSwiGetSMSStorage (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSSwiGetSMSStorage_t *pOutput)

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 Enumeration Type Documentation

9.55.2.1 enum eqmiCbkSetStatus

Enumerator

LITEQMI_QMI_CBK_PARAM_RESET

LITEQMI_QMI_CBK_PARAM_SET

LITEQMI_QMI_CBK_PARAM_NOCHANGE

9.55.3 Function Documentation

9.55.3.1 int pack_sms_GetSMSCAddress (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Gets the SMS center address pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.2 int pack_sms_SaveSMS (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_sms_SaveSMS_t * *reqParam*)

Saves an SMS message to device memory pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
	reqParam	packed request

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.3 int pack_sms_SendSMS (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_sms_SendSMS_t * *reqParam*)

send sms list pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
	reqParam	packed request

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.4 int pack_sms_SetNewSMSCallback (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_sms_SetNewSMSCallback_t *reqParam*)

set new sms callback pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
	reqParam	packed request

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.5 int pack_sms_SetSMSCAddress (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_sms_SetSMSCAddress_t * *reqParam*)

Set the SMS center address pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
	reqParam	packed request

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.6 int pack_sms_SLQSDeleteSMS (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_sms_SLQSDeleteSMS_t * *reqParam*)

delete sms pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
	reqParam	packed request

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.7 int pack_sms_SLQSGetIndicationRegister (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Gets registration state of different WMS indications pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.8 int pack_sms_SLQSGetMessageWaiting (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Gets the message waiting information pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.9 int pack_sms_SLQSGetSMS (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_sms_SLQSGetSMS_t * *reqParam*)

get sms pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
	reqParam	packed request

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.10 int pack_sms_SLQSGetSmsBroadcastConfig (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_sms_SLQSGetSmsBroadcastConfig_t * *reqParam*)

Provides Information about the SMS BroadcastConfiguration pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
	reqParam	packed request

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.11 int pack_sms_SLQSGetSMSList (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_sms_SLQSGetSMSList_t * *reqParam*)

get sms list pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
	reqParam	packed request

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.12 int pack_sms_SLQSGetTransLayerInfo (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)

Gets information about the transport layer pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.13 int pack_sms_SLQSGetTransNWRegInfo (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)

Gets transport layer network registration info pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.14 int pack_sms_SLQSModifySMSStatus (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_sms_SLQSModifySMSStatus_t * reqParam)

modify sms status pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
	reqParam	packed request

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.15 int pack_sms_SLQSSendAsyncSMS (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_sms_SLQSSendAsyncSMS_t * *reqParam*)

Sends an SMS message for immediate over-the-air transmission pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
	reqParam	packed request

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.16 int pack_sms_SLQSSetIndicationRegister (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_sms_SLQSSetIndicationRegister_t * *reqParam*)

Sets the registration state of different WMS indications pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
	reqParam	packed request

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.17 int pack_sms_SLQSSetSmsBroadcastActivation (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_sms_SLQSSetSmsBroadcastActivation_t * *reqParam*)

Enables or disables the reception of broadcast SMS messages pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
	reqParam	packed request

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.18 int pack_sms_SLQSSetSmsBroadcastConfig (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_sms_SLQSSetSmsBroadcastConfig_t * *reqParam*)

Sets the information about the SMS BroadcastConfiguration pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
	reqParam	packed request

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.19 int pack_sms_SLQSSetSmsStorage (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_sms_SLQSSetSmsStorage_t * *reqParam*)

Sets the SMS Storage on the device pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
	reqParam	packed request

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.20 int pack_sms_SLQSSmsGetMaxStorageSize (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_sms_SLQSSmsGetMaxStorageSize_t * *reqParam*)

Get the maximum number of messages that can be stored in the specified memory storage. Also it provides the number of slots currently available pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
	reqParam	packed request

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.21 int pack_sms_SLQSSmsGetMessageProtocol (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Gets the message protocol currently in use for the WMS client pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.22 int pack_sms_SLQSSmsSetRoutes (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_sms_SLQSSmsSetRoutes_t * *reqParam*)

Sets the action performed on SMS message receipt for specified message routes. It also specifies the action performed on SMS receipt of status reports pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
	reqParam	packed request

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.23 int pack_sms_SLQSSwiGetSMSStorage (pack_qmi_t * *pCtx,* uint8_t * *pReqBuf,* uint16_t * *pLen*)

Gets the current SMS configuration that is applied to all incoming and outgoing messages pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.24 int unpack_sms_GetSMSCAddress (uint8_t * *pResp,* uint16_t *respLen,* unpack_sms_GetSMSCAddress_t * *pOutput*)

Gets the SMS center address unpack.

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.25 int unpack_sms_SaveSMS (uint8_t * *pResp,* uint16_t *respLen,* unpack_sms_SaveSMS_t * *pOutput*)

Saves an SMS message to device memory unpack.

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.26 int unpack_sms_SendSMS (uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SendSMS_t * *pOutput*)

send sms unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.27 int unpack_sms_SetNewSMSCallback (uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SetNewSMS-Callback_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

9.55.3.28 int unpack_sms_SetNewSMSCallback_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SetNewSMSCallback_ind_t * *pOutput*)

set new sms callback indication unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.29 int unpack_sms_SetSMSCAddress (uint8_t * *pResp*, uint16_t *respLen*)

Set the SMS center address unpack.

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.30 int unpack_sms_SLQSDeleteSMS (uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SLQSDeleteSMS_t * *pOutput*)

delete sms unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.31 int unpack_sms_SLQSGetIndicationRegister (uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SLQSGetIndicationRegister_t * *pOutput*)

Gets registration state of different WMS indications unpack.

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.32 int unpack_sms_SLQSGetMessageWaiting (uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SLQSGetMessageWaiting_t * *pOutput*)

Gets the message waiting information unpack.

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.33 int unpack_sms_SLQSGetSMS (uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SLQSGetSMS_t * *pOutput*)

get sms unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.34 int unpack_sms_SLQSGetSmsBroadcastConfig (uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SLQSGetSmsBroadcastConfig_t * *pOutput*)

Provides Information about the SMS BroadcastConfiguration unpack.

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.35 int unpack_sms_SLQSGetSMSList (uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SLQSGetSMSList_t * *pOutput*)

get sms list unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.36 int unpack_sms_SLQSGetTransLayerInfo (uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SLQSGetTransLayerInfo_t * *pOutput*)

Gets information about the transport layer unpack.

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.37 int unpack_sms_SLQSGetTransNWRegInfo (uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SLQSGetTransNWRegInfo_t * *pOutput*)

Gets transport layer network registration info unpack.

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.38 int unpack_sms_SLQSModifySMSStatus (uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SLQSModifySMSStatus_t * *pOutput*)

modify sms status unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.3.39 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

9.55.3.40 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

9.55.3.41 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

9.55.3.42 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

9.55.3.43 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

9.55.3.44 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

9.55.3.45 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

9.55.3.46 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

9.55.3.47 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

9.55.3.48 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

pCtx	[out] • See pack_qmi_t for more information
pReqBuf	[in,out] • Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
pLen	[in,out] • On input, size of pReqBuf • On ouptut, number of bytes actually packed

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 20 seconds

9.58.2.2 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

pCtx	[out] • See pack_qmi_t for more information
pReqBuf	[in,out] • Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
pLen	[in,out] • On input, size of pReqBuf • On output, number of bytes actually packed
reqParam	[in] • See pack_swioma_SLQSOMADMCancelSession_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 20 seconds

9.58.2.3 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

pCtx	[out] • See pack_qmi_t for more information
pReqBuf	[in,out] • Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
pLen	[in,out] • On input, size of pReqBuf • On ouptut, number of bytes actually packed
reqParam	[in] • See pack_swioma_SLQSOMADMGetSessionInfo_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 20 seconds

9.58.2.4 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 SLQSOMADMGet-Settings2

Parameters

pCtx	[out] • See pack_qmi_t for more information

pReqBuf	[in,out] • Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
pLen	[in,out] • On input, size of pReqBuf • On 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

pCtx	[out] • See pack_qmi_t for more information
pReqBuf	[in,out] • Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
pLen	[in,out] • On input, size of pReqBuf • On ouptut, number of bytes actually packed
reqParam	[in] • See pack_swioma_SLQSOMADMSendSelection_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 20 seconds

9.58.2.6 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

pCtx	[out] • See pack_qmi_t for more information
pReqBuf	[in,out] • Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
pLen	[in,out] • On input, size of pReqBuf • On ouptut, number of bytes actually packed
reqParam	[in] • See pack_swioma_SLQSOMADMSetSettings_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 20 seconds

9.58.2.7 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

pCtx	[out] • See pack_qmi_t for more information
pReqBuf	[in,out] • Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
pLen	[in,out] • On input, size of pReqBuf • On ouptut, number of bytes actually packed
reqParam	[in] • See pack_swioma_SLQSOMADMStartSession_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 20 seconds

9.58.2.8 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

pResp	[in] • Response from modem
respLen	[in] • Length of pResp from modem

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

- Please use eQMI_SWIOMA_EVENT_IND indication to identify this event from SWIOMA service read function

9.58.2.9 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

pResp	[in] • Response from modem
respLen	[in] • Length of pResp from modem
pOutput	[out] • See unpack_swioma_SLQSOMADMAlertCallback_ind_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.58.2.10 int unpack_swioma_SLQSOMADMCancelSession (uint8_t * *pResp*, uint16_t *respLen*)

Function to pack cancel OMA-DM session command This maps to SLQSOMADMCancelSession

Parameters

pResp	[in] • Response from modem
respLen	[in] • Length of pResp from modem

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.58.2.11 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

pResp	[in] • Response from modem
respLen	[in] • Length of pResp from modem
pOutput	[out] • See unpack_swioma_SLQSOMADMGetSessionInfo_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.58.2.12 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

pResp	[in] • Response from modem
respLen	[in] • Length of pResp from modem
pOutput	[out] • See unpack_swioma_SLQSOMADMGetSettings_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.58.2.13 int unpack_swioma_SLQSOMADMSendSelection (uint8_t * *pResp*, uint16_t *respLen*)

Function to unpack OMA-DM send selection command This maps to SLQSOMADMSendSelection2

Parameters

pResp	[in] • Response from modem
respLen	[in] • Length of pResp from modem

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.58.2.14 int unpack_swioma_SLQSOMADMSetSettings (uint8_t * *pResp*, uint16_t *respLen*)

Function to unpack OMA-DM set settings command This maps to SLQSOMADMSetSettings3

Parameters

pResp	[in] • Response from modem

respLen	[in] • Length of pResp from modem

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.58.2.15 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

pResp	[in] • Response from modem
respLen	[in] • Length of pResp from modem
pOutput	[out] • See unpack_swioma_SLQSOMADMStartSession_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.59 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*)

Pack get slots status.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.11 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_SLQSUIMEventRegister_t * *pOutput*)

UIM Status Change callback enable unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.29 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 Detailed Description

9.61.2 S1

- 3GPP Mobile Radio Interface Layer 3 Specification : Core Network Protocols; Stage 3 (Release 7)
- 3GPP TS 24.008 V7.0.0 (2005-06)

9.61.3 Macro Definition Documentation

9.61.3.1 #define BARRING_PASSWORD_LENGTH 4

9.61.3.2 #define MAX_VOICE_CALL_NO_LEN 81

9.61.3.3 #define MAX_VOICE_DESCRIPTION_LENGTH 255

9.61.3.4 #define MAXVOICEUSSDLENGTH 182

9.61.3.5 #define VOICE_MAX_NO_OF_CALLS 20

9.61.4 Enumeration Type Documentation

9.61.4.1 enum liteServiceClassInformation

Service Class information

Enumerator

LITE_VOICE_SUPS_SRV_CLASS_NONE

LITE_VOICE_SUPS_SRV_CLASS_VOICE

LITE_VOICE_SUPS_SRV_CLASS_DATA

LITE_VOICE_SUPS_SRV_CLASS_FAX

LITE_VOICE_SUPS_SRV_CLASS_SMS

LITE_VOICE_SUPS_SRV_CLASS_DATACIRCUITSYNC

LITE_VOICE_SUPS_SRV_CLASS_DATACIRCUITASYNC

LITE_VOICE_SUPS_SRV_CLASS_PACKETACCESS

LITE_VOICE_SUPS_SRV_CLASS_PADACCESS

9.61.5 Function Documentation

9.61.5.1 int pack_voice_AnswerUSSD (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_AnswerUSSD_t * *reqArg*)

Responds to a USSD request from the network pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.5.2 int pack_voice_CancelUSSD (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Cancels an in-progress USSD operation pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.5.3 int pack_voice_OriginateUSSD (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_OriginateUSSD_t * *reqArg*)

Initiates a USSD operation pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.5.4 int pack_voice_SLQSOriginateUSSD (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSOriginateUSSD_t * *reqArg*)

Initiates a USSD session pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.5.5 int pack_voice_SLQSVoiceALSSelectLine (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceALSSelectLine_t * *reqArg*)

allows the user to select the preferred line pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.5.6 int pack_voice_SLQSVoiceALSSetLineSwitching (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceALSSetLineSwitching_t * *reqArg*)

sets the line switch setting on the card pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.5.7 int pack_voice_SLQSVoiceAnswerCall (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceAnswerCall_t * *pReq*)

Answers an incoming voice call pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReq*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.5.8 int pack_voice_SLQSVoiceBindSubscription (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceBindSubscription_t * *reqArg*)

binds a subscription type to a specific voice client ID pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.5.9 int pack_voice_SLQSVoiceBurstDTMF (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceBurstDTMF_t * *pReq*)

Sends a burst Dual-Tone Multi frequency (DTMF) pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReq*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.5.10 int pack_voice_SLQSVoiceDialCall (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceDialCall_t * *pReq*)

Originates a voice call (MO call) pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReq*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.5.11 int pack_voice_SLQSVoiceEndCall (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceEndCall_t * *pReq*)

Ends a voice call pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReq*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.5.12 int pack_voice_SLQSVoiceGetAllCallInfo (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

information associated with all the calls originating or terminating pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.5.13 int pack_voice_SLQSVoiceGetCallBarring (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceGetCallBarring_t * *pReq*)

Status of Call Barring Supplementary Service pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReq*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.5.14 int pack_voice_SLQSVoiceGetCallForwardingStatus (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_voice_SLQSVoiceGetCallForwardingStatus_t * pReq)

Status of Call Forwarding Supplementary Service pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReq*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.5.15 int pack_voice_SLQSVoiceGetCallInfo (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_voice_SLQSVoiceGetCallInfo_t * pReq)

information associated with a call pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReq*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.5.16 int pack_voice_SLQSVoiceGetCallWaiting (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_voice_SLQSVoiceGetCallWaiting_t * pReq)

Status of Call Waiting Supplementary Service pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReq*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.5.17 int pack_voice_SLQSVoiceGetCLIP (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

status of the Calling Line Identification Presentation (CLIP) pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.5.18 int pack_voice_SLQSVoiceGetCLIR (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

status of the Calling Line Identification Restriction (CLIR) pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.5.19 int pack_voice_SLQSVoiceGetCNAP (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

status of the Calling Name Presentation(CNAP) pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.5.20 int pack_voice_SLQSVoiceGetCOLP (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

status of the Connected Line Identification Presentation pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.5.21 int pack_voice_SLQSVoiceGetCOLR (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

status of the Connected Line Identification Restriction pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.5.22 int pack_voice_SLQSVoiceGetConfig (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceGetConfig_t * *reqArg*)

retrieves various configuration parameters that control the modem behavior pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.5.23 int pack_voice_SLQSVoiceIndicationRegister (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceIndicationRegister_t * *reqArg*)

Sets the registration state for different QMI_VOICE indications pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.5.24 int pack_voice_SLQSVoiceManageCalls (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceManageCalls_t * *pReq*)

Manages the calls by using the supplementary service pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReq*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.5.25 int pack_voice_SLQSVoiceOrigUSSDNoWait (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceOrigUSSDNoWait_t * *reqArg*)

initiates a USSD operation No Wait pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.5.26 int pack_voice_SLQSVoiceSendFlash (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceSendFlash_t * *pReq*)

sends a simple flash message pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReq*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.5.27 int pack_voice_SLQSVoiceSetCallBarringPassword (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceSetCallBarringPassword_t * *pReq*)

Sets a Call Barring Password pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReq*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.5.28 int pack_voice_SLQSVoiceSetConfig (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceSetConfig_t * *pReq*)

configuration parameters that control the modem behavior pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReq*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.5.29 int pack_voice_SLQSVoiceSetPreferredPrivacy (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceSetPreferredPrivacy_t * *reqArg*)

sets the voice privacy preference pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.5.30 int pack_voice_SLQSVoiceSetSUPSService (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceSetSUPSService_t * *pReq*)

call-independent supplementary services pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReq*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.5.31 int pack_voice_SLQSVoiceStartContDTMF (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceStartContDTMF_t * *pReq*)

Starts a continuous DTMF pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReq*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.5.32 int pack_voice_SLQSVoiceStopContDTMF (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceStopContDTMF_t * *pReq*)

Stops a continuous DTMF pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReq*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.5.33 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

9.61.5.34 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

9.61.5.35 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

9.61.5.36 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

9.61.5.37 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

9.61.5.38 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

9.61.5.39 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

9.61.5.40 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

9.61.5.41 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

9.61.5.42 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

9.61.5.43 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

9.61.5.44 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

9.61.5.45 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

9.61.5.46 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

9.61.5.47 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

9.61.5.48 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

9.61.5.49 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

9.61.5.50 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

9.61.5.51 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

9.61.5.52 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

9.61.5.53 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

9.61.5.54 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

9.61.5.55 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

9.61.5.56 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

9.61.5.57 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

9.61.5.58 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

9.61.5.59 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

9.61.5.60 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

9.61.5.61 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

9.61.5.62 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

9.61.5.63 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

9.61.5.64 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

Note

return eQCWWAN_ERR_NULL_TLV then callID value is invalid.

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 wds_transferStatInd
- struct pack_wds_SLQSGetDUNCallInfo_t
- struct connectionStatus
- struct dunchannelRate

- struct unpack_wds_SLQSGetDUNCallInfo_t
- struct qmiWDSDataBearerTechnology
- struct unpack_wds_SLQSGetDataBearerTechnology_t
- struct pack_wds_SLQSGetDataBearerTechnology_t
- struct pack_wds_SLQSSetIPFamilyPreference_t
- struct unpack_wds_SLQSSetIPFamilyPreference_t
- struct pack_wds_SetDefaultProfileNum_t
- struct pack_wds_GetDefaultProfileNum_t
- struct unpack_wds_GetDefaultProfileNum_t
- struct wdsDhcpv4ProfileId
- struct wdsDhcpv4HwConfig
- struct wdsDhcpv4Option
- struct wdsDhcpv4OptionList
- struct pack_wds_SLQSSGetDHCPv4ClientConfig_t
- struct unpack_wds_SLQSSGetDHCPv4ClientConfig_t
- struct pack_wds_GetPacketStatistics_t
- struct unpack_wds_GetPacketStatistics_t
- struct unpack_wds_GetByteTotals_t
- struct unpack_wds_SLQSGetCurrentChannelRate_t
- struct unpack_wds_SLQSSGetLoopback_t
- struct pack_wds_SLQSSSetLoopback_t
- struct wds_DataULongTlv
- struct wds_DataULongLongTlv
- struct unpack_RMTransferStatistics_ind_t
- struct pack_wds_DHCPv4ClientLeaseChange_t
- struct wds_DHCPProfileIdTlv
- struct wds_DHCPLeaseStateTlv
- struct wds_IPv4AdTlv
- struct wds_DHCPOpt
- struct wds_DHCPLeaseOptTlv
- struct unpack_wds_DHCPv4ClientLease_ind_t
- struct pack_wds_SetMobileIP_t
- struct pack_wds_SetMobileIPParameters_t
- struct pack_wds_SetAutoconnect_t
- struct unpack_wds_GetAutoconnect_t
- struct wds_TrStatInd
- struct pack_wds_SLQSWdsSetEventReport_t
- struct wds_DHCPv4ProfileId
- struct wds_DHCPv4HWConfig
- struct wds_DHCPv4Option
- struct wds_DHCPv4OptionList
- struct pack_wds_SLQSSSetDHCPv4ClientConfig_t
- struct unpack_wds_GetDataBearerTechnology_t
- struct qmuxIPTable

Macros

- #define IPV6_ADDRESS_ARRAY_SIZE 8
- #define MAX_WDS_3GPP_CONF_LTE_ATTACH_PROFILE_LIST_SIZE 24
- #define PACK_WDS_IPV4 4
- #define PACK_WDS_IPV6 6
- #define BYT_STAT_STAT_MASK 0X000000C0
- #define WDS_DHCP_MAX_NUM_OPTIONS 30
- #define WDS_DHCP_OPTION_DATA_BUF_SIZE 2048 /∗ current max size of raw message in SDK process is 2048 ∗/

Typedefs

- typedef union unpackWdsProfileParam UnpackQmiProfileInfo

Enumerations

- enum liteQmiDataBearerMasks {
  QMI_LITE_WDS_CURRENT_CALL_DB_MASK = 0x01,
  QMI_LITE_WDS_LAST_CALL_DB_MASK = 0x02 }

Functions

- int pack_wds_SLQSStartDataSession (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSStartDataSession_t *reqArg)
- int unpack_wds_SLQSStartDataSession (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSStartDataSession_t *pOutput)
- int unpack_wds_SLQSSetPacketSrvStatusCallback (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSSetPacketSrvStatusCallback_t *pOutput)
- int pack_wds_SLQSStopDataSession (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSStopDataSession_t *reqArg)
- int unpack_wds_SLQSStopDataSession (uint8_t *pResp, uint16_t respLen)
- int unpack_wds_SLQSGetRuntimeSettings (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSGetRuntimeSettings_t *pOutput)
- int unpack_wds_SLQSSetWdsEventCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSSetWdsEventCallback_ind_t *pOutput)
- int unpack_wds_SLQSSetWdsEventCallback (uint8_t *pResp, uint16_t respLen)
- int pack_wds_SLQSSetWdsEventCallback (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSSetWdsEventCallback_t *reqArg)
- int pack_wds_SLQSGetRuntimeSettings (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSGetRuntimeSettings_t *reqArg)
- int pack_wds_SLQSCreateProfile (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSCreateProfile_t *reqArg)
- int unpack_wds_SLQSCreateProfile (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSCreateProfile_t *pOutput)
- int pack_wds_SLQSModifyProfile (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSModifyProfile_t *reqArg)
- int unpack_wds_SLQSModifyProfile (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSModifyProfile_t *pOutput)
- int pack_wds_SLQSGetProfileSettings (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSGetProfileSettings_t *reqArg)
- int unpack_wds_SLQSGetProfileSettings (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSGetProfileSettings_t *pOutput)
- int pack_wds_GetSessionState (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_wds_GetSessionState (uint8_t *pResp, uint16_t respLen, unpack_wds_GetSessionState_t *pOutput)
- int pack_wds_GetDefaultProfile (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetDefaultProfile_t *reqParam)
- int unpack_wds_GetDefaultProfile (uint8_t *pResp, uint16_t respLen, unpack_wds_GetDefaultProfile_t *pOutput)
- int pack_wds_GetConnectionRate (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_wds_GetConnectionRate (uint8_t *pResp, uint16_t respLen, unpack_wds_GetConnectionRate_t *pOutput)
- int pack_wds_GetPacketStatus (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetPacketStatus_t *reqParam)

- int unpack_wds_GetPacketStatus (uint8_t *pResp, uint16_t respLen, unpack_wds_GetPacketStatus_t *pOutput)
- int pack_wds_GetSessionDuration (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetSessionDuration_t *reqParam)
- int unpack_wds_GetSessionDuration (uint8_t *pResp, uint16_t respLen, unpack_wds_GetSessionDuration_t *pOutput)
- int pack_wds_GetDormancyState (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetDormancyState_t *reqParam)
- int unpack_wds_GetDormancyState (uint8_t *pResp, uint16_t respLen, unpack_wds_GetDormancyState_t *pOutput)
- int pack_wds_SLQSDeleteProfile (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSDeleteProfile_t *reqParam)
- int unpack_wds_SLQSDeleteProfile (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSDeleteProfile_t *pOutput)
- int pack_wds_SetDefaultProfile (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SetDefaultProfile_t *reqParam)
- int unpack_wds_SetDefaultProfile (uint8_t *pResp, uint16_t respLen)
- int pack_wds_SLQSGet3GPPConfigItem (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_wds_SLQSGet3GPPConfigItem (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSGet3GPPConfigItem_t *pOutput)
- int pack_wds_SLQSSet3GPPConfigItem (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSSet3GPPConfigItem_t *reqParam)
- int unpack_wds_SLQSSet3GPPConfigItem (uint8_t *pResp, uint16_t respLen)
- int pack_wds_GetMobileIP (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetMobileIP_t *pReqParam)
- int unpack_wds_GetMobileIP (uint8_t *pResp, uint16_t respLen, unpack_wds_GetMobileIP_t *pOutput)
- int pack_wds_GetMobileIPProfile (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetMobileIPProfile_t *reqParam)
- int unpack_wds_GetMobileIPProfile (uint8_t *pResp, uint16_t respLen, unpack_wds_GetMobileIPProfile_t *pOutput)
- int pack_wds_SLQSGetCurrDataSystemStat (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSGetCurrDataSystemStat_t *pReqParam)
- int unpack_wds_SLQSGetCurrDataSystemStat (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSGetCurrDataSystemStat_t *pOutput)
- int pack_wds_GetLastMobileIPError (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetLastMobileIPError_t *pReqParam)
- int unpack_wds_GetLastMobileIPError (uint8_t *pResp, uint16_t respLen, unpack_wds_GetLastMobileIPError_t *pOutput)
- int pack_wds_RMSetTransferStatistics (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_RMSetTransferStatistics_t *reqParam)
- int unpack_wds_RMSetTransferStatistics (uint8_t *pResp, uint16_t respLen, unpack_wds_RMSetTransferStatistics_t *pOutput)
- int pack_wds_SetMobileIPProfile (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SetMobileIPProfile_t *reqParam)
- int unpack_wds_SetMobileIPProfile (uint8_t *pResp, uint16_t respLen, unpack_wds_SetMobileIPProfile_t *pOutput)
- int pack_wds_SLQSWdsSwiPDPRuntimeSettings (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSWdsSwiPDPRuntimeSettings_t *reqParam)
- int unpack_wds_SLQSWdsSwiPDPRuntimeSettings (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t *pOutput)
- int pack_wds_SLQSGetDUNCallInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSGetDUNCallInfo_t *reqParam)
- int unpack_wds_SLQSGetDUNCallInfo (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSGetDUNCallInfo_t *pOutput)
- int pack_wds_SLQSGetDataBearerTechnology (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSGetDataBearerTechnology_t *pReqParam)

- int unpack_wds_SLQSGetDataBearerTechnology (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSGetDataBearerTechnology_t *pOutput)
- int pack_wds_SLQSSetIPFamilyPreference (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSSetIPFamilyPreference_t *pReqParam)
- int unpack_wds_SLQSSetIPFamilyPreference (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSSetIPFamilyPreference_t *pOutput)
- int pack_wds_SetDefaultProfileNum (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SetDefaultProfileNum_t *pReqParam)
- int unpack_wds_SetDefaultProfileNum (uint8_t *pResp, uint16_t respLen)
- int pack_wds_GetDefaultProfileNum (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetDefaultProfileNum_t *pReqParam)
- int unpack_wds_GetDefaultProfileNum (uint8_t *pResp, uint16_t respLen, unpack_wds_GetDefaultProfileNum_t *pOutput)
- int pack_wds_SLQSSGetDHCPv4ClientConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSSGetDHCPv4ClientConfig_t *pReq)
- int unpack_wds_SLQSSGetDHCPv4ClientConfig (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSSGetDHCPv4ClientConfig_t *pOutput)
- int pack_wds_GetPacketStatistics (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetPacketStatistics_t *pReq)
- int unpack_wds_GetPacketStatistics (uint8_t *pResp, uint16_t respLen, unpack_wds_GetPacketStatistics_t *pOutput)
- int pack_wds_GetByteTotals (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_wds_GetByteTotals (uint8_t *pResp, uint16_t respLen, unpack_wds_GetByteTotals_t *pOutput)
- int pack_wds_SLQSGetCurrentChannelRate (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_wds_SLQSGetCurrentChannelRate (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSGetCurrentChannelRate_t *pOutput)
- int pack_wds_SLQSSGetLoopback (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_wds_SLQSSGetLoopback (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSSGetLoopback_t *pOutput)
- int pack_wds_SLQSSSetLoopback (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSSSetLoopback_t *reqArg)
- int unpack_wds_SLQSSSetLoopback (uint8_t *pResp, uint16_t respLen)
- int unpack_wds_RMTransferStatistics_ind (uint8_t *pResp, uint16_t respLen, unpack_RMTransferStatistics_ind_t *pOutput)
- int pack_wds_DHCPv4ClientLeaseChange (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_DHCPv4ClientLeaseChange_t *reqArg)
- int unpack_wds_DHCPv4ClientLeaseChange (uint8_t *pResp, uint16_t respLen)
- int unpack_wds_DHCPv4ClientLease_ind (uint8_t *pResp, uint16_t respLen, unpack_wds_DHCPv4ClientLease_ind_t *pOutput)
- int pack_wds_SetMobileIP (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SetMobileIP_t *reqArg)
- int unpack_wds_SetMobileIP (uint8_t *pResp, uint16_t respLen)
- int pack_wds_SetMobileIPParameters (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SetMobileIPParameters_t *reqArg)
- int unpack_wds_SetMobileIPParameters (uint8_t *pResp, uint16_t respLen)
- int pack_wds_SetAutoconnect (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SetAutoconnect_t *reqArg)
- int unpack_wds_SetAutoconnect (uint8_t *pResp, uint16_t respLen)
- int pack_wds_GetAutoconnect (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_wds_GetAutoconnect (uint8_t *pResp, uint16_t respLen, unpack_wds_GetAutoconnect_t *pOutput)
- int pack_wds_SLQSWdsSetEventReport (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSWdsSetEventReport_t *reqArg)
- int unpack_wds_SLQSWdsSetEventReport (uint8_t *pResp, uint16_t respLen)
- int pack_wds_SLQSWdsGoDormant (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_wds_SLQSWdsGoDormant (uint8_t *pResp, uint16_t respLen)

- int pack_wds_SLQSWdsGoActive (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_wds_SLQSWdsGoActive (uint8_t *pResp, uint16_t respLen)
- int pack_wds_SLQSResetPacketStatics (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_wds_SLQSResetPacketStatics (uint8_t *pResp, uint16_t respLen)
- int pack_wds_SLQSSSetDHCPv4ClientConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSSSetDHCPv4ClientConfig_t *reqArg)
- int unpack_wds_SLQSSSetDHCPv4ClientConfig (uint8_t *pResp, uint16_t respLen)
- int pack_wds_GetDataBearerTechnology (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_wds_GetDataBearerTechnology (uint8_t *pResp, uint16_t respLen, unpack_wds_GetDataBearerTechnology_t *pOutput)
- int pack_wds_SetMuxID (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, uint8_t *pMuxID)
- int unpack_wds_SetMuxID (uint8_t *pResp, uint16_t respLen)

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 • See LibpackProfile3GPP
SlqsProfile3GPP2	3GPP2 profile • See LibpackProfile3GPP2

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*)

get default profile pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.7 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	*pReqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.20 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_SetMuxID (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, uint8_t * *pMuxID*)

Set MUX ID pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pMuxID*	MUX ID. • 0x80 to 0x88

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.24 int pack_wds_SLQSCreateProfile (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSCreateProfile_t * *reqArg*)

Create Profile pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.25 int pack_wds_SLQSDeleteProfile (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSDeleteProfile_t * *reqParam*)

delete stored profile pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.26 int pack_wds_SLQSGet3GPPConfigItem (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

get 3Gpp config items pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.27 int pack_wds_SLQSGetCurrDataSystemStat (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSGetCurrDataSystemStat_t * *pReqParam*)

get current data system pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.28 int pack_wds_SLQSGetCurrentChannelRate (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

get current Tx/Rx channel bitrate of the current packet data pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.29 int pack_wds_SLQSGetDataBearerTechnology (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSGetDataBearerTechnology_t * *pReqParam*)

get data bearer technology pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.30 int pack_wds_SLQSGetDUNCallInfo (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSGetDUNCallInfo_t * *reqParam*)

get dun call info pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.31 int pack_wds_SLQSGetProfileSettings (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSGetProfileSettings_t * *reqArg*)

Get Profile Settings pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.32 int pack_wds_SLQSGetRuntimeSettings (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSGetRuntimeSettings_t * *reqArg*)

get runtime settings pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.33 int pack_wds_SLQSModifyProfile (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSModifyProfile_t * *reqArg*)

Modify Profile pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.34 int pack_wds_SLQSResetPacketStatics (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Reset packet data transfer statistics pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.35 int pack_wds_SLQSSet3GPPConfigItem (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSSet3GPPConfigItem_t * *reqParam*)

set 3Gpp config items pack

Parameters

in, out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.36 int pack_wds_SLQSSetIPFamilyPreference (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSSetIPFamilyPreference_t * *pReqParam*)

Set IP Family Preference pack

Parameters

in, out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.37 int pack_wds_SLQSSetWdsEventCallback (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSSetWdsEventCallback_t * *reqArg*)

set event callback pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: No

9.62.4.38 int pack_wds_SLQSSGetDHCPv4ClientConfig (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSSGetDHCPv4ClientConfig_t * *pReq*)

get DHCPv4 Client Config pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReq*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.39 int pack_wds_SLQSSGetLoopback (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

get the value of loopback mode and multiplier pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.40 int pack_wds_SLQSSSetDHCPv4ClientConfig (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSSSetDHCPv4ClientConfig_t * *reqArg*)

Gets the DHCP Client V4 Configuration pack

Parameters

in, out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

see qmerrno.h for eQCWWAN_xxx error values

9.62.4.41 int pack_wds_SLQSSSetLoopback (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSSSetLoopback_t * *reqArg*)

Enable/disable Data Loopback Mode and set the value of loopback multiplier pack

Parameters

in, out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.42 int pack_wds_SLQSStartDataSession (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSStartDataSession_t * *reqArg*)

Start data session

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.43 int pack_wds_SLQSStopDataSession (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSStopDataSession_t * *reqArg*)

stop data session pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.44 int pack_wds_SLQSWdsGoActive (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Gets the device into Active state pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.45 int pack_wds_SLQSWdsGoDormant (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Gets the device into dormant state pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.46 int pack_wds_SLQSWdsSetEventReport (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSWdsSetEventReport_t * *reqArg*)

Sets the event report parameters pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.47 int pack_wds_SLQSWdsSwiPDPRuntimeSettings (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSWdsSwiPDPRuntimeSettings_t * *reqParam*)

swi pdp runtime settings pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.48 int unpack_wds_DHCPv4ClientLease_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_DHCPv4ClientLease_ind_t * *pOutput*)

DHCP lease state has changed indication unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.49 int unpack_wds_DHCPv4ClientLeaseChange (uint8_t * *pResp*, uint16_t *respLen*)

DHCPv4 lease state changes unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.50 int unpack_wds_GetAutoconnect (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetAutoconnect_t * *pOutput*)

Gets auto connect data session setting unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.51 int unpack_wds_GetByteTotals (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetByteTotals_t * *pOutput*)

get Rx/Tx byte counts since the start of the last packet data session unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.52 int unpack_wds_GetConnectionRate (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetConnectionRate_t * *pOutput*)

get connection rate unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.53 int unpack_wds_GetDataBearerTechnology (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetDataBearerTechnology_t * *pOutput*)

get current data bearer technology unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.54 int unpack_wds_GetDefaultProfile (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetDefaultProfile_t * *pOutput*)

get default profile unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.55 int unpack_wds_GetDefaultProfileNum (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetDefaultProfileNum_t * *pOutput*)

get default profile number unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.56 int unpack_wds_GetDormancyState (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetDormancyState_t * *pOutput*)

get dormancy state unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.57 int unpack_wds_GetLastMobileIPError (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetLastMobileIP-Error_t * *pOutput*)

get current data system unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.58 int unpack_wds_GetMobileIP (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetMobileIP_t * *pOutput*)

get mobile ip mode unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.59 int unpack_wds_GetMobileIPProfile (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetMobileIPProfile_t * *pOutput*)

get mobile ip profile unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.60 int unpack_wds_GetPacketStatistics (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetPacketStatistics_t * *pOutput*)

gets current packet transfer counter values unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.61 int unpack_wds_GetPacketStatus (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetPacketStatus_t * *pOutput*)

get packet status unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.62 int unpack_wds_GetSessionDuration (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetSessionDuration_t * *pOutput*)

get session duration unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.63 int unpack_wds_GetSessionState (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetSessionState_t * *pOutput*)

get session state unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.64 int unpack_wds_RMSetTransferStatistics (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_RMSetTransferStatistics_t * *pOutput*)

RM set transfer statistics unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.65 int unpack_wds_RMTransferStatistics_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_RMTransferStatistics_ind_t * *pOutput*)

RM transfer statistics indication unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.66 int unpack_wds_SetAutoconnect (uint8_t * *pResp*, uint16_t *respLen*)

Auto connect data session parameters unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.67 int unpack_wds_SetDefaultProfile (uint8_t * *pResp*, uint16_t *respLen*)

set default profile unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.68 int unpack_wds_SetDefaultProfileNum (uint8_t * *pResp*, uint16_t *respLen*)

set default profile number unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.69 int unpack_wds_SetMobileIP (uint8_t * *pResp*, uint16_t *respLen*)

Sets the current mobile IP setting unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.70 int unpack_wds_SetMobileIPParameters (uint8_t * *pResp*, uint16_t *respLen*)

Sets the specified mobile IP parameters unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.71 int unpack_wds_SetMobileIPProfile (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SetMobileIPProfile_t * *pOutput*)

set mobile ip profile unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.72 int unpack_wds_SetMuxID (uint8_t * *pResp*, uint16_t *respLen*)

Set MUX ID unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.73 int unpack_wds_SLQSCreateProfile (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SLQSCreateProfile_t * *pOutput*)

Create Profile unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response structure to fill

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.74 int unpack_wds_SLQSDeleteProfile (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SLQSDeleteProfile_t * *pOutput*)

delete stored profile unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.75 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

9.62.4.76 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

9.62.4.77 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

9.62.4.78 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

9.62.4.79 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

9.62.4.80 int unpack_wds_SLQSGetProfileSettings (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SLQSGetProfileSettings_t * *pOutput*)

get session state unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response structure to fill

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.81 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

9.62.4.82 int unpack_wds_SLQSModifyProfile (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SLQSModifyProfile_t * *pOutput*)

Modify Profile unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response structure to fill

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.83 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

9.62.4.84 int unpack_wds_SLQSSet3GPPConfigItem (uint8_t * *pResp*, uint16_t *respLen*)

set 3GPP config items unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

9.62.4.85 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

9.62.4.86 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

9.62.4.87 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

9.62.4.88 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

9.62.4.89 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

9.62.4.90 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

9.62.4.91 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

9.62.4.92 int unpack_wds_SLQSSetLoopback (uint8_t * *pResp*, uint16_t *respLen*)

Enable/disable Data Loopback Mode and set the value of loopback multiplier unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.93 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

9.62.4.94 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

9.62.4.95 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

9.62.4.96 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

9.62.4.97 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

9.62.4.98 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, 1270
_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
pFwPromptUpdate, 85
pFwPromptdownload, 85
pNotifStore, 85
pPeroidInfo, 85
pPollingTimer, 85
_SLQSAVMSSettings, 85
pAPNInfo, 85
pAutoConnect, 85
pAutoReboot, 85
pConnectionRetryTimers, 85
pFwAutoSDM, 85
pFwPromptUpdate, 85
pFwPromptdownload, 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, 1262
AAAState
unpack_wds_GetMobileIPProfile_t, 1262
ABSOLUTE_VALIDITY
qaGobiApiSms.h, 1824
ACT_CODE_MAX_SIZE
dms.h, 1442
ALS
getAllCallInformation, 256
voice_getAllCallInformation, 1323
AMSSString
unpack_dms_GetFirmwareRevision_t, 1047
unpack_dms_GetFirmwareRevisions_t, 1048
APNName
unpack_wds_SLQSGetRuntimeSettings_t, 1278
AVMS (SWIAVMS), 59
absoluteValidity
cdmaMsgDecodingParams, 166
cdmaMsgDecodingParamsExt, 169
accelAcceptReady
qaGobiApiCbk.h, 1595
accelAcceptReady_s, 113
batchPerSec, 113
injectEnable, 113
samplesPerBatch, 113
accelTemp
pack_loc_SLQSLOCInjectSensorData_t, 624
accelTempAcceptReady
qaGobiApiCbk.h, 1596
accelTempAcceptReady_s, 113
batchPerSec, 114
injectEnable, 114
samplesPerBatch, 114
acceleroData
pack_loc_SLQSLOCInjectSensorData_t, 624
acceleroTimeSrc
pack_loc_SLQSLOCInjectSensorData_t, 624
AccessMac
nas_protocolSubtypeElement, 495
protocolSubtypeElement, 773
accolc
pack_nas_SetACCOLC_t, 629
accuracy
pack_pds_SetPDSDefaults_t, 656
ackIndicator
SMSTransferRouteMTMessage, 926
sMSTransferRouteMTMessageInfo, 926
acqOrdeLen
acqOrderPref, 115
nas_acqOrderPref, 420
acqOrderPref, 114
acqOrdeLen, 115
pAcqOrder, 115

acroamsetting
pack_wds_SetAutoconnect_t, 717
slqsautoconnect, 882
acsetting
pack_wds_SetAutoconnect_t, 717
slqsautoconnect, 882
actCode
pack_dms_ActivateAutomatic_t, 597
ActPilotPNElement, 115
ActSetPilotPN, 115
ActSetPilotPNStrength, 115
ActSetCnt
nas_NetworkStat1x, 481
NetworkStat1x, 581
ActSetPilotPN
ActPilotPNElement, 115
nas_ActPilotPNElement, 421
ActSetPilotPNStrength
ActPilotPNElement, 115
nas_ActPilotPNElement, 421
action
pack_nas_InitiateDomainAttach_t, 629
slqsautoconnect, 882
ssdatasession_params, 931
ActivateAutomatic
qaGobiApiDms.h, 1690
activated_ind
_qaQmi3GPP2BroadcastCfgInfo, 71
_qaQmi3GPPBroadcastCfgInfo, 71
sms_qaQmi3GPP2BroadcastCfgInfo, 905
sms_qaQmi3GPPBroadcastCfgInfo, 905
activationState
pack_dms_SetActivationStatusCallback_t, 599
activationStatus
dms_ActivationStatusTlv, 228
ActivationStatusTlv
unpack_dms_SetEventReport_ind_t, 1061
activeBandClass
nas_RFInfoTlv, 501
RFBandInfoElements, 824
activeChannel
nas_RFInfoTlv, 501
RFBandInfoElements, 824
activeInd
messageWaitingInfoContent, 415
sms_messageWaitingInfoContent, 903
ActiveStatus
CLIPResp, 182
CLIRResp, 183
CNAPResp, 185
COLPResp, 186
COLRResp, 186
voice_CLIPResp, 1315
voice_CLIRResp, 1316
voice_CNAPResp, 1316
voice_COLPResp, 1317
voice_COLRResp, 1318
ActiveTechPref
unpack_nas_GetNetworkPreference_t, 1111
AddCDMASysInfo, 115
geoSysIdx, 116
regPrd, 116
AddSysInfo, 116
cellBroadcastCap, 116
geoSysIdx, 116
addr
IPv4Addr, 331
IPv6Addr, 332
unpack_qos_IPv4Addr_t, 1174
unpack_qos_IPv6Addr_t, 1174
address
IPv4Info, 332
IPv6Info, 335
loc_IPv4Info, 372
loc_IPv6Info, 372
loc_urlAddr, 379
unpack_wds_GetMobileIPProfile_t, 1262
urlAddr, 1289
addressSize
unpack_sms_GetSMSCAddress_t, 1196
aid
uim_refreshevent, 993
uim_sessionInformation, 995
uim_UIMSessionInformation, 999
UIMRefreshEvent, 1013
UIMSessionInformation, 1016
aidLength
appStats, 124
appStatus, 127
uim_appStatus, 978
uim_refreshevent, 993
uim_sessionInformation, 995
uim_UIMSessionInformation, 999
UIMRefreshEvent, 1013
UIMSessionInformation, 1016
aidVal
appStats, 124
appStatus, 127
uim_appStatus, 979
aidingIndicatorMask
loc_sensorDataUsage, 377
sensorDataUsage_s, 843
t_sensor, 955
airTimer, 116
airTimerValue, 117
namID, 117
airTimerValue
airTimer, 117
voice_airTimer, 1297
alertPitch
signalInfo, 877
alertingPattern
arrAlertingPattern, 128
voice_arrAlertingPattern, 1300
AlertingType
arrAlertingType, 129

voice_arrAlertingType, 1301
alertmsg
omaDmConfigTlv, 587
omaDmConfigTlvExt, 590
unpack_omaDmConfigTlv_t, 1158
alertmsglength
omaDmConfigTlv, 587
omaDmConfigTlvExt, 590
unpack_omaDmConfigTlv_t, 1158
allCallsAlphaIDInfo, 117
AlphaIDInfo, 117
callID, 117
allCallsAlphaIDInfoArr
arrAlphaID, 129
voice_arrAlphaID, 1301
allCallsDiagInfo, 117
callID, 118
DiagInfo, 118
AllCallsUUSInfo
arrUUSInfo, 135
voice_arrUUSInfo, 1307
allCallsUUSInfo, 118
callID, 118
uusInfo, 118
alphaDcs
alphaIDInfo, 119
voice_alphaIDInfo, 1299
AlphaID
CatAlPhaIdentifierTlv, 158
AlphaIDInfo
allCallsAlphaIDInfo, 117
voice_allCallsAlphaIDInfo, 1297
alphaIDInfo, 118
alphaDcs, 119
alphaLen, 119
alphaText, 119
alphaIDLen
SMSAsyncRawSend_s, 912
AlphaIDLength
CatAlPhaIdentifierTlv, 158
alphaLen
alphaIDInfo, 119
voice_alphaIDInfo, 1299
alphaText
alphaIDInfo, 119
voice_alphaIDInfo, 1299
alphabet
wcdmaMsgEncodingParams, 1390
altSrcInfo_t, 120
coverage, 121
linkage, 121
source, 121
Altitude
GPSStateInfo, 292
unpack_pds_SLQSGetGPSStateInfo_t, 1172
altitudeAssumed
unpack_loc_GnssSvInfo_Ind_t, 1094
altitudeSrcInfo, 119
coverage, 120
linkage, 120
pack_loc_SLQSLOCInjectPosition_t, 622
source, 120
altitudeWrtEllipsoid
pack_loc_SLQSLOCInjectPosition_t, 622
altitudeWrtMeanSeaLevel
pack_loc_SLQSLOCInjectPosition_t, 622
ambr_dl
sApnExtraParams, 836
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 1179
ambr_dl_ext
sApnExtraParams, 836
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 1180
ambr_dl_ext2
sApnExtraParams, 836
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 1180
ambr_ul
sApnExtraParams, 836
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 1180
ambr_ul_ext
sApnExtraParams, 836
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 1180
ambr_ul_ext2
sApnExtraParams, 836
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 1180
amssSize
unpack_dms_GetFirmwareRevision_t, 1047
unpack_dms_GetFirmwareRevisions_t, 1048
AnswerUSSD
qaGobiApiVoice.h, 1933
apdoxypages.c, 1431
apnId
pack_qos_SLQSQosSwiReadApnExtraParams_t, 665
pack_qos_SLQSQosSwiReadDataStats_t, 666
sApnExtraParams, 836
sQosStat, 928
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 1180
unpack_qos_SLQSQosSwiReadDataStats_t, 1181
apnName
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 1287
apnname
unpack_wds_GetDefaultProfile_t, 1258
apnsize
unpack_wds_GetDefaultProfile_t, 1258
appNameLength
loc_LocApplicationInfo, 373
LocApplicationInfo, 380
appProviderLength

loc_LocApplicationInfo, 373
LocApplicationInfo, 380
appState
appStats, 124
appStatus, 127
uim_appStatus, 979
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, 879
slotInfo, 881
uim_slotInfo, 998
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, 979
appVersionLength
loc_LocApplicationInfo, 373
LocApplicationInfo, 380
appVersionValid
loc_LocApplicationInfo, 373
LocApplicationInfo, 380
Application
unpack_nas_GetCDMANetworkParameters_t, 1106
appversion_str
slqsfwinfo_s, 884
unpack_dms_GetFirmwareInfo_t, 1046
arfcn
GERANInfo, 254
gsmCellInfo, 294
nas_GERANInfo, 445
nas_gsmCellInfo, 446
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, 1370
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, 820
uim_refreshevent, 993
uim_registerRefresh, 994
UIMRefreshEvent, 1013
AtCmdPort
DcsUsbPortNames, 220
Audio Service (AUDIO), 53

auth
unpack_wds_GetDefaultProfile_t, 1258
authData
pack_uim_SLQSUIMAuthenticate_t, 685
UIMAuthenticateReq, 1001
AuthProt
nas_protocolSubtypeElement, 496
protocolSubtypeElement, 773
authenticateResult, 135
content, 136
contentLen, 136
Authentication
unpack_wds_SLQSGetRuntimeSettings_t, 1278
authentication
pack_wds_SetDefaultProfile_t, 718
authenticationData, 136
context, 136
data, 137
dataLen, 137
Autosdm
unpack_swioma_SLQSOMADMGetSettings_t, 1211
avgPeriod
LTESigRptCfg, 406
LTESigRptConfig, 407
nas_LTESigRptCfg, 470
nas_LTESigRptConfig, 471
azimuth
loc_satelliteInfo, 376
satelliteInfo, 838

bAPNLength
_QmiSwiAvmsSetSettingsAPNInfo, 80
_SLQSAVMSSettingsAPNInfo, 86
bAltitudeAssumed
gnssSvInfoNotification, 287
bAuto
pack_pds_SetPortAutomaticTracking_t, 657
pack_pds_SetServiceAutomaticTracking_t, 657
bBinaryType
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 73
bEnable
pack_dms_UIMSetPINProtection_t, 610
pack_nas_SLQSSetSignalStrengthsCallback_t, 647
bEnabled
pack_pds_SetXTRAAutomaticDownload_t, 658
bForceDownload
pack_fms_SetImagesPreference_t, 613
bICCID
slot_t, 878
UIMSlotStatus, 1018
bICCIDLength
slot_t, 878
UIMSlotStatus, 1018
bLogicalSlot
pack_uim_SLQSUIMSwitchSlot_t, 691
slot_t, 878
UIMSlotStatus, 1018
UIMSwitchSlotReq, 1020
bManSize
_SLQSSwiGetHostDevInfoParams, 96
_SLQSSwiSetHostDevInfoParams, 98
bModelSize
_SLQSSwiGetHostDevInfoParams, 96
_SLQSSwiSetHostDevInfoParams, 98
bNameSize
_SLQSSwiGetOSInfoParams, 96
_SLQSSwiSetOSInfoParams, 99
bNotification
_QmiSwiAvmsEventReportNotificationResp, 77
bNumberOfPhySlots
UIMSlotStatusChangeInfo, 1019
unpack_uim_SetUimSlotStatusChangeCallback_ind_t, 1215
BOOL
SwiDataTypes.h, 2035
BPTlv
NASQmiCbkNasSystemSelPrefInd, 567
bPWDLength
_QmiSwiAvmsSetSettingsAPNInfo, 80
_SLQSAVMSSettingsAPNInfo, 86
bPackageID
_QmiSwiAvmsEventReportPackageIDResp, 78
bPlasmaIDSize
_SLQSSwiGetHostDevInfoParams, 96
_SLQSSwiSetHostDevInfoParams, 98
bRegStatus
_QmiSwiAvmsEventReportRegStatusResp, 78
bResetStatistics
rmTrasnferStaticsReq, 825
swiRMTrasnferStaticsReq, 953
bSWVerSize
_SLQSSwiGetHostDevInfoParams, 96
_SLQSSwiSetHostDevInfoParams, 98
bSerity
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 73
bState
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 73
_QmiSwiAvmsEventReportConfigResp, 74
bType
_QmiSwiAvmsEventReportDataSessionStatusResp, 75
_QmiSwiAvmsEventReportSessionTypeResp, 79
BUILD_ID_LEN
qaGobiApiFms.h, 1715
BUILD_ID_MAX_LEN
qaGobiApiFms.h, 1715
bUnameLength
_QmiSwiAvmsSetSettingsAPNInfo, 80
_SLQSAVMSSettingsAPNInfo, 86
bUserInputRequest
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 73

_QmiSwiAvmsEventReportConfigResp, 74
_QmiSwiAvmsEventReportConnectionRequest-Resp, 75
bVersionSize
_SLQSSwiGetOSInfoParams, 96
_SLQSSwiSetOSInfoParams, 99
BYT_STAT_STAT_MASK
wds.h, 2098
BYTE
SwiDataTypes.h, 2035
band
LTEInfo, 398
nas_LTEInfo, 462
band1900
gsmCellInfo, 294
nas_gsmCellInfo, 447
band_pref
NASBandPreferenceTlv, 535
BandCapability
unpack_dms_GetBandCapability_t, 1035
bandCapability
BandCapabilityResp, 140
unpack_dms_SLQSGetBandCapability_t, 1068
BandCapabilityResp, 137
bandCapability, 140
pLteBandCapability, 140
pTdsBandCapability, 140
bandPref
pack_nas_SLQSSetBandPreference_t, 646
bandwidth
LTEInfo, 398
nas_LTEInfo, 462
baseId
CDMAInfo, 164
CDMASysInfo, 176
nas_CDMAInfo, 425
nas_CDMASysInfo, 429
baseLat
CDMAInfo, 164
CDMASysInfo, 176
nas_CDMAInfo, 425
nas_CDMASysInfo, 429
baseLong
CDMAInfo, 164
CDMASysInfo, 177
nas_CDMAInfo, 426
nas_CDMASysInfo, 429
BasestationID
qaQmiServingSystemParam, 778
unpack_nas_SLQSGetServingSystem_t, 1129
BasestationLatitude
qaQmiServingSystemParam, 778
unpack_nas_SLQSGetServingSystem_t, 1129
BasestationLongitude
qaQmiServingSystemParam, 778
unpack_nas_SLQSGetServingSystem_t, 1129
batchPerSec
accelAcceptReady_s, 113
accelTempAcceptReady_s, 114
gyroAcceptReady_s, 299
gyroTempAcceptReady_s, 300
BdsSV, 140
id, 140
mask, 140
BdsSVInfo, 140
len, 141
pSV, 141
BearerID
unpack_qos_QosFlowInfo_t, 1177
bearerID
_packetSrvStatus, 70
unpack_wds_SLQSSetPacketSrvStatusCallback_-t, 1281
bearerId
sQosFlowStat, 927
unpack_QosFlowStat_t, 1193
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1287
BinaryUpdateSessionInfo
_SwiAvmsEventInd, 99
bootSize
unpack_dms_GetFirmwareRevisions_t, 1048
BootString
unpack_dms_GetFirmwareRevisions_t, 1048
bootversion_str
slqsfwinfo_s, 884
unpack_dms_GetFirmwareInfo_t, 1046
Broadcast
unpack_nas_GetCDMANetworkParameters_t, 1106
broadcastActivate
pack_sms_SLQSSetSmsBroadcastActivation_t, 675
BroadcastConfig, 141
fromServiceId, 142
selected, 142
toServiceId, 142
broadcastConfig
_qaQmi3GPPBroadcastCfgInfo, 71
sms_qaQmi3GPPBroadcastCfgInfo, 905
bsInfoValid
CDMASysInfo, 177
nas_CDMASysInfo, 429
bsPRev
CDMASysInfo, 177
nas_CDMASysInfo, 429
bsPRevValid
CDMASysInfo, 177
nas_CDMASysInfo, 429
bsic
GERANInfo, 254
nas_GERANInfo, 445
bsicId
gsmCellInfo, 294
nas_gsmCellInfo, 447
bucketSz

tokenBucket, 969
unpack_qos_tokenBucket_t, 1192
buildID
CurrImageInfo, 203
FMSImageIdElement, 250
image_info_t, 311
ImageIdElement, 312
buildIDLen
CurrImageInfo, 203
image_info_t, 311
buildIDLength
FMSImageIdElement, 250
ImageIdElement, 313
buildId
FMSImageElement, 249
ImageElement, 312
buildIdLength
FMSImageElement, 250
ImageElement, 312
BurstDTMFInfo
pack_voice_SLQSVoiceBurstDTMF_t, 697
voiceBurstDTMFInfo, 1335
burstDTMFInfo, 142
digitCnt, 142
pCallID, 142
pDigitBuff, 142
ByteLoopbackMode
unpack_wds_SLQSSGetLoopback_t, 1285
WDSGetLoopbackData, 1421
ByteLoopbackMultiplier
unpack_wds_SLQSSGetLoopback_t, 1285
WDSGetLoopbackData, 1421
ByteTotalsElmntsV4
WdsByteTotals, 1412
ByteTotalsElmntsV6
WdsByteTotals, 1412

CATEventDataType, 159
eventMask, 159
pErrorMask, 159
CATSendEnvelopeCommand
qaGobiApiCat.h, 1584
CATSendTerminalResponse
qaGobiApiCat.h, 1585
CBK_DISABLE_EVENT
qaGobiApiCbk.h, 1594
CBK_ENABLE_EVENT
qaGobiApiCbk.h, 1594
CBK_NOCHANGE
qaGobiApiCbk.h, 1594
CCETlv
QmiCbkCatEventStatusReportInd, 779
CDMA_P_Rev
qaQmiServingSystemParam, 778
unpack_nas_SLQSGetServingSystem_t, 1129
CDMABroadcastConfig, 161
_qaQmi3GPP2BroadcastCfgInfo, 71
language, 162
selected, 162
serviceCategory, 162
sms_qaQmi3GPP2BroadcastCfgInfo, 905
CDMAChannel, 162
priChA, 162
priChB, 162
secChA, 162
secChB, 163
CDMAECIOThresh, 163
CDMAECIOThreshListLen, 163
pCDMAECIOThreshList, 163
CDMAECIOThreshListLen
CDMAECIOThresh, 163
nas_CDMAECIOThresh, 425
CDMAInfo, 163
baseId, 164
baseLat, 164
baseLong, 164
nid, 164
refpn, 164
sid, 164
CDMARSSIThresh, 172
CDMARSSIThreshListLen, 172
pCDMARSSIThreshList, 172
CDMARSSIThreshListLen
CDMARSSIThresh, 172
nas_CDMARSSIThresh, 426
CDMASSInfo, 172
ecio, 173
rssi, 173
unpack_nas_SLQSNasGetSigInfo_t, 1144
CDMASysInfo, 174
baseId, 176
baseLat, 176
baseLong, 177
bsInfoValid, 177
bsPRev, 177
bsPRevValid, 177
ccsSupported, 177
ccsSupportedValid, 177
cdmaSysIdValid, 177
isSysPrlMatch, 177
isSysPrlMatchValid, 177
MCC, 177
MNC, 177
networkID, 177
networkIdValid, 177
pRevInUse, 177
pRevInUseValid, 177
packetZone, 177
packetZoneValid, 177
sysInfoCDMA, 177
systemID, 177
CDMASysInfoExt, 177
imsi_11_12, 178
MCC, 178
CDMASystemInfoExt
qaQmiServingSystemParam, 778
unpack_nas_SLQSGetServingSystem_t, 1129

CHAR
SwiDataTypes.h, 2035
CK_MAX_SIZE
dms.h, 1442
CLIPResp, 182
ActiveStatus, 182
ProvisionStatus, 182
CLIRResp, 182
ActiveStatus, 183
ProvisionStatus, 183
CNAPResp, 184
ActiveStatus, 185
ProvisionStatus, 185
COLPResp, 185
ActiveStatus, 186
ProvisionStatus, 186
COLRResp, 186
ActiveStatus, 186
ProvisionStatus, 186
CONFIG_LEN
qaGobiApiSms.h, 1824
CQIValueCW0
LteCQIParm, 394
unpack_nas_SLQSSwiGetLteCQI_t, 1153
CQIValueCW1
LteCQIParm, 394
unpack_nas_SLQSSwiGetLteCQI_t, 1153
CSDomain
unpack_nas_GetServingNetwork_t, 1114
CSGID, 196
id, 197
mcc, 197
mnc, 197
mncPcsDigits, 197
rat, 197
CUGIndex
CUGInfo, 198
voice_CUGInfo, 1320
CUGInfo, 197
CUGIndex, 198
SuppOA, 198
SuppPrefCUG, 198
CallBacK registration (CBK), 41
CallBarStatus
qaQmiServingSystemParam, 778
unpack_nas_SLQSGetServingSystem_t, 1129
callBarStatus, 143
csBarStatus, 144
psBarStatus, 144
CallBarringSysInfo, 142
csBarStatus, 143
psBarStatus, 143
callDuration
unpack_wds_GetSessionDuration_t, 1266
CallEndReason
DUNCallInfoInd, 236
callEndReason
arrCallEndReason, 131
unpack_wds_SLQSGetDUNCallInfo_t, 1275
voice_arrCallEndReason, 1302
CallFWExtInfo
getCallFWExtInfo, 263
voice_getCallFWExtInfo, 1324
callFWExtInfo, 148
noReplyTimer, 150
numLen, 150
numPlan, 150
numType, 150
number, 150
PI, 150
SI, 150
SvcClass, 150
SvcStatus, 150
CallFWInfo
getCallFWInfo, 264
voice_getCallFWInfo, 1325
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, 235
getAllCallRmtPtyName, 257
getAllCallRmtPtyNum, 257
pack_voice_SLQSVoiceGetCallInfo_t, 701
pack_voice_SLQSVoiceStopContDTMF_t, 712
peerNumberInfo, 745
unpack_voice_SLQSVoiceStopContDTMF_t, 1253
voice_allCallsAlphaIDInfo, 1297
voice_allCallsDiagInfo, 1298
voice_allCallsUUSInfo, 1298
voice_arrAlertingPattern, 1300
voice_arrAlertingType, 1301
voice_arrCallEndReason, 1302
voice_arrSvcOption, 1306
voice_callInfo, 1313
voice_getAllCallRmtPtyName, 1323
voice_getAllCallRmtPtyNum, 1324
voice_peerNumberInfo, 1327
voiceCallInfoReq, 1335
voiceInfoRec, 1364
voiceOTASPStatusInfo, 1367
voicePrivacyInfo, 1368

voiceStopContDTMFinfo, 1380
voiceSUPSNotification, 1384
callInfo, 151
callID, 152
callState, 152
callType, 153
direction, 153
mode, 153
callNumber
pack_voice_SLQSVoiceDialCall_t, 698
voiceCallRequestParams, 1340
callState
callInfo, 152
voice_callInfo, 1313
callType
callInfo, 153
voice_callInfo, 1314
calledPartyInfo, 144
numLen, 145
numPlan, 145
numType, 145
number, 145
PI, 146
SI, 146
CalledPartyNum
arrCalledPartyNum, 130
voice_arrCalledPartyNum, 1302
calledPartySubAdd, 146
extBit, 146
oddEvenInd, 146
subAddr, 146
subAddrLen, 146
subAddrType, 146
callerID
callerIDInfo, 147
connectNumInfo, 191
voice_connectNumInfo, 1319
callerIDInfo, 147
callerID, 147
callerIDLen, 147
PI, 147
callerIDLen
callerIDInfo, 147
connectNumInfo, 191
voice_connectNumInfo, 1319
callerName
remotePartyName, 821
voice_remotePartyName, 1329
Callinfo
getAllCallInformation, 256
voice_getAllCallInformation, 1323
callingPartyInfo, 153
numLen, 154
numPlan, 154
numType, 154
number, 154
PI, 154
SI, 154
CancelUSSD
qaGobiApiVoice.h, 1933
Card Application Toolkit (CAT), 44
cardResult, 154
sw1, 155
sw2, 155
cardState
slotInf, 879
slotInfo, 881
uim_slotInfo, 998
cardStatus, 155
index1xPri, 156
index1xSec, 156
indexGwPri, 156
indexGwSec, 156
numSlot, 156
SlotInfo, 156
Carrier
fwinfo_s, 253
carrier
CurrentImgList, 201
unpack_dms_SLQSSwiGetFirmwareCurr_t, 1071
carrier_str
slqsfwinfo_s, 884
unpack_dms_GetFirmwareInfo_t, 1046
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, 200
CatCommonEventTlv, 158
CatEvent, 158
EventID, 158
EventLength, 158
TlvPresent, 158
CatEndPS
currentCatEvent, 200
CatEndProactiveSessionTlv, 158
EndProactiveSession, 159
CatEvIDData
currentCatEvent, 200
CatEvent
CatCommonEventTlv, 158
CatEventIDDataTlv, 159
Data, 159
DataLength, 159
ReferenceID, 159
CatEventListTlv, 159
SetupEventList, 160

CatEventLst
currentCatEvent, 200
CatRefresh
currentCatEvent, 200
CatRefreshTlv, 160
RefreshMode, 160
RefreshStage, 160
causeCode
SMSAsyncRawSend_s, 912
ccSUPSType, 160
reason, 161
svcType, 161
ccsSupported
CDMASysInfo, 177
nas_CDMASysInfo, 429
ccsSupportedValid
CDMASysInfo, 177
nas_CDMASysInfo, 429
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
cdmaMsgDecodingParamsExt, 167
absoluteValidity, 169
mcTimeStamp, 169
messageLength, 169
pAlertPriority, 170
pCallbkAddr, 170
pCallbkAddrLength, 170
pDisplayMode, 170
pIsUDHPresent, 170
pLanguage, 170
pMessage, 170
pMessageID, 170
pPartNum, 170
pPriority, 170
pPrivacy, 170
pReadAcknowledgementReq, 170
pReferenceNum, 170
pRelativeValidity, 170
pSenderAddr, 170
pSenderAddrLength, 170
pTextMsg, 170
pTextMsgLength, 170
pTotalNum, 170
pUserAcknowledgementReq, 170
cdmaMsgEncodingParams, 170
messageId, 172
pCallbackAddr, 172
pDestAddr, 172
pEncodingAlphabet, 172
pMessage, 172
pMessageSize, 172
pPriority, 172
pRelValidity, 172
pTextMsg, 172
textMsgLength, 172
cdmaSSInfo, 173
ecio, 173
rssi, 173
cdmaSysIdValid
CDMASysInfo, 177
nas_CDMASysInfo, 429
cell_resel_priority
infoInterFreq, 330
nas_infoInterFreq, 457
cellBroadcastCap
AddSysInfo, 116
nas_AddSysInfo, 422
CellDb, 178
mask, 178
CellID
qaQmiServingSystemParam, 778
unpack_nas_SLQSGetServingSystem_t, 1129
cellID
GERANInfo, 255
nas_GERANInfo, 445
nas_UMTSInfo, 525
UMTSInfo, 1023
cellId
GSMSysInfo, 298
LTESysInfo, 412
nas_GSMSysInfo, 450
nas_LTESysInfo, 475
nas_WCDMASysInfo, 533
WCDMASysInfo, 1393
cellIdValid
gsmCellInfo, 294
GSMSysInfo, 298
LTESysInfo, 412
nas_gsmCellInfo, 447
nas_GSMSysInfo, 450
nas_LTESysInfo, 475
nas_WCDMASysInfo, 534
WCDMASysInfo, 1394
cellInterFreqParams
infoInterFreq, 330
nas_infoInterFreq, 457
cellIsTDD
nas_umtsLTENbrCell, 527

umtsLTENbrCell, 1025
CellParams
LTEInfoIntrafreq, 400
nas_LTEInfoIntrafreq, 465
cellParams, 178
pci, 179
rsrp, 179
rsrq, 179
rssi, 179
srxlev, 179
cellReselPriority
lteGsmCellInfo, 396
LTEInfoIntrafreq, 400
lteWcdmaCellInfo, 414
nas_lteGsmCellInfo, 460
nas_LTEInfoIntrafreq, 465
nas_lteWcdmaCellInfo, 476
cells_len
infoInterFreq, 330
lteGsmCellInfo, 396
nas_infoInterFreq, 457
nas_lteGsmCellInfo, 460
cellsLen
LTEInfoIntrafreq, 400
lteWcdmaCellInfo, 414
nas_LTEInfoIntrafreq, 465
nas_lteWcdmaCellInfo, 476
chaddr
wds_DHCPv4HWConfig, 1400
WdsDHCPv4HWConfig, 1417
wdsDhcpv4HwConfig, 1417
chaddrLen
wds_DHCPv4HWConfig, 1400
WdsDHCPv4HWConfig, 1417
wdsDhcpv4HwConfig, 1417
changePIN
pack_uim_ChangePin_t, 682
UIMChangePinReq, 1002
changeUIMPIN, 179
oldPINLen, 180
oldPINVal, 180
pinID, 180
pinLen, 180
pinVal, 180
ChannelRate, 181
CurrChanRxRate, 181
CurrChanTxRate, 181
DUNCallInfoInd, 236
MaxChanRxRate, 181
MaxChanTxRate, 181
channelRate, 180
CurrChanRxRate, 181
CurrChanTxRate, 181
unpack_wds_SLQSGetDUNCallInfo_t, 1275
Chipset
DeviceConfigDetail, 224
nas_DeviceConfigDetail, 440
ckLen
depersonalizationInformation, 222
uim_depersonalizationInformation, 984
ckVal
depersonalizationInformation, 222
uim_depersonalizationInformation, 984
clear
pack_dms_SLQSSwiGetCrashInfo_t, 605
ClkInfo, 183
mask, 184
codingScheme
nas_PLMNNetworkNameData, 494
PLMNNetworkNameData, 756
remotePartyName, 821
voice_remotePartyName, 1330
CommInfo, 186
imsRegState, 188
modemMode, 188
psState, 188
systemMode, 188
temperature, 188
common.h
eAUDIO, 1434
eCAT, 1434
eCTL, 1434
eDMS, 1434
eIMS, 1434
eIMSA, 1434
eIND, 1435
eLOC, 1434
eLOG_DEBUG, 1434
eLOG_FATAL, 1434
eLOG_INFO, 1434
eLOG_WARN, 1434
eNAS, 1434
ePDS, 1434
eQOS, 1434
eREQ, 1435
eRMS, 1434
eRSP, 1435
eSAR, 1434
eSMS, 1434
eSWIAUDIO, 1434
eSWILOC, 1434
eSWIM2MCMD, 1434
eSWIOMA, 1434
eSWIOMAEXT, 1434
eTIMEOUT_10_S, 1434
eTIMEOUT_20_S, 1434
eTIMEOUT_2_S, 1434
eTIMEOUT_300_S, 1434
eTIMEOUT_30_S, 1434
eTIMEOUT_5_S, 1434
eTIMEOUT_60_S, 1434
eTIMEOUT_8_S, 1434
eTIMEOUT_DEFAULT, 1434
eTMD, 1434
eUIM, 1434
eVOICE, 1434

eWDS, 1434
common.h, 1431
eLOG_LEVEL, 1433
eQMI_SVC, 1434
eTimeout, 1434
fill_pack_ctx, 1435
fill_sdu_hdr, 1435
get_version, 1435
glog, 1436
gloglvl, 1436
helper_get_resp_ctx, 1435
helper_get_xid, 1435
helper_set_log_func, 1435
helper_set_log_lvl, 1435
liteqmi_GetVersion, 1435
liteqmi_helper_decode7bitAsciiEncString, 1435
liteqmi_log, 1436
logger, 1433
MINREQBKLEN, 1433
MSGID_AND_LEN, 1433
MSGID_DONT_CARE, 1433
msgtype, 1434
SDU_HDR_LEN, 1433
UNUSEDPARAM, 1433
unpack_result_code_only, 1436
commonInfo
swiModemStatusResp, 940
unpack_nas_SLQSNasSwiModemStatus_t, 1148
ConcSvcInfo
unpack_nas_SLQSGetServingSystem_t, 1129
concSvcInfo
qaQmiServingSystemParam, 778
confidence
pack_loc_SLQSLOCSetCradleMountConfig_t, 626
Config
_SwiAvmsEventInd, 99
conn_status
unpack_wds_SLQSSetPacketSrvStatusCallback_t, 1281
ConnRateElmntsV4
WdsConnectionRate, 1414
ConnRateElmntsV6
WdsConnectionRate, 1414
connStatus
_packetSrvStatus, 70
connectNumInfo, 190
callerID, 191
callerIDLen, 191
numPlan, 191
numPresInd, 191
numType, 191
screeningInd, 191
ConnectedPartyNum
arrConnectPartyNum, 132
voice_arrConnectPartyNum, 1303
ConnectionRequest
_SwiAvmsEventInd, 99
ConnectionStatus, 188
MDMCallDuration, 189
MDMConnStatus, 189
connectionStatus, 189
MDMCallDuration, 189
MDMConnStatus, 190
unpack_wds_GetSessionState_t, 1267
unpack_wds_SLQSGetDUNCallInfo_t, 1275
connetionState
imsaPdpStatusInfo, 316
content
authenticateResult, 136
readResult, 817
uim_authenticateResult, 979
uim_readResult, 991
contentLen
authenticateResult, 136
readResult, 817
uim_authenticateResult, 979
uim_readResult, 991
context
authenticationData, 136
uim_authenticationData, 980
contextId
pack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 737
swiPDPRuntimeSettingsReq, 942
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 1287
contextType
pack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 737
swiPDPRuntimeSettingsReq, 942
ControlMac
nas_protocolSubtypeElement, 496
protocolSubtypeElement, 773
Count1
RankIndicatorInd, 816
Count2
RankIndicatorInd, 816
countryInitials
nas_PLMNNetworkNameData, 494
PLMNNetworkNameData, 756
coverage
altitudeSrcInfo, 120
altSrcInfo_t, 121
cpich_ecno
nas_wcdmaCellInfo, 529
wcdmaCellInfo, 1384
cpich_rscp
nas_wcdmaCellInfo, 529
wcdmaCellInfo, 1384
cradleMountConfigStatus
QmiCbkLocCradleMountInd, 786
crashAction
pack_dms_SetCrashAction_t, 599
crashData
CrashInfo, 192
crashInformation, 194

crashId
CrashInfo, 192
crashInformation, 194
CrashInfo, 191
crashData, 192
crashId, 192
crashStrLen, 192
gcDumpStrLen, 192
numCrashes, 192
pCrashString, 192
pGCDumpString, 192
crashInfo
crashInfoParams, 193
crashInfoParam
unpack_dms_SLQSSwiGetCrashInfo_t, 1069
CrashInfoParams, 193
pCrashInfo, 193
pDevCrashStatus, 193
crashInfoParams, 192
crashInfo, 193
crashStatus, 193
crashInformation, 193
crashData, 194
crashId, 194
crashString, 194
crashStrlen, 194
gcdumpString, 194
gcdumpStrlen, 194
numCrashes, 194
crashStatus
crashInfoParams, 193
crashStrLen
CrashInfo, 192
crashString
crashInformation, 194
crashStrlen
crashInformation, 194
CreateProfileIn, 194
curProfile, 195
pProfileID, 195
pProfileType, 195
CreateProfileOut, 195
pExtErrorCode, 196
pProfileIndex, 196
pProfileType, 196
csAttachState
nas_servSystem, 510
NASServingSystemInfo, 569
ServingSystemInfo, 846
servSystem, 848
csBarStatus
CallBarringSysInfo, 143
callBarStatus, 144
nas_CallBarringSysInfo, 422
nas_callBarStatus, 423
CsgId, 196
csgId, 196
TlvPresent, 196
csgId
CsgId, 196
cur_carr_name
slqsfwinfo_s, 884
unpack_dms_GetFirmwareInfo_t, 1046
cur_carr_rev
slqsfwinfo_s, 884
unpack_dms_GetFirmwareInfo_t, 1046
curAMRConfig, 198
gsmAmrStat, 199
wcdmaAmrStat, 199
curDataBearerTechnology
unpack_wds_SLQSGetDataBearerTechnology_t, 1272
curProfile
_GetProfileSettingOut, 65
CreateProfileIn, 195
ModifyProfileIn, 418
pack_wds_SLQSModifyProfile_t, 729
UnPackGetProfileSettingOut, 1288
CurrChanRxRate
ChannelRate, 181
channelRate, 181
dunchannelRate, 237
CurrChanTxRate
ChannelRate, 181
channelRate, 181
dunchannelRate, 237
currDBTechAvail
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1283
CurrDataSysStat, 199
pCurrNetworkInfo, 200
pNetworkInfoLen, 200
pPrefNetwork, 200
CurrImageInfo, 202
buildID, 203
buildIDLen, 203
imageType, 203
uniqueID, 203
currNWInfo
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1283
CurrNetworkInfo, 203
NetworkType, 204
RATMask, 205
SOMask, 205
currNetworkInfo, 205
NetworkType, 206
RATMask, 206
SOMask, 206
unpack_wds_SLQSGetCurrDataSystemStat_t, 1271
current_channel_rx_rate
unpack_wds_SLQSGetCurrentChannelRate_t, 1271
WDSSWICurrentChannelRates, 1429
current_channel_tx_rate

unpack_wds_SLQSGetCurrentChannelRate_t, 1271
WDSSWICurrentChannelRates, 1429
currentCatEvent, 200
CatAlphaIdtfr, 200
CatEndPS, 200
CatEvIDData, 200
CatEventLst, 200
CatRefresh, 200
currentChannelRXRate
unpack_wds_GetConnectionRate_t, 1255
currentChannelTXRate
unpack_wds_GetConnectionRate_t, 1255
currentDataBearer
pack_wds_SLQSSetWdsEventCallback_t, 732
CurrentImgList, 200
carrier, 201
fwvers, 201
numEntries, 201
pCurrImgInfo, 201
pkgver, 201
priver, 201
currentMitigationLvl
QmiCbkTmdMitiLvlRptInd, 801
currentNetwork
dataBearerTechnology, 216
qmiWDSDataBearerTechnology, 806
CurrentPLMN
qaQmiServingSystemParam, 778
unpack_nas_SLQSGetServingSystem_t, 1129
currentPLMN, 201
MCC, 202
MNC, 202
netDescr, 202
netDescrLength, 202
cust_attr
custSettingInfo, 211
DMScustSettingInfo, 230
cust_id
custSettingInfo, 211
DMScustSettingInfo, 230
DMSgetCustomInput, 231
getCustomInput, 265
pack_dms_GetCustFeaturesV2_t, 598
pack_dms_SetCustFeaturesV2_t, 602
setCustomSettingV2, 855
cust_value
custSettingInfo, 211
DMScustSettingInfo, 230
pack_dms_SetCustFeaturesV2_t, 602
setCustomSettingV2, 855
custFeaturesInfo, 206
GpsEnable, 208
pDHCPRelayEnabled, 208
pDisableIMSI, 208
pGPSLPM, 208
pGPSSel, 208
pIPFamSupport, 208
pIsVoiceEnabled, 208
pRMAutoConnect, 208
pSMSSupport, 208
qaGobiApiDms.h, 1683
custFeaturesSetting, 209
pDHCPRelayEnabled, 210
pGPSEnable, 210
pGPSLPM, 210
pGPSSel, 210
pIsVoiceEnabled, 210
qaGobiApiDms.h, 1685
custSetting
custSettingList, 211
DMScustSettingList, 230
custSettingInfo, 210
cust_attr, 211
cust_id, 211
cust_value, 211
id_length, 211
value_length, 211
custSettingList, 211
custSetting, 211
list_type, 211
num_instances, 212
CustomSCP
unpack_nas_GetCDMANetworkParameters_t, 1106
CwtMute
GetM2MAudioProfileResp, 278
GetM2MAVMuteResp, 280

DEVICE_STATE_BOOT
qaGobiApiCbk.h, 1635
DEVICE_STATE_DISCONNECTED
qaGobiApiCbk.h, 1635
DEVICE_STATE_READY
qaGobiApiCbk.h, 1635
DEVICE_STATE_SERIAL_ONLY
qaGobiApiCbk.h, 1635
dBTechAvail
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1283
dBTechExtRatValue
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1283
dBTechExtSoMask
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1283
dBTechnology
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1283
dBtechExtAvail
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1283
dBtechnologyExt
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1283
DEFAULTBYTEVALUE
qaGobiApiPds.h, 1798

DEFAULTLONGVALUE
qaGobiApiPds.h, 1798
DEFAULTWORDVALUE
qaGobiApiPds.h, 1798
DEREGISTER_EVENT
qaGobiApiCbk.h, 1594
DEREGISTER_SRV
qaGobiApiCbk.h, 1594
DEVICE_OFFLINE
qaGobiApiFms.h, 1715
DEVICE_RESET
qaGobiApiFms.h, 1715
DEVICE_SHUTDOWN
qaGobiApiFms.h, 1715
DHCPOption, 225
optCode, 225
optValLen, 225
pOptVal, 225
DHCPOptionList, 225
numOpt, 226
pOptions, 226
DHCPRelayEnabled
pack_dms_SetCustFeature_t, 601
unpack_dms_GetCustFeature_t, 1038
DHCPv4LeaseOptTlv
unpack_wds_DHCPv4ClientLease_ind_t, 1253
DHCPv4LeaseStateTlv
unpack_wds_DHCPv4ClientLease_ind_t, 1253
dLatitude
unpack_pds_SetEventReport_Ind_t, 1167
dLongitude
unpack_pds_SetEventReport_Ind_t, 1167
DMS_IMGDETAILS_LEN
dms.h, 1442
DMS_PM_FACTORY
dms.h, 1442
DMS_PM_LOW
dms.h, 1443
DMS_PM_OFFLINE
dms.h, 1443
DMS_PM_ONLINE
dms.h, 1443
DMS_PM_RESET
dms.h, 1443
DMS_PM_SHUT_DOWN
dms.h, 1443
DMScustSettingInfo, 229
cust_attr, 230
cust_id, 230
cust_value, 230
id_length, 230
value_length, 230
DMScustSettingList, 230
custSetting, 230
list_type, 230
num_instances, 230
DMSgetCustomFeatureV2, 230
pCustSettingInfo, 231
pCustSettingList, 231
pGetCustomInput, 231
DMSgetCustomInput, 231
cust_id, 231
list_type, 231
DRCCover
DRCParams, 234
nas_DRCParams, 441
DRCParams, 234
DRCCover, 234
DRCValue, 234
DRCValue
DRCParams, 234
nas_DRCParams, 441
DTMFEvent
DTMFInfo, 235
DTMFInfo, 234
callID, 235
DTMFEvent, 235
digitBuff, 235
digitCnt, 235
DTMFInformation
voiceDTMFEventInfo, 1342
DTMFInterdigitInterval
DTMFLengths, 236
voice_DTMFLengths, 1322
DTMFLengths, 235
DTMFInterdigitInterval, 236
DTMFPulseWidth, 236
DTMFPulseWidth
DTMFLengths, 236
voice_DTMFLengths, 1322
DTMFdigit
pack_voice_SLQSVoiceStartContDTMF_t, 712
voiceContDTMFinfo, 1341
DTMInd
qaQmiServingSystemParam, 779
unpack_nas_SLQSGetServingSystem_t, 1129
DUNCallInfoInd, 236
CallEndReason, 236
ChannelRate, 236
DataBearerTech, 236
DormancyStatus, 236
MdmConnStatus, 236
RXOKBytesCount, 236
TXOKBytesCount, 236
Data
CatEventIDDataTlv, 159
data
authenticationData, 137
SMSCAddress, 913
sMSCAddressInfo, 913
SMSEtwsMessage, 914
sMSEtwsMessageInfo, 915
SMSTransferRouteMTMessage, 926
sMSTransferRouteMTMessageInfo, 926
SwiOTAMsg_s, 941
uim_authenticationData, 980

data_buf
NASOTAMessageTlv, 558
data_len
NASOTAMessageTlv, 558
SwiOTAMsg_s, 941
dataBearer
pack_wds_SLQSSetWdsEventCallback_t, 732
dataBearerMask
dataBearers, 212
unpack_wds_SLQSGetDataBearerTechnology_t, 1272
DataBearerTech, 212
DUNCallInfoInd, 236
ratValue, 214
soMask, 214
techType, 214
dataBearerTech
unpack_wds_SLQSGetDUNCallInfo_t, 1275
DataBearerTechExt, 214
pBearerTech, 215
pLastBearerTech, 215
dataBearerTechExt
pack_wds_SLQSSetWdsEventCallback_t, 732
dataBearerTechnology, 215
currentNetwork, 216
ratMask, 216
soMask, 216
dataBearers, 212
dataBearerMask, 212
pCurDataBearerTechnology, 212
pLastCallDataBearerTechnology, 212
dataCapabilities
dataSrvCapabilities, 217
nas_dataSrvCapabilities, 436
dataCapabilitiesLen
dataSrvCapabilities, 217
nas_dataSrvCapabilities, 436
DataCaps
unpack_nas_GetServingNetwork_t, 1114
unpack_nas_GetServingNetworkCapabilities_t, 1115
dataCaps
unpack_nas_SetDataCapabilitiesCallback_ind_t, 1117
DataCapsLen
unpack_nas_GetServingNetwork_t, 1114
unpack_nas_GetServingNetworkCapabilities_t, 1115
dataCapsSize
unpack_nas_SetDataCapabilitiesCallback_ind_t, 1117
dataLen
authenticationData, 137
uim_authenticationData, 980
DataLength
CatEventIDDataTlv, 159
DataRate
unpack_qos_swiQosFlow_t, 1190
dataRate, 216
dataRateMax, 217
guaranteedRate, 217
dataRateMax
dataRate, 217
unpack_qos_dataRate_t, 1174
dataServiceCaCapability
unpack_dms_GetDeviceCapabilities_t, 1041
DataServiceCapability
unpack_dms_GetDeviceCap_t, 1040
DataSessionStatus
_SwiAvmsEventInd, 99
DataSrvCapabilities
qaQmiServingSystemParam, 779
unpack_nas_SLQSGetServingSystem_t, 1129
dataSrvCapabilities, 217
dataCapabilities, 217
dataCapabilitiesLen, 217
DataStatusDetail, 218
IPAddress, 219
LastErrCode, 219
dataSysStatAvail
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1283
dataSystemStatus
pack_wds_SLQSSetWdsEventCallback_t, 732
DataULongLongTlv, 220
TlvPresent, 220
ullData, 220
DataULongTlv, 220
TlvPresent, 220
ulData, 220
Date
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1209
wcdmaLongMsgDecodingParams, 1387
wcdmaMsgDecodingParams, 1389
DateLength
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1209
day
nas_timeInfo, 522
nas_UniversalTime, 528
timeInfo, 965
UniversalTime, 1033
dayLtSavingAdj
nas_timeInfo, 522
timeInfo, 965
dayOfWeek
nas_timeInfo, 522
nas_UniversalTime, 528
timeInfo, 965
UniversalTime, 1033
daylightSavings
nas_qaQmi3Gpp2TimeZone, 496
qaQmi3Gpp2TimeZone, 774
DcsUsbPortNames, 220
AtCmdPort, 220

DmPort, 220
NmeaPort, 220
defaultPDNEnabled
unpack_wds_SLQSGet3GPPConfigItem_t, 1270
DefaultRoamInd
unpack_nas_SLQSGetServingSystem_t, 1129
defaultRoamInd
qaQmiServingSystemParam, 779
delAssistDataStatus, 220
status, 221
delayClass
GPRSQoS, 288
GPRSRequestedQoS, 289
LibPackGPRSRequestedQoS, 336
wds_GPRSQoS, 1403
DeleteStoredImage
qaGobiApiFms.h, 1718
deliveryErrSDU
LibPackUMTSQoS, 364
UMTSMinQoS, 1027
UMTSQoS, 1030
wds_UMTSMinQoS, 1411
depersonalizationInformation, 221
ckLen, 222
ckVal, 222
feature, 222
operation, 222
depersonilisationInfo
pack_uim_SLQSUIMDepersonalization_t, 686
UIMDepersonalizationReq, 1003
Description
SlqsNas3GppNetworkInfo, 885
description
omaDmFotaTlv, 591
omaDmFotaTlvExt, 594
unpack_omaDmFotaTlv_t, 1160
descriptionlength
omaDmFotaTlv, 591
omaDmFotaTlvExt, 594
unpack_omaDmFotaTlv_t, 1160
Desription
nas_QmiNas3GppNetworkInfo, 498
destPortRangeEnd
LibPackTFTIDParams, 361
TFTIDParams, 963
destPortRangeStart
LibPackTFTIDParams, 362
TFTIDParams, 963
detailSvcInfo, 222
hdrHybrid, 223
hdrSrvStatus, 223
isSysForbidden, 223
srvCapability, 223
srvStatus, 224
DetailedSvcInfo
qaQmiServingSystemParam, 779
unpack_nas_SLQSGetServingSystem_t, 1129
dev
qmifwinfo_s, 802
DevCrashState
unpack_dms_GetCrashAction_t, 1036
Device
SetM2MAudioAVCFGReq, 861
Device Connectivity Service (DCS), 37
Device Management Service (DMS), 39
device_state_enum
qaGobiApiCbk.h, 1635
DeviceConfigDetail, 224
Chipset, 224
HWVersion, 224
QLIC, 225
Technology, 225
deviceID
_MitigationDevInfo, 67
deviceIdLen
_MitigationDevInfo, 67
DiagInfo
allCallsDiagInfo, 118
arrDiagInfo, 132
voice_allCallsDiagInfo, 1298
voice_arrDiagInfo, 1304
diagInfo, 226
diagInfoLen, 226
diagnosticInfo, 226
diagInfoLen
diagInfo, 226
voice_diagInfo, 1321
diagnosticInfo
diagInfo, 226
voice_diagInfo, 1321
digitBuff
DTMFInfo, 235
digitCnt
burstDTMFInfo, 142
DTMFInfo, 235
voice_burstDTMFInfo, 1307
dirNum, 226
dirNum, 227
dirNumLen, 227
dirNum, 227
nas_dirNum, 440
dirNumLen
dirNum, 227
nas_dirNum, 440
direction
callInfo, 153
voice_callInfo, 1314
DisableIMSI
pack_dms_SetCustFeature_t, 601
unpack_dms_GetCustFeature_t, 1038
dispType
extDispRecInfo, 243
displayCondition
nas_serviceProviderName, 509
serviceProviderName, 845
dl_bw_value

nas_PhyCaAggPcellInfo, 487
nas_PhyCaAggScellDlBw, 487
nas_PhyCaAggScellInfo, 491
NASPhyCaAggPcellInfo, 558
NASPhyCaAggScellDlBw, 559
NASPhyCaAggScellInfo, 561
PhyCaAggPcellInfo, 747
PhyCaAggScellDlBw, 748
PhyCaAggScellInfo, 752
DmPort
DcsUsbPortNames, 220
dms.h, 1436
ACT_CODE_MAX_SIZE, 1442
CK_MAX_SIZE, 1442
DMS_IMGDETAILS_LEN, 1442
DMS_PM_FACTORY, 1442
DMS_PM_LOW, 1443
DMS_PM_OFFLINE, 1443
DMS_PM_ONLINE, 1443
DMS_PM_RESET, 1443
DMS_PM_SHUT_DOWN, 1443
ERI_DATA_MAX_SIZE, 1443
MAX_BUILD_ID_LEN, 1443
MEID_MAX_SIZE, 1443
pack_dms_ActivateAutomatic, 1444
pack_dms_GetActivationState, 1444
pack_dms_GetBandCapability, 1444
pack_dms_GetCrashAction, 1445
pack_dms_GetCustFeature, 1445
pack_dms_GetCustFeaturesV2, 1445
pack_dms_GetDeviceCap, 1446
pack_dms_GetDeviceCapabilities, 1446
pack_dms_GetDeviceHardwareRev, 1446
pack_dms_GetDeviceMfr, 1447
pack_dms_GetDeviceSerialNumbers, 1447
pack_dms_GetFSN, 1449
pack_dms_GetFirmwareInfo, 1448
pack_dms_GetFirmwareRevision, 1448
pack_dms_GetFirmwareRevisions, 1448
pack_dms_GetHardwareRevision, 1449
pack_dms_GetIMSI, 1449
pack_dms_GetManufacturer, 1450
pack_dms_GetModelID, 1450
pack_dms_GetNetworkTime, 1450
pack_dms_GetOfflineReason, 1451
pack_dms_GetPRLVersion, 1451
pack_dms_GetPower, 1451
pack_dms_GetSerialNumbers, 1452
pack_dms_GetUSBComp, 1452
pack_dms_GetVoiceNumber, 1452
pack_dms_ResetToFactoryDefaults, 1453
pack_dms_SLQSDmsSwiGetResetInfo, 1456
pack_dms_SLQSDmsSwiIndicationRegister, 1456
pack_dms_SLQSGetBandCapability, 1457
pack_dms_SLQSGetERIFile, 1457
pack_dms_SLQSSwiClearDyingGaspStatistics, 1458
pack_dms_SLQSSwiGetCrashInfo, 1458
pack_dms_SLQSSwiGetDyingGaspCfg, 1458
pack_dms_SLQSSwiGetDyingGaspStatistics, 1459
pack_dms_SLQSSwiGetFirmwareCurr, 1459
pack_dms_SLQSSwiGetFwUpdateStatus, 1459
pack_dms_SLQSSwiGetHostDevInfo, 1460
pack_dms_SLQSSwiGetOSInfo, 1460
pack_dms_SLQSSwiGetSerialNoExt, 1460
pack_dms_SLQSSwiSetDyingGaspCfg, 1461
pack_dms_SLQSSwiSetHostDevInfo, 1461
pack_dms_SLQSSwiSetOSInfo, 1461
pack_dms_SLQSUIMGetState, 1462
pack_dms_SetActivationStatusCallback, 1453
pack_dms_SetCrashAction, 1454
pack_dms_SetCustFeature, 1454
pack_dms_SetCustFeaturesV2, 1454
pack_dms_SetEventReport, 1455
pack_dms_SetFirmwarePreference, 1455
pack_dms_SetPower, 1455
pack_dms_SetUSBComp, 1456
pack_dms_UIMChangePIN, 1462
pack_dms_UIMGetControlKeyStatus, 1462
pack_dms_UIMGetICCID, 1463
pack_dms_UIMGetPINStatus, 1463
pack_dms_UIMSetControlKeyProtection, 1464
pack_dms_UIMSetPINProtection, 1464
pack_dms_UIMUnblockControlKey, 1464
pack_dms_UIMUnblockPIN, 1465
pack_dms_UIMVerifyPIN, 1465
pack_dms_ValidateSPC, 1465
SPC_SIZE, 1443
UNIQUE_ID_LEN, 1443
unpack_dms_ActivateAutomatic, 1466
unpack_dms_GetActivationState, 1466
unpack_dms_GetBandCapability, 1467
unpack_dms_GetCrashAction, 1467
unpack_dms_GetCustFeature, 1467
unpack_dms_GetCustFeaturesV2, 1468
unpack_dms_GetDeviceCap, 1468
unpack_dms_GetDeviceCapabilities, 1468
unpack_dms_GetDeviceHardwareRev, 1469
unpack_dms_GetDeviceMfr, 1469
unpack_dms_GetDeviceSerialNumbers, 1469
unpack_dms_GetFSN, 1471
unpack_dms_GetFirmwareInfo, 1470
unpack_dms_GetFirmwareRevision, 1470
unpack_dms_GetFirmwareRevisions, 1470
unpack_dms_GetHardwareRevision, 1471
unpack_dms_GetIMSI, 1471
unpack_dms_GetManufacturer, 1472
unpack_dms_GetModelID, 1472
unpack_dms_GetNetworkTime, 1472
unpack_dms_GetOfflineReason, 1473
unpack_dms_GetPRLVersion, 1473
unpack_dms_GetPower, 1473
unpack_dms_GetSerialNumbers, 1474
unpack_dms_GetUSBComp, 1474
unpack_dms_GetVoiceNumber, 1474

unpack_dms_ResetToFactoryDefaults, 1475
unpack_dms_SLQSDmsSwiGetResetInfo, 1478
unpack_dms_SLQSDmsSwiGetResetInfo_Ind, 1478
unpack_dms_SLQSDmsSwiIndicationRegister, 1479
unpack_dms_SLQSGetBandCapability, 1479
unpack_dms_SLQSGetERIFile, 1480
unpack_dms_SLQSSwiClearDyingGaspStatistics, 1480
unpack_dms_SLQSSwiGetCrashInfo, 1480
unpack_dms_SLQSSwiGetDyingGaspCfg, 1481
unpack_dms_SLQSSwiGetDyingGaspStatistics, 1481
unpack_dms_SLQSSwiGetFirmwareCurr, 1481
unpack_dms_SLQSSwiGetFwUpdateStatus, 1482
unpack_dms_SLQSSwiGetHostDevInfo, 1482
unpack_dms_SLQSSwiGetOSInfo, 1482
unpack_dms_SLQSSwiGetSerialNoExt, 1483
unpack_dms_SLQSSwiSetDyingGaspCfg, 1483
unpack_dms_SLQSSwiSetHostDevInfo, 1483
unpack_dms_SLQSSwiSetOSInfo, 1484
unpack_dms_SLQSUIMGetState, 1484
unpack_dms_SetActivationStatusCallback, 1475
unpack_dms_SetCrashAction, 1475
unpack_dms_SetCustFeature, 1476
unpack_dms_SetCustFeaturesV2, 1476
unpack_dms_SetEventReport, 1476
unpack_dms_SetEventReport_ind, 1477
unpack_dms_SetFirmwarePreference, 1477
unpack_dms_SetPower, 1477
unpack_dms_SetUSBComp, 1478
unpack_dms_UIMChangePIN, 1484
unpack_dms_UIMGetControlKeyStatus, 1485
unpack_dms_UIMGetICCID, 1485
unpack_dms_UIMGetPINStatus, 1485
unpack_dms_UIMSetControlKeyProtection, 1486
unpack_dms_UIMSetPINProtection, 1486
unpack_dms_UIMUnblockControlKey, 1486
unpack_dms_UIMUnblockPIN, 1487
unpack_dms_UIMVerifyPIN, 1487
unpack_dms_ValidateSPC, 1488
dms_ActivationStatusTlv, 227
activationStatus, 228
TlvPresent, 228
dms_OperatingModeTlv, 228
operatingMode, 228
TlvPresent, 228
dmsCurrentPRLInfo, 228
pPRLPreference, 229
pPRLVersion, 229
qaGobiApiDms.h, 1686
dmsIndicationRegisterReq, 232
pSwiGetResetInd, 232
dmsSwiGetResetInfo, 232
source, 233
type, 233
Domain, 233
domainLen, 233
domainName, 233
domain
DomainNameList, 234
wds_DomainNameList, 1403
domainLen
Domain, 233
wds_Domain, 1402
DomainList
unpack_wds_SLQSGetRuntimeSettings_t, 1278
domainName
Domain, 233
wds_Domain, 1402
DomainNameList, 233
domain, 234
numInstances, 234
dormancyStatAvail
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1283
dormancyState
unpack_wds_GetDormancyState_t, 1259
DormancyStatus
DUNCallInfoInd, 236
dormancyStatus
pack_wds_SLQSSetWdsEventCallback_t, 732
unpack_wds_SLQSGetDUNCallInfo_t, 1275
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1283
downLink
NSSAudioCtrl, 586
DownloadToSlot
qaGobiApiFms.h, 1718
dscp
QosMap, 816
dtmSupp
GSMSysInfo, 298
nas_GSMSysInfo, 450
dtmSuppValid
GSMSysInfo, 298
nas_GSMSysInfo, 450
dunchannelRate, 237
CurrChanRxRate, 237
CurrChanTxRate, 237
MaxChanRxRate, 237
MaxChanTxRate, 237
Duration
pack_nas_SetNetworkPreference_t, 631
unpack_nas_GetNetworkPreference_t, 1111

eAUDIO
common.h, 1434
eCAT
common.h, 1434
eCTL
common.h, 1434
eDMS
common.h, 1434
eGOBI_DEV_SERIES_9X07_9X50
qaGobiApiFms.h, 1716

eGOBI_DEV_SERIES_9X15
qaGobiApiFms.h, 1716
eGOBI_DEV_SERIES_9X30
qaGobiApiFms.h, 1716
eGOBI_DEV_SERIES_G3K
qaGobiApiFms.h, 1716
eGOBI_DEV_SERIES_NON_GOBI
qaGobiApiFms.h, 1716
eGOBI_DEV_SERIES_SIERRA_GOBI
qaGobiApiFms.h, 1716
eGOBI_DEV_SERIES_UNKNOWN
qaGobiApiFms.h, 1716
eGOBI_IMG_CAR_3
qaGobiApiFms.h, 1717
eGOBI_IMG_CAR_AERIS
qaGobiApiFms.h, 1717
eGOBI_IMG_CAR_ALLTEL
qaGobiApiFms.h, 1716
eGOBI_IMG_CAR_AMX_TELCEL
qaGobiApiFms.h, 1717
eGOBI_IMG_CAR_ATT
qaGobiApiFms.h, 1716
eGOBI_IMG_CAR_BELL
qaGobiApiFms.h, 1716
eGOBI_IMG_CAR_BHARTI
qaGobiApiFms.h, 1717
eGOBI_IMG_CAR_BRASIL_VIVO
qaGobiApiFms.h, 1717
eGOBI_IMG_CAR_CHINA_MOBILE
qaGobiApiFms.h, 1717
eGOBI_IMG_CAR_CHINA_TELECOM
qaGobiApiFms.h, 1716
eGOBI_IMG_CAR_CHINA_UNICOM
qaGobiApiFms.h, 1716
eGOBI_IMG_CAR_EMOBILE
qaGobiApiFms.h, 1717
eGOBI_IMG_CAR_FACTORY
qaGobiApiFms.h, 1716
eGOBI_IMG_CAR_GENERIC
qaGobiApiFms.h, 1716
eGOBI_IMG_CAR_GENERIC_CDMA
qaGobiApiFms.h, 1716
eGOBI_IMG_CAR_IUSACELL
qaGobiApiFms.h, 1716
eGOBI_IMG_CAR_KDDI
qaGobiApiFms.h, 1716
eGOBI_IMG_CAR_KT_FREETEL
qaGobiApiFms.h, 1717
eGOBI_IMG_CAR_LEAP
qaGobiApiFms.h, 1716
eGOBI_IMG_CAR_METROPCS
qaGobiApiFms.h, 1716
eGOBI_IMG_CAR_NETCOM
qaGobiApiFms.h, 1717
eGOBI_IMG_CAR_NORF
qaGobiApiFms.h, 1716
eGOBI_IMG_CAR_NTT_DOCOMO
qaGobiApiFms.h, 1717
eGOBI_IMG_CAR_O2
qaGobiApiFms.h, 1717
eGOBI_IMG_CAR_OMH
qaGobiApiFms.h, 1716
eGOBI_IMG_CAR_ORANGE
qaGobiApiFms.h, 1717
eGOBI_IMG_CAR_RELIANCE1
qaGobiApiFms.h, 1716
eGOBI_IMG_CAR_RELIANCE2
qaGobiApiFms.h, 1717
eGOBI_IMG_CAR_ROGERS
qaGobiApiFms.h, 1717
eGOBI_IMG_CAR_SFR
qaGobiApiFms.h, 1717
eGOBI_IMG_CAR_SINGTEL_OPTUS
qaGobiApiFms.h, 1717
eGOBI_IMG_CAR_SK_TELCOM1
qaGobiApiFms.h, 1716
eGOBI_IMG_CAR_SK_TELCOM2
qaGobiApiFms.h, 1717
eGOBI_IMG_CAR_SOFTBANK
qaGobiApiFms.h, 1717
eGOBI_IMG_CAR_SPRINT
qaGobiApiFms.h, 1716
eGOBI_IMG_CAR_SWISSCOM
qaGobiApiFms.h, 1717
eGOBI_IMG_CAR_TATA
qaGobiApiFms.h, 1716
eGOBI_IMG_CAR_TELCOM_ITALIA
qaGobiApiFms.h, 1717
eGOBI_IMG_CAR_TELCOM_NZ
qaGobiApiFms.h, 1716
eGOBI_IMG_CAR_TELEFONICA
qaGobiApiFms.h, 1717
eGOBI_IMG_CAR_TELENOR
qaGobiApiFms.h, 1717
eGOBI_IMG_CAR_TELIASONERA
qaGobiApiFms.h, 1717
eGOBI_IMG_CAR_TELSTRA1
qaGobiApiFms.h, 1716
eGOBI_IMG_CAR_TELSTRA2
qaGobiApiFms.h, 1717
eGOBI_IMG_CAR_TELUS
qaGobiApiFms.h, 1716
eGOBI_IMG_CAR_TMOBILE
qaGobiApiFms.h, 1716
eGOBI_IMG_CAR_US
qaGobiApiFms.h, 1716
eGOBI_IMG_CAR_VERIZON
qaGobiApiFms.h, 1716
eGOBI_IMG_CAR_VODAFONE
qaGobiApiFms.h, 1716
eGOBI_IMG_GPS_ASSISTED
qaGobiApiFms.h, 1717
eGOBI_IMG_GPS_NO_XTRA
qaGobiApiFms.h, 1717
eGOBI_IMG_GPS_NONE
qaGobiApiFms.h, 1717

eGOBI_IMG_GPS_STAND_ALONE
qaGobiApiFms.h, 1717
eGOBI_IMG_REG_ASIA
qaGobiApiFms.h, 1717
eGOBI_IMG_REG_AUS
qaGobiApiFms.h, 1717
eGOBI_IMG_REG_EU
qaGobiApiFms.h, 1717
eGOBI_IMG_REG_GLOBAL
qaGobiApiFms.h, 1717
eGOBI_IMG_REG_LA
qaGobiApiFms.h, 1717
eGOBI_IMG_REG_NA
qaGobiApiFms.h, 1717
eGOBI_IMG_TECH_CDMA
qaGobiApiFms.h, 1718
eGOBI_IMG_TECH_UMTS
qaGobiApiFms.h, 1718
eGobi_DEV_SERIES_MC83
qaGobiApiFms.h, 1716
eIMS
common.h, 1434
eIMSA
common.h, 1434
eIND
common.h, 1435
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_100
nas.h, 1517
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_15
nas.h, 1517
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_25
nas.h, 1517
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_50
nas.h, 1517
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_6
nas.h, 1517
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_75
nas.h, 1517
eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED
nas.h, 1517
eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED
nas.h, 1517
eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_DECONFIGURED
nas.h, 1517
eLOC
common.h, 1434
eLOG_DEBUG
common.h, 1434
eLOG_FATAL
common.h, 1434
eLOG_INFO
common.h, 1434
eLOG_WARN
common.h, 1434
eNAS
common.h, 1434
eNAS_LTE_CPHY_CA_BW_NRB_100
qaGobiApiNas.h, 1765
eNAS_LTE_CPHY_CA_BW_NRB_15
qaGobiApiNas.h, 1765
eNAS_LTE_CPHY_CA_BW_NRB_25
qaGobiApiNas.h, 1765
eNAS_LTE_CPHY_CA_BW_NRB_50
qaGobiApiNas.h, 1765
eNAS_LTE_CPHY_CA_BW_NRB_6
qaGobiApiNas.h, 1765
eNAS_LTE_CPHY_CA_BW_NRB_75
qaGobiApiNas.h, 1765
eNAS_LTE_CPHY_CA_BW_NRB_LITE_100
nas.h, 1517
eNAS_LTE_CPHY_CA_BW_NRB_LITE_15
nas.h, 1517
eNAS_LTE_CPHY_CA_BW_NRB_LITE_25
nas.h, 1517
eNAS_LTE_CPHY_CA_BW_NRB_LITE_50
nas.h, 1517
eNAS_LTE_CPHY_CA_BW_NRB_LITE_6
nas.h, 1517
eNAS_LTE_CPHY_CA_BW_NRB_LITE_75
nas.h, 1517
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED
qaGobiApiNas.h, 1765
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED_LITE
nas.h, 1517
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED
qaGobiApiNas.h, 1765
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED_LITE
nas.h, 1517
eNAS_LTE_CPHY_SCELL_STATE_DECONFIGURED
qaGobiApiNas.h, 1765
eNAS_LTE_CPHY_SCELL_STATE_DECONFIGURED_LITE
nas.h, 1517
eNAS_RADIO_IF_GSM
qaGobiApiNas.h, 1764
eNAS_RADIO_IF_LTE
qaGobiApiNas.h, 1764
eNAS_RADIO_IF_TDSCDMA
qaGobiApiNas.h, 1764
eNAS_RADIO_IF_UMTS
qaGobiApiNas.h, 1764
ePDS
common.h, 1434
eQA_QMI_SVC_NA
qaGobiApiCbk.h, 1635
eQA_QMI_SVC_NAS
qaGobiApiCbk.h, 1635
eQA_QMI_SVC_WDS
qaGobiApiCbk.h, 1635

eQCWWAN_ERR_API_MUTEX_TIMEOUT
qmerrno.h, 1998
eQCWWAN_ERR_BUFFER_SZ
qmerrno.h, 1997
eQCWWAN_ERR_CANCEL_OP
qmerrno.h, 1998
eQCWWAN_ERR_DRIVER
qmerrno.h, 1998
eQCWWAN_ERR_ENUM_BEGIN
qmerrno.h, 1997
eQCWWAN_ERR_ENUM_END
qmerrno.h, 1998
eQCWWAN_ERR_FILE_COPY
qmerrno.h, 1997
eQCWWAN_ERR_FILE_OPEN
qmerrno.h, 1997
eQCWWAN_ERR_GENERAL
qmerrno.h, 1997
eQCWWAN_ERR_INTERNAL
qmerrno.h, 1997
eQCWWAN_ERR_INVALID_ARG
qmerrno.h, 1997
eQCWWAN_ERR_INVALID_DEVID
qmerrno.h, 1997
eQCWWAN_ERR_INVALID_FILE
qmerrno.h, 1997
eQCWWAN_ERR_INVALID_QMI_RSP
qmerrno.h, 1997
eQCWWAN_ERR_INVALID_XID
qmerrno.h, 1998
eQCWWAN_ERR_MALFORMED_QMI_RSP
qmerrno.h, 1997
eQCWWAN_ERR_MEMORY
qmerrno.h, 1997
eQCWWAN_ERR_MULTIPLE_DEVICES
qmerrno.h, 1998
eQCWWAN_ERR_MULTIPLE_SMS_UNSUPPORTED
qmerrno.h, 1998
eQCWWAN_ERR_NO_CANCELABLE_OP
qmerrno.h, 1998
eQCWWAN_ERR_NO_CONNECTION
qmerrno.h, 1997
eQCWWAN_ERR_NO_DEVICE
qmerrno.h, 1997
eQCWWAN_ERR_NO_SIGNAL
qmerrno.h, 1998
eQCWWAN_ERR_NONE
qmerrno.h, 1997
eQCWWAN_ERR_NULL_TLV
qmerrno.h, 2001
eQCWWAN_ERR_OFFLINE
qmerrno.h, 1998
eQCWWAN_ERR_PDU_GENERATION
qmerrno.h, 1998
eQCWWAN_ERR_QMI_ABORTED
qmerrno.h, 1998
eQCWWAN_ERR_QMI_ACCESS_DENIED
qmerrno.h, 2000
eQCWWAN_ERR_QMI_ACK_NOT_SENT
qmerrno.h, 2000
eQCWWAN_ERR_QMI_ARG_TOO_LONG
qmerrno.h, 1998
eQCWWAN_ERR_QMI_AUTHENTICATION_FAILED
qmerrno.h, 1999
eQCWWAN_ERR_QMI_AUTHENTICATION_LOCK
qmerrno.h, 1999
eQCWWAN_ERR_QMI_BUNDLING_NOT_SUPPORTED
qmerrno.h, 2000
eQCWWAN_ERR_QMI_CALL_FAILED
qmerrno.h, 1998
eQCWWAN_ERR_QMI_CARD_BUSY_RSP
qmerrno.h, 2001
eQCWWAN_ERR_QMI_CARD_CALL_CONTROL_FAILED
qmerrno.h, 2000
eQCWWAN_ERR_QMI_CAT_END
qmerrno.h, 2001
eQCWWAN_ERR_QMI_CAT_START
qmerrno.h, 2001
eQCWWAN_ERR_QMI_CAUSE_CODE
qmerrno.h, 1999
eQCWWAN_ERR_QMI_CLIENT_IDS_EXHAUSTED
qmerrno.h, 1998
eQCWWAN_ERR_QMI_CONNECT
qmerrno.h, 1997
eQCWWAN_ERR_QMI_DEVICE_IN_USE
qmerrno.h, 1998
eQCWWAN_ERR_QMI_DEVICE_NOT_READY
qmerrno.h, 1999
eQCWWAN_ERR_QMI_DEVICE_STORAGE_FULL
qmerrno.h, 1999
eQCWWAN_ERR_QMI_DISABLED
qmerrno.h, 1999
eQCWWAN_ERR_QMI_ENCODING
qmerrno.h, 1999
eQCWWAN_ERR_QMI_ENVELOPE_CMD_FAILURE
qmerrno.h, 2001
eQCWWAN_ERR_QMI_EVENT_REG_FAILED
qmerrno.h, 2001
eQCWWAN_ERR_QMI_EXTENDED_INTERNAL
qmerrno.h, 2000
eQCWWAN_ERR_QMI_FDN_RESTRICT
qmerrno.h, 2000
eQCWWAN_ERR_QMI_FLOW_SUSPENDED
qmerrno.h, 1999
eQCWWAN_ERR_QMI_GENERAL
qmerrno.h, 1999
eQCWWAN_ERR_QMI_HARDWARE_RESTRICTED
qmerrno.h, 2000
eQCWWAN_ERR_QMI_IFACE
qmerrno.h, 1997
eQCWWAN_ERR_QMI_INCOMPATIBLE_STATE
qmerrno.h, 2000
eQCWWAN_ERR_QMI_INCORRECT_FLOW_FILTER
qmerrno.h, 1999

eQCWWAN_ERR_QMI_INCORRECT_PIN
qmerrno.h, 1998
eQCWWAN_ERR_QMI_INFO_UNAVAILABLE
qmerrno.h, 2000
eQCWWAN_ERR_QMI_INJECT_TIMEOUT
qmerrno.h, 2000
eQCWWAN_ERR_QMI_INSUFFICIENT_RESOURCE-S
qmerrno.h, 1999
eQCWWAN_ERR_QMI_INTERFACE_NOT_FOUND
qmerrno.h, 1999
eQCWWAN_ERR_QMI_INTERNAL
qmerrno.h, 1998
eQCWWAN_ERR_QMI_INVALID_ARG
qmerrno.h, 1999
eQCWWAN_ERR_QMI_INVALID_CLIENT_ID
qmerrno.h, 1998
eQCWWAN_ERR_QMI_INVALID_DATA_FORMAT
qmerrno.h, 1999
eQCWWAN_ERR_QMI_INVALID_ENVELOPE_CMD
qmerrno.h, 2001
eQCWWAN_ERR_QMI_INVALID_HANDLE
qmerrno.h, 1998
eQCWWAN_ERR_QMI_INVALID_ID
qmerrno.h, 1999
eQCWWAN_ERR_QMI_INVALID_INDEX
qmerrno.h, 1999
eQCWWAN_ERR_QMI_INVALID_IP_FAMILY_PREF
qmerrno.h, 1999
eQCWWAN_ERR_QMI_INVALID_MCAST_HANDLE
qmerrno.h, 1999
eQCWWAN_ERR_QMI_INVALID_MESSAGE_ID
qmerrno.h, 1999
eQCWWAN_ERR_QMI_INVALID_OPERATION
qmerrno.h, 2000
eQCWWAN_ERR_QMI_INVALID_PDP_TYPE
qmerrno.h, 1998
eQCWWAN_ERR_QMI_INVALID_PINID
qmerrno.h, 1998
eQCWWAN_ERR_QMI_INVALID_PROFILE
qmerrno.h, 1998
eQCWWAN_ERR_QMI_INVALID_PROFILE_TYPE
qmerrno.h, 1998
eQCWWAN_ERR_QMI_INVALID_PS_ATTACH_ACTI-ON
qmerrno.h, 1999
eQCWWAN_ERR_QMI_INVALID_QMI_CMD
qmerrno.h, 2000
eQCWWAN_ERR_QMI_INVALID_QOS_ID
qmerrno.h, 1999
eQCWWAN_ERR_QMI_INVALID_REGISTER_ACTIO-N
qmerrno.h, 1999
eQCWWAN_ERR_QMI_INVALID_SERVICE_TYPE
qmerrno.h, 1998
eQCWWAN_ERR_QMI_INVALID_TECH_PREF
qmerrno.h, 1998
eQCWWAN_ERR_QMI_INVALID_TERMINAL_RSP
qmerrno.h, 2001
eQCWWAN_ERR_QMI_INVALID_TRANSITION
qmerrno.h, 1999
eQCWWAN_ERR_QMI_INVALID_TX_ID
qmerrno.h, 1998
eQCWWAN_ERR_QMI_MALFORMED_MSG
qmerrno.h, 1998
eQCWWAN_ERR_QMI_MAX
qmerrno.h, 2000
eQCWWAN_ERR_QMI_MAX_MCAST_REQUESTS_I-N_USE
qmerrno.h, 1999
eQCWWAN_ERR_QMI_MAX_QOS_REQUESTS_IN_-USE
qmerrno.h, 1999
eQCWWAN_ERR_QMI_MESSAGE_DELIVERY_FAIL-URE
qmerrno.h, 1999
eQCWWAN_ERR_QMI_MESSAGE_NOT_SENT
qmerrno.h, 1999
eQCWWAN_ERR_QMI_MISSING_ARG
qmerrno.h, 1998
eQCWWAN_ERR_QMI_MSG_BLOCKED
qmerrno.h, 2000
eQCWWAN_ERR_QMI_NETWORK_ABORTED
qmerrno.h, 2000
eQCWWAN_ERR_QMI_NETWORK_NOT_READY
qmerrno.h, 1999
eQCWWAN_ERR_QMI_NETWORK_QOS_UNAWARE
qmerrno.h, 1999
eQCWWAN_ERR_QMI_NO_EFFECT
qmerrno.h, 1998
eQCWWAN_ERR_QMI_NO_ENTRY
qmerrno.h, 1999
eQCWWAN_ERR_QMI_NO_FREE_PROFILE
qmerrno.h, 1998
eQCWWAN_ERR_QMI_NO_MEMORY
qmerrno.h, 1998
eQCWWAN_ERR_QMI_NO_NETWORK_FOUND
qmerrno.h, 1998
eQCWWAN_ERR_QMI_NO_RADIO
qmerrno.h, 2000
eQCWWAN_ERR_QMI_NO_SUBSCRIPTION
qmerrno.h, 2000
eQCWWAN_ERR_QMI_NO_THRESHOLDS
qmerrno.h, 1998
eQCWWAN_ERR_QMI_NOT_A_MCAST_IFACE
qmerrno.h, 1999
eQCWWAN_ERR_QMI_NOT_PROVISIONED
qmerrno.h, 1998
eQCWWAN_ERR_QMI_NOT_SUPPORTED
qmerrno.h, 2000
eQCWWAN_ERR_QMI_OFFSET
qmerrno.h, 1998
eQCWWAN_ERR_QMI_OP_DEVICE_UNSUPPORTE-D
qmerrno.h, 1998

eQCWWAN_ERR_QMI_OP_NETWORK_UNSUPPORTED
qmerrno.h, 1998
eQCWWAN_ERR_QMI_OP_PARTIAL_FAILURE
qmerrno.h, 2000
eQCWWAN_ERR_QMI_OUT_OF_CALL
qmerrno.h, 1998
eQCWWAN_ERR_QMI_PIN_BLOCKED
qmerrno.h, 1999
eQCWWAN_ERR_QMI_PIN_PERM_BLOCKED
qmerrno.h, 1999
eQCWWAN_ERR_QMI_POLICY_MISMATCH
qmerrno.h, 2000
eQCWWAN_ERR_QMI_REQ
qmerrno.h, 1997
eQCWWAN_ERR_QMI_REQ_SCH
qmerrno.h, 1997
eQCWWAN_ERR_QMI_REQ_TO
qmerrno.h, 1997
eQCWWAN_ERR_QMI_REQUESTED_NUM_UNSUPPORTED
qmerrno.h, 1999
eQCWWAN_ERR_QMI_RSP
qmerrno.h, 1997
eQCWWAN_ERR_QMI_RSP_TO
qmerrno.h, 1997
eQCWWAN_ERR_QMI_SEGMENT_ORDER
qmerrno.h, 2000
eQCWWAN_ERR_QMI_SEGMENT_TOO_LONG
qmerrno.h, 2000
eQCWWAN_ERR_QMI_SESSION_INACTIVE
qmerrno.h, 1999
eQCWWAN_ERR_QMI_SESSION_INVALID
qmerrno.h, 1999
eQCWWAN_ERR_QMI_SESSION_OWNERSHIP
qmerrno.h, 1999
eQCWWAN_ERR_QMI_SIM_FILE_NOT_FOUND
qmerrno.h, 2000
eQCWWAN_ERR_QMI_SIM_NOT_INITIALIZED
qmerrno.h, 1999
eQCWWAN_ERR_QMI_SMSC_ADDR
qmerrno.h, 2000
eQCWWAN_ERR_QMI_SUPS_FAILURE_CAUSE
qmerrno.h, 2000
eQCWWAN_ERR_QMI_TPDU_TYPE
qmerrno.h, 2000
eQCWWAN_ERR_QMI_UNABORTABLE_TRANSACTION
qmerrno.h, 1998
eQCWWAN_ERR_QMI_UNKNOWN
qmerrno.h, 1999
eQCWWAN_ERR_QMI_WIDTH
qmerrno.h, 2001
eQCWWAN_ERR_RESET
qmerrno.h, 1998
eQCWWAN_ERR_SWICM_AM_VERS_ERROR
qmerrno.h, 2000
eQCWWAN_ERR_SWICM_CALL_IN_PROGRESS
qmerrno.h, 2000
eQCWWAN_ERR_SWICM_END
qmerrno.h, 2001
eQCWWAN_ERR_SWICM_FAILED_TO_KILL_SDK_PROCESS
qmerrno.h, 2000
eQCWWAN_ERR_SWICM_INVALID_SESSION_ID
qmerrno.h, 2000
eQCWWAN_ERR_SWICM_INVALID_V4_SESSION_ID
qmerrno.h, 2000
eQCWWAN_ERR_SWICM_INVALID_V6_SESSION_ID
qmerrno.h, 2000
eQCWWAN_ERR_SWICM_NOT_IMPLEMENTED
qmerrno.h, 2000
eQCWWAN_ERR_SWICM_QMI_CLNT_NOT_SUPPORTED
qmerrno.h, 2000
eQCWWAN_ERR_SWICM_QMI_SVC_NOT_SUPPORTED
qmerrno.h, 2000
eQCWWAN_ERR_SWICM_SM_NO_AVAILABLE_SESSIONS
qmerrno.h, 2000
eQCWWAN_ERR_SWICM_SOCKET_IN_USE
qmerrno.h, 2000
eQCWWAN_ERR_SWICM_START
qmerrno.h, 2000
eQCWWAN_ERR_SWICM_TIMEOUT
qmerrno.h, 2000
eQCWWAN_ERR_SWICM_V4DWN_V6DWN
qmerrno.h, 2000
eQCWWAN_ERR_SWICM_V4DWN_V6UP
qmerrno.h, 2000
eQCWWAN_ERR_SWICM_V4UP_V6DWN
qmerrno.h, 2000
eQCWWAN_ERR_SWICM_V4UP_V6UP
qmerrno.h, 2000
eQCWWAN_ERR_SWIDCS_APP_DISCONNECTED
qmerrno.h, 2001
eQCWWAN_ERR_SWIDCS_DEVNODE_NOT_FOUND
qmerrno.h, 2001
eQCWWAN_ERR_SWIDCS_END
qmerrno.h, 2001
eQCWWAN_ERR_SWIDCS_FILEIO_ERR
qmerrno.h, 2001
eQCWWAN_ERR_SWIDCS_IOCTL_ERR
qmerrno.h, 2001
eQCWWAN_ERR_SWIDCS_START
qmerrno.h, 2001
eQCWWAN_ERR_SWIIM_CORRUPTED_FW_IMAGE
qmerrno.h, 2001
eQCWWAN_ERR_SWIIM_END
qmerrno.h, 2001
eQCWWAN_ERR_SWIIM_FILE_NOT_FOUND
qmerrno.h, 2001

eQCWWAN_ERR_SWIIM_FIRMWARE_NOT_DOWNLOADED
qmerrno.h, 2001
eQCWWAN_ERR_SWIIM_FW_ENTER_DOWNLOAD_MODE
qmerrno.h, 2001
eQCWWAN_ERR_SWIIM_FW_FLASH_COMPLETE
qmerrno.h, 2001
eQCWWAN_ERR_SWIIM_FW_INVALID_SLOT_INDEX
qmerrno.h, 2001
eQCWWAN_ERR_SWIIM_FW_PREFERENCE_MISMATCH
qmerrno.h, 2001
eQCWWAN_ERR_SWIIM_FW_SAME_AS_CURRENT_ACTIVE_IMAGE
qmerrno.h, 2001
eQCWWAN_ERR_SWIIM_FW_UPDATE_FAIL
qmerrno.h, 2001
eQCWWAN_ERR_SWIIM_FW_UPDATE_SUCCESS
qmerrno.h, 2001
eQCWWAN_ERR_SWIIM_FW_WAIT_FOR_REBOOT
qmerrno.h, 2001
eQCWWAN_ERR_SWIIM_INVALID_CRASH_STATE
qmerrno.h, 2001
eQCWWAN_ERR_SWIIM_INVALID_PATH
qmerrno.h, 2001
eQCWWAN_ERR_SWIIM_OPENING_DIR
qmerrno.h, 2001
eQCWWAN_ERR_SWIIM_OPENING_FILE
qmerrno.h, 2001
eQCWWAN_ERR_SWIIM_START
qmerrno.h, 2001
eQCWWAN_ERR_SWISM_END
qmerrno.h, 2001
eQCWWAN_ERR_SWISMS_BEARER_DATA_NOT_FOUND
qmerrno.h, 2001
eQCWWAN_ERR_SWISMS_MSG_CORRUPTED
qmerrno.h, 2001
eQCWWAN_ERR_SWISMS_MSG_LEN_TOO_LONG
qmerrno.h, 2001
eQCWWAN_ERR_SWISMS_SMSC_NUM_CORRUPTED
qmerrno.h, 2001
eQCWWAN_ERR_SWISMS_START
qmerrno.h, 2001
eQMI_LOC_SESS_STATUS_FAILURE
loc.h, 1497
eQMI_LOC_SESS_STATUS_IN_PROGRESS
loc.h, 1497
eQMI_LOC_SESS_STATUS_SUCCESS
loc.h, 1497
eQMI_LOC_SESS_STATUS_TIMEOUT
loc.h, 1497
eQOS
common.h, 1434
eREQ
common.h, 1435
eRMS
common.h, 1434
eRSP
common.h, 1435
eSAR
common.h, 1434
eSMS
common.h, 1434
eSWIAUDIO
common.h, 1434
eSWILOC
common.h, 1434
eSWIM2MCMD
common.h, 1434
eSWIOMA
common.h, 1434
eSWIOMAEXT
common.h, 1434
eSYS_SRV_DOMAIN_CAMPED
qaGobiApiNas.h, 1764
eSYS_SRV_DOMAIN_CS_ONLY
qaGobiApiNas.h, 1764
eSYS_SRV_DOMAIN_CS_PS
qaGobiApiNas.h, 1764
eSYS_SRV_DOMAIN_NO_SRV
qaGobiApiNas.h, 1764
eSYS_SRV_DOMAIN_PS_ONLY
qaGobiApiNas.h, 1764
eSYS_SRV_DOMAIN_UNKNOWN
qaGobiApiNas.h, 1764
eSetServiceAutomaticTrackingDisable
qaGobiApiPds.h, 1798
eSetServiceAutomaticTrackingEnable
qaGobiApiPds.h, 1798
eTIMEOUT_10_S
common.h, 1434
eTIMEOUT_20_S
common.h, 1434
eTIMEOUT_2_S
common.h, 1434
eTIMEOUT_300_S
common.h, 1434
eTIMEOUT_30_S
common.h, 1434
eTIMEOUT_5_S
common.h, 1434
eTIMEOUT_60_S
common.h, 1434
eTIMEOUT_8_S
common.h, 1434
eTIMEOUT_DEFAULT
common.h, 1434
eTLV_3GPP_NETWORK_INFO
qaNasPerformNetworkScan.h, 1995
eTLV_CBK_ALPHA_IDENTIFIER
qaCbkCatEventReportInd.h, 1579
eTLV_CBK_DISPLAY_TEXT

qaCbkCatEventReportInd.h, 1579
eTLV_CBK_END_PROACTIVE_SESSION
qaCbkCatEventReportInd.h, 1579
eTLV_CBK_GET_IN_KEY
qaCbkCatEventReportInd.h, 1579
eTLV_CBK_GET_INPUT
qaCbkCatEventReportInd.h, 1579
eTLV_CBK_LANGUAGE_NOTIFICATION
qaCbkCatEventReportInd.h, 1579
eTLV_CBK_REFRESH
qaCbkCatEventReportInd.h, 1579
eTLV_CBK_SELECT_ITEM
qaCbkCatEventReportInd.h, 1579
eTLV_CBK_SETUP_EVENT_LIST
qaCbkCatEventReportInd.h, 1579
eTLV_CBK_SETUP_IDLE_MODE_TEXT
qaCbkCatEventReportInd.h, 1579
eTLV_CBK_SETUP_MENU
qaCbkCatEventReportInd.h, 1579
eTLV_END_PROACTIVE_SESSION_LENGTH
qaCbkCatEventReportInd.h, 1579
eTLV_IND_OMA_DM_CONFIG
qaCbkSwiOmaDmEventReportInd.h, 1580
eTLV_IND_OMA_DM_FOTA
qaCbkSwiOmaDmEventReportInd.h, 1580
eTLV_IND_OMA_DM_NOT
qaCbkSwiOmaDmEventReportInd.h, 1580
eTLV_REFRESH_LENGTH
qaCbkCatEventReportInd.h, 1579
eTLV_RF_BAND_INFO
qaNasGetRFBandInfo.h, 1993
eTLV_SETUP_EVENT_LIST_LENGTH
qaCbkCatEventReportInd.h, 1579
eTMD
common.h, 1434
eUIM
common.h, 1434
eVOICE
common.h, 1434
eWDS
common.h, 1434
eWDS_ERR_PROFILE_REG_3GPP2_ERR_INVALID_IDENT_FOR_PROFILE
qmerrno.h, 2002
eWDS_ERR_PROFILE_REG_3GPP_ACCESS_ERR
qmerrno.h, 2002
eWDS_ERR_PROFILE_REG_3GPP_CONTEXT_NOT_DEFINED
qmerrno.h, 2002
eWDS_ERR_PROFILE_REG_3GPP_ERR_OUT_OF_PROFILES
qmerrno.h, 2002
eWDS_ERR_PROFILE_REG_3GPP_INVAL_PROFILE_FAMILY
qmerrno.h, 2002
eWDS_ERR_PROFILE_REG_3GPP_READ_ONLY_FLAG_SET
qmerrno.h, 2002
eWDS_ERR_PROFILE_REG_3GPP_VALID_FLAG_NOT_SET
qmerrno.h, 2002
eWDS_ERR_PROFILE_REG_END
qmerrno.h, 2002
eWDS_ERR_PROFILE_REG_INVAL_PROFILE_FAMILY
qmerrno.h, 2002
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL
qmerrno.h, 2002
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_HNDL
qmerrno.h, 2002
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_IDENT
qmerrno.h, 2002
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_OP
qmerrno.h, 2002
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_PROFILE_NUM
qmerrno.h, 2002
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_PROFILE_TYPE
qmerrno.h, 2002
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_SUBS_ID
qmerrno.h, 2002
eWDS_ERR_PROFILE_REG_RESULT_ERR_LEN_INVALID
qmerrno.h, 2002
eWDS_ERR_PROFILE_REG_RESULT_ERR_LIB_NOT_INITED
qmerrno.h, 2002
eWDS_ERR_PROFILE_REG_RESULT_FAIL
qmerrno.h, 2002
eWDS_ERR_PROFILE_REG_RESULT_LIST_END
qmerrno.h, 2002
ECIOThresListLen
ECIOThresh, 238
nas_ECIOThresh, 442
ECIOThresh, 238
ECIOThresListLen, 238
pECIOThresList, 238
ECTCallState
ECTNum, 239
ECTNum, 239
ECTCallState, 239
number, 239
presentationInd, 239
eDevState
qaGobiApiCbk.h, 1596
eDevice
sGetDeviceSeriesResult, 874
eGetDeviceSeries
qaGobiApiFms.h, 1719
eGobiDeviceSeries
qaGobiApiFms.h, 1716

eGobiImageCarrier
qaGobiApiFms.h, 1716
eGobiImageGPS
qaGobiApiFms.h, 1717
eGobiImageRegion
qaGobiApiFms.h, 1717
eGobiImageTech
qaGobiApiFms.h, 1717
eLOG_LEVEL
common.h, 1433
EMTlv
NASQmiCbkNasSystemSelPrefInd, 567
eQCWWANError
qmerrno.h, 1997
eQMI_SVC
common.h, 1434
eQMISARRFState
qaGobiApiSar.h, 1820
eQaQMIService
qaGobiApiCbk.h, 1635
ERI_DATA_MAX_SIZE
dms.h, 1443
ERIFileparams, 240
pFile, 241
pFileSize, 241
qaGobiApiDms.h, 1687
eSMSEventType
qaGobiApiCbk.h, 1596
ESNString
unpack_dms_GetDeviceSerialNumbers_t, 1044
eSYS_SRV_DOMAIN
qaGobiApiNas.h, 1764
ETWSPLMNInfo
eTWSPLMNInfoTlv, 242
eTWSPLMNInfoTlv, 242
ETWSPLMNInfo, 242
TlvPresent, 242
ETWSPLMNTlv
unpack_sms_SetNewSMSCallback_ind_t, 1198
ETWSTlv
unpack_sms_SetNewSMSCallback_ind_t, 1198
eTimeout
common.h, 1434
EVENT_MASK_CARD
qaGobiApiCbk.h, 1594
eValid
LibPackTFTIDParams, 362
TFTIDParams, 963
EarMute
GetAudioProfileResp, 261
GetM2MAudioProfileResp, 278
GetM2MAVMuteResp, 280
SetAudioProfileReq, 853
SetM2MAVMuteReq, 864
earfcn
infoInterFreq, 330
LTEInfoIntrafreq, 400
ltePCI, 403
nas_infoInterFreq, 457
nas_LTEInfoIntrafreq, 465
nas_ltePCI, 467
nas_umtsLTENbrCell, 527
umtsLTENbrCell, 1025
earfcn0
lteEARFCN, 394
nas_lteEARFCN, 459
earfcn1
lteEARFCN, 395
nas_lteEARFCN, 459
ecio
CDMASSInfo, 173
cdmaSSInfo, 173
ecioListElement, 238
HDRSSInfo, 306
hdrSSInfo, 307
nas_ecioListElement, 441
nas_rxInfo, 505
nas_UMTSInfo, 525
rxInfo, 831
TDSCDMASigInfoExt, 958
tdscdmaSigInfoExt, 959
UMTSInfo, 1023
ecioDelta
nas_SLQSSignalStrengthsIndReq, 513
SLQSSignalStrengthsIndReq, 894
ecioInfo
nas_SLQSSignalStrengthsInformation, 514
SLQSSignalStrengthsInformation, 895
ecioList
slqsSignalStrengthInfo, 892
unpack_nas_SLQSGetSignalStrength_t, 1132
ecioListElement, 237
ecio, 238
radioIf, 238
ecioListLen
slqsSignalStrengthInfo, 892
unpack_nas_SLQSGetSignalStrength_t, 1132
ecioThresholdList
nas_SLQSSignalStrengthsIndReq, 513
SLQSSignalStrengthsIndReq, 894
ecioThresholdListLen
nas_SLQSSignalStrengthsIndReq, 513
SLQSSignalStrengthsIndReq, 894
egprsSupp
GSMSysInfo, 298
nas_GSMSysInfo, 451
egprsSuppValid
GSMSysInfo, 298
nas_GSMSysInfo, 451
elevation
loc_satelliteInfo, 376
satelliteInfo, 838
EmerMode
NASEmergencyModeTlv, 537
emmConnState
LTEInfo, 398

nas_LTEInfo, 462
emmState
LTEInfo, 398
nas_LTEInfo, 462
emmSubState
LTEInfo, 398
nas_LTEInfo, 462
Enable
SetM2MAudioLPBKReq, 862
enable
pack_pds_SetPDSState_t, 656
pack_qos_SLQSSetQosEventCallback_t, 666
enabled
pack_rms_SetSMSWake_t, 667
unpack_rms_GetSMSWake_t, 1194
unpack_wds_GetMobileIPProfile_t, 1262
EnabledStatus
unpack_pds_SetPdsState_Ind_t, 1168
EncryptProt
nas_protocolSubtypeElement, 496
protocolSubtypeElement, 773
EncryptedPIN1
pack_uim_ChangePin_t, 682
pack_uim_SetPinProtection_t, 684
pack_uim_UnblockPin_t, 692
encryptedPIN1, 239
pin1Len, 240
pin1Val, 240
EndProactiveSession
CatEndProactiveSessionTlv, 159
EngineState
GPSStateInfo, 292
unpack_pds_SLQSGetGPSStateInfo_t, 1172
engineState
QmiCbkLocEngineStateInd, 786
unpack_loc_EngineState_Ind_t, 1091
entries
t_Sv, 956
eqmiCbkSetStatus
sms.h, 2017
eriData
eriDataparams, 240
eriDataLen
eriDataparams, 240
eriDataparams, 240
eriData, 240
eriDataLen, 240
eriFile
unpack_dms_SLQSGetERIFile_t, 1068
error
unpack_wds_GetLastMobileIPError_t, 1260
errorClass
SMSAsyncRawSend_s, 912
errorRate
errorRateListElement, 242
nas_errorRateListElement, 443
errorRateInfo
nas_SLQSSignalStrengthsInformation, 514
SLQSSignalStrengthsInformation, 896
errorRateList
slqsSignalStrengthInfo, 892
unpack_nas_SLQSGetSignalStrength_t, 1132
errorRateListElement, 241
errorRate, 242
radioIf, 242
errorRateListLen
slqsSignalStrengthInfo, 892
unpack_nas_SLQSGetSignalStrength_t, 1132
errorState
slotInf, 879
slotInfo, 881
uim_slotInfo, 998
esn
unpack_dms_GetSerialNumbers_t, 1055
esnSize
serialNumbersInfo, 844
unpack_dms_GetDeviceSerialNumbers_t, 1044
EspSpi
unpack_qos_swiQosFilter_t, 1186
EtwsMessageInfo
sMSEtwsMessageTlv, 916
event
unpack_qos_SLQSSetQosPriEventCallback_ind_t, 1182
unpack_qos_SLQSSetQosStatusCallback_ind_t, 1184
event_Index
QmiCbkCatEventStatusReportInd, 779
EventID
CatCommonEventTlv, 158
EventLength
CatCommonEventTlv, 158
eventMask
CATEventDataType, 159
pack_uim_SLQSUIMEventRegister_t, 686
UIMEventRegisterReqResp, 1004
unpack_uim_SLQSUIMEventRegister_t, 1217
eventRegister
LOCEventRegisterReqResp, 384
pack_loc_EventRegister_t, 616
eventType
unpack_swioma_SLQSOMADMAlertCallback_ind_t, 1207
evrcCapability
prefVoiceSO, 760
voice_prefVoiceSO, 1329
executingImage
FMSImageIDEntries, 251
ImageIDEntries, 313
exponent
pktErrRate, 753
unpack_qos_pktErrRate_t, 1175
extBit
calledPartySubAdd, 146
voice_calledPartySubAdd, 1308
extDispInfo

extDispRecInfo, 243
extDispInfoLen
extDispRecInfo, 243
extDispRecInfo, 242
dispType, 243
extDispInfo, 243
extDispInfoLen, 243
ExtErrorCode
PackCreateProfileOut, 738
extPowerState
LOCExtPowerStateReqResp, 384
pack_loc_SetExtPowerState_t, 616
extendedErrorCode
unpack_wds_SLQSDeleteProfile_t, 1268

FIRST_INSTANCE
qaGobiApiCbk.h, 1594
FLOAT
SwiDataTypes.h, 2035
FMSImageElement, 249
buildId, 249
buildIdLength, 250
imageId, 250
imageType, 250
FMSImageIDEntries, 251
executingImage, 251
imageIDElement, 251
imageIDSize, 251
imageType, 251
maxImages, 251
FMSImageIdElement, 250
buildID, 250
buildIDLength, 250
failureCount, 250
imageID, 250
storageIndex, 250
FMSImageList, 251
imageIDEntries, 252
listSize, 252
FMSPrefImageList, 252
listEntries, 252
listSize, 252
FORBIDDEN_INDEX
qaNasPerformNetworkScan.h, 1994
FOTAUpdate
_SLQSOMADMSettingsReqParams, 94
_SLQSOMADMSettingsReqParams3, 95
pack_swioma_SLQSOMADMSetSettings_t, 681
unpack_swioma_SLQSOMADMGetSettings_t, 1211
FOTAdownload
_SLQSOMADMSettingsReqParams, 93
_SLQSOMADMSettingsReqParams3, 94
pack_swioma_SLQSOMADMSetSettings_t, 681
unpack_swioma_SLQSOMADMGetSettings_t, 1211
FSNumber
FactorySequenceNumber, 243
facility
pack_dms_UIMGetControlKeyStatus_t, 608
pack_dms_UIMSetControlKeyProtection_t, 609
pack_dms_UIMUnblockControlKey_t, 610
facilityCk
pack_dms_UIMSetControlKeyProtection_t, 609
pack_dms_UIMUnblockControlKey_t, 610
facilityState
pack_dms_UIMSetControlKeyProtection_t, 609
unpack_dms_UIMGetControlKeyStatus_t, 1078
FactorySequenceNumber, 243
FSNumber, 243
failureCount
FMSImageIdElement, 250
ImageIdElement, 313
failureReason
ssdatasession_params, 931
failureReasonv4
ssdatasession_params, 932
failureReasonv6
ssdatasession_params, 932
family
pack_wds_GetDefaultProfileNum_t, 713
pack_wds_SetDefaultProfileNum_t, 719
feature
depersonalizationInformation, 222
personalizationStatus, 746
uim_depersonalizationInformation, 984
uim_personalizationStatus, 990
fileAttributes, 243
fileID, 246
fileSize, 246
fileType, 246
rawLen, 246
rawValue, 246
recordCount, 246
recordSize, 246
secActivate, 246
secActivateMask, 246
secDeactivate, 246
secDeactivateMask, 246
secIncrease, 247
secIncreaseMask, 247
secRead, 247
secReadMask, 247
secWrite, 247
secWriteMask, 247
fileID
fileAttributes, 246
fileInfo, 247
uim_fileAttributes, 988
uim_fileInfo, 989
fileIndex
pack_uim_ReadTransparent_t, 683
pack_uim_SLQSUIMGetFileAttributes_t, 687
UIMGetFileAttributesReq, 1006
UIMReadTransparentReq, 1010
fileInfo, 247
fileID, 247

path, 247
pathLen, 247
fileSize
fileAttributes, 246
uim_fileAttributes, 988
fileType
fileAttributes, 246
uim_fileAttributes, 988
fill_pack_ctx
common.h, 1435
fill_sdu_hdr
common.h, 1435
filterId
LibPackTFTIDParams, 362
TFTIDParams, 963
Firmware Management Service (FMS), 46
FirmwareID
fwinfo_s, 253
FirmwareUpdatStat, 247
pImgType, 249
pLogString, 249
pLogStringLen, 249
pRefData, 249
pRefString, 249
pRefStringLen, 249
ResCode, 249
fix_rate
pack_swiloc_SwiLocSetAutoStart_t, 678
SwiLocGetAutoStartResp, 938
SwiLocSetAutoStartReq, 939
unpack_swiloc_SwiLocGetAutoStart_t, 1206
fix_rate_reported
SwiLocGetAutoStartResp, 938
unpack_swiloc_SwiLocGetAutoStart_t, 1206
fix_type
pack_swiloc_SwiLocSetAutoStart_t, 678
SwiLocGetAutoStartResp, 938
SwiLocSetAutoStartReq, 940
unpack_swiloc_SwiLocGetAutoStart_t, 1206
fix_type_reported
SwiLocGetAutoStartResp, 938
unpack_swiloc_SwiLocGetAutoStart_t, 1206
fixAccuracy
pack_pds_StartPDSTrackingSessionExt_t, 664
fixCount
pack_pds_StartPDSTrackingSessionExt_t, 664
fixInterval
pack_pds_StartPDSTrackingSessionExt_t, 664
fixTimeout
pack_pds_StartPDSTrackingSessionExt_t, 664
flags
sensorData, 841
sensorData_t, 842
flowLabel
LibPackTFTIDParams, 362
TFTIDParams, 964
fms.h, 1488
GetValidFwPriCombinations, 1489
pack_fms_GetImagesPreference, 1490
pack_fms_GetStoredImages, 1490
pack_fms_SetImagesPreference, 1490
unpack_fms_GetImagesPreference, 1490
unpack_fms_GetStoredImages, 1491
unpack_fms_SetImagesPreference, 1491
Forbidden
nas_QmiNas3GppNetworkInfo, 498
SlqsNas3GppNetworkInfo, 885
forceFlag
pack_pds_SLQSPDSInjectAbsoluteTimeReference-_t, 659
ForceRev0
unpack_nas_GetCDMANetworkParameters_t, 1106
ForceXTRADownload
qaGobiApiPds.h, 1799
format
SMSTransferRouteMTMessage, 926
sMSTransferRouteMTMessageInfo, 926
ForwardMac
nas_protocolSubtypeElement, 496
protocolSubtypeElement, 773
fqdnAddr
PCSCFFQDNAddress, 740
wds_PCSCFFQDNAddress, 1405
fqdnLen
PCSCFFQDNAddress, 740
wds_PCSCFFQDNAddress, 1405
freeSlots
sms_maxStorageSizeResp, 903
smsMaxStorageSizeResp, 919
freq
nas_PhyCaAggPcellInfo, 487
nas_PhyCaAggScellIndType, 488
nas_PhyCaAggScellInfo, 491
NASPhyCaAggPcellInfo, 558
NASPhyCaAggScellIndType, 560
NASPhyCaAggScellInfo, 561
PhyCaAggPcellInfo, 747
PhyCaAggScellIndType, 749
PhyCaAggScellInfo, 752
freqsLen
LTEInfoInterfreq, 398
LTEInfoNeighboringGSM, 401
LTEInfoNeighboringWCDMA, 402
nas_LTEInfoInterfreq, 463
nas_LTEInfoNeighboringGSM, 466
nas_LTEInfoNeighboringWCDMA, 467
fromServiceId
BroadcastConfig, 142
sms_BroadcastConfig, 898
fumoResultCode
omaDmFotaTlvExt, 594
function
pack_swiloc_SwiLocSetAutoStart_t, 678
SwiLocGetAutoStartResp, 938
SwiLocSetAutoStartReq, 940

unpack_swiloc_SwiLocGetAutoStart_t, 1206
function_reported
SwiLocGetAutoStartResp, 938
unpack_swiloc_SwiLocGetAutoStart_t, 1206
FwAutoCheck
unpack_swioma_SLQSOMADMGetSettings_t, 1211
FwAvailability
unpack_swioma_SLQSOMADMStartSession_t, 1212
fwdloadsize
omaDmFotaTlv, 591
unpack_omaDmFotaTlv_t, 1160
fwinfo_s, 252
Carrier, 253
FirmwareID, 253
GPSCapability, 253
Region, 253
Technology, 253
fwloadComplete
omaDmFotaTlv, 592
unpack_omaDmFotaTlv_t, 1160
fwvers
CurrentImgList, 201
unpack_dms_SLQSSwiGetFirmwareCurr_t, 1071

g
qmifwinfo_s, 802
gDlBitRate
LibPackQosClassID, 360
QosClassID, 812
GERANInfo, 253
arfcn, 254
bsic, 254
cellID, 255
insNmrCellInfo, 255
lac, 255
nmrInst, 255
plmn, 255
rxLev, 255
timingAdvance, 255
GPRSGrantedQoS
unpack_wds_SLQSGetRuntimeSettings_t, 1278
GPRSQoS, 287
delayClass, 288
meanThroughputClass, 288
peakThroughputClass, 288
precedenceClass, 288
reliabilityClass, 288
GPRSRequestedQoS, 288
delayClass, 289
meanThroughputClass, 289
peakThroughputClass, 289
precedenceClass, 289
reliabilityClass, 289
GPSCapability
fwinfo_s, 253
GPSLPM
pack_dms_SetCustFeature_t, 601
unpack_dms_GetCustFeature_t, 1038
GPSSel
pack_dms_SetCustFeature_t, 601
unpack_dms_GetCustFeature_t, 1038
GPSStateInfo, 289
Altitude, 292
EngineState, 292
glo_almanac_sv_msk, 292
glo_ephemeris_sv_msk, 292
glo_health_sv_msk, 292
glo_visible_sv_msk, 292
gps_almanac_sv_msk, 292
gps_ephemeris_sv_msk, 292
gps_health_sv_msk, 292
gps_visible_sv_msk, 292
HorizontalUncertainty, 292
Iono_valid, 292
Latitude, 292
Longitude, 292
sbas_almanac_sv_msk, 292
sbas_ephemeris_sv_msk, 292
sbas_health_sv_msk, 292
sbas_visible_sv_msk, 293
Time_uncert_ms, 293
TimeStmp_gps_week, 293
TimeStmp_tow_ms, 293
ValidMask, 293
VerticalUncertainty, 293
xtra_start_gps_minutes, 293
xtra_start_gps_week, 293
xtra_valid_duration_hours, 293
GSMRSSIThresh, 295
GSMRSSIThreshListLen, 295
pGSMRSSIThreshList, 295
GSMRSSIThreshListLen
GSMRSSIThresh, 295
nas_GSMRSSIThresh, 447
GSMSSInfo
unpack_nas_SLQSNasGetSigInfo_t, 1144
GSMSrvStatusInfo, 295
isPrefDataPath, 296
srvStatus, 296
trueSrvStatus, 296
GSMSysInfo, 296
cellId, 298
cellIdValid, 298
dtmSupp, 298
dtmSuppValid, 298
egprsSupp, 298
egprsSuppValid, 298
lac, 298
lacValid, 299
MCC, 299
MNC, 299
networkIdValid, 299
regRejectInfoValid, 299
rejCause, 299
rejectSrvDomain, 299

sysInfoGSM, 299
gUlBitRate
LibPackQosClassID, 360
QosClassID, 812
GWAOPTlv
NASQmiCbkNasSystemSelPrefInd, 567
GWAcqOrderPref
NASGWAcqOrderPrefTlv, 546
GWAddressV4
unpack_wds_SLQSGetRuntimeSettings_t, 1278
gcDumpStrLen
CrashInfo, 192
gcdumpString
crashInformation, 194
gcdumpStrlen
crashInformation, 194
Generator
GetAudioProfileReq, 260
GetAudioVolTLBConfigReq, 262
GetM2MAudioProfileResp, 278
GetM2MAudioVolumeReq, 279
SetAudioProfileReq, 853
SetAudioVolTLBConfigReq, 854
SetM2MAudioVolumeReq, 864
geoSysIdx
AddCDMASysInfo, 116
AddSysInfo, 116
nas_AddCDMASysInfo, 421
nas_AddSysInfo, 422
geranArfcn
geranInstInfo, 255
nas_geranInstInfo, 445
geranBsicBcc
geranInstInfo, 255
nas_geranInstInfo, 445
geranBsicNcc
geranInstInfo, 255
nas_geranInstInfo, 446
geranInst
nas_UMTSInfo, 525
UMTSInfo, 1023
GeranInstInfo
nas_UMTSInfo, 525
UMTSInfo, 1023
geranInstInfo, 255
geranArfcn, 255
geranBsicBcc, 255
geranBsicNcc, 255
geranRssi, 255
geranRssi
geranInstInfo, 255
nas_geranInstInfo, 446
get_version
common.h, 1435
GetACCOLC
qaGobiApiNas.h, 1765
GetANAAAAuthenticationStatus
qaGobiApiNas.h, 1765
GetActivationState
qaGobiApiDms.h, 1691
getAllCallInfo
arrCallInfo, 131
voice_arrCallInfo, 1303
getAllCallInformation, 256
ALS, 256
Callinfo, 256
isEmpty, 256
GetAllCallRmtPtyName
arrRemotePartyName, 133
voice_arrRemotePartyName, 1305
getAllCallRmtPtyName, 256
callID, 257
RemotePartyName, 257
getAllCallRmtPtyNum, 257
callID, 257
RemotePartyNum, 257
GetAudioPathConfigReq, 257
Item, 258
Profile, 258
GetAudioPathConfigResp, 258
pCodecSTGain, 259
pDTMFTXGain, 260
pECMode, 260
pMICGainSelect, 260
pNSEnable, 260
pRXAGCList, 260
pRXAVCAGCSwitch, 260
pRXAVCList, 260
pRXPCMIIRFltr, 260
pTXAGCList, 260
pTXAVCSwitch, 260
pTXGain, 260
pTXPCMIIRFltr, 260
GetAudioProfileReq, 260
Generator, 260
GetAudioProfileResp, 260
EarMute, 261
MicMute, 261
Profile, 261
Volume, 261
GetAudioVolTLBConfigReq, 262
Generator, 262
Item, 262
Profile, 262
Volume, 262
GetAudioVolTLBConfigResp, 262
ResCode, 263
GetAutoconnect
qaGobiApiWds.h, 1957
GetByteTotals
qaGobiApiWds.h, 1957
GetCDMANetworkParameters
qaGobiApiNas.h, 1766
getCallFWExtInfo, 263
CallFWExtInfo, 263
numInstances, 263

getCallFWInfo, 263
CallFWInfo, 264
numInstances, 264
GetConnectionRate
qaGobiApiWds.h, 1958
GetCustomFeatureV2
unpack_dms_GetCustFeaturesV2_t, 1039
getCustomFeatureV2, 264
pCustSettingInfo, 264
pCustSettingList, 264
pGetCustomInput, 264
getCustomInput, 265
cust_id, 265
list_type, 265
getDUNCallInfoReq, 265
Mask, 266
pReportChannelRate, 266
pReportConnStatus, 266
pReportDataBearerTech, 266
pReportDormStatus, 266
pTransferStatInd, 266
getDUNCallInfoResp, 267
pCallEndReason, 269
pChannelRate, 269
pConnectionStatus, 269
pDataBearerTech, 269
pDormancyStatus, 269
pLastCallDataBearerTech, 269
pLastCallRXOKBytesCnt, 269
pLastCallTXOKBytesCnt, 269
pMdmCallDurationActive, 269
pRXOKBytesCount, 269
pTXOKBytesCount, 269
GetDataBearerTechnology
qaGobiApiWds.h, 1959
GetDefaultProfile
qaGobiApiWds.h, 1959
GetDefaultProfileLTE
qaGobiApiWds.h, 1961
GetDefaultProfileNum
qaGobiApiWds.h, 1963
GetDeviceCapabilities
qaGobiApiDms.h, 1691
GetDormancyState
qaGobiApiWds.h, 1963
getDyingGaspCfg, 269
pDestSMSContent, 270
pDestSMSNum, 270
getDyingGaspStatistics, 270
pSMSAttemptedFlag, 270
pTimeStamp, 271
GetErrRateResp, 271
pCDMAFrameErrRate, 271
pGSMBER, 271
pHDRPackErrRate, 271
pWCDMABER, 271
GetFirmwareRevision
qaGobiApiDms.h, 1692
GetFirmwareRevisions
qaGobiApiDms.h, 1693
GetHRPDStatsResp, 272
pDRCParams, 272
pPilotSetData, 272
pUATI, 272
GetHardwareRevision
qaGobiApiDms.h, 1693
GetHomeNetwork
qaGobiApiNas.h, 1767
GetHomeNetwork3GPP2
qaGobiApiNas.h, 1768
GetIMSSMSConfigParams, 272
pPhoneCtxtURI, 273
pPhoneCtxtURILen, 273
pSMSFormat, 273
pSMSOverIPNwInd, 273
pSettingResp, 273
GetIMSUserConfigParams, 273
pIMSDomain, 274
pIMSDomainLen, 274
pSettingResp, 274
GetIMSVoIPConfigResp, 274
pAmrMode, 276
pAmrOctetAligned, 276
pAmrWBMode, 276
pAmrWBOctetAligned, 276
pAmrWbEnable, 276
pMinSessionExpiryTimer, 276
pRTPRTCPInactTimer, 276
pRingBackTimer, 276
pRingingTimer, 276
pScrAmrEnable, 276
pScrAmrWbEnable, 276
pSessionExpiryTimer, 276
pSettingResp, 276
GetIPAddressLTE
qaGobiApiWds.h, 1964
GetImageStore
qaGobiApiFms.h, 1720
GetImagesPreference
qaGobiApiFms.h, 1719
getIndicationRegResp
qaGobiApiSms.h, 1825
GetInstIDResp, 276
pIPFamily, 276
pInstanceID, 276
GetLastMobileIPError
qaGobiApiWds.h, 1964
GetM2MAVMuteReq, 279
Profile, 279
GetM2MAVMuteResp, 280
CwtMute, 280
EarMute, 280
MicMute, 280
GetM2MAudioProfileReq, 277
pGenerator, 277
GetM2MAudioProfileResp, 277

CwtMute, 278
EarMute, 278
Generator, 278
MicMute, 278
Profile, 278
Volume, 278
GetM2MAudioVolumeReq, 278
Generator, 279
Profile, 279
GetM2MAudioVolumeResp, 279
Level, 279
GetM2MSpkrGainReq, 280
Profile, 281
GetM2MSpkrGainResp, 281
Value, 281
GetManufacturer
qaGobiApiDms.h, 1694
GetMobileIP
qaGobiApiWds.h, 1965
GetMobileIPProfile
qaGobiApiWds.h, 1965
GetModelID
qaGobiApiDms.h, 1694
getMsgWaitingInfo, 281
msgWaitInfo, 282
numInstances, 282
GetNetworkPreference
qaGobiApiNas.h, 1770
GetNetworkTime
qaGobiApiDms.h, 1695
GetNetworkTimeResp, 282
p3GPP2TimeInfo, 282
p3GPPTimeInfo, 282
GetOfflineReason
qaGobiApiDms.h, 1695
GetPDSDefaults
qaGobiApiPds.h, 1799
GetPDSState
qaGobiApiPds.h, 1800
GetPRLVersion
qaGobiApiDms.h, 1696
GetPacketStatistics
qaGobiApiWds.h, 1967
GetPacketStatus
qaGobiApiWds.h, 1967
GetPortAutomaticTracking
qaGobiApiPds.h, 1800
GetPower
qaGobiApiDms.h, 1696
GetProfileSettingIn
qaGobiApiWds.h, 1953
GetProfileSettingOut
qaGobiApiWds.h, 1954
GetQmapIsSupported
qaGobiApiDcs.h, 1674
GetRFInfo
qaGobiApiNas.h, 1771
GetRegMgrConfigParams, 282
pIMSTestMode, 283
pPCSCFPort, 283
pPriCSCFPortName, 283
pPriCSCFPortNameLen, 283
pSettingResp, 283
GetSIPConfigResp, 283
pSIPLocalPort, 284
pSettingResp, 284
pSigCompEnabled, 284
pSubscribeTimer, 284
pTimerSIPReg, 284
pTimerT1, 284
pTimerT2, 284
pTimerTf, 284
GetSMSCAddress
qaGobiApiSms.h, 1828
GetSMSWake
qaGobiApiRms.h, 1819
GetSerialNumbers
qaGobiApiDms.h, 1697
GetServiceAutomaticTracking
qaGobiApiPds.h, 1801
GetServingNetwork
qaGobiApiNas.h, 1771
GetServingNetworkCapabilities
qaGobiApiNas.h, 1772
GetSessionDuration
qaGobiApiWds.h, 1968
GetSessionIDResp, 283
pSessionIDv4, 283
pSessionIDv6, 283
GetSessionState
qaGobiApiWds.h, 1968
GetSignalStrengths
qaGobiApiNas.h, 1773
GetStoredImages
qaGobiApiFms.h, 1720
getTransLayerInfoResp
qaGobiApiSms.h, 1825
getTransNWRegInfoResp
qaGobiApiSms.h, 1826
GetValidFwPriCombinations
fms.h, 1489
GetVoiceNumber
qaGobiApiDms.h, 1698
GetXTRAAutomaticDownload
qaGobiApiPds.h, 1801
GetXTRANetwork
qaGobiApiPds.h, 1802
GetXTRAValidity
qaGobiApiPds.h, 1802
glo_almanac_sv_msk
GPSStateInfo, 292
unpack_pds_SLQSGetGPSStateInfo_t, 1172
glo_ephemeris_sv_msk
GPSStateInfo, 292
unpack_pds_SLQSGetGPSStateInfo_t, 1172
glo_health_sv_msk

GPSStateInfo, 292
unpack_pds_SLQSGetGPSStateInfo_t, 1172
glo_visible_sv_msk
GPSStateInfo, 292
unpack_pds_SLQSGetGPSStateInfo_t, 1172
globalCellId
LTEInfoIntrafreq, 400
nas_LTEInfoIntrafreq, 465
glog
common.h, 1436
gloglvl
common.h, 1436
GnssData, 285
mask, 286
gnssSvId
loc_satelliteInfo, 376
satelliteInfo, 838
gnssSvInfoNotification, 286
bAltitudeAssumed, 287
pSatelliteInfo, 287
gnssSvUsedList
loc_svUsedforFix, 379
svUsedforFix_s, 935
gnssSvUsedList_len
loc_svUsedforFix, 379
svUsedforFix_s, 935
Gpp2TimeZone
qaQmiServingSystemParam, 779
unpack_nas_SLQSGetServingSystem_t, 1129
GppNetworkDSTAdjustment
qaQmiServingSystemParam, 779
unpack_nas_SLQSGetServingSystem_t, 1129
GppTimeZone
qaQmiServingSystemParam, 779
unpack_nas_SLQSGetServingSystem_t, 1129
gps_almanac_sv_msk
GPSStateInfo, 292
unpack_pds_SLQSGetGPSStateInfo_t, 1172
gps_ephemeris_sv_msk
GPSStateInfo, 292
unpack_pds_SLQSGetGPSStateInfo_t, 1172
gps_health_sv_msk
GPSStateInfo, 292
unpack_pds_SLQSGetGPSStateInfo_t, 1172
gps_visible_sv_msk
GPSStateInfo, 292
unpack_pds_SLQSGetGPSStateInfo_t, 1172
GpsEnable
custFeaturesInfo, 208
pack_dms_SetCustFeature_t, 601
unpack_dms_GetCustFeature_t, 1038
gpsTime
qaGobiApiCbk.h, 1597
gpsTime_s, 293
gpsTimeOfWeekMs, 293
gpsWeek, 293
gpsTimeOfWeekMs
gpsTime_s, 293
loc_gpsTime, 371
t_gpsTime, 955
gpsWeek
gpsTime_s, 293
loc_gpsTime, 371
t_gpsTime, 955
grntDownlinkBitrate
LibPackUMTSQoS, 364
UMTSMinQoS, 1027
UMTSQoS, 1030
wds_UMTSMinQoS, 1411
grntUplinkBitrate
LibPackUMTSQoS, 364
UMTSMinQoS, 1028
UMTSQoS, 1030
wds_UMTSMinQoS, 1411
gsmAmrStat
curAMRConfig, 199
voice_curAMRConfig, 1321
GsmCellInfo
lteGsmCellInfo, 396
nas_lteGsmCellInfo, 460
gsmCellInfo, 293
arfcn, 294
band1900, 294
bsicId, 294
cellIdValid, 294
rssi, 294
srxlev, 294
gsmUmtsDl
NasSwiIndReg, 572
pack_nas_SLQSNasSwiIndicationRegister_t, 644
gsmUmtsUl
NasSwiIndReg, 572
pack_nas_SLQSNasSwiIndicationRegister_t, 644
guaranteedRate
dataRate, 217
unpack_qos_dataRate_t, 1174
gwAddressV6
IPV6GWAddressInfo, 334
wds_IPV6GWAddressInfo, 1405
gwV6PrefixLen
IPV6GWAddressInfo, 334
wds_IPV6GWAddressInfo, 1405
gyroAcceptReady
qaGobiApiCbk.h, 1597
gyroAcceptReady_s, 299
batchPerSec, 299
injectEnable, 299
samplesPerBatch, 300
gyroData
pack_loc_SLQSLOCInjectSensorData_t, 624
gyroTemp
pack_loc_SLQSLOCInjectSensorData_t, 624
gyroTempAcceptReady
qaGobiApiCbk.h, 1597
gyroTempAcceptReady_s, 300
batchPerSec, 300

injectEnable, 300
samplesPerBatch, 300
gyroTimeSrc
pack_loc_SLQSLOCInjectSensorData_t, 625

HASPI
unpack_wds_GetMobileIPProfile_t, 1262
HAState
unpack_wds_GetMobileIPProfile_t, 1262
HDOP
loc_precisionDilution, 374
precisionDilution_s, 758
HDRECIOThresh, 300
HDRECIOThreshListLen, 301
pHDRECIOThreshList, 301
HDRECIOThreshListLen
HDRECIOThresh, 301
nas_HDRECIOThresh, 451
HDRIOThresh, 301
HDRIOThreshListLen, 301
pHDRIOThreshList, 301
HDRIOThreshListLen
HDRIOThresh, 301
nas_HDRIOThresh, 452
HDRPersonalityInd, 301
pCurrentPersonality, 302
pPersonalityListLength, 302
pProtocolSubtypeElement, 302
HDRPersonalityResp, 302
pCurrentPersonality, 302
pPersonalityListLength, 302
pProtocolSubtypeElement, 302
HDRProtSubtypResp, 302
pAppSubType, 303
pCurrentPrsnlty, 303
pPersonalityListLength, 303
pProtoSubTypElmnt, 303
HDRRSSIThresh, 303
HDRRSSIThreshListLen, 304
pHDRRSSIThreshList, 304
HDRRSSIThreshListLen
HDRRSSIThresh, 304
nas_HDRRSSIThresh, 452
HDRSINRThresListLen
HDRSINRThresh, 304
nas_HDRSINRThresh, 453
HDRSINRThresh, 304
HDRSINRThresListLen, 304
pHDRSINRThresList, 304
HDRSINRThreshListLen
HDRSINRThreshold, 305
nas_HDRSINRThreshold, 454
HDRSINRThreshold, 304
HDRSINRThreshListLen, 305
pHDRSINRThreshList, 305
HDRSSInfo, 305
ecio, 306
io, 306
rssi, 306
sinr, 306
unpack_nas_SLQSNasGetSigInfo_t, 1144
HDRSysInfo, 307
hdrActiveProt, 309
hdrActiveProtValid, 309
hdrPersonality, 309
hdrPersonalityValid, 309
is856SysId, 309
is856SysIdValid, 309
isSysPrlMatch, 309
isSysPrlMatchValid, 309
sysInfoHDR, 309
HTTPStatus
_SwiAvmsEventInd, 100
HWVersion
DeviceConfigDetail, 224
nas_DeviceConfigDetail, 440
HardwareControlledMode
unpack_dms_GetPower_t, 1053
has_PositionDataNMEA
unpack_pds_SetEventReport_Ind_t, 1167
has_SessionStatus
unpack_pds_SetEventReport_Ind_t, 1167
has_accelTemp
pack_loc_SLQSLOCInjectSensorData_t, 625
has_acceleroTimeSrc
pack_loc_SLQSLOCInjectSensorData_t, 625
has_accleroData
pack_loc_SLQSLOCInjectSensorData_t, 625
has_altitudeSrcInfo
pack_loc_SLQSLOCInjectPosition_t, 622
has_altitudeWrtEllipsoid
pack_loc_SLQSLOCInjectPosition_t, 622
has_altitudeWrtMeanSeaLevel
pack_loc_SLQSLOCInjectPosition_t, 622
has_confidence
pack_loc_SLQSLOCSetCradleMountConfig_t, 626
has_dLatitude
unpack_pds_SetEventReport_Ind_t, 1167
has_dLongitude
unpack_pds_SetEventReport_Ind_t, 1167
has_gyroData
pack_loc_SLQSLOCInjectSensorData_t, 625
has_gyroTemp
pack_loc_SLQSLOCInjectSensorData_t, 625
has_gyroTimeSrc
pack_loc_SLQSLOCInjectSensorData_t, 625
has_horConfidence
pack_loc_SLQSLOCInjectPosition_t, 622
has_horReliability
pack_loc_SLQSLOCInjectPosition_t, 622
has_horUncCircular
pack_loc_SLQSLOCInjectPosition_t, 622
has_latitude
pack_loc_SLQSLOCInjectPosition_t, 622
has_longitude
pack_loc_SLQSLOCInjectPosition_t, 622
has_opaqueId

pack_loc_SLQSLOCInjectSensorData_t, 625
has_posSrc
unpack_pds_SetEventReport_Ind_t, 1167
has_positionSrc
pack_loc_SLQSLOCInjectPosition_t, 622
has_rawHorConfidence
pack_loc_SLQSLOCInjectPosition_t, 622
has_rawHorUncCircular
pack_loc_SLQSLOCInjectPosition_t, 622
has_serverAddrTypeMask
pack_loc_SLQSLOCGetServer_t, 618
has_timestampAge
pack_loc_SLQSLOCInjectPosition_t, 622
has_timestampUtc
pack_loc_SLQSLOCInjectPosition_t, 622
has_vertConfidence
pack_loc_SLQSLOCInjectPosition_t, 622
has_vertRelicability
pack_loc_SLQSLOCInjectPosition_t, 622
has_vertUnc
pack_loc_SLQSLOCInjectPosition_t, 623
hdrActiveProt
HDRSysInfo, 309
nas_HDRSysInfo, 456
hdrActiveProtValid
HDRSysInfo, 309
nas_HDRSysInfo, 456
hdrHybrid
detailSvcInfo, 223
nas_detailSvcInfo, 439
HdrPersonality
unpack_nas_SLQSGetServingSystem_t, 1129
hdrPersonality
HDRSysInfo, 309
nas_HDRSysInfo, 456
NASServingSystemInfo, 569
qaQmiServingSystemParam, 779
ServingSystemInfo, 846
hdrPersonalityValid
HDRSysInfo, 309
nas_HDRSysInfo, 456
hdrSSInfo, 306
ecio, 307
io, 307
rssi, 307
sinr, 307
hdrSrvStatus
detailSvcInfo, 223
nas_detailSvcInfo, 439
healthStatus
loc_satelliteInfo, 376
satelliteInfo, 838
helper_get_resp_ctx
common.h, 1435
helper_get_xid
common.h, 1435
helper_set_log_func
common.h, 1435
helper_set_log_lvl
common.h, 1435
homeOrigVoiceSO
prefVoiceSO, 760
voice_prefVoiceSO, 1329
homePageVoiceSO
prefVoiceSO, 760
voice_prefVoiceSO, 1329
homeSIDNID, 309
numInstances, 310
SidNid, 310
horConfidence
pack_loc_SLQSLOCInjectPosition_t, 623
horReliability
pack_loc_SLQSLOCInjectPosition_t, 623
horUncCircular
pack_loc_SLQSLOCInjectPosition_t, 623
HorizontalUncertainty
GPSStateInfo, 292
unpack_pds_SLQSGetGPSStateInfo_t, 1172
hotSwap
hotSwapStatus, 310
uim_hotSwapStatus, 989
hotSwapLength
hotSwapStatus, 310
uim_hotSwapStatus, 989
hotSwapStatus, 310
hotSwap, 310
hotSwapLength, 310
hour
nas_timeInfo, 522
nas_UniversalTime, 528
timeInfo, 965
UniversalTime, 1033
hsCallStatus
nas_WCDMASysInfo, 534
WCDMASysInfo, 1394
hsCallStatusValid
nas_WCDMASysInfo, 534
WCDMASysInfo, 1394
hsInd
nas_WCDMASysInfo, 534
WCDMASysInfo, 1394
hsIndValid
nas_WCDMASysInfo, 534
WCDMASysInfo, 1394
hwType
wds_DHCPv4HWConfig, 1400
WdsDHCPv4HWConfig, 1417
wdsDhcpv4HwConfig, 1417
hwVer
unpack_dms_GetHardwareRevision_t, 1049

iFaceTab
PCMparams, 739
iFaceTabLen
PCMparams, 739
iGetByteTotals
qaGobiApiWds.h, 1969

iGetConnectionRate
qaGobiApiWds.h, 1969
iGetPacketStatistics
qaGobiApiWds.h, 1969
iLTEbandValue
nas_PhyCaAggPcellInfo, 487
nas_PhyCaAggScellInfo, 491
NASPhyCaAggPcellInfo, 558
NASPhyCaAggScellInfo, 561
PhyCaAggPcellInfo, 747
PhyCaAggScellInfo, 752
IMCNflag
unpack_wds_SLQSGetRuntimeSettings_t, 1278
IMEIString
unpack_dms_GetDeviceSerialNumbers_t, 1044
IMG_ID_LEN
qaGobiApiFms.h, 1715
IMGDETAILS_LEN
qaGobiApiFms.h, 1715
IMS Service (IMS), 55
IMSAIndRegisterInfo, 314
pPdpStatusConfig, 315
pRatHandoverStatusConfig, 315
pRegStatusConfig, 315
pServiceStatusConfig, 315
IMSARegistrationStatus, 317
pImsRegErrCode, 317
pImsRegStatus, 317
pNewImsRegStatus, 317
IMSAServiceStatus, 318
pSmsServiceRat, 320
pSmsServiceStatus, 320
pUtServiceRat, 320
pUtServiceStatus, 320
pVoipServiceRat, 320
pVoipServiceStatus, 320
pVsServiceRat, 320
pVsServiceStatus, 320
pVtServiceRat, 320
pVtServiceStatus, 320
IMSASupportedFieldsResp, 320
pIndFieldsList, 321
pReqFieldsList, 321
pRespFieldsList, 321
IMSASupportedMsgInfo, 321
pSupportedMsgList, 321
IMSI_M_S1_LENGTH
qaGobiApiNas.h, 1752
IMSI_M_S2_LENGTH
qaGobiApiNas.h, 1752
IMSInfo
sMSOnIMSTlv, 923
IMSTlv
unpack_sms_SetNewSMSCallback_ind_t, 1198
INDEX_ZERO
qaNasPerformNetworkScan.h, 1994
INT32
SwiDataTypes.h, 2035
INT8
SwiDataTypes.h, 2035
INVALID_INSTACNE
qaGobiApiCbk.h, 1594
IOThresListLen
IOThresh, 331
nas_IOThresh, 458
IOThresh, 330
IOThresListLen, 331
pIOThresList, 331
IPAddress
DataStatusDetail, 219
nas_DataStatusDetail, 437
IPAddressV6
IPV6AddressInfo, 333
ipv6AddressInfo, 333
wds_IPV6AddressInfo, 1404
IPFamSupport
pack_dms_SetCustFeature_t, 601
unpack_dms_GetCustFeature_t, 1038
IPFamilyPreference
pack_wds_SLQSSetIPFamilyPreference_t, 731
unpack_wds_SLQSGetRuntimeSettings_t, 1278
IPSECSPI
LibPackTFTIDParams, 362
TFTIDParams, 964
IPV4
qaGobiApiCbk.h, 1594
IPV4V6
qaGobiApiCbk.h, 1594
IPV6
qaGobiApiCbk.h, 1594
IPV6AddrInfo
unpack_wds_SLQSGetRuntimeSettings_t, 1278
IPV6AddressInfo, 333
IPAddressV6, 333
IPV6PrefixLen, 333
IPV6GWAddrInfo
unpack_wds_SLQSGetRuntimeSettings_t, 1278
IPV6GWAddressInfo, 334
gwAddressV6, 334
gwV6PrefixLen, 334
IPV6PrefixLen
IPV6AddressInfo, 333
ipv6AddressInfo, 333
wds_IPV6AddressInfo, 1404
IPv4
unpack_wds_SLQSGetRuntimeSettings_t, 1278
IPv4Addr, 331
addr, 331
subnetMask, 331
wds_IPv4AdTlv, 1404
IPv4AddrTlv
unpack_wds_DHCPv4ClientLease_ind_t, 1253
IPv4DstAddr
unpack_qos_swiQosFilter_t, 1186
IPv4Info, 331
address, 332

port, 332
TlvPresent, 332
IPv4SrcAddr
unpack_qos_swiQosFilter_t, 1186
IPv4Tos
unpack_qos_swiQosFilter_t, 1186
IPv6Addr, 332
addr, 332
prefixLen, 332
IPv6DstAddr
unpack_qos_swiQosFilter_t, 1186
IPv6Info, 334
address, 335
port, 335
TlvPresent, 335
IPv6Label
unpack_qos_swiQosFilter_t, 1187
IPv6SrcAddr
unpack_qos_swiQosFilter_t, 1187
IPv6TrafCls, 335
mask, 335
unpack_qos_swiQosFilter_t, 1187
val, 335
iSLQSMISetIPFamilyPreference
qaGobiApiWds.h, 1969
iSLQSSetDUNCallInfoCallback
qaGobiApiCbk.h, 1636
iSLQSSetSignalStrengthsCallback
qaGobiApiCbk.h, 1636
iSLQSSetWdsFirstInstEventCallback
qaGobiApiCbk.h, 1636
iSLQSSetWdsSecondInstEventCallback
qaGobiApiCbk.h, 1636
iSLQSSetWdsThirdInstEventCallback
qaGobiApiCbk.h, 1636
iSLQSSetWdsXferStatsFirstInstCallback
qaGobiApiCbk.h, 1636
iSLQSSetWdsXferStatsSecondInstCallback
qaGobiApiCbk.h, 1636
iSetCATEventCallback
qaGobiApiCbk.h, 1636
iSetSignalStrengthCallback
qaGobiApiCbk.h, 1636
Id
unpack_qos_swiQosFilter_t, 1186
id
BdsSV, 140
CSGID, 197
loc_BdsSV, 367
loc_SV, 378
nas_CSGID, 434
pack_dms_UIMChangePIN_t, 607
pack_dms_UIMSetPINProtection_t, 610
pack_dms_UIMUnblockPIN_t, 611
pack_dms_UIMVerifyPIN_t, 611
QosFlowInfoState, 815
SV, 934
swiQosModifyReq, 951
unpack_qos_QosFlowInfoState_t, 1178
unpack_qos_SLQSSetQosStatusCallback_ind_t, 1184
id_length
custSettingInfo, 211
DMScustSettingInfo, 230
IdleState
nas_protocolSubtypeElement, 496
protocolSubtypeElement, 773
image_info_t, 310
buildID, 311
buildIDLen, 311
imageType, 311
uniqueID, 311
ImageElement, 311
buildId, 312
buildIdLength, 312
imageId, 312
imageType, 312
imageID
FMSImageIdElement, 250
ImageIdElement, 313
imageIDElement
FMSImageIDEntries, 251
ImageIDEntries, 313
ImageIDEntries, 313
executingImage, 313
imageIDElement, 313
imageIDSize, 313
imageType, 313
maxImages, 313
imageIDEntries
FMSImageList, 252
ImageList, 314
imageIDSize
FMSImageIDEntries, 251
ImageIDEntries, 313
imageId
FMSImageElement, 250
ImageElement, 312
ImageIdElement, 312
buildID, 312
buildIDLength, 313
failureCount, 313
imageID, 313
storageIndex, 313
ImageList, 314
imageIDEntries, 314
listSize, 314
imageList
unpack_fms_GetStoredImages_t, 1083
ImageListSize
unpack_fms_GetImagesPreference_t, 1082
imageListSize
pack_fms_SetImagesPreference_t, 613
imageType
CurrImageInfo, 203
FMSImageElement, 250

FMSImageIDEntries, 251
image_info_t, 311
ImageElement, 312
ImageIDEntries, 313
ImageTypes
unpack_fms_SetImagesPreference_t, 1083
ImageTypesSize
unpack_fms_SetImagesPreference_t, 1083
imagelistSize
unpack_fms_GetStoredImages_t, 1083
imei_no
unpack_dms_GetSerialNumbers_t, 1055
imeiSize
serialNumbersInfo, 844
unpack_dms_GetDeviceSerialNumbers_t, 1044
imeiSvnSize
serialNumbersInfo, 844
unpack_dms_GetDeviceSerialNumbers_t, 1044
ImeiSvnString
unpack_dms_GetDeviceSerialNumbers_t, 1044
imeisv_svn
unpack_dms_GetSerialNumbers_t, 1055
imgType
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 1073
imsCfgIndRegisterInfo, 322
pRegMgrConfigEvents, 323
pSIPConfigEvents, 323
pSMSConfigEvents, 323
pUserConfigEvents, 323
pVoIPConfigEvents, 323
imsRegMgrConfigInfo, 323
pCSCFPortName, 324
pIMSTestMode, 324
pPriCSCFPort, 324
imsRegState
CommInfo, 188
nas_CommInfo, 433
imsSIPConfigInfo, 324
pSIPLocalPort, 325
pSigCompEnabled, 325
pSubscribeTimer, 325
pTimerSIPReg, 325
pTimerT1, 325
pTimerT2, 325
pTimerTf, 325
imsSMSConfigInfo, 325
pPhoneCtxtURI, 326
pSMSFormat, 326
pSMSOverIPNwInd, 326
imsUserConfigInfo, 326
pIMSDomain, 326
imsVoIPConfigInfo, 326
pAmrMode, 328
pAmrOctetAligned, 328
pAmrWBMode, 328
pAmrWBOctetAligned, 328
pAmrWbEnable, 328
pMinSessionExpiryTimer, 328
pRTPRTCPInactTimer, 329
pRingBackTimer, 328
pRingingTimer, 329
pScrAmrEnable, 329
pScrAmrWbEnable, 329
pSessionExpiryTimer, 329
imsaPdpStatusInfo, 315
connetionState, 316
pFailErrorCode, 316
imsaRatStatusInfo, 316
pErrorCodeStr, 316
pRATStatus, 316
pSrcRAT, 316
pTgtRAT, 316
imsaRegStatusInfo, 317
pImsRegStatus, 318
pRegStatusErrorCode, 318
pbIMSRegistered, 318
imsaSvcStatusInfo, 321
pSMSSvcRAT, 322
pSMSSvcStatus, 322
pUTSvcRAT, 322
pUTSvcStatus, 322
pVOIPSvcRAT, 322
pVOIPSvcStatus, 322
pVTSvcRAT, 322
pVTSvcStatus, 322
imsi
unpack_dms_GetIMSI_t, 1049
imsi_11_12
CDMASysInfoExt, 178
nas_CDMASysInfoExt, 430
imsiM1112
minBasedIMSI, 416
nas_minBasedIMSI, 477
imsiMS1
minBasedIMSI, 416
nas_minBasedIMSI, 477
imsiMS2
minBasedIMSI, 416
nas_minBasedIMSI, 477
imsiT1112
nas_trueIMSI, 523
trueIMSI, 973
imsiTS1
nas_trueIMSI, 523
trueIMSI, 973
imsiTS2
nas_trueIMSI, 523
trueIMSI, 973
imsiTaddrNum
nas_trueIMSI, 523
trueIMSI, 973
InUse
nas_QmiNas3GppNetworkInfo, 498
SlqsNas3GppNetworkInfo, 885
includes_pcs_digit

nas_QmisNasPcsDigit, 499
SlqsNasPcsDigit, 886
IndFieldsList, 329
indicationFields, 329
indicationFieldsLen, 329
index
pack_wds_GetMobileIPProfile_t, 715
pack_wds_SetDefaultProfileNum_t, 719
pack_wds_SetMobileIPProfile_t, 722
swiQosFilter, 946
swiQosFlow, 949
swiQosReq, 952
unpack_qos_swiQosFilter_t, 1186
unpack_qos_swiQosFlow_t, 1190
unpack_wds_GetDefaultProfileNum_t, 1259
index1xPri
cardStatus, 156
uim_cardStatus, 982
index1xSec
cardStatus, 156
uim_cardStatus, 982
indexGwPri
cardStatus, 156
uim_cardStatus, 982
indexGwSec
cardStatus, 156
uim_cardStatus, 982
indicationFields
IndFieldsList, 329
indicationFieldsLen
IndFieldsList, 329
Info
unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t, 1149
unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t, 1150
infoInterFreq, 329
cell_resel_priority, 330
cellInterFreqParams, 330
cells_len, 330
earfcn, 330
threshXHigh, 330
threshXLow, 330
InfoInterfreq
LTEInfoInterfreq, 398
nas_LTEInfoInterfreq, 463
InitiateDomainAttach
qaGobiApiNas.h, 1774
InitiateNetworkRegistration
qaGobiApiNas.h, 1774
injectEnable
accelAcceptReady_s, 113
accelTempAcceptReady_s, 114
gyroAcceptReady_s, 299
gyroTempAcceptReady_s, 300
injectSensorDataStatus
QmiCbkLocInjectSensorDataInd, 790
injectTimeSyncStatus
QmiCbkLocInjectTimeInd, 791
insNmrCellInfo
GERANInfo, 255
nas_GERANInfo, 445
instanceId
ssdatasession_params, 932
instancesSize
unpack_nas_GetRFInfo_t, 1112
interval
pack_pds_SetPDSDefaults_t, 656
pack_pds_SetXTRAAutomaticDownload_t, 658
pack_wds_SLQSSetWdsEventCallback_t, 732
TransferStatsDataType, 971
Io
slqsSignalStrengthInfo, 892
unpack_nas_SLQSGetSignalStrength_t, 1132
io
HDRSSInfo, 306
hdrSSInfo, 307
nas_SLQSSignalStrengthsInformation, 514
SLQSSignalStrengthsInformation, 896
ioDelta
nas_SLQSSignalStrengthsIndReq, 513
SLQSSignalStrengthsIndReq, 894
Iono_valid
GPSStateInfo, 292
unpack_pds_SLQSGetGPSStateInfo_t, 1172
ip
WdsIpAddressInfoReq, 1422
ipAddress
pack_wds_SetDefaultProfile_t, 719
qmuxIPTable, 811
ipFamily
_packetSrvStatus, 70
unpack_wds_SLQSSetPacketSrvStatusCallback_t, 1281
ipVersion
LibPackTFTIDParams, 362
TFTIDParams, 964
ipaddr
unpack_wds_GetDefaultProfile_t, 1258
ipaddrv6
unpack_wds_GetDefaultProfile_t, 1258
ipfamily
ssdatasession_params, 932
ipv4Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 1287
ipv4GWAddress
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 1287
ipv6Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 1287
ipv6AddressInfo, 333
IPAddressV6, 333
IPV6PrefixLen, 333
ipv6GWAddress

unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1287
is856SysId
HDRSysInfo, 309
nas_HDRSysInfo, 456
is856SysIdValid
HDRSysInfo, 309
nas_HDRSysInfo, 456
is_DataRate_Available
unpack_qos_swiQosFlow_t, 1190
is_EspSpi_Available
unpack_qos_swiQosFilter_t, 1187
is_IPv4DstAddr_Available
unpack_qos_swiQosFilter_t, 1187
is_IPv4SrcAddr_Available
unpack_qos_swiQosFilter_t, 1187
is_IPv4Tos_Available
unpack_qos_swiQosFilter_t, 1187
is_IPv6DstAddr_Available
unpack_qos_swiQosFilter_t, 1187
is_IPv6Label_Available
unpack_qos_swiQosFilter_t, 1187
is_IPv6SrcAddr_Available
unpack_qos_swiQosFilter_t, 1187
is_IPv6TrafCls_Available
unpack_qos_swiQosFilter_t, 1187
is_Id_Available
unpack_qos_swiQosFilter_t, 1187
is_Jitter_Available
unpack_qos_swiQosFlow_t, 1191
is_Latency_Available
unpack_qos_swiQosFlow_t, 1191
is_LteBandCapability_Available
unpack_dms_SLQSGetBandCapability_t, 1068
is_LteQci_Available
unpack_qos_swiQosFlow_t, 1191
is_MaxAllowedPktSz_Available
unpack_qos_swiQosFlow_t, 1191
is_MinPolicedPktSz_Available
unpack_qos_swiQosFlow_t, 1191
is_NxtHdrProto_Available
unpack_qos_swiQosFilter_t, 1187
is_PktErrRate_Available
unpack_qos_swiQosFlow_t, 1191
is_Precedence_Available
unpack_qos_swiQosFilter_t, 1187
is_ProfileId3GPP2_Available
unpack_qos_swiQosFlow_t, 1191
is_RxQFlowGranted_Available
unpack_qos_QosFlowInfo_t, 1177
is_TCPDstPort_Available
unpack_qos_swiQosFilter_t, 1187
is_TCPSrcPort_Available
unpack_qos_swiQosFilter_t, 1187
is_TdsBandCapability_Available
unpack_dms_SLQSGetBandCapability_t, 1068
is_TokenBucket_Available
unpack_qos_swiQosFlow_t, 1191
is_TrafficClass_Available
unpack_qos_swiQosFlow_t, 1191
is_TranDstPort_Available
unpack_qos_swiQosFilter_t, 1187
is_TranSrcPort_Available
unpack_qos_swiQosFilter_t, 1187
is_TxQFlowGranted_Available
unpack_qos_QosFlowInfo_t, 1177
is_UDPDstPort_Available
unpack_qos_swiQosFilter_t, 1187
is_UDPSrcPort_Available
unpack_qos_swiQosFilter_t, 1187
is_val_3GPP2Pri_Available
unpack_qos_swiQosFlow_t, 1191
is_val_3GPPImCn_Available
unpack_qos_swiQosFlow_t, 1191
is_val_3GPPResResidualBER_Available
unpack_qos_swiQosFlow_t, 1191
is_val_3GPPSigInd_Available
unpack_qos_swiQosFlow_t, 1191
is_val_3GPPTraHdlPri_Available
unpack_qos_swiQosFlow_t, 1191
isEmpty
getAllCallInformation, 256
voice_getAllCallInformation, 1323
isInTraffic
nas_txInfo, 523
txInfo, 974
isModByCC
SUPSInfo, 933
isNewFlow
QosFlowInfoState, 815
unpack_qos_QosFlowInfoState_t, 1178
isPrefDataPath
GSMSrvStatusInfo, 296
nas_GSMSrvStatusInfo, 448
nas_SrvStatusInfo, 515
SrvStatusInfo, 929
isRadioTuned
nas_rxInfo, 505
nas_RxSigInfo, 506
rxInfo, 831
RxSigInfo, 834
isSysForbidden
detailSvcInfo, 223
nas_detailSvcInfo, 439
nas_sysInfoCommon, 517
sysInfoCommon, 955
isSysForbiddenValid
nas_sysInfoCommon, 517
sysInfoCommon, 955
isSysPrlMatch
CDMASysInfo, 177
HDRSysInfo, 309
nas_CDMASysInfo, 429
nas_HDRSysInfo, 456
isSysPrlMatchValid
CDMASysInfo, 177

HDRSysInfo, 309
nas_CDMASysInfo, 429
nas_HDRSysInfo, 456
IsVoiceEnabled
pack_dms_SetCustFeature_t, 601
unpack_dms_GetCustFeature_t, 1038
Item
GetAudioPathConfigReq, 258
GetAudioVolTLBConfigReq, 262
SetAudioVolTLBConfigReq, 854

Jitter
unpack_qos_swiQosFlow_t, 1191

KeyExchange
nas_protocolSubtypeElement, 496
protocolSubtypeElement, 773

LITE_VOICE_SUPS_SRV_CLASS_DATA
voice.h, 2069
LITE_VOICE_SUPS_SRV_CLASS_DATACIRCUITAS-
YNC
voice.h, 2070
LITE_VOICE_SUPS_SRV_CLASS_DATACIRCUITSY-
NC
voice.h, 2070
LITE_VOICE_SUPS_SRV_CLASS_FAX
voice.h, 2069
LITE_VOICE_SUPS_SRV_CLASS_NONE
voice.h, 2069
LITE_VOICE_SUPS_SRV_CLASS_PACKETACCESS
voice.h, 2070
LITE_VOICE_SUPS_SRV_CLASS_PADACCESS
voice.h, 2070
LITE_VOICE_SUPS_SRV_CLASS_SMS
voice.h, 2070
LITE_VOICE_SUPS_SRV_CLASS_VOICE
voice.h, 2069
LITEQMI_QMI_CBK_PARAM_NOCHANGE
sms.h, 2017
LITEQMI_QMI_CBK_PARAM_RESET
sms.h, 2017
LITEQMI_QMI_CBK_PARAM_SET
sms.h, 2017
LBPTlv
NASQmiCbkNasSystemSelPrefInd, 567
LEN
qaGobiApiDcs.h, 1674
LOCEVENTMASKGNSSSVINFO
loc.h, 1495
LOCEVENTMASKNMEA
loc.h, 1496
LOCEVENTMASKWIFIREQ
loc.h, 1496
LOCEventRegisterReqResp, 382
eventRegister, 384
LOCExtPowerStateReqResp, 384
extPowerState, 384
LOCStartReq, 391
pApplicationInfo, 392
pConfigAltitudeAssumed, 392
pHorizontalAccuracyLvl, 392
pIntermediateReportState, 393
pMinIntervalTime, 393
pRecurrenceType, 393
SessionId, 393
LOCStopReq, 393
sessionId, 393
LPCSTR
SwiDataTypes.h, 2035
LTEAttachProfile
unpack_wds_SLQSGet3GPPConfigItem_t, 1270
LTEAttachProfileList
unpack_wds_SLQSGet3GPPConfigItem_t, 1270
LTEAttachProfileListLen
_slqs3GPPConfigItem, 83
pack_wds_SLQSSet3GPPConfigItem_t, 730
unpack_wds_SLQSGet3GPPConfigItem_t, 1270
LTEBandPref
NASLTEBandPreferenceTlv, 550
LTECphyCaInfo
unpack_nas_SlqsGetLTECphyCAInfo_t, 1122
LTEInfo, 396
band, 398
bandwidth, 398
emmConnState, 398
emmState, 398
emmSubState, 398
RXChan, 398
TXChan, 398
LTEInfoInterfreq, 398
freqsLen, 398
InfoInterfreq, 398
ueInIdle, 398
LTEInfoIntrafreq, 398
CellParams, 400
cellReselPriority, 400
cellsLen, 400
earfcn, 400
globalCellId, 400
plmn, 400
sIntraSearch, 400
sNonIntraSearch, 400
servingCellId, 400
tac, 400
threshServingLow, 400
ueInIdle, 400
LTEInfoNeighboringGSM, 401
freqsLen, 401
LteGsmCellInfo, 401
ueInIdle, 401
LTEInfoNeighboringWCDMA, 401
freqsLen, 402
LTEWCDMACellInfo, 402
ueInIdle, 402
LTERSRPThresh, 404
LTERSRPThreshListLen, 404

pLTERSRPThreshList, 404
LTERSRPThreshListLen
LTERSRPThresh, 404
nas_LTERSRPThresh, 468
LTERSRQThresh, 404
LTERSRQThreshListLen, 405
pLTERSRQThreshList, 405
LTERSRQThreshListLen
LTERSRQThresh, 405
nas_LTERSRQThresh, 469
LTERSSIThresh, 405
LTERSSIThreshListLen, 405
pLTERSSIThreshList, 405
LTERSSIThreshListLen
LTERSSIThresh, 405
nas_LTERSSIThresh, 469
LTESNRThresListLen
LTESNRThresh, 408
nas_LTESNRThresh, 472
LTESNRThresh, 408
LTESNRThresListLen, 408
pLTESNRThresList, 408
LTESNRThreshListLen
LTESNRThreshold, 409
nas_LTESNRThreshold, 473
LTESNRThreshold, 408
LTESNRThreshListLen, 409
pLTESNRThreshList, 409
LTESSInfo, 409
rsrp, 410
rsrq, 410
rssi, 410
snr, 410
unpack_nas_SLQSNasGetSigInfo_t, 1144
LTESigRptCfg, 406
avgPeriod, 406
rptRate, 406
LTESigRptConfig, 406
avgPeriod, 407
rptRate, 407
LTESysInfo, 411
cellId, 412
cellIdValid, 412
lac, 412
lacValid, 412
MCC, 412
MNC, 413
networkIdValid, 413
regRejectInfoValid, 413
rejCause, 413
rejectSrvDomain, 413
sysInfoLTE, 413
tac, 413
tacValid, 413
LTEWCDMACellInfo
LTEInfoNeighboringWCDMA, 402
nas_LTEInfoNeighboringWCDMA, 467
Lac
qaQmiServingSystemParam, 779
unpack_nas_SLQSGetServingSystem_t, 1129
lac
GERANInfo, 255
GSMSysInfo, 298
LTESysInfo, 412
nas_GERANInfo, 445
nas_GSMSysInfo, 451
nas_LTESysInfo, 475
nas_UMTSInfo, 525
nas_WCDMASysInfo, 534
UMTSInfo, 1023
WCDMASysInfo, 1394
lac1
nas_OperatorPLMNData, 485
OperatorPLMNData, 596
lac2
nas_OperatorPLMNData, 485
OperatorPLMNData, 596
lacValid
GSMSysInfo, 299
LTESysInfo, 412
nas_GSMSysInfo, 451
nas_LTESysInfo, 475
nas_WCDMASysInfo, 534
WCDMASysInfo, 1394
language
CDMABroadcastConfig, 162
sms_CDMABroadcastConfig, 899
lastCallDataBearerTech
unpack_wds_SLQSGetDUNCallInfo_t, 1275
lastCallDataBearerTechnology
unpack_wds_SLQSGetDataBearerTechnology_t, 1272
lastCallRXOKBytesCnt
unpack_wds_SLQSGetDUNCallInfo_t, 1275
lastCallTXOKBytesCnt
unpack_wds_SLQSGetDUNCallInfo_t, 1275
LastErrCode
DataStatusDetail, 219
nas_DataStatusDetail, 437
Latency
unpack_qos_swiQosFlow_t, 1191
Latitude
GPSStateInfo, 292
unpack_pds_SLQSGetGPSStateInfo_t, 1172
latitude
pack_loc_SLQSLOCInjectPosition_t, 623
leapSeconds
nas_qaQmi3Gpp2TimeZone, 497
qaQmi3Gpp2TimeZone, 774
leaseState
wds_DHCPLeaseStateTlv, 1398
len
BdsSVInfo, 141
loc_BdsSVInfo, 367
loc_SVInfo, 378
SVInfo, 935

t_Sv, 956
unpack_nas_GetSignalStrengths_t, 1115
length
readTransparentInfo, 817
SMSCAddress, 913
sMSCAddressInfo, 913
SMSEtwsMessage, 914
sMSEtwsMessageInfo, 915
SMSTransferRouteMTMessage, 926
sMSTransferRouteMTMessageInfo, 926
uim_readTransparentInfo, 991
Level
GetM2MAudioVolumeResp, 279
SetM2MAudioVolumeReq, 864
LibPackGPRSRequestedQoS, 335
delayClass, 336
meanThroughputClass, 336
peakThroughputClass, 336
precedenceClass, 336
reliabilityClass, 336
LibPackQosClassID, 359
gDlBitRate, 360
gUlBitRate, 360
maxDlBitRate, 360
maxUlBitRate, 360
QCI, 360
LibPackTFTIDParams, 360
destPortRangeEnd, 361
destPortRangeStart, 362
eValid, 362
filterId, 362
flowLabel, 362
IPSECSPI, 362
ipVersion, 362
nextHeader, 362
pSourceIP, 362
sourceIPMask, 362
srcPortRangeEnd, 362
srcPortRangeStart, 362
tosMask, 362
LibPackUMTSQoS, 362
deliveryErrSDU, 364
grntDownlinkBitrate, 364
grntUplinkBitrate, 364
maxDownlinkBitrate, 364
maxSDUSize, 364
maxUplinkBitrate, 364
qosDeliveryOrder, 364
resBerRatio, 365
sduErrorRatio, 365
trafficClass, 365
trafficPriority, 365
transferDelay, 365
LibPackUMTSReqQoSSigInd, 365
SigInd, 365
UMTSReqQoS, 365
LibPackprofile_3GPP, 348
pAPNClass, 353
pAPNDisabledFlag, 353
pAPNName, 353
pAPNnameSize, 353
pAddrAllocPref, 353
pAuthenticationPref, 353
pGPRSMinimumQoS, 353
pGPRSRequestedQos, 353
pIPv4AddrPref, 353
pIPv6AddPref, 353
pImCnFlag, 353
pPDNInactivTimeout, 353
pPDPtype, 353
pPassword, 353
pPasswordSize, 353
pPcscfAddrUsingDhcp, 353
pPcscfAddrUsingPCO, 353
pPdpAccessConFlag, 353
pPdpContext, 353
pPdpDataCompType, 353
pPdpHdrCompType, 353
pPriDNSIPv4AddPref, 353
pPriDNSIPv6addpref, 353
pPrimaryID, 353
pProfilename, 354
pProfilenameSize, 354
pQosClassID, 354
pSecDNSIPv4AddPref, 354
pSecDNSIPv6addpref, 354
pSecondaryFlag, 354
pTFTID1Params, 354
pTFTID2Params, 354
pUMTSMinQoS, 354
pUMTSMinQosSigInd, 354
pUMTSReqQoS, 354
pUMTSReqQoSSigInd, 354
pUsername, 354
pUsernameSize, 354
LibPackprofile_3GPP2, 354
pAPNClass3GPP2, 358
pAPNEnabled3GPP2, 358
pAllowLinger, 358
pApnString, 358
pApnStringSize, 358
pAppPriority, 358
pAppType, 358
pAuthPassword, 358
pAuthPassword_tSize, 358
pAuthProtocol, 358
pAuthRetryCount, 359
pAuthTimeout, 359
pDataMode, 359
pDataRate, 359
pIpcpAckTimeout, 359
pIpcpCreqRetryCount, 359
pIsPcscfAddressNedded, 359
pLcpAckTimeout, 359
pLcpCreqRetryCount, 359
pNegoDnsSrvrPref, 359

pPDNInactivTimeout3GPP2, 359
pPdnType, 359
pPppSessCloseTimer1x, 359
pPppSessCloseTimerDO, 359
pPriV6DnsAddress, 359
pPrimaryV4DnsAddress, 359
pRATType, 359
pSecV6DnsAddress, 359
pSecondaryV4DnsAddress, 359
pUserId, 359
pUserIdSize, 359
LibpackProfile3GPP, 336
pAPNClass, 341
pAPNDisabledFlag, 341
pAPNName, 341
pAPNnameSize, 342
pAddrAllocPref, 341
pAuthenticationPref, 342
pGPRSMinimumQoS, 342
pGPRSRequestedQos, 342
pIPv4AddrPref, 342
pIPv6AddPref, 342
pImCnFlag, 342
pPDNInactivTimeout, 342
pPDPtype, 342
pPassword, 342
pPasswordSize, 342
pPcscfAddrUsingDhcp, 342
pPcscfAddrUsingPCO, 342
pPdpAccessConFlag, 342
pPdpContext, 342
pPdpDataCompType, 342
pPdpHdrCompType, 342
pPriDNSIPv4AddPref, 342
pPriDNSIPv6addpref, 342
pPrimaryID, 342
pProfilename, 342
pProfilenameSize, 342
pQosClassID, 342
pSecDNSIPv4AddPref, 342
pSecDNSIPv6addpref, 342
pSecondaryFlag, 342
pTFTID1Params, 342
pTFTID2Params, 342
pUMTSMinQoS, 343
pUMTSMinQosSigInd, 343
pUMTSReqQoS, 343
pUMTSReqQoSSigInd, 343
pUsername, 343
pUsernameSize, 343
LibpackProfile3GPP2, 343
pAPNClass3GPP2, 347
pAPNEnabled3GPP2, 347
pAllowLinger, 347
pApnString, 347
pApnStringSize, 347
pAppPriority, 347
pAppType, 347
pAuthPassword, 347
pAuthPasswordSize, 347
pAuthProtocol, 347
pAuthRetryCount, 347
pAuthTimeout, 347
pDataMode, 347
pDataRate, 347
pIpcpAckTimeout, 347
pIpcpCreqRetryCount, 348
pIsPcscfAddressNedded, 348
pLcpAckTimeout, 348
pLcpCreqRetryCount, 348
pNegoDnsSrvrPref, 348
pPDNInactivTimeout3GPP2, 348
pPdnType, 348
pPppSessCloseTimer1x, 348
pPppSessCloseTimerDO, 348
pPriV6DnsAddress, 348
pPrimaryV4DnsAddress, 348
pRATType, 348
pSecV6DnsAddress, 348
pSecondaryV4DnsAddress, 348
pUserId, 348
pUserIdSize, 348
lineCtrlInfo, 365
polarityIncluded, 366
pwrDenialTime, 366
revPolarity, 366
toggleMode, 366
lineValue
pack_voice_SLQSVoiceALSSelectLine_t, 695
voiceALSSelectLineInfo, 1333
linkage
altitudeSrcInfo, 120
altSrcInfo_t, 121
list_type
custSettingList, 211
DMScustSettingList, 230
DMSgetCustomInput, 231
getCustomInput, 265
pack_dms_GetCustFeaturesV2_t, 598
listEntries
FMSPrefImageList, 252
PrefImageList, 758
listSize
FMSImageList, 252
FMSPrefImageList, 252
ImageList, 314
PrefImageList, 758
liteQmiDataBearerMasks
wds.h, 2098
liteServiceClassInformation
voice.h, 2069
liteqmi_GetVersion
common.h, 1435
liteqmi_helper_decode7bitAsciiEncString
common.h, 1435
liteqmi_log

common.h, 1436
loc.h
eQMI_LOC_SESS_STATUS_FAILURE, 1497
eQMI_LOC_SESS_STATUS_IN_PROGRESS, 1497
eQMI_LOC_SESS_STATUS_SUCCESS, 1497
eQMI_LOC_SESS_STATUS_TIMEOUT, 1497
loc.h, 1491
LOCEVENTMASKNMEA, 1496
LOCEVENTMASKWIFIREQ, 1496
MAX_TEMP_DATA_LEN, 1496
pack_loc_DeleteAssistData, 1497
pack_loc_EventRegister, 1497
pack_loc_SLQSLOCGetBestAvailPos, 1498
pack_loc_SLQSLOCGetServer, 1498
pack_loc_SLQSLOCInjectPosition, 1499
pack_loc_SLQSLOCInjectSensorData, 1499
pack_loc_SLQSLOCInjectUTCTime, 1500
pack_loc_SLQSLOCSetCradleMountConfig, 1500
pack_loc_SetExtPowerState, 1497
pack_loc_SetOperationMode, 1498
pack_loc_Start, 1500
pack_loc_Stop, 1501
unpack_loc_BestAvailPos_Ind, 1501
unpack_loc_DeleteAssistData, 1501
unpack_loc_DeleteAssistData_Ind, 1502
unpack_loc_EngineState_Ind, 1502
unpack_loc_EventNMEA_Ind, 1503
unpack_loc_EventRegister, 1503
unpack_loc_GetServer_Ind, 1503
unpack_loc_GnssSvInfo_Ind, 1504
unpack_loc_PositionRpt_Ind, 1504
unpack_loc_SLQSLOCGetBestAvailPos, 1506
unpack_loc_SLQSLOCGetServer, 1506
unpack_loc_SLQSLOCInjectPosition, 1506
unpack_loc_SLQSLOCInjectSensorData, 1507
unpack_loc_SLQSLOCInjectUTCTime, 1507
unpack_loc_SLQSLOCSetCradleMountConfig, 1507
unpack_loc_SetExtPowerConfig_Ind, 1504
unpack_loc_SetExtPowerState, 1505
unpack_loc_SetOperationMode, 1505
unpack_loc_SetOperationMode_Ind, 1505
unpack_loc_Start, 1508
unpack_loc_Stop, 1508
loc_BdsSV, 366
id, 367
mask, 367
loc_BdsSVInfo, 367
len, 367
pSV, 367
loc_CellDb, 367
mask, 368
loc_ClkInfo, 368
mask, 369
loc_GnssData, 369
mask, 370
loc_IPv4Info, 371
address, 372
port, 372
TlvPresent, 372
loc_IPv6Info, 372
address, 372
port, 372
TlvPresent, 372
loc_LocApplicationInfo, 372
appNameLength, 373
appProviderLength, 373
appVersionLength, 373
appVersionValid, 373
pAppName, 373
pAppProvider, 373
pAppVersion, 373
loc_SV, 377
id, 378
mask, 378
system, 378
loc_SVInfo, 378
len, 378
pSV, 378
loc_gpsTime, 371
gpsTimeOfWeekMs, 371
gpsWeek, 371
loc_precisionDilution, 373
HDOP, 374
PDOP, 374
VDOP, 374
loc_satelliteInfo, 374
azimuth, 376
elevation, 376
gnssSvId, 376
healthStatus, 376
snr, 376
svInfoMask, 376
svListLen, 376
svStatus, 376
system, 376
validMask, 376
loc_sensorDataUsage, 376
aidingIndicatorMask, 377
usageMask, 377
loc_svUsedforFix, 378
gnssSvUsedList, 379
gnssSvUsedList_len, 379
loc_urlAddr, 379
address, 379
TlvPresent, 379
LocApplicationInfo, 380
appNameLength, 380
appProviderLength, 380
appVersionLength, 380
appVersionValid, 380
pAppName, 381
pAppProvider, 381
pAppVersion, 381
LocDelAssDataReq, 381

pBdsSVInfo, 381
pCellDb, 381
pClkInfo, 381
pGnssData, 381
pSVInfo, 382
LocGetServerReq, 384
pServerAddrTypeMask, 385
serverType, 385
LocInjectPositionReq, 385
pAltitudeSrcInfo, 388
pAltitudeWrtEllipsoid, 388
pAltitudeWrtMeanSeaLevel, 388
pHorConfidence, 388
pHorReliability, 389
pHorUncCircular, 389
pLatitude, 389
pLongitude, 389
pPositionSrc, 389
pRawHorConfidence, 389
pRawHorUncCircular, 389
pTimestampAge, 389
pTimestampUtc, 389
pVertConfidence, 389
pVertReliability, 389
pVertUnc, 389
LocInjectSensorDataReq, 389
pAcceleroData, 390
pAcceleroTempData, 390
pAcceleroTimeSrc, 390
pGyroData, 390
pGyroTempData, 390
pGyroTimeSrc, 390
pOpaqueIdentifier, 390
LocSetCradleMountReq, 390
pConfidence, 391
state, 391
localTimeOffset
nas_qaQmi3Gpp2TimeZone, 497
qaQmi3Gpp2TimeZone, 774
Location Service(LOC), 57
logString
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 1073
logger
common.h, 1433
longName
nas_PLMNNetworkNameData, 494
nasPLMNNameResp, 564
PLMNNetworkNameData, 756
unpack_nas_SLQSGetPLMNName_t, 1125
longNameCI
nasPLMNNameResp, 564
unpack_nas_SLQSGetPLMNName_t, 1125
longNameEn
nasPLMNNameResp, 564
unpack_nas_SLQSGetPLMNName_t, 1125
longNameLen
nas_PLMNNetworkNameData, 494
nasPLMNNameResp, 564
PLMNNetworkNameData, 756
unpack_nas_SLQSGetPLMNName_t, 1125
longNameSB
nasPLMNNameResp, 564
unpack_nas_SLQSGetPLMNName_t, 1125
longNameSpareBits
nas_PLMNNetworkNameData, 494
PLMNNetworkNameData, 756
Longitude
GPSStateInfo, 292
unpack_pds_SLQSGetGPSStateInfo_t, 1172
longitude
pack_loc_SLQSLOCInjectPosition_t, 623
loopbackMode
pack_wds_SLQSSSetLoopback_t, 733
loopbackMultiplier
pack_wds_SLQSSSetLoopback_t, 733
LteBandCapability
unpack_dms_SLQSGetBandCapability_t, 1068
LteCQIParm, 393
CQIValueCW0, 394
CQIValueCW1, 394
ValidityCW0, 394
ValidityCW1, 394
lteEARFCN, 394
earfcn0, 394
earfcn1, 395
status, 395
lteEmmDl
NasSwiIndReg, 572
pack_nas_SLQSNasSwiIndicationRegister_t, 644
lteEmmUl
NasSwiIndReg, 572
pack_nas_SLQSNasSwiIndicationRegister_t, 644
lteEsmDl
NasSwiIndReg, 572
pack_nas_SLQSNasSwiIndicationRegister_t, 644
lteEsmUl
NasSwiIndReg, 572
pack_nas_SLQSNasSwiIndicationRegister_t, 644
LteGsmCellInfo
LTEInfoNeighboringGSM, 401
nas_LTEInfoNeighboringGSM, 466
lteGsmCellInfo, 395
cellReselPriority, 396
cells_len, 396
GsmCellInfo, 396
nccPermitted, 396
threshGsmHigh, 396
threshGsmLow, 396
LteNasReleaseInfo
qaGobiApiCbk.h, 1598
LteNasReleaseInfo_s, 402
nas_major, 402
nas_minor, 402
nas_release, 402
ltePCI, 403

earfcn, 403
pci, 403
status, 403
LteQci
unpack_qos_swiQosFlow_t, 1191
lteRsrpDelta
nas_SLQSSignalStrengthsIndReq, 513
SLQSSignalStrengthsIndReq, 894
lteRsrpinfo
nas_SLQSSignalStrengthsInformation, 514
SLQSSignalStrengthsInformation, 896
lteRsrpinformation, 403
rsrplevel, 404
lteSSInfo, 410
rsrp, 410
rsrq, 410
rssi, 410
snr, 410
LteSccRxInfoResp, 405
pSccRxInfo, 405
lteSnrDelta
nas_SLQSSignalStrengthsIndReq, 513
SLQSSignalStrengthsIndReq, 894
lteSnrinfo
nas_SLQSSignalStrengthsInformation, 514
SLQSSignalStrengthsInformation, 896
lteSnrinformation, 407
snrlevel, 408
lteWcdmaCellInfo, 413
cellReselPriority, 414
cellsLen, 414
threshXhigh, 414
threshXlow, 414
uarfcn, 414
WCDMACellInfo, 414
ltersrp
slqsSignalStrengthInfo, 892
unpack_nas_SLQSGetSignalStrength_t, 1132
ltesnr
slqsSignalStrengthInfo, 892
unpack_nas_SLQSGetSignalStrength_t, 1132

m_FwBuildId
CarrierImage_t, 157
SWI_STRUCT_CarrierImage, 936
m_FwImageId
CarrierImage_t, 157
SWI_STRUCT_CarrierImage, 936
m_PriBuildId
CarrierImage_t, 157
SWI_STRUCT_CarrierImage, 936
m_PriImageId
CarrierImage_t, 157
SWI_STRUCT_CarrierImage, 937
m_nCarrierId
CarrierImage_t, 157
SWI_STRUCT_CarrierImage, 936
m_nFolderId
CarrierImage_t, 157
SWI_STRUCT_CarrierImage, 936
m_nStorage
CarrierImage_t, 157
SWI_STRUCT_CarrierImage, 936
MACIndex
nas_NetworkStatEVDO, 482
NetworkStatEVDO, 583
MAX_BUILD_ID_LEN
dms.h, 1443
MAX_CALL_NO_LEN
qaGobiApiVoice.h, 1932
MAX_CONTENT_LENGTH
qaGobiApiUim.h, 1919
MAX_CUST_ID_LEN
qaGobiApiDms.h, 1683
MAX_FSN_LENGTH
qaGobiApiDms.h, 1683
MAX_ICCID_LENGTH
qaGobiApiUim.h, 1919
uim.h, 2049
MAX_MSE_TWS_MSG
sms.h, 2017
MAX_NMEA_STR_LEN
qaGobiApiCbk.h, 1594
MAX_NO_OF_CALLS
qaGobiApiCbk.h, 1594
qaGobiApiVoice.h, 1932
MAX_NO_OF_FILES
qaGobiApiCbk.h, 1595
MAX_NO_OF_SLOTS
qaGobiApiCbk.h, 1595
qaGobiApiUim.h, 1919
uim.h, 2049
MAX_NO_OF_UUSINFO
qaGobiApiCbk.h, 1595
MAX_PATH_LENGTH
qaGobiApiCbk.h, 1595
qaGobiApiUim.h, 1919
MAX_PILOT_SETS
qaGobiApiNas.h, 1752
MAX_PUK_LENGTH
qaGobiApiUim.h, 1919
MAX_SENSOR_DATA_LEN
loc.h, 1496
MAX_SLOTS_STATUS
qaGobiApiUim.h, 1919
uim.h, 2049
MAX_SMS_LIST_SIZE
sms.h, 2017
MAX_SMS_ROUTES
qaGobiApiSms.h, 1824
MAX_TEMP_DATA_LEN
loc.h, 1496
qaGobiApiLoc.h, 1741
MAXUSSDLENGTH
qaGobiApiCbk.h, 1595
qaGobiApiVoice.h, 1932
MAXVOICEUSSDLENGTH

voice.h, 2069
MCC
SlqsNas3GppNetworkRAT, 87
CDMASysInfo, 177
CDMASysInfoExt, 178
currentPLMN, 202
GSMSysInfo, 299
LTESysInfo, 412
nas_CDMASysInfo, 429
nas_CDMASysInfoExt, 430
nas_currentPLMN, 435
nas_GSMSysInfo, 451
nas_LTESysInfo, 475
nas_QmiNas3GppNetworkInfo, 498
nas_QmiNas3GppNetworkRAT, 498
nas_QmisNasPcsDigit, 499
nas_WCDMASysInfo, 534
SlqsNas3GppNetworkInfo, 885
SlqsNasPcsDigit, 886
unpack_nas_GetServingNetwork_t, 1114
WCDMASysInfo, 1394
MDMCallDuration
ConnectionStatus, 189
connectionStatus, 189
MDMConnStatus
ConnectionStatus, 189
connectionStatus, 190
MEID_MAX_SIZE
dms.h, 1443
MEIDString
unpack_dms_GetDeviceSerialNumbers_t, 1044
MIN
unpack_dms_GetVoiceNumber_t, 1058
MINREQBKLEN
common.h, 1433
MMTlv
unpack_sms_SetNewSMSCallback_ind_t, 1198
MNC
SlqsNas3GppNetworkRAT, 87
CDMASysInfo, 177
currentPLMN, 202
GSMSysInfo, 299
LTESysInfo, 413
nas_CDMASysInfo, 430
nas_currentPLMN, 435
nas_GSMSysInfo, 451
nas_LTESysInfo, 475
nas_QmiNas3GppNetworkInfo, 498
nas_QmiNas3GppNetworkRAT, 499
nas_QmisNasPcsDigit, 499
nas_WCDMASysInfo, 534
SlqsNas3GppNetworkInfo, 885
SlqsNasPcsDigit, 886
unpack_nas_GetServingNetwork_t, 1114
WCDMASysInfo, 1394
MNRInfo, 417
mcc, 417
mnc, 417
rat, 417
MPTlv
NASQmiCbkNasSystemSelPrefInd, 567
MSGID_AND_LEN
common.h, 1433
MSGID_DONT_CARE
common.h, 1433
MTMessageInfo
newMTMessageTlv, 584
manString
pack_dms_SLQSSwiSetHostDevInfo_t, 606
unpack_dms_SLQSSwiGetHostDevInfo_t, 1074
manufacturer
unpack_dms_GetManufacturer_t, 1050
Mask
getDUNCallInfoReq, 266
pack_wds_SLQSGetDUNCallInfo_t, 726
mask
BdsSV, 140
CellDb, 178
ClkInfo, 184
GnssData, 286
IPv6TrafCls, 335
loc_BdsSV, 367
loc_CellDb, 368
loc_ClkInfo, 369
loc_GnssData, 370
loc_SV, 378
SV, 934
Tos, 970
unpack_qos_IPv6TrafCls_t, 1175
unpack_qos_Tos_t, 1193
max
_SLQSAVMSSettingsPeriodsInfo, 87
max_channel_rx_rate
unpack_wds_SLQSGetCurrentChannelRate_t, 1272
WDSSWICurrentChannelRates, 1429
max_channel_tx_rate
unpack_wds_SLQSGetCurrentChannelRate_t, 1272
WDSSWICurrentChannelRates, 1429
max_dist
pack_swiloc_SwiLocSetAutoStart_t, 678
SwiLocGetAutoStartResp, 938
SwiLocSetAutoStartReq, 940
unpack_swiloc_SwiLocGetAutoStart_t, 1206
max_dist_reported
SwiLocGetAutoStartResp, 938
unpack_swiloc_SwiLocGetAutoStart_t, 1206
max_time
pack_swiloc_SwiLocSetAutoStart_t, 678
SwiLocGetAutoStartResp, 938
SwiLocSetAutoStartReq, 940
unpack_swiloc_SwiLocGetAutoStart_t, 1206
max_time_reported
SwiLocGetAutoStartResp, 938
unpack_swiloc_SwiLocGetAutoStart_t, 1206

MaxAllowedPktSz
unpack_qos_swiQosFlow_t, 1191
MaxChanRxRate
ChannelRate, 181
dunchannelRate, 237
MaxChanTxRate
ChannelRate, 181
dunchannelRate, 237
maxChannelRXRate
unpack_wds_GetConnectionRate_t, 1255
maxChannelTXRate
unpack_wds_GetConnectionRate_t, 1255
maxDlBitRate
LibPackQosClassID, 360
QosClassID, 812
maxDownlinkBitrate
LibPackUMTSQoS, 364
UMTSMinQoS, 1028
UMTSQoS, 1030
wds_UMTSMinQoS, 1411
maxImages
FMSImageIDEntries, 251
ImageIDEntries, 313
maxMitigationLevel
mitigationDevList, 416
MaxRXChannelRate
unpack_dms_GetDeviceCap_t, 1040
maxRxChannelRate
unpack_dms_GetDeviceCapabilities_t, 1041
maxSDUSize
LibPackUMTSQoS, 364
UMTSMinQoS, 1028
UMTSQoS, 1030
wds_UMTSMinQoS, 1411
maxStorageSize
sms_maxStorageSizeResp, 903
smsMaxStorageSizeResp, 919
MaxTXChannelRate
unpack_dms_GetDeviceCap_t, 1040
maxTxChannelRate
unpack_dms_GetDeviceCapabilities_t, 1041
maxUlBitRate
LibPackQosClassID, 360
QosClassID, 812
maxUplinkBitrate
LibPackUMTSQoS, 364
UMTSMinQoS, 1028
UMTSQoS, 1030
wds_UMTSMinQoS, 1411
mcTimeStamp
cdmaMsgDecodingParams, 166
cdmaMsgDecodingParamsExt, 169
mcc
CSGID, 197
MNRInfo, 417
nas_CSGID, 434
nas_MNRInfo, 478
nas_netSelectionPref, 479
nas_OperatorPLMNData, 485
nasPLMNNameReq, 562
netSelectionPref, 577
OperatorPLMNData, 596
pack_nas_SLQSGetPLMNName_t, 634
PlmnID, 754
unpack_nas_GetHomeNetwork_t, 1110
mccM
minBasedIMSI, 416
nas_minBasedIMSI, 477
mccT
nas_trueIMSI, 523
trueIMSI, 973
mdmCallDurationActive
unpack_wds_SLQSGetDUNCallInfo_t, 1275
MdmConnStatus
DUNCallInfoInd, 236
meanThroughputClass
GPRSQoS, 288
GPRSRequestedQoS, 289
LibPackGPRSRequestedQoS, 336
wds_GPRSQoS, 1403
meid
unpack_dms_GetSerialNumbers_t, 1055
meidLength
_SLQSSwiGetSerialNoExtParams, 97
meidSize
serialNumbersInfo, 844
unpack_dms_GetDeviceSerialNumbers_t, 1044
meidString
unpack_dms_SLQSSwiGetSerialNoExt_t, 1075
message
unpack_sms_SLQSGetSMS_t, 1200
message_type
NASOTAMessageTlv, 558
messageClass
sms_routeEntry, 906
smsRouteEntry, 924
messageFailureCode
slqssendsmsparams_s, 889
unpack_sms_SendSMS_t, 1197
messageFormat
pack_sms_SaveSMS_t, 668
pack_sms_SendSMS_t, 669
slqssendasyncsmsparams_s, 888
slqssendsmsparams_s, 889
sms_sendAsyncsmsParams, 908
unpack_sms_SLQSGetSMS_t, 1200
messageID
slqssendsmsparams_s, 889
SMSAsyncRawSend_s, 912
unpack_sms_SendSMS_t, 1197
messageId
cdmaMsgEncodingParams, 172
messageIndex
pack_sms_SLQSGetSMS_t, 671
pack_sms_SLQSModifySMSStatus_t, 673
qmiSmsMessageList, 805

SMSMTMessage, 921
sMSMTMessageInfo, 922
messageLength
cdmaMsgDecodingParams, 166
cdmaMsgDecodingParamsExt, 169
messageList
unpack_sms_SLQSGetSMSList_t, 1202
messageListSize
unpack_sms_SLQSGetSMSList_t, 1202
messageMode
SMSMemoryInfo, 920
SMSMessageMode, 920
sMSMessageModeInfo, 920
unpack_sms_SLQSWmsMemoryFullCallBack_ind_t, 1204
MessageModeInfo
messageModeTlv, 414
messageModeTlv, 414
MessageModeInfo, 414
TlvPresent, 414
messageSize
pack_sms_SaveSMS_t, 668
pack_sms_SendSMS_t, 669
slqssendasyncsmsparams_s, 888
slqssendsmsparams_s, 889
sms_sendAsyncsmsParams, 908
unpack_sms_SLQSGetSMS_t, 1200
wcdmaMsgEncodingParams, 1390
messageTag
pack_sms_SLQSModifySMSStatus_t, 673
qmiSmsMessageList, 805
unpack_sms_SLQSGetSMS_t, 1200
messageType
sms_routeEntry, 906
smsRouteEntry, 924
messageWaitingInfoContent, 414
activeInd, 415
msgCount, 415
msgType, 415
MicMute
GetAudioProfileResp, 261
GetM2MAudioProfileResp, 278
GetM2MAVMuteResp, 280
SetAudioProfileReq, 853
SetM2MAVMuteReq, 864
min
_SLQSAVMSSettingsPeriodsInfo, 87
minBasedIMSI, 415
imsiM1112, 416
imsiMS1, 416
imsiMS2, 416
mccM, 416
MinPolicedPktSz
unpack_qos_swiQosFlow_t, 1191
minSize
unpack_dms_GetVoiceNumber_t, 1058
minute
nas_timeInfo, 522
nas_UniversalTime, 528
timeInfo, 965
UniversalTime, 1033
mipMode
unpack_wds_GetMobileIP_t, 1260
mipStatus
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1283
mipstatAvail
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1283
mitigationDevID
TmdDeRegNotMitigationLvlReq, 966
TmdGetMitigationLvlReq, 967
TmdMitigationLvlIndReq, 968
TmdRegNotMitigationLvlReq, 969
mitigationDevIDLen
TmdDeRegNotMitigationLvlReq, 966
TmdGetMitigationLvlReq, 967
TmdMitigationLvlIndReq, 968
TmdRegNotMitigationLvlReq, 969
mitigationDevId
mitigationDevList, 417
mitigationDevIdLen
mitigationDevList, 417
MitigationDevInfo
QmiCbkTmdMitiLvlRptInd, 801
mitigationDevList, 416
maxMitigationLevel, 416
mitigationDevId, 417
mitigationDevIdLen, 417
mnc
CSGID, 197
MNRInfo, 417
nas_CSGID, 434
nas_MNRInfo, 478
nas_netSelectionPref, 479
nas_OperatorPLMNData, 485
nasPLMNNameReq, 562
netSelectionPref, 578
OperatorPLMNData, 596
pack_nas_SLQSGetPLMNName_t, 634
PlmnID, 754
unpack_nas_GetHomeNetwork_t, 1110
mncPcsDigits
CSGID, 197
nas_CSGID, 434
mobileCountryCode
SMSEtwsPlmn, 916
sMSEtwsPlmnInfo, 917
mobileIP
pack_wds_SLQSSetWdsEventCallback_t, 732
mobileNetworkCode
SMSEtwsPlmn, 916
sMSEtwsPlmnInfo, 917
mode
callInfo, 153
pack_dms_SetEventReport_t, 602

pack_dms_SetPower_t, 603
pack_loc_SetOperationMode_t, 617
pack_sms_SLQSGetSmsBroadcastConfig_t, 672
pack_sms_SLQSSetSmsBroadcastActivation_t, 675
pack_sms_SLQSSetSmsBroadcastConfig_t, 675
pack_wds_SetMobileIP_t, 720
uim_refreshevent, 993
UIMRefreshEvent, 1013
voice_callInfo, 1314
ModePref
NASModePreferenceTlv, 551
modelString
pack_dms_SLQSSwiSetHostDevInfo_t, 606
unpack_dms_SLQSSwiGetHostDevInfo_t, 1074
modelid
unpack_dms_GetModelID_t, 1050
modelid_str
slqsfwinfo_s, 884
unpack_dms_GetFirmwareInfo_t, 1046
modemMode
CommInfo, 188
nas_CommInfo, 433
modemTempNotification
qaGobiApiCbk.h, 1598
ModemTempState
_modemTempNotification, 68
ModemTemperature
_modemTempNotification, 68
modemindex
pack_fms_SetImagesPreference_t, 613
ModifyProfileIn, 417
curProfile, 418
pProfileID, 418
pProfileType, 418
ModifyProfileOut, 418
pExtErrorCode, 418
month
nas_timeInfo, 522
nas_UniversalTime, 528
timeInfo, 965
UniversalTime, 1033
msgCount
messageWaitingInfoContent, 415
sms_messageWaitingInfoContent, 903
msgDelFailureCause
SMSAsyncRawSend_s, 912
msgDelFailureType
SMSAsyncRawSend_s, 912
msgProtocol
sms_msgProtocolResp, 904
smsMsgprotocolResp, 921
msgType
messageWaitingInfoContent, 415
sms_messageWaitingInfoContent, 903
msgWaitInfo
getMsgWaitingInfo, 282
msgWaitingInfo, 419
sms_getMsgWaitingInfo, 900
msgWaitingInfo, 418
msgWaitInfo, 419
numInstances, 419
msgid
pack_qmi_t, 665
unpack_qmi_t, 1173
msgtype
common.h, 1434
Mtu
unpack_wds_SLQSGetRuntimeSettings_t, 1278
MultDisc
nas_protocolSubtypeElement, 496
protocolSubtypeElement, 773
multiplier
pktErrRate, 753
unpack_qos_pktErrRate_t, 1175
muxID
qmuxIPTable, 811

NAI
unpack_wds_GetMobileIPProfile_t, 1262
NAM_NAME_LENGTH
qaGobiApiNas.h, 1752
NAS_NAM_NAME_LENGTH
nas.h, 1516
NAS_PLMN_LENGTH
nas.h, 1516
NAS_SRV
qaGobiApiCbk.h, 1595
NASBandPreferenceTlv, 535
band_pref, 535
TlvPresent, 535
NASEmergencyModeTlv, 536
EmerMode, 537
TlvPresent, 537
NASGWAcqOrderPrefTlv, 545
GWAcqOrderPref, 546
TlvPresent, 546
NASLTEBandPreferenceTlv, 550
LTEBandPref, 550
TlvPresent, 550
NASLteNasReleaseInfoTlv, 550
nas_major, 551
nas_minor, 551
nas_release, 551
TlvPresent, 551
NASModePreferenceTlv, 551
ModePref, 551
TlvPresent, 552
NASNetSelPreferenceTlv, 552
NetSelPref, 552
TlvPresent, 552
NASOTAMessageTlv, 557
data_buf, 558
data_len, 558
message_type, 558
TlvPresent, 558
NASPRLPreferenceTlv, 565

PRLPref, 565
TlvPresent, 565
NASPhyCaAggPcellInfo, 558
dl_bw_value, 558
freq, 558
iLTEbandValue, 558
pci, 558
TlvPresent, 559
NASPhyCaAggScellDlBw, 559
dl_bw_value, 559
TlvPresent, 559
NASPhyCaAggScellIndType, 560
freq, 560
pci, 560
scell_state, 560
TlvPresent, 560
NASPhyCaAggScellIndex, 559
scell_idx, 559
TlvPresent, 560
NASPhyCaAggScellInfo, 560
dl_bw_value, 561
freq, 561
iLTEbandValue, 561
pci, 561
scell_state, 561
TlvPresent, 561
NASQmiCbkNasSwiOTAMessageInd, 565
nasRelInfoTlv, 565
otaMsgTlv, 566
timeTlv, 566
NASQmiCbkNasSystemSelPrefInd, 566
BPTlv, 567
EMTlv, 567
GWAOPTlv, 567
LBPTlv, 567
MPTlv, 567
NSPTlv, 567
PRLPTlv, 567
RPTlv, 567
SDPTlv, 567
NASRoamPreferenceTlv, 567
RoamPref, 567
TlvPresent, 567
NASServDomainPrefTlv, 567
SrvDomainPref, 568
TlvPresent, 568
NASServingSystemInfo, 568
csAttachState, 569
hdrPersonality, 569
psAttachState, 569
radioInterfaceList, 569
radioInterfaceNo, 569
registrationState, 569
selectedNetwork, 569
NASTimeInfoTlv, 576
time, 576
TlvPresent, 576
NMEAData
unpack_loc_EventNMEA_Ind_t, 1092
NSPTlv
NASQmiCbkNasSystemSelPrefInd, 567
NSSAudioCtrl, 586
downLink, 586
upLink, 586
NUM_OF_SET
qaGobiApiCbk.h, 1595
qaGobiApiSms.h, 1824
NWProfile, 586
pProfSz, 587
pProfValues, 587
tech, 587
NWQoSStatus
unpack_qos_SLQSQosGetNetworkStatus_t, 1179
NWRegStat
_transNWRegInfoNotification, 112
naiSize
unpack_wds_GetMobileIPProfile_t, 1262
namID
airTimer, 117
nasGet3GPP2SubscriptionInfoReq, 537
pack_nas_SLQSNasGet3GPP2Subscription_t, 640
prefVoiceSO, 760
roamTimer, 827
voice_airTimer, 1297
voice_prefVoiceSO, 1329
voice_roamTimer, 1331
namName, 419
namName, 419
namNameLen, 419
namName, 419
nas_namName, 478
namNameLen
namName, 419
nas_namName, 478
Name
unpack_nas_GetServingNetwork_t, 1114
name
unpack_nas_GetHomeNetwork_t, 1110
unpack_wds_GetDefaultProfile_t, 1258
nameLen
remotePartyName, 821
voice_remotePartyName, 1330
namePI
remotePartyName, 821
voice_remotePartyName, 1330
nameSize
unpack_nas_GetHomeNetwork3GPP2_t, 1108
unpack_nas_GetServingNetwork_t, 1114
nameString
pack_dms_SLQSSwiSetOSInfo_t, 607
unpack_dms_SLQSSwiGetOSInfo_t, 1074
namelength
omaDmFotaTlv, 592
omaDmFotaTlvExt, 594
unpack_omaDmFotaTlv_t, 1160

namesize
unpack_wds_GetDefaultProfile_t, 1258
nas.h
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_100, 1517
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_15, 1517
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_25, 1517
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_50, 1517
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_6, 1517
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_75, 1517
eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED, 1517
eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED, 1517
eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_DECONFIGURED, 1517
eNAS_LTE_CPHY_CA_BW_NRB_LITE_100, 1517
eNAS_LTE_CPHY_CA_BW_NRB_LITE_15, 1517
eNAS_LTE_CPHY_CA_BW_NRB_LITE_25, 1517
eNAS_LTE_CPHY_CA_BW_NRB_LITE_50, 1517
eNAS_LTE_CPHY_CA_BW_NRB_LITE_6, 1517
eNAS_LTE_CPHY_CA_BW_NRB_LITE_75, 1517
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED_LITE, 1517
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED_LITE, 1517
eNAS_LTE_CPHY_SCELL_STATE_DECONFIGURED_LITE, 1517
nas.h, 1508
NAS_PLMN_LENGTH, 1516
pack_nas_GetACCOLC, 1517
pack_nas_GetANAAAAuthenticationStatus, 1518
pack_nas_GetCDMANetworkParameters, 1518
pack_nas_GetHomeNetwork, 1518
pack_nas_GetHomeNetwork3GPP2, 1519
pack_nas_GetNetworkPreference, 1519
pack_nas_GetRFInfo, 1519
pack_nas_GetServingNetwork, 1519
pack_nas_GetServingNetworkCapabilities, 1520
pack_nas_GetSignalStrengths, 1520
pack_nas_InitiateDomainAttach, 1520
pack_nas_PerformNetworkScan, 1521
pack_nas_SLQSConfigSigInfo, 1523
pack_nas_SLQSGetErrorRate, 1523
pack_nas_SLQSGetNetworkTime, 1524
pack_nas_SLQSGetOperatorNameData, 1524
pack_nas_SLQSGetPLMNName, 1524
pack_nas_SLQSGetServingSystem, 1525
pack_nas_SLQSGetSignalStrength, 1525
pack_nas_SLQSGetSysInfo, 1525
pack_nas_SLQSGetSysSelectionPref, 1526
pack_nas_SLQSInitiateNetworkRegistration, 1526
pack_nas_SLQSNASSwiGetChannelLock, 1529
pack_nas_SLQSNASSwiSetChannelLock, 1530
pack_nas_SLQSNasConfigSigInfo2, 1526
pack_nas_SLQSNasGet3GPP2Subscription, 1527
pack_nas_SLQSNasGetCellLocationInfo, 1527
pack_nas_SLQSNasGetHDRColorCode, 1528
pack_nas_SLQSNasGetSigInfo, 1528
pack_nas_SLQSNasGetTxRxInfo, 1528
pack_nas_SLQSNasIndicationRegisterExt, 1529
pack_nas_SLQSNasSwiIndicationRegister, 1529
pack_nas_SLQSNasSwiModemStatus, 1530
pack_nas_SLQSSetBandPreference, 1531
pack_nas_SLQSSetSignalStrengthsCallback, 1531
pack_nas_SLQSSetSysSelectionPref, 1531
pack_nas_SLQSSwiGetHDRPersonality, 1532
pack_nas_SLQSSwiGetHDRProtSubtype, 1532
pack_nas_SLQSSwiGetHRPDStats, 1532
pack_nas_SLQSSwiGetLteCQI, 1533
pack_nas_SLQSSwiGetLteSccRxInfo, 1533
pack_nas_SLQSSwiNetworkDebug, 1533
pack_nas_SLQSSwiPSDetach, 1534
pack_nas_SetACCOLC, 1521
pack_nas_SetCDMANetworkParameters, 1522
pack_nas_SetLURejectCallback, 1522
pack_nas_SetNetworkPreference, 1522
pack_nas_SetRFInfoCallback, 1523
pack_nas_SlqsGetLTECphyCAInfo, 1524
unpack_nas_GetACCOLC, 1534
unpack_nas_GetANAAAAuthenticationStatus, 1534
unpack_nas_GetCDMANetworkParameters, 1535
unpack_nas_GetHomeNetwork, 1535
unpack_nas_GetHomeNetwork3GPP2, 1536
unpack_nas_GetNetworkPreference, 1536
unpack_nas_GetRFInfo, 1536
unpack_nas_GetServingNetwork, 1537
unpack_nas_GetServingNetworkCapabilities, 1537
unpack_nas_GetSignalStrengths, 1537
unpack_nas_InitiateDomainAttach, 1538
unpack_nas_PerformNetworkScan, 1538
unpack_nas_SLQSConfigSigInfo, 1542
unpack_nas_SLQSGetErrorRate, 1542
unpack_nas_SLQSGetNetworkTime, 1543
unpack_nas_SLQSGetOperatorNameData, 1543
unpack_nas_SLQSGetPLMNName, 1543
unpack_nas_SLQSGetServingSystem, 1544
unpack_nas_SLQSGetSignalStrength, 1544
unpack_nas_SLQSGetSysInfo, 1545
unpack_nas_SLQSGetSysSelectionPref, 1545
unpack_nas_SLQSInitiateNetworkRegistration, 1545
unpack_nas_SLQSNASSwiGetChannelLock, 1549
unpack_nas_SLQSNASSwiSetChannelLock, 1550
unpack_nas_SLQSNasConfigSigInfo2, 1546
unpack_nas_SLQSNasGet3GPP2Subscription, 1546

unpack_nas_SLQSNasGetCellLocationInfo, 1546
unpack_nas_SLQSNasGetHDRColorCode, 1547
unpack_nas_SLQSNasGetSigInfo, 1547
unpack_nas_SLQSNasGetTxRxInfo, 1548
unpack_nas_SLQSNasIndicationRegisterExt, 1548
unpack_nas_SLQSNasNetworkTimeCallBack_ind, 1548
unpack_nas_SLQSNasSigInfoCallback_ind, 1549
unpack_nas_SLQSNasSwiIndicationRegister, 1549
unpack_nas_SLQSNasSwiModemStatus, 1550
unpack_nas_SLQSNasSwiOTAMessageCallback-_ind, 1550
unpack_nas_SLQSNasSysInfoCallback_ind, 1551
unpack_nas_SLQSNasTimerCallback_ind, 1551
unpack_nas_SLQSSetBandPreference, 1551
unpack_nas_SLQSSetSignalStrengthsCallback, 1552
unpack_nas_SLQSSetSysSelectionPref, 1552
unpack_nas_SLQSSetSysSelectionPrefCallBack-_ind, 1552
unpack_nas_SLQSSwiGetHDRPersonality, 1552
unpack_nas_SLQSSwiGetHDRProtSubtype, 1553
unpack_nas_SLQSSwiGetHRPDStats, 1553
unpack_nas_SLQSSwiGetLteCQI, 1554
unpack_nas_SLQSSwiGetLteSccRxInfo, 1554
unpack_nas_SLQSSwiNetworkDebug, 1554
unpack_nas_SLQSSwiPSDetach, 1555
unpack_nas_SetACCOLC, 1538
unpack_nas_SetCDMANetworkParameters, 1539
unpack_nas_SetDataCapabilitiesCallback_ind, 1539
unpack_nas_SetEventReportInd, 1539
unpack_nas_SetLURejectCallback, 1540
unpack_nas_SetNasLTECphyCaIndCallback_ind, 1540
unpack_nas_SetNetworkPreference, 1540
unpack_nas_SetRFInfoCallback, 1541
unpack_nas_SetRoamingIndicatorCallback_ind, 1541
unpack_nas_SetServingSystemCallback_ind, 1541
unpack_nas_SlqsGetLTECphyCAInfo, 1542
unpack_valid_nas_GetCDMANetworkParameters, 1555
unpack_valid_nas_SLQSGetServingSystem, 1555
unpack_valid_nas_SLQSGetSignalStrength, 1556
unpack_valid_nas_SLQSNasGetSigInfo, 1557
nas_ActPilotPNElement, 420
ActSetPilotPN, 421
ActSetPilotPNStrength, 421
nas_AddCDMASysInfo, 421
geoSysIdx, 421
regPrd, 421
nas_AddSysInfo, 421
cellBroadcastCap, 422
geoSysIdx, 422
nas_CDMAChannel, 423
priChA, 424
priChB, 424
secChA, 424
secChB, 424
nas_CDMAECIOThresh, 424
CDMAECIOThreshListLen, 425
pCDMAECIOThreshList, 425
nas_CDMAInfo, 425
baseId, 425
baseLat, 425
baseLong, 426
nid, 426
refpn, 426
sid, 426
nas_CDMARSSIThresh, 426
CDMARSSIThreshListLen, 426
pCDMARSSIThreshList, 426
nas_CDMASysInfo, 426
baseId, 429
baseLat, 429
baseLong, 429
bsInfoValid, 429
bsPRev, 429
bsPRevValid, 429
ccsSupported, 429
ccsSupportedValid, 429
cdmaSysIdValid, 429
isSysPrlMatch, 429
isSysPrlMatchValid, 429
MCC, 429
MNC, 430
networkID, 430
networkIdValid, 430
pRevInUse, 430
pRevInUseValid, 430
packetZone, 430
packetZoneValid, 430
sysInfoCDMA, 430
systemID, 430
nas_CDMASysInfoExt, 430
imsi_11_12, 430
MCC, 430
nas_CSGID, 433
id, 434
mcc, 434
mnc, 434
mncPcsDigits, 434
rat, 434
nas_CallBarringSysInfo, 422
csBarStatus, 422
psBarStatus, 422
nas_CommInfo, 431
imsRegState, 433
modemMode, 433
psState, 433
systemMode, 433
temperature, 433

nas_DRCParams, 440
DRCCover, 441
DRCValue, 441
nas_DataStatusDetail, 436
IPAddress, 437
LastErrCode, 437
nas_DeviceConfigDetail, 439
Chipset, 440
HWVersion, 440
QLIC, 440
Technology, 440
nas_ECIOThresh, 442
ECIOThresListLen, 442
pECIOThresList, 442
nas_GERANInfo, 443
arfcn, 445
bsic, 445
cellID, 445
insNmrCellInfo, 445
lac, 445
nmrInst, 445
plmn, 445
rxLev, 445
timingAdvance, 445
nas_GSMRSSIThresh, 447
GSMRSSIThreshListLen, 447
pGSMRSSIThreshList, 447
nas_GSMSrvStatusInfo, 447
isPrefDataPath, 448
srvStatus, 448
trueSrvStatus, 448
nas_GSMSysInfo, 448
cellId, 450
cellIdValid, 450
dtmSupp, 450
dtmSuppValid, 450
egprsSupp, 451
egprsSuppValid, 451
lac, 451
lacValid, 451
MCC, 451
MNC, 451
networkIdValid, 451
regRejectInfoValid, 451
rejCause, 451
rejectSrvDomain, 451
sysInfoGSM, 451
nas_HDRECIOThresh, 451
HDRECIOThreshListLen, 451
pHDRECIOThreshList, 451
nas_HDRIOThresh, 451
HDRIOThreshListLen, 452
pHDRIOThreshList, 452
nas_HDRRSSIThresh, 452
HDRRSSIThreshListLen, 452
pHDRRSSIThreshList, 452
nas_HDRSINRThresh, 453
HDRSINRThresListLen, 453
pHDRSINRThresList, 453
nas_HDRSINRThreshold, 453
HDRSINRThreshListLen, 454
pHDRSINRThreshList, 454
nas_HDRSysInfo, 454
hdrActiveProt, 456
hdrActiveProtValid, 456
hdrPersonality, 456
hdrPersonalityValid, 456
is856SysId, 456
is856SysIdValid, 456
isSysPrlMatch, 456
isSysPrlMatchValid, 456
sysInfoHDR, 456
nas_IOThresh, 458
IOThresListLen, 458
pIOThresList, 458
nas_LTEInfo, 460
band, 462
bandwidth, 462
emmConnState, 462
emmState, 462
emmSubState, 462
RXChan, 462
TXChan, 462
nas_LTEInfoInterfreq, 463
freqsLen, 463
InfoInterfreq, 463
ueInIdle, 463
nas_LTEInfoIntrafreq, 463
CellParams, 465
cellReselPriority, 465
cellsLen, 465
earfcn, 465
globalCellId, 465
plmn, 465
sIntraSearch, 465
sNonIntraSearch, 465
servingCellId, 465
tac, 465
threshServingLow, 465
ueInIdle, 465
nas_LTEInfoNeighboringGSM, 465
freqsLen, 466
LteGsmCellInfo, 466
ueInIdle, 466
nas_LTEInfoNeighboringWCDMA, 466
freqsLen, 467
ueInIdle, 467
nas_LTERSRPThresh, 468
LTERSRPThreshListLen, 468
pLTERSRPThreshList, 468
nas_LTERSRQThresh, 468
LTERSRQThreshListLen, 469
pLTERSRQThreshList, 469
nas_LTERSSIThresh, 469
LTERSSIThreshListLen, 469
pLTERSSIThreshList, 469

nas_LTESNRThresh, 472
LTESNRThresListLen, 472
pLTESNRThresList, 472
nas_LTESNRThreshold, 472
LTESNRThreshListLen, 473
pLTESNRThreshList, 473
nas_LTESigRptCfg, 469
avgPeriod, 470
rptRate, 470
nas_LTESigRptConfig, 470
avgPeriod, 471
rptRate, 471
nas_LTESysInfo, 473
cellId, 475
cellIdValid, 475
lac, 475
lacValid, 475
MCC, 475
MNC, 475
networkIdValid, 475
regRejectInfoValid, 475
rejCause, 475
rejectSrvDomain, 475
sysInfoLTE, 475
tac, 475
tacValid, 475
nas_MNRInfo, 477
mcc, 478
mnc, 478
rat, 478
nas_NetworkStat1x, 479
ActSetCnt, 481
NeighborSetCnt, 481
pActPilotPNElements, 481
pNeighborSetPilotPN, 481
RX_EC_IO, 481
RX_PWR, 481
SO, 481
State, 481
TX_PWR, 481
nas_NetworkStatEVDO, 481
MACIndex, 482
PER, 483
pSectorID, 483
PilotEnergy, 483
RX_PWR, 483
SNR, 483
SectorIDLen, 483
State, 483
nas_OperatorPLMNData, 484
lac1, 485
lac2, 485
mcc, 485
mnc, 485
PLMNRecID, 485
nas_PLMNNetworkName, 492
numInstance, 493
PLMNNetName, 493
nas_PLMNNetworkNameData, 493
codingScheme, 494
countryInitials, 494
longName, 494
longNameLen, 494
longNameSpareBits, 494
shortName, 494
shortNameLen, 494
shortNameSpareBits, 494
nas_PhyCaAggPcellInfo, 486
dl_bw_value, 487
freq, 487
iLTEbandValue, 487
pci, 487
TlvPresent, 487
nas_PhyCaAggScellDlBw, 487
dl_bw_value, 487
TlvPresent, 487
nas_PhyCaAggScellIndType, 488
freq, 488
pci, 488
scell_state, 488
TlvPresent, 488
nas_PhyCaAggScellIndex, 487
scell_idx, 488
TlvPresent, 488
nas_PhyCaAggScellInfo, 489
dl_bw_value, 491
freq, 491
iLTEbandValue, 491
pci, 491
scell_state, 491
TlvPresent, 491
nas_PilotSetData, 491
NumPilots, 491
pPilotSetInfo, 491
nas_PilotSetParams, 492
PilotPN, 492
PilotStrength, 492
PilotType, 492
nas_QmiNas3GppNetworkInfo, 497
Desription, 498
Forbidden, 498
InUse, 498
MCC, 498
MNC, 498
Preferred, 498
Roaming, 498
nas_QmiNas3GppNetworkRAT, 498
MCC, 498
MNC, 499
RAT, 499
nas_QmisNasPcsDigit, 499
includes_pcs_digit, 499
MCC, 499
MNC, 499
nas_RFInfoTlv, 500
activeBandClass, 501

activeChannel, 501
radioInterface, 501
radioInterfaceSize, 501
TlvPresent, 501
nas_RSRPThresh, 502
pRSRPThresList, 502
RSRPThresListLen, 502
nas_RSRQThresh, 503
pRSRQThresList, 503
RSRQThresListLen, 503
nas_RSSIThresh, 503
pRSSIThresList, 504
RSSIThresListLen, 504
nas_RejectReasonTlv, 499
rejectCause, 500
serviceDomain, 500
TlvPresent, 500
nas_RxSigInfo, 505
isRadioTuned, 506
rsrp, 506
rxChainIndex, 506
rxPower, 506
nas_SLQSSignalStrengthsIndReq, 511
ecioDelta, 513
ecioThresholdList, 513
ecioThresholdListLen, 513
ioDelta, 513
lteRsrpDelta, 513
lteSnrDelta, 513
rsrqDelta, 513
rxSignalStrengthDelta, 513
sinrDelta, 513
sinrThresholdList, 513
sinrThresholdListLen, 513
nas_SLQSSignalStrengthsInformation, 513
ecioInfo, 514
errorRateInfo, 514
io, 514
lteRsrpinfo, 514
lteSnrinfo, 514
rsrqInfo, 514
rxSignalStrengthInfo, 514
sinr, 514
nas_SLQSSignalStrengthsTlv, 514
sSLQSSignalStrengthsInfo, 514
TlvPresent, 514
nas_SccRxInfo, 507
numInstances, 508
rsrq, 508
sigInfo, 508
snr, 508
TlvPresent, 508
nas_SignalStrengthTlv, 511
radioInterface, 511
signalStrength, 511
TlvPresent, 511
nas_SrvStatusInfo, 515
isPrefDataPath, 515
srvStatus, 515
nas_TDSCDMAECIOThresh, 518
nas_TDSCDMARSCPThresh, 518
nas_TDSCDMARSSIThresh, 519
nas_TDSCDMASINRCONFThresh, 519
nas_TDSCDMASINRThresh, 520
nas_UMTSInfo, 523
cellID, 525
ecio, 525
geranInst, 525
GeranInstInfo, 525
lac, 525
plmn, 525
psc, 525
rscp, 525
UMTSInstInfo, 525
uarfcn, 525
umtsInst, 525
nas_UMTSinstInfo, 525
umtsEcio, 526
umtsPsc, 526
umtsRscp, 526
umtsUarfcn, 526
nas_UniversalTime, 527
day, 528
dayOfWeek, 528
hour, 528
minute, 528
month, 528
second, 528
year, 528
nas_WCDMAECIOThresh, 529
nas_WCDMAInfoLTENeighborCell, 529
umtsLTENbrCellLen, 530
wcdmaRRCState, 530
nas_WCDMARSSIThresh, 530
nas_WCDMASysInfo, 531
cellId, 533
cellIdValid, 534
hsCallStatus, 534
hsCallStatusValid, 534
hsInd, 534
hsIndValid, 534
lac, 534
lacValid, 534
MCC, 534
MNC, 534
networkIdValid, 534
psc, 534
pscValid, 534
regRejectInfoValid, 534
rejCause, 534
rejectSrvDomain, 534
sysInfoWCDMA, 534
nas_acqOrderPref, 420
acqOrdeLen, 420
pAcqOrder, 420
nas_callBarStatus, 423

csBarStatus, 423
psBarStatus, 423
nas_cellParams, 430
pci, 431
rsrp, 431
rsrq, 431
rssi, 431
srxlev, 431
nas_currentPLMN, 434
MCC, 435
MNC, 435
netDescr, 435
netDescrLength, 435
nas_dataSrvCapabilities, 435
dataCapabilities, 436
dataCapabilitiesLen, 436
nas_detailSvcInfo, 438
hdrHybrid, 439
hdrSrvStatus, 439
isSysForbidden, 439
srvCapability, 439
srvStatus, 439
nas_dirNum, 440
dirNum, 440
dirNumLen, 440
nas_ecioListElement, 441
ecio, 441
radioIf, 442
nas_errorRateListElement, 442
errorRate, 443
radioIf, 443
nas_geranInstInfo, 445
geranArfcn, 445
geranBsicBcc, 445
geranBsicNcc, 446
geranRssi, 446
nas_gsmCellInfo, 446
arfcn, 446
band1900, 447
bsicId, 447
cellIdValid, 447
rssi, 447
srxlev, 447
nas_homeSIDNID, 456
numInstances, 456
SidNid, 456
nas_infoInterFreq, 456
cell_resel_priority, 457
cellInterFreqParams, 457
cells_len, 457
earfcn, 457
threshXHigh, 457
threshXLow, 458
nas_lteEARFCN, 458
earfcn0, 459
earfcn1, 459
status, 459
nas_lteGsmCellInfo, 459
cellReselPriority, 460
cells_len, 460
GsmCellInfo, 460
nccPermitted, 460
threshGsmHigh, 460
threshGsmLow, 460
nas_ltePCI, 467
earfcn, 467
pci, 467
status, 467
nas_lteRsrpinformation, 467
rsrplevel, 468
nas_lteSnrinformation, 471
snrlevel, 471
nas_lteWcdmaCellInfo, 475
cellReselPriority, 476
cellsLen, 476
threshXhigh, 476
threshXlow, 476
uarfcn, 476
WCDMACellInfo, 476
nas_major
LteNasReleaseInfo_s, 402
NASLteNasReleaseInfoTlv, 551
nas_minBasedIMSI, 476
imsiM1112, 477
imsiMS1, 477
imsiMS2, 477
mccM, 477
nas_minor
LteNasReleaseInfo_s, 402
NASLteNasReleaseInfoTlv, 551
nas_namName, 478
namName, 478
namNameLen, 478
nas_netSelectionPref, 478
mcc, 479
mnc, 479
netReg, 479
nas_nmrCellInfo, 483
nmrArfcn, 484
nmrBsic, 484
nmrCellID, 484
nmrLac, 484
nmrPlmn, 484
nmrRxLev, 484
nas_operatorNameString, 484
PLMNName, 484
nas_operatorPLMNList, 485
numInstance, 486
PLMNData, 486
nas_protocolSubtypeElement, 494
AccessMac, 495
AuthProt, 496
ControlMac, 496
EncryptProt, 496
ForwardMac, 496
IdleState, 496

KeyExchange, 496
MultDisc, 496
PhysicalLayer, 496
ReverseMac, 496
SecProt, 496
VirtStream, 496
nas_qaQmi3Gpp2TimeZone, 496
daylightSavings, 496
leapSeconds, 497
localTimeOffset, 497
nas_release
LteNasReleaseInfo_s, 402
NASLteNasReleaseInfoTlv, 551
nas_roamIndList, 501
numInstances, 502
radioInterface, 502
roamIndicator, 502
nas_rsrqInformation, 502
radioIf, 503
rsrq, 503
nas_rxInfo, 504
ecio, 505
isRadioTuned, 505
phase, 505
rscp, 505
rsrp, 505
rxPower, 505
nas_rxSignalStrengthListElement, 506
radioIf, 507
rxSignalStrength, 507
nas_servSystem, 509
csAttachState, 510
numRadioInterfaces, 510
psAttachState, 510
radioInterface, 510
regState, 510
selNetwork, 510
nas_serviceProviderName, 508
displayCondition, 509
spn, 509
spnLength, 509
nas_sidNid, 510
nid, 511
sid, 511
nas_sysInfoCommon, 515
isSysForbidden, 517
isSysForbiddenValid, 517
roamStatus, 517
roamStatusValid, 517
srvCapability, 517
srvCapabilityValid, 517
srvDomain, 517
srvDomainValid, 517
nas_timeInfo, 520
day, 522
dayLtSavingAdj, 522
dayOfWeek, 522
hour, 522
minute, 522
month, 522
radioInterface, 522
second, 522
timeZone, 522
TlvPresent, 522
year, 522
nas_trueIMSI, 522
imsiT1112, 523
imsiTS1, 523
imsiTS2, 523
imsiTaddrNum, 523
mccT, 523
nas_txInfo, 523
isInTraffic, 523
txPower, 523
nas_umtsLTENbrCell, 526
cellIsTDD, 527
earfcn, 527
pci, 527
rsrp, 527
rsrq, 527
srxlev, 527
nas_wcdmaCellInfo, 528
cpich_ecno, 529
cpich_rscp, 529
psc, 529
srxlev, 529
nas_wcdmaUARFCN, 534
status, 535
uarfcn, 535
nasCellLocationInfoResp, 535
pCDMAInfo, 536
pGERANInfo, 536
pLTEInfoInterfreq, 536
pLTEInfoIntrafreq, 536
pLTEInfoNeighboringGSM, 536
pLTEInfoNeighboringWCDMA, 536
pUMTSCellID, 536
pUMTSInfo, 536
pWCDMAInfoLTENeighborCell, 536
nasGet3GPP2SubscriptionInfoReq, 537
namID, 537
nasGet3GPP2SubscriptionInfoResp, 537
pCDMAChannel, 538
pDirNum, 538
pHomeSIDNID, 538
pMinBasedIMSI, 538
pNAMNameInfo, 538
pTrueIMSI, 538
nasGetHDRColorCodeResp, 538
pColorCode, 539
nasGetLTECphyCa, 539
sPhyCaAggPcellInfo, 539
sPhyCaAggScellDlBw, 539
sPhyCaAggScellIndType, 539
sPhyCaAggScellIndex, 539
sPhyCaAggScellInfo, 539

NasGetLTECphyCaInfo, 539
PhyCaAggPcellInfo, 540
PhyCaAggScellDlBw, 540
PhyCaAggScellIndType, 540
PhyCaAggScellIndex, 540
PhyCaAggScellInfo, 540
nasGetLTECphyCaResp, 540
pPhyCaAggPcellInfo, 540
pPhyCaAggScellDlBw, 540
pPhyCaAggScellIndType, 540
pPhyCaAggScellIndex, 540
pPhyCaAggScellInfo, 540
nasGetSigInfoResp, 540
pCDMASSInfo, 541
pGSMSSInfo, 541
pHDRSSInfo, 541
pLTESSInfo, 541
pTDSCDMASigInfoExt, 541
pTDSCDMASigInfoRscp, 541
pWCDMASSInfo, 541
nasGetSysInfoResp, 541
pAddCDMASysInfo, 543
pAddGSMSysInfo, 543
pAddHDRSysInfo, 543
pAddLTESysInfo, 544
pAddWCDMASysInfo, 544
pCDMASrvStatusInfo, 544
pCDMASysInfo, 544
pGSMCallBarringSysInfo, 544
pGSMCipherDomainSysInfo, 544
pGSMSrvStatusInfo, 544
pGSMSysInfo, 544
pHDRSrvStatusInfo, 544
pHDRSysInfo, 544
pLTESrvStatusInfo, 544
pLTESysInfo, 544
pLTEVoiceSupportSysInfo, 544
pWCDMACallBarringSysInfo, 544
pWCDMACipherDomainSysInfo, 544
pWCDMASrvStatusInfo, 544
pWCDMASysInfo, 544
nasGetTxRxInfoReq, 544
radio_if, 545
nasGetTxRxInfoResp, 545
pRXChain0Info, 545
pRXChain1Info, 545
pTXInfo, 545
nasIndicationRegisterReq, 546
pDDTMInd, 549
pDualStandByPrefInd, 549
pErrorRateInd, 549
pHDRNewUATIAssInd, 549
pHDRSessionCloseInd, 549
pLTECphyCa, 549
pManagedRoamingInd, 549
pNetworkRejectInd, 549
pNetworkTimeInd, 549
pServingSystemInd, 549
pSignalStrengthInd, 549
pSubscriptionInfoInd, 549
pSuppressSysInfoInd, 549
pSysInfoInd, 549
pSystemSelectionInd, 549
nasInitNetworkReg, 549
pChangeDuration, 550
pMNRInfo, 550
pMncPcsDigitStatus, 550
regAction, 550
nasNetworkReject, 552
pCsgId, 555
pPlmnId, 555
radioIf, 555
rejectCause, 555
serviceDomain, 555
nasNetworkTime, 555
pDayltSavAdj, 556
pRadioInterface, 556
pTimeZone, 556
universalTime, 556
nasOperatorNameResp, 556
pNITZInformation, 557
pOperatorNameString, 557
pOperatorPLMNList, 557
pPLMNNetworkName, 557
pSrvcProviderName, 557
nasPLMNNameReq, 561
mcc, 562
mnc, 562
pMncPcsStatus, 562
nasPLMNNameResp, 562
longName, 564
longNameCI, 564
longNameEn, 564
longNameLen, 564
longNameSB, 564
shortName, 564
shortNameCI, 564
shortNameEn, 564
shortNameLen, 564
shortNameSB, 564
spn, 564
spnEncoding, 564
spnLength, 565
nasRelInfoTlv
NASQmiCbkNasSwiOTAMessageInd, 565
nasSigInfo, 569
pCDMASigInfo, 570
pGSMSigInfo, 570
pHDRSigInfo, 570
pLTESigInfo, 570
pRscp, 570
pTDSCDMASigInfoExt, 570
pWCDMASigInfo, 570
nasSwiGetChannelLockResp, 570
pLteEARFCN, 571
pLtePCI, 571

pWcdmaUARFCN, 571
NasSwiIndReg, 571
gsmUmtsDl, 572
gsmUmtsUl, 572
lteEmmDl, 572
lteEmmUl, 572
lteEsmDl, 572
lteEsmUl, 572
pRankIndicatorInd, 572
pTimer, 572
nasSwiSetChannelLockReq, 572
pLteEARFCN, 573
pLtePCI, 573
pWcdmaUARFCN, 573
nasSysInfo, 573
pAddCDMASysInfo, 575
pAddGSMSysInfo, 575
pAddHDRSysInfo, 575
pAddLTESysInfo, 575
pAddWCDMASysInfo, 575
pCDMASrvStatusInfo, 575
pCDMASysInfo, 575
pGSMCallBarringSysInfo, 575
pGSMCipherDomainSysInfo, 575
pGSMSrvStatusInfo, 575
pGSMSysInfo, 575
pHDRSrvStatusInfo, 575
pHDRSysInfo, 576
pLTESrvStatusInfo, 576
pLTESysInfo, 576
pLTEVoiceSupportSysInfo, 576
pSysInfoNoChange, 576
pWCDMACallBarringSysInfo, 576
pWCDMACipherDomainSysInfo, 576
pWCDMASrvStatusInfo, 576
pWCDMASysInfo, 576
nasTimers, 576
t3396_apn, 577
t3396_plmn_id, 577
t3396_val, 577
nccPermitted
lteGsmCellInfo, 396
nas_lteGsmCellInfo, 460
NeighborSetCnt
nas_NetworkStat1x, 481
NetworkStat1x, 581
netDescr
currentPLMN, 202
nas_currentPLMN, 435
netDescrLength
currentPLMN, 202
nas_currentPLMN, 435
netInfoLen
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1283
netReg
nas_netSelectionPref, 479
netSelectionPref, 578
NetSelPref
NASNetSelPreferenceTlv, 552
netSelectionPref, 577
mcc, 577
mnc, 578
netReg, 578
NetStats, 578
rx_bytes, 578
rx_errors, 578
rx_overflows, 579
rx_packets, 579
tx_bytes, 579
tx_errors, 579
tx_overflows, 579
tx_packets, 579
Network Access Service (NAS), 40
NetworkDebugResp, 579
pDataStatusDetail, 579
pDeviceConfigDetail, 579
pNetworkStat1x, 579
pNetworkStatEVDO, 580
pObjectVer, 580
NetworkID
qaQmiServingSystemParam, 779
unpack_nas_SLQSGetServingSystem_t, 1129
networkID
CDMASysInfo, 177
nas_CDMASysInfo, 430
networkIdValid
CDMASysInfo, 177
GSMSysInfo, 299
LTESysInfo, 413
nas_CDMASysInfo, 430
nas_GSMSysInfo, 451
nas_LTESysInfo, 475
nas_WCDMASysInfo, 534
WCDMASysInfo, 1394
networkInfoLen
unpack_wds_SLQSGetCurrDataSystemStat_t, 1271
NetworkStat1x, 580
ActSetCnt, 581
NeighborSetCnt, 581
pActPilotPNElements, 581
pNeighborSetPilotPN, 581
RX_EC_IO, 582
RX_PWR, 582
SO, 582
State, 582
TX_PWR, 582
NetworkStatEVDO, 582
MACIndex, 583
PER, 583
pSectorID, 583
PilotEnergy, 583
RX_PWR, 583
SNR, 583
SectorIDLen, 583

State, 583
NetworkType
CurrNetworkInfo, 204
currNetworkInfo, 206
wds_currNetworkInfo, 1396
NewMMTlv
unpack_sms_SetNewSMSCallback_ind_t, 1198
newMTMessageTlv, 583
MTMessageInfo, 584
TlvPresent, 584
newPINLen
uim_unblockUIMPIN, 999
unblockUIMPIN, 1032
newPINVal
uim_unblockUIMPIN, 999
unblockUIMPIN, 1032
newPasswd
pack_voice_SLQSVoiceSetCallBarringPassword_t, 707
voiceSetCallBarringPwdInfo, 1371
newPasswdAgain
pack_voice_SLQSVoiceSetCallBarringPassword_t, 707
voiceSetCallBarringPwdInfo, 1371
newPin
pack_dms_UIMUnblockPIN_t, 611
newPwd
newPwdData, 584
newPwdAgain
newPwdData, 584
newPwdData, 584
newPwd, 584
newPwdAgain, 584
newValue
pack_dms_UIMChangePIN_t, 607
nextHeader
LibPackTFTIDParams, 362
TFTIDParams, 964
nid
CDMAInfo, 164
nas_CDMAInfo, 426
nas_sidNid, 511
sidNid, 875
unpack_nas_GetHomeNetwork_t, 1110
NmeaPort
DcsUsbPortNames, 220
nmrArfcn
nas_nmrCellInfo, 484
nmrCellInfo, 585
nmrBsic
nas_nmrCellInfo, 484
nmrCellInfo, 585
nmrCellID
nas_nmrCellInfo, 484
nmrCellInfo, 586
nmrCellInfo, 584
nmrArfcn, 585
nmrBsic, 585
nmrCellID, 586
nmrLac, 586
nmrPlmn, 586
nmrRxLev, 586
nmrInst
GERANInfo, 255
nas_GERANInfo, 445
nmrLac
nas_nmrCellInfo, 484
nmrCellInfo, 586
nmrPlmn
nas_nmrCellInfo, 484
nmrCellInfo, 586
nmrRxLev
nas_nmrCellInfo, 484
nmrCellInfo, 586
noReplyTimer
callFWExtInfo, 150
callFWInfo, 151
voice_callFWExtInfo, 1311
voice_callFWInfo, 1312
Non-service specific APIs (SWI), 51
notifType
voiceSUPSNotification, 1384
Notification
_SwiAvmsEventInd, 100
notification
omaDmNotificationsTlv, 595
unpack_omaDmNotificationsTlv_t, 1161
notificationType
SMSEtwsMessage, 914
sMSEtwsMessageInfo, 915
notused
unpack_dms_SetCrashAction_t, 1059
num_instances
_qaQmi3GPP2BroadcastCfgInfo, 71
_qaQmi3GPPBroadcastCfgInfo, 71
custSettingList, 212
DMScustSettingList, 230
sms_qaQmi3GPP2BroadcastCfgInfo, 905
sms_qaQmi3GPPBroadcastCfgInfo, 905
numApp
slotInf, 879
slotInfo, 881
uim_slotInfo, 998
numCrashes
CrashInfo, 192
crashInformation, 194
numEntries
CurrentImgList, 201
unpack_dms_SLQSSwiGetFirmwareCurr_t, 1071
numFeatures
personalizationStatus, 746
uim_personalizationStatus, 990
numFiles
registerRefresh, 820
uim_registerRefresh, 994
NumFlows

unpack_qos_SLQSSetQosEventCallback_ind_t, 1182
numInstance
nas_operatorPLMNList, 486
nas_PLMNNetworkName, 493
operatorPLMNList, 597
PLMNNetworkName, 755
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, 234
getCallFWExtInfo, 263
getCallFWInfo, 264
getMsgWaitingInfo, 282
homeSIDNID, 310
msgWaitingInfo, 419
nas_homeSIDNID, 456
nas_roamIndList, 502
nas_SccRxInfo, 508
PCSCFFQDNAddressList, 740
PCSCFIPv4ServerAddressList, 741
roamIndList, 826
SccRxInfo, 839
sms_getMsgWaitingInfo, 900
voice_arrAlertingPattern, 1300
voice_arrAlertingType, 1301
voice_arrAlphaID, 1301
voice_arrCalledPartyNum, 1302
voice_arrCallEndReason, 1302
voice_arrCallInfo, 1303
voice_arrConnectPartyNum, 1303
voice_arrDiagInfo, 1304
voice_arrRedirPartyNum, 1304
voice_arrRemotePartyName, 1305
voice_arrRemotePartyNum, 1306
voice_arrSvcOption, 1306
voice_arrUUSInfo, 1307
voice_getCallFWExtInfo, 1324
voice_getCallFWInfo, 1325
wds_DomainNameList, 1403
wds_PCSCFFQDNAddressList, 1406
wds_PCSCFIPv4ServerAddressList, 1407
numLen
calledPartyInfo, 145
callFWExtInfo, 150
callFWInfo, 151
callingPartyInfo, 154
peerNumberInfo, 745
redirNumInfo, 819
remotePartyNum, 822
voice_callFWExtInfo, 1311
voice_callFWInfo, 1312
voice_peerNumberInfo, 1327
voice_remotePartyNum, 1330
numOfFiles
uim_refreshevent, 993
UIMRefreshEvent, 1013
numOfRoutes
sms_setRoutesReq, 910
smsSetRoutesReq, 925
numOpt
DHCPOptionList, 226
wds_DHCPLeaseOptTlv, 1397
wds_DHCPv4OptionList, 1401
WdsDHCPv4OptionList, 1420
wdsDhcpv4OptionList, 1419
numPI
peerNumberInfo, 745
voice_peerNumberInfo, 1327
NumPilots
nas_PilotSetData, 491
PilotSetData, 752
numPlan
calledPartyInfo, 145
callFWExtInfo, 150
callingPartyInfo, 154
connectNumInfo, 191
peerNumberInfo, 745
redirNumInfo, 819
voice_callFWExtInfo, 1311
voice_connectNumInfo, 1319
voice_peerNumberInfo, 1327
numPresInd
connectNumInfo, 191
voice_connectNumInfo, 1319
numQosFlow
sQosStat, 928
unpack_qos_SLQSQosSwiReadDataStats_t, 1181
numRadioInterfaces
nas_servSystem, 510
servSystem, 848
NumRxFilters
unpack_qos_QosFlowInfo_t, 1177
numSI
peerNumberInfo, 745
voice_peerNumberInfo, 1327
numSlot
cardStatus, 156
uim_cardStatus, 982
NumSupUSBComps
unpack_dms_GetUSBComp_t, 1057
NumTxFilters
unpack_qos_QosFlowInfo_t, 1177
numType
calledPartyInfo, 145

callFWExtInfo, 150
callingPartyInfo, 154
connectNumInfo, 191
peerNumberInfo, 746
redirNumInfo, 819
voice_callFWExtInfo, 1311
voice_connectNumInfo, 1320
voice_peerNumberInfo, 1327
number
calledPartyInfo, 145
callFWExtInfo, 150
callFWInfo, 151
callingPartyInfo, 154
ECTNum, 239
peerNumberInfo, 745
redirNumInfo, 819
voice_callFWExtInfo, 1311
voice_callFWInfo, 1312
voice_peerNumberInfo, 1327
numberPlan
callFwdTypeAndPlan, 148
voice_callFwdTypeAndPlan, 1309
numberType
callFwdTypeAndPlan, 148
voice_callFwdTypeAndPlan, 1309
NxtHdrProto
unpack_qos_swiQosFilter_t, 1187

OKtoRefresh
pack_uim_SLQSUIMRefreshOK_t, 690
UIMRefreshOKReq, 1015
OMADMCancelSession
qaGobiApiOmadm.h, 1794
OMADMEnabled
unpack_swioma_SLQSOMADMGetSettings_t, 1211
OMADMGetPendingNIA
qaGobiApiOmadm.h, 1795
OMADMGetSessionInfo
qaGobiApiOmadm.h, 1795
OMADMStartSession
qaGobiApiOmadm.h, 1797
OTASPStatus
voiceOTASPStatusInfo, 1367
oddEvenInd
calledPartySubAdd, 146
voice_calledPartySubAdd, 1308
OfflineReason
unpack_dms_GetPower_t, 1053
offset
readTransparentInfo, 817
uim_readTransparentInfo, 991
oldPINLen
changeUIMPIN, 180
uim_changeUIMPIN, 983
oldPINVal
changeUIMPIN, 180
uim_changeUIMPIN, 983
oldPasswd
pack_voice_SLQSVoiceSetCallBarringPassword_t, 707
voiceSetCallBarringPwdInfo, 1371
oldValue
pack_dms_UIMChangePIN_t, 607
omaDmConfig
sessionInfo, 849
sessionInfoExt, 849
omaDmConfigTlv, 587
alertmsg, 587
alertmsglength, 587
state, 587
userInputReq, 587
userInputTimeout, 587
omaDmConfigTlvExt, 588
alertmsg, 590
alertmsglength, 590
state, 590
userInputReq, 590
userInputTimeout, 590
omaDmFota
sessionInfo, 849
sessionInfoExt, 849
omaDmFotaTlv, 590
description, 591
descriptionlength, 591
fwdloadsize, 591
fwloadComplete, 592
namelength, 592
package_name, 592
sessionType, 592
severity, 592
state, 592
updateCompleteStatus, 592
userInputReq, 592
userInputTimeout, 592
version, 592
versionlength, 592
omaDmFotaTlvExt, 592
description, 594
descriptionlength, 594
fumoResultCode, 594
namelength, 594
package_name, 594
packageSize, 594
receivedBytes, 594
reserved, 594
state, 594
userInputTimeout, 594
version, 594
versionlength, 594
omaDmNotifications
sessionInfo, 849
omaDmNotificationsTlv, 594
notification, 595
sessionStatus, 595
opaqueId
pack_loc_SLQSLOCInjectSensorData_t, 625

Open Mobile Alliance Service (OMA), 47
operatingMode
dms_OperatingModeTlv, 228
OperatingModeTlv
unpack_dms_SetEventReport_ind_t, 1061
operation
depersonalizationInformation, 222
pack_pds_SetPDSDefaults_t, 656
uim_depersonalizationInformation, 984
OperationMode
unpack_dms_GetPower_t, 1053
operatorNameString, 595
PLMNName, 595
OperatorPLMNData, 596
lac1, 596
lac2, 596
mcc, 596
mnc, 596
PLMNRecID, 596
operatorPLMNList, 596
numInstance, 597
PLMNData, 597
optCode
DHCPOption, 225
wds_DHCPOpt, 1399
wds_DHCPv4Option, 1400
WdsDHCPv4Option, 1418
wdsDhcpv4Option, 1418
optList
wds_DHCPLeaseOptTlv, 1397
optListData
wds_DHCPLeaseOptTlv, 1397
optVal
wds_DHCPv4Option, 1400
WdsDHCPv4Option, 1419
wdsDhcpv4Option, 1418
optValLen
DHCPOption, 225
wds_DHCPOpt, 1399
wds_DHCPv4Option, 1401
WdsDHCPv4Option, 1419
wdsDhcpv4Option, 1418
OriginateUSSD
qaGobiApiVoice.h, 1934
otaMsgTlv
NASQmiCbkNasSwiOTAMessageInd, 566

p1Status
unpack_dms_UIMGetPINStatus_t, 1080
p1UnblockRetriesLeft
unpack_dms_UIMGetPINStatus_t, 1080
p1VerifyRetriesLeft
unpack_dms_UIMGetPINStatus_t, 1080
p2Status
unpack_dms_UIMGetPINStatus_t, 1080
p2UnblockRetriesLeft
unpack_dms_UIMGetPINStatus_t, 1080
p2VerifyRetriesLeft
unpack_dms_UIMGetPINStatus_t, 1080
p3GPP2Pri
swiQosFlow, 949
p3GPP2TimeInfo
GetNetworkTimeResp, 282
unpack_nas_SLQSGetNetworkTime_t, 1122
p3GPPImCn
swiQosFlow, 949
p3GPPResResidualBER
swiQosFlow, 949
p3GPPSigInd
swiQosFlow, 949
p3GPPTimeInfo
GetNetworkTimeResp, 282
unpack_nas_SLQSGetNetworkTime_t, 1122
p3GPPTraHdlPri
swiQosFlow, 949
p3GppNetworkInfoInstances
unpack_nas_PerformNetworkScan_t, 1117
p3GppNetworkInstanceSize
unpack_nas_PerformNetworkScan_t, 1117
p3gppRelease
_slqs3GPPConfigItem, 83
pack_wds_SLQSSet3GPPConfigItem_t, 730
pAAASPI
pack_wds_SetMobileIPProfile_t, 723
PACK_WDS_IPV4
wds.h, 2098
PACK_WDS_IPV6
wds.h, 2098
pAMRStatus
pack_voice_SLQSVoiceGetConfig_t, 703
voiceGetConfigReq, 1359
pAPN
_SLQSAVMSSettingsAPNInfo, 86
pAPNClass
LibpackProfile3GPP, 341
LibPackprofile_3GPP, 353
Profile3GPP, 765
pAPNClass3GPP2
LibpackProfile3GPP2, 347
LibPackprofile_3GPP2, 358
Profile3GPP2, 770
pAPNDisabledFlag
LibpackProfile3GPP, 341
LibPackprofile_3GPP, 353
Profile3GPP, 765
pAPNEnabled3GPP2
LibpackProfile3GPP2, 347
LibPackprofile_3GPP2, 358
Profile3GPP2, 770
pAPNInfo
_SLQSAVMSSetSettingsReqPara, 85
_SLQSAVMSSettings, 85
pAPNName
LibpackProfile3GPP, 341
LibPackprofile_3GPP, 353
Profile3GPP, 765
qmiWdsRunTimeSettings, 810

swiPDPRuntimeSettingsResp, 944
pAPNnameSize
LibpackProfile3GPP, 342
LibPackprofile_3GPP, 353
Profile3GPP, 765
PASSWORD_LENGTH
qaGobiApiVoice.h, 1933
pAVRXAVCHeadroom
RXAVCList, 830
pAVRXAVCSens
RXAVCList, 831
pAccelAcceptReady
QmiCbkLocSensorStreamingInd, 798
pAccelSamplesAccepted
QmiCbkLocInjectSensorDataInd, 790
pAccelTempAcceptReady
QmiCbkLocSensorStreamingInd, 798
pAccelTempSamplesAccepted
QmiCbkLocInjectSensorDataInd, 790
pAcceleroData
LocInjectSensorDataReq, 390
pAcceleroTempData
LocInjectSensorDataReq, 390
pAcceleroTimeSrc
LocInjectSensorDataReq, 390
pAccuracy
unpack_pds_GetPDSDefaults_t, 1162
pAcqOrder
acqOrderPref, 115
nas_acqOrderPref, 420
pAcqOrderPref
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 652
pActPilotPNElements
nas_NetworkStat1x, 481
NetworkStat1x, 581
pAddCDMASysInfo
nasGetSysInfoResp, 543
nasSysInfo, 575
unpack_nas_SLQSGetSysInfo_t, 1135
unpack_nas_SLQSSysInfoCallback_ind_t, 1157
pAddGSMSysInfo
nasGetSysInfoResp, 543
nasSysInfo, 575
unpack_nas_SLQSGetSysInfo_t, 1135
unpack_nas_SLQSSysInfoCallback_ind_t, 1157
pAddHDRSysInfo
nasGetSysInfoResp, 543
nasSysInfo, 575
unpack_nas_SLQSGetSysInfo_t, 1135
unpack_nas_SLQSSysInfoCallback_ind_t, 1157
pAddLTESysInfo
nasGetSysInfoResp, 544
nasSysInfo, 575
unpack_nas_SLQSGetSysInfo_t, 1135
unpack_nas_SLQSSysInfoCallback_ind_t, 1157
pAddWCDMASysInfo
nasGetSysInfoResp, 544
nasSysInfo, 575
unpack_nas_SLQSGetSysInfo_t, 1135
unpack_nas_SLQSSysInfoCallback_ind_t, 1157
pAddrAllocPref
LibpackProfile3GPP, 341
LibPackprofile_3GPP, 353
Profile3GPP, 765
pAddress
pack_wds_SetMobileIPProfile_t, 723
pAirTimer
pack_voice_SLQSVoiceGetConfig_t, 703
voiceGetConfigReq, 1359
pAirTimerCnt
unpack_voice_SLQSVoiceGetConfig_t, 1247
voiceGetConfigResp, 1361
pAirTimerConfig
pack_voice_SLQSVoiceSetConfig_t, 709
voiceSetConfigReq, 1373
pAirTimerStatus
unpack_voice_SLQSVoiceSetConfig_t, 1250
voiceSetConfigResp, 1375
pAlertPriority
cdmaMsgDecodingParams, 166
cdmaMsgDecodingParamsExt, 170
pAlertType
unpack_voice_SLQSVoiceGetCallInfo_t, 1236
voiceCallInfoResp, 1338
pAlertingPattern
unpack_voice_SLQSVoiceGetCallInfo_t, 1236
voiceCallInfoResp, 1338
pAllowLinger
LibpackProfile3GPP2, 347
LibPackprofile_3GPP2, 358
Profile3GPP2, 770
pAlphaID
SMSAsyncRawSend_s, 912
pAlphaIDInfo
unpack_voice_SLQSOriginateUSSD_t, 1225
unpack_voice_SLQSVoiceDialCall_t, 1227
unpack_voice_SLQSVoiceGetCallBarring_t, 1232
unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 1233
unpack_voice_SLQSVoiceGetCallInfo_t, 1236
unpack_voice_SLQSVoiceGetCallWaiting_t, 1238
unpack_voice_SLQSVoiceGetCLIP_t, 1239
unpack_voice_SLQSVoiceGetCLIR_t, 1240
unpack_voice_SLQSVoiceGetCNAP_t, 1242
unpack_voice_SLQSVoiceGetCOLP_t, 1243
unpack_voice_SLQSVoiceGetCOLR_t, 1245
unpack_voice_SLQSVoiceSetCallBarringPassword-_t, 1249
unpack_voice_SLQSVoiceSetSUPSService_t, 1252
USSResp, 1294
voiceCallInfoResp, 1338
voiceCallResponseParams, 1341
voiceGetCallBarringResp, 1347
voiceGetCallFWResp, 1350

voiceGetCallWaitInfo, 1351
voiceGetCLIPResp, 1352
voiceGetCLIRResp, 1353
voiceGetCNAPResp, 1355
voiceGetCOLPResp, 1356
voiceGetCOLRResp, 1357
voiceSetCallBarringPwdResp, 1372
voiceSetSUPSServiceResp, 1379
voiceSUPSInfo, 1382
pAlphaIdentifier
USSDNoWaitIndicationInfo, 1293
pAltitudeAssumed
QmiCbkLocPositionReportInd, 796
unpack_loc_PositionRpt_Ind_t, 1100
pAltitudeSrcInfo
LocInjectPositionReq, 388
pAltitudeWrtEllipsoid
LocInjectPositionReq, 388
pack_pds_SLQSPDSInjectPositionData_t, 661
PDSPositionData, 743
QmiCbkLocBestAvailPosInd, 784
QmiCbkLocPositionReportInd, 796
unpack_loc_BestAvailPos_Ind_t, 1088
unpack_loc_PositionRpt_Ind_t, 1100
pAltitudeWrtMeanSeaLevel
LocInjectPositionReq, 388
QmiCbkLocBestAvailPosInd, 784
QmiCbkLocPositionReportInd, 796
unpack_loc_BestAvailPos_Ind_t, 1088
unpack_loc_PositionRpt_Ind_t, 1100
pAltitudeWrtSealevel
pack_pds_SLQSPDSInjectPositionData_t, 661
PDSPositionData, 743
pAmrMode
GetIMSVoIPConfigResp, 276
imsVoIPConfigInfo, 328
SetIMSVoIPConfigReq, 860
pAmrOctetAligned
GetIMSVoIPConfigResp, 276
imsVoIPConfigInfo, 328
SetIMSVoIPConfigReq, 860
pAmrWBMode
GetIMSVoIPConfigResp, 276
imsVoIPConfigInfo, 328
SetIMSVoIPConfigReq, 860
pAmrWBOctetAligned
GetIMSVoIPConfigResp, 276
imsVoIPConfigInfo, 328
SetIMSVoIPConfigReq, 860
pAmrWbEnable
GetIMSVoIPConfigResp, 276
imsVoIPConfigInfo, 328
SetIMSVoIPConfigReq, 860
pApnString
LibpackProfile3GPP2, 347
LibPackprofile_3GPP2, 358
Profile3GPP2, 770
pApnStringSize
LibpackProfile3GPP2, 347
LibPackprofile_3GPP2, 358
Profile3GPP2, 770
pApnname
pack_wds_SetDefaultProfile_t, 719
pAppName
loc_LocApplicationInfo, 373
LocApplicationInfo, 381
pAppPriority
LibpackProfile3GPP2, 347
LibPackprofile_3GPP2, 358
Profile3GPP2, 770
pAppProvider
loc_LocApplicationInfo, 373
LocApplicationInfo, 381
pAppSubType
HDRProtSubtypResp, 303
unpack_nas_SLQSSwiGetHDRProtSubtype_t, 1151
pAppType
LibpackProfile3GPP2, 347
LibPackprofile_3GPP2, 358
Profile3GPP2, 770
pAppVersion
loc_LocApplicationInfo, 373
LocApplicationInfo, 381
pApplication
pack_nas_SetCDMANetworkParameters_t, 630
pApplicationInfo
LOCStartReq, 392
pack_loc_Start_t, 627
pArrAlertingPattern
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1230
voiceGetAllCallInfo, 1345
voiceSetAllCallStatusCbkInfo, 1370
pArrAlertingType
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1230
voiceGetAllCallInfo, 1345
voiceSetAllCallStatusCbkInfo, 1370
pArrAlphaID
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1230
voiceGetAllCallInfo, 1345
voiceSetAllCallStatusCbkInfo, 1370
pArrCallEndReason
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1230
voiceGetAllCallInfo, 1345
voiceSetAllCallStatusCbkInfo, 1370
pArrCallInfo
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1230
voiceGetAllCallInfo, 1345
pArrCalledPartyNum
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1230
voiceGetAllCallInfo, 1345
voiceSetAllCallStatusCbkInfo, 1370
pArrConnectPartyNum
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1230
voiceGetAllCallInfo, 1345
voiceSetAllCallStatusCbkInfo, 1370

pArrDiagInfo
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1230
voiceGetAllCallInfo, 1345
voiceSetAllCallStatusCbkInfo, 1370
pArrRedirPartyNum
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1230
voiceGetAllCallInfo, 1345
voiceSetAllCallStatusCbkInfo, 1370
pArrRemotePartyName
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1230
voiceGetAllCallInfo, 1345
voiceSetAllCallStatusCbkInfo, 1370
pArrRemotePartyNum
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1230
voiceGetAllCallInfo, 1345
voiceSetAllCallStatusCbkInfo, 1370
pArrSvcOption
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1230
voiceGetAllCallInfo, 1345
voiceSetAllCallStatusCbkInfo, 1370
pArrUUSInfo
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1230
voiceGetAllCallInfo, 1346
pAuth
pack_wds_SLQSStartDataSession_t, 734
pAuthPassword
LibpackProfile3GPP2, 347
LibPackprofile_3GPP2, 358
Profile3GPP2, 770
pAuthPassword_tSize
LibPackprofile_3GPP2, 358
pAuthPasswordSize
LibpackProfile3GPP2, 347
Profile3GPP2, 770
pAuthProtocol
LibpackProfile3GPP2, 347
LibPackprofile_3GPP2, 358
Profile3GPP2, 770
pAuthRetryCount
LibpackProfile3GPP2, 347
LibPackprofile_3GPP2, 359
Profile3GPP2, 771
pAuthTimeout
LibpackProfile3GPP2, 347
LibPackprofile_3GPP2, 359
Profile3GPP2, 771
pAuthenticateResult
UIMAuthenticateResp, 1002
unpack_uim_SLQSUIMAuthenticate_t, 1216
pAuthentication
qmiWdsRunTimeSettings, 810
ssdatasession_params, 932
pAuthenticationPref
LibpackProfile3GPP, 342
LibPackprofile_3GPP, 353
Profile3GPP, 765
pAutoAnsStatus
unpack_voice_SLQSVoiceSetConfig_t, 1250
voiceSetConfigResp, 1375
pAutoAnswer
pack_voice_SLQSVoiceGetConfig_t, 703
pack_voice_SLQSVoiceSetConfig_t, 709
voiceGetConfigReq, 1359
voiceSetConfigReq, 1374
pAutoAnswerStat
unpack_voice_SLQSVoiceGetConfig_t, 1247
voiceGetConfigResp, 1361
pAutoConnect
_SLQSAVMSSetSettingsReqPara, 85
_SLQSAVMSSettings, 85
pAutoReboot
_SLQSAVMSSetSettingsReqPara, 85
_SLQSAVMSSettings, 85
pAutoSelection
UIMGetConfigurationResp, 1006
unpack_uim_SLQSUIMGetConfiguration_t, 1218
pAutosdm
_SLQSOMADMSettings, 93
_SLQSOMADMSettingsReqParams, 94
_SLQSOMADMSettingsReqParams3, 95
pack_swioma_SLQSOMADMSetSettings_t, 681
pBandPref
_sysSelectPrefInfo, 104
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 652
unpack_nas_SLQSGetSysSelectionPref_t, 1140
pBdsSVInfo
LocDelAssDataReq, 381
pack_loc_Delete_Assist_Data_t, 614
pBearerID
QosFlowInfo, 814
pBearerId
swiPDPRuntimeSettingsResp, 944
pBearerTech
DataBearerTechExt, 215
pBroadcast
pack_nas_SetCDMANetworkParameters_t, 630
pBroadcastConfig
pack_sms_SLQSSetSmsBroadcastConfig_t, 675
unpack_sms_SLQSGetSmsBroadcastConfig_t, 1201
pBurstDTMFLengths
pack_voice_SLQSVoiceBurstDTMF_t, 697
voiceBurstDTMFInfo, 1335
pCCResType
unpack_voice_SLQSVoiceGetCallBarring_t, 1232
unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 1233
unpack_voice_SLQSVoiceGetCallWaiting_t, 1238
unpack_voice_SLQSVoiceGetCLIP_t, 1239
unpack_voice_SLQSVoiceGetCLIR_t, 1240
unpack_voice_SLQSVoiceGetCNAP_t, 1242
unpack_voice_SLQSVoiceGetCOLP_t, 1243
unpack_voice_SLQSVoiceGetCOLR_t, 1245
unpack_voice_SLQSVoiceSetCallBarringPassword-_t, 1249

voiceGetCallBarringResp, 1348
voiceGetCallFWResp, 1350
voiceGetCallWaitInfo, 1351
voiceGetCLIPResp, 1352
voiceGetCLIRResp, 1354
voiceGetCNAPResp, 1355
voiceGetCOLPResp, 1356
voiceGetCOLRResp, 1357
voiceSetCallBarringPwdResp, 1372
pCCResultType
unpack_voice_SLQSVoiceDialCall_t, 1227
unpack_voice_SLQSVoiceSetSUPSService_t, 1252
voiceCallResponseParams, 1341
voiceSetSUPSServiceResp, 1379
pCCSUPSType
unpack_voice_SLQSVoiceDialCall_t, 1227
unpack_voice_SLQSVoiceGetCallBarring_t, 1232
unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 1233
unpack_voice_SLQSVoiceGetCallWaiting_t, 1238
unpack_voice_SLQSVoiceGetCLIP_t, 1239
unpack_voice_SLQSVoiceGetCLIR_t, 1241
unpack_voice_SLQSVoiceGetCNAP_t, 1242
unpack_voice_SLQSVoiceGetCOLP_t, 1243
unpack_voice_SLQSVoiceGetCOLR_t, 1245
unpack_voice_SLQSVoiceSetCallBarringPassword-_t, 1249
unpack_voice_SLQSVoiceSetSUPSService_t, 1252
voiceCallResponseParams, 1341
voiceGetCallBarringResp, 1348
voiceGetCallFWResp, 1350
voiceGetCallWaitInfo, 1351
voiceGetCLIPResp, 1352
voiceGetCLIRResp, 1354
voiceGetCNAPResp, 1355
voiceGetCOLPResp, 1356
voiceGetCOLRResp, 1357
voiceSetCallBarringPwdResp, 1372
voiceSetSUPSServiceResp, 1379
pCCSuppsType
unpack_voice_SLQSOriginateUSSD_t, 1225
USSResp, 1294
pCDMABroadcastConfig
pack_sms_SLQSSetSmsBroadcastConfig_t, 675
unpack_sms_SLQSGetSmsBroadcastConfig_t, 1201
pCDMAChannel
nasGet3GPP2SubscriptionInfoResp, 538
unpack_nas_SLQSNasGet3GPP2Subscription_t, 1141
pCDMAECIODelta
pack_nas_SLQSNasConfigSigInfo2_t, 638
setSignalStrengthInfo, 871
pCDMAECIOThresh
pack_nas_SLQSNasConfigSigInfo2_t, 638
setSignalStrengthInfo, 871
pCDMAECIOThreshList
CDMAECIOThresh, 163
nas_CDMAECIOThresh, 425
pCDMAFrameErrRate
GetErrRateResp, 271
unpack_nas_SLQSGetErrorRate_t, 1121
pCDMAInfo
nasCellLocationInfoResp, 536
unpack_nas_SLQSNasGetCellLocationInfo_t, 1142
pCDMARSSIDelta
pack_nas_SLQSNasConfigSigInfo2_t, 638
setSignalStrengthInfo, 871
pCDMARSSIThresh
pack_nas_SLQSNasConfigSigInfo2_t, 638
setSignalStrengthInfo, 871
pCDMARSSIThreshList
CDMARSSIThresh, 172
nas_CDMARSSIThresh, 426
pCDMASSInfo
nasGetSigInfoResp, 541
pCDMASigInfo
nasSigInfo, 570
unpack_nas_SLQSNasSigInfoCallback_ind_t, 1147
pCDMASrvStatusInfo
nasGetSysInfoResp, 544
nasSysInfo, 575
unpack_nas_SLQSGetSysInfo_t, 1135
unpack_nas_SLQSSysInfoCallback_ind_t, 1157
pCDMASysInfo
nasGetSysInfoResp, 544
nasSysInfo, 575
unpack_nas_SLQSGetSysInfo_t, 1135
unpack_nas_SLQSSysInfoCallback_ind_t, 1157
pCLIPResp
unpack_voice_SLQSVoiceGetCLIP_t, 1239
voiceGetCLIPResp, 1352
pCLIPstatus
voiceSUPSInfo, 1382
pCLIRCause
voiceInfoRec, 1364
pCLIRResp
unpack_voice_SLQSVoiceGetCLIR_t, 1241
voiceGetCLIRResp, 1354
pCLIRType
pack_voice_SLQSVoiceDialCall_t, 698
voiceCallRequestParams, 1340
pCLIRstatus
voiceSUPSInfo, 1382
PCMparams, 739
iFaceTab, 739
iFaceTabLen, 739
pCNAPResp
unpack_voice_SLQSVoiceGetCNAP_t, 1242
voiceGetCNAPResp, 1355
pCNAPstatus
voiceSUPSInfo, 1382

pCOLPResp
unpack_voice_SLQSVoiceGetCOLP_t, 1243
voiceGetCOLPResp, 1356
pCOLPstatus
voiceSUPSInfo, 1382
pCOLRResp
unpack_voice_SLQSVoiceGetCOLR_t, 1245
voiceGetCOLRResp, 1357
pCOLRstatus
voiceSUPSInfo, 1382
PCSCFAddrPCO
unpack_wds_SLQSGetRuntimeSettings_t, 1278
PCSCFFQDNAddrList
unpack_wds_SLQSGetRuntimeSettings_t, 1278
PCSCFFQDNAddress, 740
fqdnAddr, 740
fqdnLen, 740
PCSCFFQDNAddressList, 740
numInstances, 740
pcsfFQDNAddress, 740
PCSCFIPv4ServerAddressList, 741
numInstances, 741
pscsfIPv4Addr, 741
pCSCFPortName
imsRegMgrConfigInfo, 324
SetRegMgrConfigReq, 867
pCSCFPortNameLen
SetRegMgrConfigReq, 867
pCSGID
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 652
pCUGIndex
voiceSUPSNotification, 1384
pCUGInfo
pack_voice_SLQSVoiceDialCall_t, 698
voiceCallRequestParams, 1340
pCallBarPasswd
voiceSUPSInfo, 1382
pCallBarringPasswd
pack_voice_SLQSVoiceSetSUPSService_t, 711
voiceSetSUPSServiceReq, 1378
pCallEndReason
getDUNCallInfoResp, 269
pCallFWNum
voiceSUPSInfo, 1382
pCallFWTimerVal
voiceSUPSInfo, 1382
pCallForwardingNumber
pack_voice_SLQSVoiceSetSUPSService_t, 711
voiceSetSUPSServiceReq, 1378
pCallFwdInfo
voiceSUPSInfo, 1382
pCallFwdTypeAndPlan
pack_voice_SLQSVoiceSetSUPSService_t, 711
voiceSetSUPSServiceReq, 1378
pCallID
burstDTMFInfo, 142
pack_voice_SLQSVoiceManageCalls_t, 705
pack_voice_SLQSVoiceSendFlash_t, 706
pack_voice_SLQSVoiceStartContDTMF_t, 712
unpack_voice_SLQSVoiceBurstDTMF_t, 1226
unpack_voice_SLQSVoiceDialCall_t, 1227
unpack_voice_SLQSVoiceGetCallBarring_t, 1232
unpack_voice_SLQSVoiceGetCallForwarding-
Status_t, 1233
unpack_voice_SLQSVoiceGetCallWaiting_t, 1238
unpack_voice_SLQSVoiceGetCLIP_t, 1239
unpack_voice_SLQSVoiceGetCLIR_t, 1240
unpack_voice_SLQSVoiceGetCNAP_t, 1242
unpack_voice_SLQSVoiceGetCOLP_t, 1243
unpack_voice_SLQSVoiceGetCOLR_t, 1245
unpack_voice_SLQSVoiceSendFlash_t, 1248
unpack_voice_SLQSVoiceSetCallBarringPassword-
_t, 1249
unpack_voice_SLQSVoiceSetSUPSService_t,
1252
unpack_voice_SLQSVoiceStartContDTMF_t, 1252
voice_burstDTMFInfo, 1307
voiceCallResponseParams, 1341
voiceContDTMFinfo, 1341
voiceFlashInfo, 1343
voiceGetCallBarringResp, 1348
voiceGetCallFWResp, 1350
voiceGetCallWaitInfo, 1351
voiceGetCLIPResp, 1352
voiceGetCLIRResp, 1353
voiceGetCNAPResp, 1355
voiceGetCOLPResp, 1356
voiceGetCOLRResp, 1357
voiceManageCallsReq, 1365
voiceSetCallBarringPwdResp, 1372
voiceSetSUPSServiceResp, 1379
voiceSUPSInfo, 1382
pCallId
pack_voice_SLQSVoiceAnswerCall_t, 696
pack_voice_SLQSVoiceEndCall_t, 699
unpack_voice_SLQSOriginateUSSD_t, 1225
unpack_voice_SLQSVoiceAnswerCall_t, 1226
unpack_voice_SLQSVoiceEndCall_t, 1228
USSResp, 1294
voiceAnswerCall, 1334
pCallInfo
unpack_voice_SLQSVoiceGetCallInfo_t, 1236
voiceCallInfoResp, 1338
pCallPartySubAdd
pack_voice_SLQSVoiceDialCall_t, 698
voiceCallRequestParams, 1340
pCallType
pack_voice_SLQSVoiceDialCall_t, 698
voiceCallRequestParams, 1340
pCallWaitInd
voiceInfoRec, 1364
pCallbackAddr
cdmaMsgEncodingParams, 172
pCallbkAddr
cdmaMsgDecodingParams, 166

cdmaMsgDecodingParamsExt, 170
pCallbkAddrLength
cdmaMsgDecodingParams, 166
cdmaMsgDecodingParamsExt, 170
pCalledPartyInfo
voiceInfoRec, 1364
pCallerIDInfo
voiceInfoRec, 1364
pCallerNameInfo
voiceInfoRec, 1364
pCallingPartyInfo
voiceInfoRec, 1364
pCardResult
UIMAuthenticateResp, 1002
UIMGetFileAttributesResp, 1007
UIMReadTransparentResp, 1011
unpack_uim_ReadTransparent_t, 1214
unpack_uim_SLQSUIMAuthenticate_t, 1216
unpack_uim_SLQSUIMGetFileAttributes_t, 1219
pCardStatus
UIMGetCardStatusResp, 1004
unpack_uim_GetCardStatus_t, 1213
unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t, 1220
pCcResultType
unpack_voice_SLQSOriginateUSSD_t, 1225
USSResp, 1294
pCellDataMask
pack_pds_ResetPDSData_t, 654
pCellDb
LocDelAssDataReq, 381
pack_loc_Delete_Assist_Data_t, 614
pChangeDuration
nasInitNetworkReg, 550
pack_nas_SLQSInitiateNetworkRegistration_t, 635
pChannelRate
getDUNCallInfoResp, 269
pChgDuration
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 652
pClkInfo
LocDelAssDataReq, 381
pack_loc_Delete_Assist_Data_t, 614
pCodecSTGain
GetAudioPathConfigResp, 259
SetAudioPathConfigReq, 851
pColorCode
nasGetHDRColorCodeResp, 539
unpack_nas_SLQSNasGetHDRColorCode_t, 1143
pConfidence
LocSetCradleMountReq, 391
pConfigAltitudeAssumed
LOCStartReq, 392
pack_loc_Start_t, 627
pConfigurationMask
pack_uim_SLQSUIMGetConfiguration_t, 686
UIMGetConfigurationReq, 1005
pConnectNumInfo
unpack_voice_SLQSVoiceGetCallInfo_t, 1236
voiceCallInfoResp, 1338
voiceInfoRec, 1364
pConnectionRetryTimers
_SLQSAVMSSetSettingsReqPara, 85
_SLQSAVMSSettings, 85
pConnectionStatus
getDUNCallInfoResp, 269
pContextId
swiPDPRuntimeSettingsResp, 944
pCrashInfo
CrashInfoParams, 193
pCrashString
CrashInfo, 192
pCreateProfileOut
unpack_wds_SLQSCreateProfile_t, 1268
pCsgId
nasNetworkReject, 555
pCurAMRConfig
unpack_voice_SLQSVoiceGetConfig_t, 1247
voiceGetConfigResp, 1361
pCurDataBearerTechnology
dataBearers, 212
pCurPrefVoiceSO
unpack_voice_SLQSVoiceGetConfig_t, 1247
voiceGetConfigResp, 1361
pCurProfile
pack_wds_SLQSCreateProfile_t, 724
pCurVoiceDomainPref
unpack_voice_SLQSVoiceGetConfig_t, 1247
voiceGetConfigResp, 1361
pCurVoicePrivacyPref
unpack_voice_SLQSVoiceGetConfig_t, 1247
voiceGetConfigResp, 1361
pCurrChannelRateInd
pack_wds_SLQSWdsSetEventReport_t, 736
wdsSetEventReportReq, 1428
pCurrDataBearerTechInd
pack_wds_SLQSWdsSetEventReport_t, 736
wdsSetEventReportReq, 1428
pCurrImgInfo
CurrentImgList, 201
unpack_dms_SLQSSwiGetFirmwareCurr_t, 1071
pCurrNetworkInfo
CurrDataSysStat, 200
pCurrPrefDataSysInd
pack_wds_SLQSWdsSetEventReport_t, 736
wdsSetEventReportReq, 1428
pCurrTTYMode
unpack_voice_SLQSVoiceGetConfig_t, 1247
voiceGetConfigResp, 1361
pCurrentChannelRXRate
WdsConnectionRateElmnts, 1415
pCurrentChannelTXRate
WdsConnectionRateElmnts, 1415
pCurrentPersonality
HDRPersonalityInd, 302
HDRPersonalityResp, 302

unpack_nas_SLQSSwiGetHDRPersonality_t, 1151
pCurrentPrsnlty
HDRProtSubtypResp, 303
unpack_nas_SLQSSwiGetHDRProtSubtype_t, 1151
pCurrentmitigationLvl
TmdGetMitigationLvlResp, 968
pCustSettingInfo
DMSgetCustomFeatureV2, 231
getCustomFeatureV2, 264
pCustSettingList
DMSgetCustomFeatureV2, 231
getCustomFeatureV2, 264
pCustomSCP
pack_nas_SetCDMANetworkParameters_t, 630
pCwtMute
SetM2MAudioProfileReq, 863
SetM2MAVMuteReq, 864
pDDTMInd
nasIndicationRegisterReq, 549
pack_nas_SLQSNasIndicationRegisterExt_t, 643
pDHCPRelayEnabled
custFeaturesInfo, 208
custFeaturesSetting, 210
PDOP
loc_precisionDilution, 374
precisionDilution_s, 758
PDPType
unpack_wds_SLQSGetRuntimeSettings_t, 1278
pDRCParams
GetHRPDStatsResp, 272
unpack_nas_SLQSSwiGetHRPDStats_t, 1152
PDS_SRV
qaGobiApiCbk.h, 1595
PDSInjectTimeReference
qaGobiApiPds.h, 1803
PDSPosMethodStateReq, 743
pWifiState, 744
pXtraDataState, 744
pXtraTimeState, 744
PDSPositionData, 741
pAltitudeWrtEllipsoid, 743
pAltitudeWrtSealevel, 743
pHorizontalConfidence, 743
pHorizontalUncCircular, 743
pLatitude, 743
pLongitude, 743
pPositionSource, 743
pTimeStamp, 743
pTimeType, 743
pVerticalConfidence, 743
pVerticalUnc, 743
pDTMFTXGain
GetAudioPathConfigResp, 260
SetAudioPathConfigReq, 851
pDataBearer
QosEventInfo, 813
slqsWdsEventInfo, 897
unpack_wds_GetDataBearerTechnology_t, 1256
pDataBearerTech
getDUNCallInfoResp, 269
pDataBearerTechInd
pack_wds_SLQSWdsSetEventReport_t, 736
wdsSetEventReportReq, 1428
pDataCallStatusChangeInd
pack_wds_SLQSWdsSetEventReport_t, 736
wdsSetEventReportReq, 1428
pDataMode
LibpackProfile3GPP2, 347
LibPackprofile_3GPP2, 359
Profile3GPP2, 771
pDataRate
LibpackProfile3GPP2, 347
LibPackprofile_3GPP2, 359
Profile3GPP2, 771
swiQosFlow, 949
pDataSrc
voiceSUPSInfo, 1382
pDataStatusDetail
NetworkDebugResp, 579
unpack_nas_SLQSSwiNetworkDebug_t, 1154
pDataSystemStatusChangeInd
pack_wds_SLQSWdsSetEventReport_t, 736
wdsSetEventReportReq, 1428
pDate
_SLQSOMADMSessionInfo, 91
pDateLength
_SLQSOMADMSessionInfo, 91
pDayltSavAdj
nasNetworkTime, 556
unpack_nas_SLQSNasNetworkTimeCallBack_ind-_t, 1146
pDefaultPDNEnabled
_slqs3GPPConfigItem, 83
pack_wds_SLQSSet3GPPConfigItem_t, 730
pDeferTime
pack_swioma_SLQSOMADMSendSelection_t, 680
pDesription
QmiNas3GppNetworkInfo, 803
pDestAddr
cdmaMsgEncodingParams, 172
wcdmaMsgEncodingParams, 1390
pDestSMSContent
getDyingGaspCfg, 270
pack_dms_SLQSSwiSetDyingGaspCfg_t, 605
packgetDyingGaspCfg, 738
setDyingGaspCfg, 856
pDestSMSNum
getDyingGaspCfg, 270
pack_dms_SLQSSwiSetDyingGaspCfg_t, 605
packgetDyingGaspCfg, 738
setDyingGaspCfg, 856
pDetachAction
pack_nas_SLQSSwiPSDetach_t, 653
PSDetachReq, 774

pDevCrashStatus
CrashInfoParams, 193
pDeviceConfigDetail
NetworkDebugResp, 579
unpack_nas_SLQSSwiNetworkDebug_t, 1154
pDiagInfo
unpack_voice_SLQSVoiceGetCallInfo_t, 1236
voiceCallInfoResp, 1338
pDigitBuff
burstDTMFInfo, 142
voice_burstDTMFInfo, 1307
pDirNum
nasGet3GPP2SubscriptionInfoResp, 538
unpack_nas_SLQSNasGet3GPP2Subscription_t, 1141
pDisableIMSI
custFeaturesInfo, 208
pDispInfo
voiceInfoRec, 1364
pDisplayMode
cdmaMsgDecodingParams, 166
cdmaMsgDecodingParamsExt, 170
pDomainList
qmiWdsRunTimeSettings, 810
pDormancyStatus
getDUNCallInfoResp, 269
slqsWdsEventInfo, 897
pDormancyStatusInd
pack_wds_SLQSWdsSetEventReport_t, 737
wdsSetEventReportReq, 1428
pDualStandByPrefInd
nasIndicationRegisterReq, 549
pack_nas_SLQSNasIndicationRegisterExt_t, 643
pDuration
unpack_pds_GetXTRAValidity_t, 1165
pECIOThresList
ECIOThresh, 238
nas_ECIOThresh, 442
pECIOThresh
pack_nas_SLQSConfigSigInfo_t, 633
sigInfo, 876
pECMode
GetAudioPathConfigResp, 260
SetAudioPathConfigReq, 852
pECTNum
voiceSUPSNotification, 1384
PER
nas_NetworkStatEVDO, 483
NetworkStatEVDO, 583
pESNString
serialNumbersInfo, 844
pEVDOPageMonPerChangeInd
pack_wds_SLQSWdsSetEventReport_t, 737
wdsSetEventReportReq, 1428
pEarMute
SetM2MAudioProfileReq, 863
pEmerMode
_sysSelectPrefInfo, 104
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 652
unpack_nas_SLQSGetSysSelectionPref_t, 1140
pEmergencyCategory
pack_voice_SLQSVoiceDialCall_t, 698
voiceCallRequestParams, 1340
pEnableNotification
pack_wds_DHCPv4ClientLeaseChange_t, 713
WdsClientLeaseChange, 1413
pEnabled
pack_wds_SetMobileIPProfile_t, 723
pEnabledStatus
unpack_pds_GetPDSState_t, 1163
pEncodingAlphabet
cdmaMsgEncodingParams, 172
pEncryptData
pack_uim_ReadTransparent_t, 683
UIMReadTransparentReq, 1010
pEncryptedData
UIMReadTransparentResp, 1011
unpack_uim_ReadTransparent_t, 1214
pEncryptedPIN1
pack_uim_VerifyPin_t, 693
UIMPinResp, 1008
UIMVerifyPinReq, 1022
unpack_uim_ChangePin_t, 1212
unpack_uim_SetPinProtection_t, 1215
unpack_uim_UnblockPin_t, 1221
unpack_uim_VerifyPin_t, 1222
pError
USSDNoWaitIndicationInfo, 1293
pErrorCodeStr
imsaRatStatusInfo, 316
pErrorMask
CATEventDataType, 159
pErrorRateInd
nasIndicationRegisterReq, 549
pack_nas_SLQSNasIndicationRegisterExt_t, 643
pEspSpi
swiQosFilter, 946
pEtwsMessageInfo
SMSEventInfo_s, 918
pEtwsPlmnInfo
SMSEventInfo_s, 918
pExtDispInfo
voiceInfoRec, 1364
pExtDispRecInfo
voiceInfoRec, 1364
pExtErrCode
_GetProfileSettingOut, 65
UnPackGetProfileSettingOut, 1288
pExtErrorCode
CreateProfileOut, 196
ModifyProfileOut, 418
unpack_wds_SLQSModifyProfile_t, 1279
pFOTAUpdate
_SLQSOMADMSettings, 93
pFOTAdownload

_SLQSOMADMSettings, 93
pFailCause
unpack_voice_SLQSVoiceGetCallBarring_t, 1232
unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 1233
unpack_voice_SLQSVoiceGetCallWaiting_t, 1238
unpack_voice_SLQSVoiceGetCLIP_t, 1239
unpack_voice_SLQSVoiceGetCLIR_t, 1241
unpack_voice_SLQSVoiceGetCNAP_t, 1242
unpack_voice_SLQSVoiceGetCOLP_t, 1243
unpack_voice_SLQSVoiceGetCOLR_t, 1245
unpack_voice_SLQSVoiceManageCalls_t, 1247
unpack_voice_SLQSVoiceSetCallBarringPassword-_t, 1249
unpack_voice_SLQSVoiceSetSUPSService_t, 1252
voiceGetCallBarringResp, 1348
voiceGetCallFWResp, 1350
voiceGetCallWaitInfo, 1351
voiceGetCLIPResp, 1352
voiceGetCLIRResp, 1354
voiceGetCNAPResp, 1355
voiceGetCOLPResp, 1356
voiceGetCOLRResp, 1357
voiceManageCallsResp, 1366
voiceSetCallBarringPwdResp, 1372
voiceSetSUPSServiceResp, 1379
voiceSUPSInfo, 1382
pFailErrorCode
imsaPdpStatusInfo, 316
pFailureCause
USSDNoWaitIndicationInfo, 1293
pFailureReason
unpack_wds_SLQSStartDataSession_t, 1285
pFile
ERIFileparams, 241
pFileAttributes
UIMGetFileAttributesResp, 1007
unpack_uim_SLQSUIMGetFileAttributes_t, 1219
pFileSize
ERIFileparams, 241
pFixId
QmiCbkLocPositionReportInd, 796
unpack_loc_PositionRpt_Ind_t, 1100
pFlag
RXPCMIIRFltr, 833
TXPCMIIRFltr, 976
pFlashPayLd
pack_voice_SLQSVoiceSendFlash_t, 706
voiceFlashInfo, 1343
pFlashType
pack_voice_SLQSVoiceSendFlash_t, 706
voiceFlashInfo, 1343
pFollowOnDC
slqssendasyncsmsparams_s, 888
sms_sendAsyncsmsParams, 908
pForbidden
QmiNas3GppNetworkInfo, 804
pForceOnDC
slqssendasyncsmsparams_s, 888
sms_sendAsyncsmsParams, 908
pForceRev0
pack_nas_SetCDMANetworkParameters_t, 631
pFwAutoCheck
_SLQSOMADMSettings, 93
_SLQSOMADMSettingsReqParams3, 95
pack_swioma_SLQSOMADMSetSettings_t, 681
pFwAutoSDM
_SLQSAVMSSetSettingsReqPara, 85
_SLQSAVMSSettings, 85
pFwPromptUpdate
_SLQSAVMSSetSettingsReqPara, 85
_SLQSAVMSSettings, 85
pFwPromptdownload
_SLQSAVMSSetSettingsReqPara, 85
_SLQSAVMSSettings, 85
pGCDumpString
CrashInfo, 192
pGERANInfo
nasCellLocationInfoResp, 536
unpack_nas_SLQSNasGetCellLocationInfo_t, 1142
pGPRSGrantedQoS
qmiWdsRunTimeSettings, 810
pGPRSMinimumQoS
LibpackProfile3GPP, 342
LibPackprofile_3GPP, 353
Profile3GPP, 765
pGPRSRequestedQos
LibpackProfile3GPP, 342
LibPackprofile_3GPP, 353
Profile3GPP, 765
pGPSDataMask
pack_pds_ResetPDSData_t, 654
pGPSEnable
custFeaturesSetting, 210
pGPSLPM
custFeaturesInfo, 208
custFeaturesSetting, 210
pGPSSel
custFeaturesInfo, 208
custFeaturesSetting, 210
pGPSWeek
unpack_pds_GetXTRAValidity_t, 1165
pGPSWeekOffset
unpack_pds_GetXTRAValidity_t, 1165
pGSMBER
GetErrRateResp, 271
unpack_nas_SLQSGetErrorRate_t, 1121
pGSMCallBarringSysInfo
nasGetSysInfoResp, 544
nasSysInfo, 575
unpack_nas_SLQSGetSysInfo_t, 1135
unpack_nas_SLQSSysInfoCallback_ind_t, 1157
pGSMCipherDomainSysInfo
nasGetSysInfoResp, 544

nasSysInfo, 575
unpack_nas_SLQSGetSysInfo_t, 1135
unpack_nas_SLQSSysInfoCallback_ind_t, 1157
pGSMRSSIDelta
pack_nas_SLQSNasConfigSigInfo2_t, 638
setSignalStrengthInfo, 871
pGSMRSSIThresh
pack_nas_SLQSNasConfigSigInfo2_t, 638
setSignalStrengthInfo, 871
pGSMRSSIThreshList
GSMRSSIThresh, 295
nas_GSMRSSIThresh, 447
pGSMSSInfo
nasGetSigInfoResp, 541
pGSMSigInfo
nasSigInfo, 570
unpack_nas_SLQSNasSigInfoCallback_ind_t, 1147
pGSMSrvStatusInfo
nasGetSysInfoResp, 544
nasSysInfo, 575
unpack_nas_SLQSGetSysInfo_t, 1135
unpack_nas_SLQSSysInfoCallback_ind_t, 1157
pGSMSysInfo
nasGetSysInfoResp, 544
nasSysInfo, 575
unpack_nas_SLQSGetSysInfo_t, 1135
unpack_nas_SLQSSysInfoCallback_ind_t, 1157
pGWAcqOrderPref
_sysSelectPrefInfo, 104
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 652
unpack_nas_SLQSGetSysSelectionPref_t, 1140
pGWAddressV4
qmiWdsRunTimeSettings, 810
pGenerator
GetM2MAudioProfileReq, 277
SetM2MAudioProfileReq, 863
pGetCallFWExtInfo
unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 1233
voiceGetCallFWResp, 1350
pGetCallFWInfo
unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 1233
voiceGetCallFWResp, 1350
pGetCustomInput
DMSgetCustomFeatureV2, 231
getCustomFeatureV2, 264
pGetDyingGaspCfg
unpack_dms_SLQSSwiGetDyingGaspCfg_t, 1070
pGetDyingGaspStatistics
unpack_dms_SLQSSwiGetDyingGaspStatistics_t, 1070
pGetIndicationRegInfo
unpack_sms_SLQSGetIndicationRegister_t, 1199
pGetMsgWaitingInfoResp
unpack_sms_SLQSGetMessageWaiting_t, 1200
pGetTransLayerInfo
unpack_sms_SLQSGetTransLayerInfo_t, 1202
pGetTransNWRegInfo
unpack_sms_SLQSGetTransNWRegInfo_t, 1202
pGnssData
LocDelAssDataReq, 381
pack_loc_Delete_Assist_Data_t, 614
pGpsTime
QmiCbkLocBestAvailPosInd, 784
QmiCbkLocPositionReportInd, 796
unpack_loc_BestAvailPos_Ind_t, 1088
unpack_loc_PositionRpt_Ind_t, 1100
pGyroAcceptReady
QmiCbkLocSensorStreamingInd, 798
pGyroData
LocInjectSensorDataReq, 390
pGyroSamplesAccepted
QmiCbkLocInjectSensorDataInd, 790
pGyroTempAcceptReady
QmiCbkLocSensorStreamingInd, 798
pGyroTempData
LocInjectSensorDataReq, 390
pGyroTempSamplesAccepted
QmiCbkLocInjectSensorDataInd, 790
pGyroTimeSrc
LocInjectSensorDataReq, 390
pHA2002bis
pack_wds_SetMobileIPParameters_t, 721
pHAAuthenticator
pack_wds_SetMobileIPParameters_t, 721
pHASPI
pack_wds_SetMobileIPProfile_t, 723
pHDRECIODelta
pack_nas_SLQSNasConfigSigInfo2_t, 638
setSignalStrengthInfo, 871
pHDRECIOThresh
pack_nas_SLQSNasConfigSigInfo2_t, 638
setSignalStrengthInfo, 871
pHDRECIOThreshList
HDRECIOThresh, 301
nas_HDRECIOThresh, 451
pHDRIODelta
pack_nas_SLQSNasConfigSigInfo2_t, 639
setSignalStrengthInfo, 871
pHDRIOThresh
pack_nas_SLQSNasConfigSigInfo2_t, 639
setSignalStrengthInfo, 871
pHDRIOThreshList
HDRIOThresh, 301
nas_HDRIOThresh, 452
pHDRNewUATIAssInd
nasIndicationRegisterReq, 549
pack_nas_SLQSNasIndicationRegisterExt_t, 643
pHDRPackErrRate
GetErrRateResp, 271
unpack_nas_SLQSGetErrorRate_t, 1121
pHDRRSSIDelta
pack_nas_SLQSNasConfigSigInfo2_t, 639

setSignalStrengthInfo, 871
pHDRRSSIThresh
pack_nas_SLQSNasConfigSigInfo2_t, 639
setSignalStrengthInfo, 871
pHDRRSSIThreshList
HDRRSSIThresh, 304
nas_HDRRSSIThresh, 452
pHDRSINRDelta
pack_nas_SLQSNasConfigSigInfo2_t, 639
setSignalStrengthInfo, 871
pHDRSINRThresList
HDRSINRThresh, 304
nas_HDRSINRThresh, 453
pHDRSINRThresh
pack_nas_SLQSConfigSigInfo_t, 633
pack_nas_SLQSNasConfigSigInfo2_t, 639
setSignalStrengthInfo, 871
sigInfo, 876
pHDRSINRThreshList
HDRSINRThreshold, 305
nas_HDRSINRThreshold, 454
pHDRSSInfo
nasGetSigInfoResp, 541
pHDRSessionCloseInd
nasIndicationRegisterReq, 549
pack_nas_SLQSNasIndicationRegisterExt_t, 643
pHDRSigInfo
nasSigInfo, 570
unpack_nas_SLQSNasSigInfoCallback_ind_t, 1147
pHDRSrvStatusInfo
nasGetSysInfoResp, 544
nasSysInfo, 575
unpack_nas_SLQSGetSysInfo_t, 1135
unpack_nas_SLQSSysInfoCallback_ind_t, 1157
pHDRSysInfo
nasGetSysInfoResp, 544
nasSysInfo, 576
unpack_nas_SLQSGetSysInfo_t, 1135
unpack_nas_SLQSSysInfoCallback_ind_t, 1157
pHaltSubscription
UIMGetConfigurationResp, 1006
unpack_uim_SLQSUIMGetConfiguration_t, 1218
pHeading
QmiCbkLocBestAvailPosInd, 784
QmiCbkLocPositionReportInd, 796
unpack_loc_BestAvailPos_Ind_t, 1088
unpack_loc_PositionRpt_Ind_t, 1100
pHeadingUnc
QmiCbkLocBestAvailPosInd, 784
QmiCbkLocPositionReportInd, 797
unpack_loc_BestAvailPos_Ind_t, 1088
unpack_loc_PositionRpt_Ind_t, 1100
pHomeSIDNID
nasGet3GPP2SubscriptionInfoResp, 538
unpack_nas_SLQSNasGet3GPP2Subscription_t, 1141
pHorCirConf
QmiCbkLocBestAvailPosInd, 784
unpack_loc_BestAvailPos_Ind_t, 1088
pHorConfidence
LocInjectPositionReq, 388
QmiCbkLocPositionReportInd, 797
unpack_loc_PositionRpt_Ind_t, 1100
pHorEllpConf
QmiCbkLocBestAvailPosInd, 784
unpack_loc_BestAvailPos_Ind_t, 1088
pHorReliability
LocInjectPositionReq, 389
QmiCbkLocBestAvailPosInd, 784
QmiCbkLocPositionReportInd, 797
unpack_loc_BestAvailPos_Ind_t, 1088
unpack_loc_PositionRpt_Ind_t, 1100
pHorUncCircular
LocInjectPositionReq, 389
QmiCbkLocBestAvailPosInd, 784
QmiCbkLocPositionReportInd, 797
unpack_loc_BestAvailPos_Ind_t, 1088
unpack_loc_PositionRpt_Ind_t, 1100
pHorUncEllipseOrientAzimuth
QmiCbkLocBestAvailPosInd, 784
QmiCbkLocPositionReportInd, 797
unpack_loc_BestAvailPos_Ind_t, 1088
unpack_loc_PositionRpt_Ind_t, 1100
pHorUncEllipseSemiMajor
QmiCbkLocBestAvailPosInd, 784
QmiCbkLocPositionReportInd, 797
unpack_loc_BestAvailPos_Ind_t, 1088
unpack_loc_PositionRpt_Ind_t, 1100
pHorUncEllipseSemiMinor
QmiCbkLocBestAvailPosInd, 784
QmiCbkLocPositionReportInd, 797
unpack_loc_BestAvailPos_Ind_t, 1088
unpack_loc_PositionRpt_Ind_t, 1100
pHorizontalAccuracyLvl
LOCStartReq, 392
pack_loc_Start_t, 628
pHorizontalConfidence
pack_pds_SLQSPDSInjectPositionData_t, 661
PDSPositionData, 743
pHorizontalUncCircular
pack_pds_SLQSPDSInjectPositionData_t, 661
PDSPositionData, 743
pHotSwapStatus
UIMGetCardStatusResp, 1004
unpack_uim_GetCardStatus_t, 1213
pHwConfig
pack_wds_SLQSSSetDHCPv4ClientConfig_t, 733
unpack_wds_SLQSSGetDHCPv4ClientConfig_t, 1284
WdsDHCPv4Config, 1416
PI
calledPartyInfo, 146
callerIDInfo, 147
callFWExtInfo, 150
callingPartyInfo, 154

redirNumInfo, 819
voice_callFWExtInfo, 1311
PIFACEId
SetM2MAudioAVCFGReq, 861
pIMCNflag
qmiWdsRunTimeSettings, 810
pIMEIString
serialNumbersInfo, 844
pIMSDomain
GetIMSUserConfigParams, 274
imsUserConfigInfo, 326
SetIMSUserConfigReq, 857
pIMSDomainLen
GetIMSUserConfigParams, 274
SetIMSUserConfigReq, 857
pIMSTestMode
GetRegMgrConfigParams, 283
imsRegMgrConfigInfo, 324
SetRegMgrConfigReq, 867
pIOThresList
IOThresh, 331
nas_IOThresh, 458
pIOThresh
pack_nas_SLQSConfigSigInfo_t, 633
sigInfo, 876
pIPAddressV4
QmiWdsIpAddressInfo, 807
qmiWdsRunTimeSettings, 810
pIPAddressV6
QmiWdsIpAddressInfo, 807
pIPFamSupport
custFeaturesInfo, 208
pIPFamily
GetInstIDResp, 276
pIPFamilyPreference
qmiWdsRunTimeSettings, 810
pIPV6AddrInfo
qmiWdsRunTimeSettings, 810
pIPV6GWAddrInfo
qmiWdsRunTimeSettings, 810
pIPv4Addr
WdsDHCPv4ClientLeaseInd, 1415
pIPv4AddrInfo
QmiCbkLocGetServerInd, 788
unpack_loc_GetServer_Ind_t, 1093
pIPv4AddrPref
LibpackProfile3GPP, 342
LibPackprofile_3GPP, 353
Profile3GPP, 765
pIPv4Address
swiPDPRuntimeSettingsResp, 944
pIPv4DstAddr
swiQosFilter, 946
pIPv4GWAddress
swiPDPRuntimeSettingsResp, 944
pIPv4SrcAddr
swiQosFilter, 946
pIPv6AddPref
LibpackProfile3GPP, 342
LibPackprofile_3GPP, 353
Profile3GPP, 765
pIPv6AddrInfo
QmiCbkLocGetServerInd, 788
unpack_loc_GetServer_Ind_t, 1093
pIPv6Address
swiPDPRuntimeSettingsResp, 944
pIPv6DstAddr
swiQosFilter, 946
pIPv6GWAddress
swiPDPRuntimeSettingsResp, 944
pIPv6Label
swiQosFilter, 946
pIPv6SrcAddr
swiQosFilter, 946
pIPv6TrafCls
swiQosFilter, 946
pIPv6prefixlen
QmiWdsIpAddressInfo, 807
pId
swiQosFilter, 946
pIds
swiQosIds, 950
pIgnoreHotSwapSwitch
pack_uim_SLQSUIMPowerUp_t, 688
UIMPowerUpReq, 1009
pImCnFlag
LibpackProfile3GPP, 342
LibPackprofile_3GPP, 353
Profile3GPP, 765
pImageList
pack_fms_SetImagesPreference_t, 613
unpack_fms_GetImagesPreference_t, 1082
pImeiSvnString
serialNumbersInfo, 844
pImgType
FirmwareUpdatStat, 249
pImsRegErrCode
IMSARegistrationStatus, 317
pImsRegStatus
IMSARegistrationStatus, 317
imsaRegStatusInfo, 318
pInUse
QmiNas3GppNetworkInfo, 804
pIndFieldsList
IMSASupportedFieldsResp, 321
pIndicationToken
pack_uim_ChangePin_t, 682
pack_uim_ReadTransparent_t, 683
pack_uim_SetPinProtection_t, 684
pack_uim_SLQSUIMAuthenticate_t, 685
pack_uim_SLQSUIMGetFileAttributes_t, 687
pack_uim_UnblockPin_t, 692
pack_uim_VerifyPin_t, 693
UIMAuthenticateReq, 1001
UIMAuthenticateResp, 1002
UIMChangePinReq, 1002

UIMGetFileAttributesReq, 1006
UIMGetFileAttributesResp, 1007
UIMPinResp, 1008
UIMReadTransparentReq, 1010
UIMReadTransparentResp, 1011
UIMSetPinProtectionReq, 1017
UIMUnblockPinReq, 1021
UIMVerifyPinReq, 1022
unpack_uim_ChangePin_t, 1212
unpack_uim_ReadTransparent_t, 1214
unpack_uim_SetPinProtection_t, 1215
unpack_uim_SLQSUIMAuthenticate_t, 1216
unpack_uim_SLQSUIMGetFileAttributes_t, 1219
unpack_uim_UnblockPin_t, 1221
unpack_uim_VerifyPin_t, 1222
pInfo
pack_voice_AnswerUSSD_t, 694
pack_voice_OriginateUSSD_t, 694
pInstanceID
GetInstIDResp, 276
pInstanceSize
QmiNasPerformNetworkScanResp, 804
pInstances
QmiNasPerformNetworkScanResp, 804
pInstancesSize
QmiNasGetRFBandInfoResp, 804
pIntermediateReportState
LOCStartReq, 393
pack_loc_Start_t, 628
pInterval
unpack_pds_GetPDSDefaults_t, 1162
unpack_pds_GetXTRAAutomaticDownload_t, 1164
pIpcpAckTimeout
LibpackProfile3GPP2, 347
LibPackprofile_3GPP2, 359
Profile3GPP2, 771
pIpcpCreqRetryCount
LibpackProfile3GPP2, 348
LibPackprofile_3GPP2, 359
Profile3GPP2, 771
pIsPcscfAddressNedded
LibpackProfile3GPP2, 348
LibPackprofile_3GPP2, 359
Profile3GPP2, 771
pIsUDHPresent
cdmaMsgDecodingParamsExt, 170
wcdmaLongMsgDecodingParams, 1387
pIsVoiceEnabled
custFeaturesInfo, 208
custFeaturesSetting, 210
pJitter
swiQosFlow, 949
pKeyReferenceID
pack_uim_ChangePin_t, 682
pack_uim_SetPinProtection_t, 684
pack_uim_UnblockPin_t, 692
pack_uim_VerifyPin_t, 693
UIMChangePinReq, 1002
UIMSetPinProtectionReq, 1017
UIMUnblockPinReq, 1021
UIMVerifyPinReq, 1022
PLMN_LENGTH
qaGobiApiNas.h, 1752
PLMNData
nas_operatorPLMNList, 486
operatorPLMNList, 597
PLMNName
nas_operatorNameString, 484
operatorNameString, 595
PLMNNetName
nas_PLMNNetworkName, 493
PLMNNetworkName, 755
PLMNNetworkName, 754
numInstance, 755
PLMNNetName, 755
PLMNNetworkNameData, 755
codingScheme, 756
countryInitials, 756
longName, 756
longNameLen, 756
longNameSpareBits, 756
shortName, 756
shortNameLen, 756
shortNameSpareBits, 757
PLMNRecID
nas_OperatorPLMNData, 485
OperatorPLMNData, 596
pLTEAttachProfile
_slqs3GPPConfigItem, 83
pack_wds_SLQSSet3GPPConfigItem_t, 730
pLTEAttachProfileList
_slqs3GPPConfigItem, 83
pack_wds_SLQSSet3GPPConfigItem_t, 730
pLTEBandPref
_sysSelectPrefInfo, 104
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 652
unpack_nas_SLQSGetSysSelectionPref_t, 1140
pLTECphyCa
nasIndicationRegisterReq, 549
pack_nas_SLQSNasIndicationRegisterExt_t, 643
pLTEInfo
swiModemStatusResp, 940
unpack_nas_SLQSNasSwiModemStatus_t, 1148
pLTEInfoInterfreq
nasCellLocationInfoResp, 536
unpack_nas_SLQSNasGetCellLocationInfo_t, 1143
pLTEInfoIntrafreq
nasCellLocationInfoResp, 536
unpack_nas_SLQSNasGetCellLocationInfo_t, 1143
pLTEInfoNeighboringGSM
nasCellLocationInfoResp, 536

unpack_nas_SLQSNasGetCellLocationInfo_t, 1143
pLTEInfoNeighboringWCDMA
nasCellLocationInfoResp, 536
unpack_nas_SLQSNasGetCellLocationInfo_t, 1143
pLTERSRPDelta
pack_nas_SLQSNasConfigSigInfo2_t, 639
setSignalStrengthInfo, 871
pLTERSRPThresh
pack_nas_SLQSNasConfigSigInfo2_t, 639
setSignalStrengthInfo, 871
pLTERSRPThreshList
LTERSRPThresh, 404
nas_LTERSRPThresh, 468
pLTERSRQDelta
pack_nas_SLQSNasConfigSigInfo2_t, 639
setSignalStrengthInfo, 871
pLTERSRQThresh
pack_nas_SLQSNasConfigSigInfo2_t, 639
setSignalStrengthInfo, 871
pLTERSRQThreshList
LTERSRQThresh, 405
nas_LTERSRQThresh, 469
pLTERSSIDelta
pack_nas_SLQSNasConfigSigInfo2_t, 639
setSignalStrengthInfo, 872
pLTERSSIThresh
pack_nas_SLQSNasConfigSigInfo2_t, 639
setSignalStrengthInfo, 872
pLTERSSIThreshList
LTERSSIThresh, 405
nas_LTERSSIThresh, 469
pLTESNRDelta
pack_nas_SLQSNasConfigSigInfo2_t, 639
setSignalStrengthInfo, 872
pLTESNRThresList
LTESNRThresh, 408
nas_LTESNRThresh, 472
pLTESNRThresh
pack_nas_SLQSConfigSigInfo_t, 633
pack_nas_SLQSNasConfigSigInfo2_t, 639
setSignalStrengthInfo, 872
sigInfo, 876
pLTESNRThreshList
LTESNRThreshold, 409
nas_LTESNRThreshold, 473
pLTESSInfo
nasGetSigInfoResp, 541
pLTESigInfo
nasSigInfo, 570
unpack_nas_SLQSNasSigInfoCallback_ind_t, 1147
pLTESigRptCfg
pack_nas_SLQSConfigSigInfo_t, 633
sigInfo, 876
pLTESigRptConfig
pack_nas_SLQSNasConfigSigInfo2_t, 639
setSignalStrengthInfo, 872
pLTESrvStatusInfo
nasGetSysInfoResp, 544
nasSysInfo, 576
unpack_nas_SLQSGetSysInfo_t, 1135
unpack_nas_SLQSSysInfoCallback_ind_t, 1157
pLTESysInfo
nasGetSysInfoResp, 544
nasSysInfo, 576
unpack_nas_SLQSGetSysInfo_t, 1135
unpack_nas_SLQSSysInfoCallback_ind_t, 1157
pLTEVoiceSupportSysInfo
nasGetSysInfoResp, 544
nasSysInfo, 576
unpack_nas_SLQSGetSysInfo_t, 1135
unpack_nas_SLQSSysInfoCallback_ind_t, 1157
pLanguage
cdmaMsgDecodingParams, 166
cdmaMsgDecodingParamsExt, 170
pLastBearerTech
DataBearerTechExt, 215
pLastCallDataBearerTech
getDUNCallInfoResp, 269
pLastCallDataBearerTechnology
dataBearers, 212
pLastCallRXOKBytesCnt
getDUNCallInfoResp, 269
pLastCallTXOKBytesCnt
getDUNCallInfoResp, 269
pLatency
swiQosFlow, 949
pLatitude
LocInjectPositionReq, 389
pack_pds_SLQSPDSInjectPositionData_t, 661
PDSPositionData, 743
QmiCbkLocBestAvailPosInd, 785
QmiCbkLocPositionReportInd, 797
unpack_loc_BestAvailPos_Ind_t, 1089
unpack_loc_PositionRpt_Ind_t, 1100
pLcpAckTimeout
LibpackProfile3GPP2, 348
LibPackprofile_3GPP2, 359
Profile3GPP2, 771
pLcpCreqRetryCount
LibpackProfile3GPP2, 348
LibPackprofile_3GPP2, 359
Profile3GPP2, 771
pLeapSeconds
QmiCbkLocPositionReportInd, 797
unpack_loc_PositionRpt_Ind_t, 1100
pLeaseState
WdsDHCPv4ClientLeaseInd, 1415
pLineCtrlInfo
voiceInfoRec, 1364
pLinktimer
pack_sms_SendSMS_t, 669
slqssendasyncsmsparams_s, 888
slqssendsmsparams_s, 889

sms_sendAsyncsmsParams, 908
pLogString
FirmwareUpdatStat, 249
pLogStringLen
FirmwareUpdatStat, 249
pLongitude
LocInjectPositionReq, 389
pack_pds_SLQSPDSInjectPositionData_t, 661
PDSPositionData, 743
QmiCbkLocBestAvailPosInd, 785
QmiCbkLocPositionReportInd, 797
unpack_loc_BestAvailPos_Ind_t, 1089
unpack_loc_PositionRpt_Ind_t, 1100
pLoopbackMode
WDSSetLoopbackData, 1429
pLoopbackMultiplier
WDSSetLoopbackData, 1429
pLteBandCapability
BandCapabilityResp, 140
pLteEARFCN
nasSwiGetChannelLockResp, 571
nasSwiSetChannelLockReq, 573
pack_nas_SLQSNASSwiSetChannelLock_t, 645
unpack_nas_SLQSNASSwiGetChannelLock_t, 1148
pLteNasRelInfo
SwiOTAMsg_s, 941
pLtePCI
nasSwiGetChannelLockResp, 571
nasSwiSetChannelLockReq, 573
pack_nas_SLQSNASSwiSetChannelLock_t, 645
unpack_nas_SLQSNASSwiGetChannelLock_t, 1148
pLteQci
swiQosFlow, 949
pMCC
QmiNas3GppNetworkInfo, 804
unpack_nas_GetHomeNetwork3GPP2_t, 1108
pMEIDString
serialNumbersInfo, 844
pMICGainSelect
GetAudioPathConfigResp, 260
pMIPStatusInd
pack_wds_SLQSWdsSetEventReport_t, 737
wdsSetEventReportReq, 1428
pMNAAA
pack_wds_SetMobileIPProfile_t, 723
pMNC
QmiNas3GppNetworkInfo, 804
unpack_nas_GetHomeNetwork3GPP2_t, 1108
pMNCIncPCSDigStat
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 652
pMNHA
pack_wds_SetMobileIPProfile_t, 723
pMNRInfo
nasInitNetworkReg, 550
pack_nas_SLQSInitiateNetworkRegistration_t, 635
pMTMessageInfo
SMSEventInfo_s, 918
pMagneticDeviation
QmiCbkLocBestAvailPosInd, 785
QmiCbkLocPositionReportInd, 797
unpack_loc_BestAvailPos_Ind_t, 1089
unpack_loc_PositionRpt_Ind_t, 1100
pManString
_SLQSSwiGetHostDevInfoParams, 96
_SLQSSwiSetHostDevInfoParams, 98
pManagedRoamingInd
nasIndicationRegisterReq, 549
pack_nas_SLQSNasIndicationRegisterExt_t, 643
pMaxAllowedPktSz
swiQosFlow, 949
pMaxChannelRXRate
WdsConnectionRateElmnts, 1415
pMaxChannelTXRate
WdsConnectionRateElmnts, 1415
pMaxStorageSizeReq
pack_sms_SLQSSmsGetMaxStorageSize_t, 676
pMaxStorageSizeResp
unpack_sms_SLQSSmsGetMaxStorageSize_t, 1203
pMdmCallDurationActive
getDUNCallInfoResp, 269
pMeidString
_SLQSSwiGetSerialNoExtParams, 97
pMessage
cdmaMsgDecodingParams, 167
cdmaMsgDecodingParamsExt, 170
cdmaMsgEncodingParams, 172
pack_sms_SaveSMS_t, 668
pack_sms_SendSMS_t, 669
slqssendasyncsmsparams_s, 888
slqssendsmsparams_s, 889
sms_sendAsyncsmsParams, 908
wcdmaLongMsgDecodingParams, 1387
wcdmaMsgDecodingParams, 1389
pMessageID
cdmaMsgDecodingParams, 167
cdmaMsgDecodingParamsExt, 170
pMessageIndex
pack_sms_SLQSDeleteSMS_t, 670
unpack_sms_SaveSMS_t, 1197
pMessageMode
pack_sms_SLQSDeleteSMS_t, 671
pack_sms_SLQSGetSMS_t, 671
pack_sms_SLQSGetSMSList_t, 672
pack_sms_SLQSModifySMSStatus_t, 673
sms_maxStorageSizeReq, 902
smsMaxStorageSizeReq, 918
pMessageModeInfo
SMSEventInfo_s, 918
pMessageProtocol
unpack_sms_SLQSSmsGetMessageProtocol_t, 1203
pMessageSize

cdmaMsgEncodingParams, 172
pMessageTag
pack_sms_SLQSDeleteSMS_t, 671
pMicMute
SetM2MAudioProfileReq, 863
pMinBasedIMSI
nasGet3GPP2SubscriptionInfoResp, 538
unpack_nas_SLQSNasGet3GPP2Subscription_t, 1141
pMinIntervalTime
LOCStartReq, 393
pack_loc_Start_t, 628
pMinPolicedPktSz
swiQosFlow, 950
pMinSessionExpiryTimer
GetIMSVoIPConfigResp, 276
imsVoIPConfigInfo, 328
SetIMSVoIPConfigReq, 860
pMitigationDevList
TmdGetMitigationDevListResp, 967
pMitigationDevListLen
TmdGetMitigationDevListResp, 967
pMncPcsDigitStatus
nasInitNetworkReg, 550
pack_nas_SLQSInitiateNetworkRegistration_t, 635
pMncPcsStatus
nasPLMNNameReq, 562
pack_nas_SLQSGetPLMNName_t, 634
pMode
pack_wds_SetMobileIPParameters_t, 721
pModePref
_sysSelectPrefInfo, 104
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 652
unpack_nas_SLQSGetSysSelectionPref_t, 1140
pModelString
_SLQSSwiGetHostDevInfoParams, 96
_SLQSSwiSetHostDevInfoParams, 98
pMtu
qmiWdsRunTimeSettings, 810
pNAI
pack_wds_SetMobileIPProfile_t, 723
pNAMNameInfo
nasGet3GPP2SubscriptionInfoResp, 538
unpack_nas_SLQSNasGet3GPP2Subscription_t, 1141
pNID
unpack_nas_GetHomeNetwork3GPP2_t, 1108
pNITZInformation
nasOperatorNameResp, 557
unpack_nas_SLQSGetOperatorNameData_t, 1123
pNSEnable
GetAudioPathConfigResp, 260
SetAudioPathConfigReq, 852
pNSSAudioCtrl
voiceInfoRec, 1364
pNSSRelease
voiceInfoRec, 1364
pNamID
pack_voice_SLQSVoiceGetConfig_t, 703
voiceGetConfigReq, 1359
pName
pack_wds_SetDefaultProfile_t, 719
unpack_nas_GetHomeNetwork3GPP2_t, 1108
pNameString
_SLQSSwiGetOSInfoParams, 96
_SLQSSwiSetOSInfoParams, 99
pNegoDnsSrvrPref
LibpackProfile3GPP2, 348
LibPackprofile_3GPP2, 359
Profile3GPP2, 771
pNeighborSetPilotPN
nas_NetworkStat1x, 481
NetworkStat1x, 581
pNetSelPref
_sysSelectPrefInfo, 104
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 652
unpack_nas_SLQSGetSysSelectionPref_t, 1140
pNetworkInfo
_slqsNetworkScanInfo, 88
pNetworkInfoInstances
_slqsNetworkScanInfo, 88
pNetworkInfoLen
CurrDataSysStat, 200
pNetworkMode
pack_pds_SLQSGetAGPSConfig_t, 659
pack_pds_SLQSSetAGPSConfig_t, 662
pNetworkRejectInd
nasIndicationRegisterReq, 549
pack_nas_SLQSNasIndicationRegisterExt_t, 643
pNetworkStat1x
NetworkDebugResp, 579
unpack_nas_SLQSSwiNetworkDebug_t, 1154
pNetworkStatEVDO
NetworkDebugResp, 580
unpack_nas_SLQSSwiNetworkDebug_t, 1154
pNetworkTimeInd
nasIndicationRegisterReq, 549
pack_nas_SLQSNasIndicationRegisterExt_t, 643
pNewImsRegStatus
IMSARegistrationStatus, 317
pNewPwdData
voiceSUPSInfo, 1382
pNotifStore
_SLQSAVMSSetSettingsReqPara, 85
pNotificationStore
_SLQSAVMSSettings, 85
pNumSupUSBComps
USBCompParams, 1292
pNumberOfPhySlot
UIMGetSlotsStatusResp, 1007
unpack_uim_SLQSUIMGetSlotsStatus_t, 1219
pNw2DescDisp
unpack_nas_GetHomeNetwork3GPP2_t, 1109
pNw2DescEnc

unpack_nas_GetHomeNetwork3GPP2_t, 1109
pNw2DescLen
unpack_nas_GetHomeNetwork3GPP2_t, 1109
pNw2MCC
unpack_nas_GetHomeNetwork3GPP2_t, 1109
pNw2MNC
unpack_nas_GetHomeNetwork3GPP2_t, 1109
pNw2Name
unpack_nas_GetHomeNetwork3GPP2_t, 1109
pNxtHdrProto
swiQosFilter, 946
pOMADMEnabled
_SLQSAVMSSettings, 86
_SLQSOMADMSettings, 93
PORTNAM_LEN
qaGobiApiDcs.h, 1674
pOTASPStatus
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1230
unpack_voice_SLQSVoiceGetCallInfo_t, 1236
voiceCallInfoResp, 1338
voiceGetAllCallInfo, 1346
pObjectVer
NetworkDebugResp, 580
unpack_nas_SLQSSwiNetworkDebug_t, 1154
pOffLength
voiceDTMFEventInfo, 1342
pOnLength
voiceDTMFEventInfo, 1342
pOpaqueIdentifier
LocInjectSensorDataReq, 390
QmiCbkLocInjectSensorDataInd, 790
pOperation
unpack_pds_GetPDSDefaults_t, 1162
pOperatorNameString
nasOperatorNameResp, 557
unpack_nas_SLQSGetOperatorNameData_t, 1123
pOperatorPLMNList
nasOperatorNameResp, 557
unpack_nas_SLQSGetOperatorNameData_t, 1123
pOptList
wds_DHCPv4OptionList, 1401
WdsDHCPv4ClientLeaseInd, 1415
WdsDHCPv4OptionList, 1420
wdsDhcpv4OptionList, 1419
pOptVal
DHCPOption, 225
wds_DHCPOpt, 1399
pOptions
DHCPOptionList, 226
pPCMParams
SetM2MAudioAVCFGReq, 861
pPCSCFAddrPCO
qmiWdsRunTimeSettings, 810
pPCSCFFQDNAddrList
qmiWdsRunTimeSettings, 810
pPCSCFPort
GetRegMgrConfigParams, 283
pPCSDigitInfo
_slqsNetworkScanInfo, 88
pPCSDigitInstances
_slqsNetworkScanInfo, 88
pPCSInstance
unpack_nas_PerformNetworkScan_t, 1117
pPCSInstanceSize
unpack_nas_PerformNetworkScan_t, 1117
pPDNInactivTimeout
LibpackProfile3GPP, 342
LibPackprofile_3GPP, 353
Profile3GPP, 765
pPDNInactivTimeout3GPP2
LibpackProfile3GPP2, 348
LibPackprofile_3GPP2, 359
Profile3GPP2, 771
pPDPType
qmiWdsRunTimeSettings, 810
pPDPtype
LibpackProfile3GPP, 342
LibPackprofile_3GPP, 353
Profile3GPP, 765
pPDUMessage
wcdmaMsgEncodingParams, 1390
pPLMNNetworkName
nasOperatorNameResp, 557
unpack_nas_SLQSGetOperatorNameData_t, 1123
pPRLPref
_sysSelectPrefInfo, 104
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 652
unpack_nas_SLQSGetSysSelectionPref_t, 1140
pPRLPreference
dmsCurrentPRLInfo, 229
pPRLVersion
dmsCurrentPRLInfo, 229
pPWD
_SLQSAVMSSettingsAPNInfo, 86
pPacketsCountRX
QosEventInfo, 813
slqsWdsEventInfo, 897
pPacketsCountTX
QosEventInfo, 813
slqsWdsEventInfo, 897
pPartNum
cdmaMsgDecodingParamsExt, 170
wcdmaLongMsgDecodingParams, 1388
pPass
pack_wds_SLQSStartDataSession_t, 734
pPassword
LibpackProfile3GPP, 342
LibPackprofile_3GPP, 353
pack_wds_SetDefaultProfile_t, 719
Profile3GPP, 765
ssdatasession_params, 932
pPasswordSize
LibpackProfile3GPP, 342
LibPackprofile_3GPP, 353
Profile3GPP, 765

pPcscfAddrUsingDhcp
LibpackProfile3GPP, 342
LibPackprofile_3GPP, 353
Profile3GPP, 765
pPcscfAddrUsingPCO
LibpackProfile3GPP, 342
LibPackprofile_3GPP, 353
Profile3GPP, 765
pPdnType
LibpackProfile3GPP2, 348
LibPackprofile_3GPP2, 359
Profile3GPP2, 771
pPdpAccessConFlag
LibpackProfile3GPP, 342
LibPackprofile_3GPP, 353
Profile3GPP, 765
pPdpContext
LibpackProfile3GPP, 342
LibPackprofile_3GPP, 353
Profile3GPP, 765
pPdpDataCompType
LibpackProfile3GPP, 342
LibPackprofile_3GPP, 353
Profile3GPP, 765
pPdpHdrCompType
LibpackProfile3GPP, 342
LibPackprofile_3GPP, 353
Profile3GPP, 765
pPdpStatusConfig
IMSAIndRegisterInfo, 315
pPeroidInfo
_SLQSAVMSSetSettingsReqPara, 85
pPeroidsInfo
_SLQSAVMSSettings, 86
pPersonalityListLength
HDRPersonalityInd, 302
HDRPersonalityResp, 302
HDRProtSubtypResp, 303
unpack_nas_SLQSSwiGetHDRPersonality_t, 1151
unpack_nas_SLQSSwiGetHDRProtSubtype_t, 1151
pPersonalizationStatus
UIMGetConfigurationResp, 1006
unpack_uim_SLQSUIMGetConfiguration_t, 1218
pPhoneCtxtURI
GetIMSSMSConfigParams, 273
imsSMSConfigInfo, 326
SetIMSSMSConfigReq, 856
pPhoneCtxtURILen
GetIMSSMSConfigParams, 273
SetIMSSMSConfigReq, 856
pPhyCaAggPcellInfo
nasGetLTECphyCaResp, 540
pPhyCaAggScellDlBw
nasGetLTECphyCaResp, 540
pPhyCaAggScellIndType
nasGetLTECphyCaResp, 540
pPhyCaAggScellIndex
nasGetLTECphyCaResp, 540
pPhyCaAggScellInfo
nasGetLTECphyCaResp, 540
pPilotSetData
GetHRPDStatsResp, 272
unpack_nas_SLQSSwiGetHRPDStats_t, 1152
pPilotSetInfo
nas_PilotSetData, 491
PilotSetData, 752
pPkgDescLength
_SLQSOMADMSessionInfo, 91
pPkgDescription
_SLQSOMADMSessionInfo, 91
pPkgName
_SLQSOMADMSessionInfo, 91
pPkgNameLength
_SLQSOMADMSessionInfo, 91
pPktErrRate
swiQosFlow, 950
pPlasmaIDString
_SLQSSwiGetHostDevInfoParams, 96
_SLQSSwiSetHostDevInfoParams, 98
pPlmnId
nasNetworkReject, 555
pPollingTimer
_SLQSAVMSSetSettingsReqPara, 85
_SLQSAVMSSettings, 86
pPositionSource
pack_pds_SLQSPDSInjectPositionData_t, 661
PDSPositionData, 743
pPositionSrc
LocInjectPositionReq, 389
pPppSessCloseTimer1x
LibpackProfile3GPP2, 348
LibPackprofile_3GPP2, 359
Profile3GPP2, 771
pPppSessCloseTimerDO
LibpackProfile3GPP2, 348
LibPackprofile_3GPP2, 359
Profile3GPP2, 771
pPrDNSIPv4Address
swiPDPRuntimeSettingsResp, 944
pPrDNSIPv6Address
swiPDPRuntimeSettingsResp, 944
pPrPCSCFIPv4Address
swiPDPRuntimeSettingsResp, 944
pPrPCSCFIPv6Address
swiPDPRuntimeSettingsResp, 944
pPrecedence
swiQosFilter, 946
pPrecisionDilution
QmiCbkLocBestAvailPosInd, 785
QmiCbkLocPositionReportInd, 797
unpack_loc_BestAvailPos_Ind_t, 1089
unpack_loc_PositionRpt_Ind_t, 1100
pPrefNetwork
CurrDataSysStat, 200
pPrefVoiceDomain

pack_voice_SLQSVoiceSetConfig_t, 709
voiceSetConfigReq, 1374
pPrefVoicePrivacy
pack_voice_SLQSVoiceGetConfig_t, 703
voiceGetConfigReq, 1359
pPrefVoiceSO
pack_voice_SLQSVoiceGetConfig_t, 703
pack_voice_SLQSVoiceSetConfig_t, 709
voiceGetConfigReq, 1359
voiceSetConfigReq, 1374
pPrefVoiceSOStatus
unpack_voice_SLQSVoiceSetConfig_t, 1250
voiceSetConfigResp, 1375
pPreference
unpack_pds_GetXTRANetwork_t, 1165
pPreferred
QmiNas3GppNetworkInfo, 804
pPriCSCFPort
imsRegMgrConfigInfo, 324
SetRegMgrConfigReq, 867
pPriCSCFPortName
GetRegMgrConfigParams, 283
pPriCSCFPortNameLen
GetRegMgrConfigParams, 283
pPriDNSIPv4AddPref
LibpackProfile3GPP, 342
LibPackprofile_3GPP, 353
Profile3GPP, 765
pPriDNSIPv6addpref
LibpackProfile3GPP, 342
LibPackprofile_3GPP, 353
Profile3GPP, 765
pPriV6DnsAddress
LibpackProfile3GPP2, 348
LibPackprofile_3GPP2, 359
Profile3GPP2, 771
pPrimaryDNSV4
qmiWdsRunTimeSettings, 810
pPrimaryDNSV6
qmiWdsRunTimeSettings, 810
pPrimaryHA
pack_wds_SetMobileIPProfile_t, 723
pPrimaryID
LibpackProfile3GPP, 342
LibPackprofile_3GPP, 353
Profile3GPP, 765
pPrimaryV4DnsAddress
LibpackProfile3GPP2, 348
LibPackprofile_3GPP2, 359
Profile3GPP2, 771
pPriority
cdmaMsgDecodingParams, 167
cdmaMsgDecodingParamsExt, 170
cdmaMsgEncodingParams, 172
pPrivacy
cdmaMsgDecodingParams, 167
cdmaMsgDecodingParamsExt, 170
pProfSz
NWProfile, 587
pProfValues
NWProfile, 587
pProfileID
CreateProfileIn, 195
ModifyProfileIn, 418
qmiWdsRunTimeSettings, 810
unpack_wds_SLQSCreateProfile_t, 1268
pProfileId
pack_wds_SLQSCreateProfile_t, 724
pack_wds_SLQSModifyProfile_t, 729
pack_wds_SLQSSGetDHCPv4ClientConfig_t, 732
pack_wds_SLQSSSetDHCPv4ClientConfig_t, 733
WdsDHCPv4ClientLeaseInd, 1416
WdsDHCPv4Config, 1416
pProfileId3GPP
ssdatasession_params, 932
pProfileId3GPP2
ssdatasession_params, 932
swiQosFlow, 950
pProfileIndex
CreateProfileOut, 196
pProfileList
_slqs3GPPConfigItem, 83
pack_wds_SLQSSet3GPPConfigItem_t, 730
pProfileName
qmiWdsRunTimeSettings, 810
pProfileSettings
unpack_wds_SLQSGetProfileSettings_t, 1276
pProfileType
CreateProfileIn, 195
CreateProfileOut, 196
ModifyProfileIn, 418
pack_wds_SLQSCreateProfile_t, 724
pack_wds_SLQSModifyProfile_t, 729
pProfilename
LibpackProfile3GPP, 342
LibPackprofile_3GPP, 354
Profile3GPP, 766
pProfilenameSize
LibpackProfile3GPP, 342
LibPackprofile_3GPP, 354
Profile3GPP, 766
pProtoSubTypElmnt
HDRProtSubtypResp, 303
unpack_nas_SLQSSwiGetHDRProtSubtype_t, 1151
pProtocol
pack_nas_SetCDMANetworkParameters_t, 631
pProtocolSubtypeElement
HDRPersonalityInd, 302
HDRPersonalityResp, 302
unpack_nas_SLQSSwiGetHDRPersonality_t, 1151
pQFlowState
QosFlowInfo, 814
pQmiInterfaceInfo
_packetSrvStatus, 70
slqsSessionStateInfo, 890

slqsWdsEventInfo, 897
pQosClassID
LibpackProfile3GPP, 342
LibPackprofile_3GPP, 354
Profile3GPP, 766
pRAT
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 652
pRATINstance
unpack_nas_PerformNetworkScan_t, 1117
pRATInfo
_slqsNetworkScanInfo, 88
pRATInstanceSize
unpack_nas_PerformNetworkScan_t, 1117
pRATInstances
_slqsNetworkScanInfo, 88
pRATStatus
imsaRatStatusInfo, 316
pRATType
LibpackProfile3GPP2, 348
LibPackprofile_3GPP2, 359
Profile3GPP2, 771
PREFERRED_INDEX
qaNasPerformNetworkScan.h, 1994
pRFBandInfoElements
QmiNasGetRFBandInfoResp, 804
PRI_UPDATE_FAIL
qaGobiApiFms.h, 1715
PRIString
unpack_dms_GetFirmwareRevision_t, 1047
unpack_dms_GetFirmwareRevisions_t, 1048
PRLInd
qaQmiServingSystemParam, 779
unpack_nas_SLQSGetServingSystem_t, 1130
PRLPTlv
NASQmiCbkNasSystemSelPrefInd, 567
PRLPref
NASPRLPreferenceTlv, 565
pRMAutoConnect
custFeaturesInfo, 208
pRSRPThresList
nas_RSRPThresh, 502
RSRPThresh, 827
pRSRPThresh
pack_nas_SLQSConfigSigInfo_t, 633
sigInfo, 876
pRSRQThresList
nas_RSRQThresh, 503
RSRQThresh, 828
pRSRQThresh
pack_nas_SLQSConfigSigInfo_t, 633
sigInfo, 876
pRSSIThresList
nas_RSSIThresh, 504
RSSIThresh, 829
pRSSIThresh
pack_nas_SLQSConfigSigInfo_t, 633
sigInfo, 876
pRTPRTCPInactTimer
GetIMSVoIPConfigResp, 276
imsVoIPConfigInfo, 329
SetIMSVoIPConfigReq, 860
pRXAGCList
GetAudioPathConfigResp, 260
SetAudioPathConfigReq, 852
pRXAIG
RXAGCList, 830
pRXAVCAGCSwitch
GetAudioPathConfigResp, 260
SetAudioPathConfigReq, 852
pRXAVCList
GetAudioPathConfigResp, 260
SetAudioPathConfigReq, 852
pRXChain0Info
nasGetTxRxInfoResp, 545
unpack_nas_SLQSNasGetTxRxInfo_t, 1145
pRXChain1Info
nasGetTxRxInfoResp, 545
unpack_nas_SLQSNasGetTxRxInfo_t, 1145
pRXComprSlope
RXAGCList, 830
pRXComprThres
RXAGCList, 830
pRXDroppedCount
unpack_wds_GetPacketStatistics_t, 1264
WdsPktStatisticsElmnts, 1423
pRXExpSlope
RXAGCList, 830
pRXExpThres
RXAGCList, 830
pRXOKBytesCount
getDUNCallInfoResp, 269
pRXOKBytesLastCall
unpack_wds_GetPacketStatistics_t, 1264
WdsPktStatisticsElmnts, 1423
pRXOkBytesCount
unpack_wds_GetPacketStatistics_t, 1264
WdsPktStatisticsElmnts, 1423
pRXPCMIIRFltr
GetAudioPathConfigResp, 260
SetAudioPathConfigReq, 852
pRXPacketErrors
unpack_wds_GetPacketStatistics_t, 1264
WdsPktStatisticsElmnts, 1423
pRXPacketOverflows
unpack_wds_GetPacketStatistics_t, 1264
WdsPktStatisticsElmnts, 1423
pRXPacketSuccesses
unpack_wds_GetPacketStatistics_t, 1264
WdsPktStatisticsElmnts, 1423
pRXStaticGain
RXAGCList, 830
pRXTotalBytes
unpack_wds_GetByteTotals_t, 1254
WdsByteTotalsElmnts, 1413
pRadioInterface

nasNetworkTime, 556
unpack_nas_SLQSNasNetworkTimeCallBack_ind_t, 1146
pRankIndicatorInd
NasSwiIndReg, 572
pack_nas_SLQSNasSwiIndicationRegister_t, 644
pRatHandoverStatusConfig
IMSAIndRegisterInfo, 315
pRawHorConfidence
LocInjectPositionReq, 389
pRawHorUncCircular
LocInjectPositionReq, 389
pReRegPeriod
pack_wds_SetMobileIPParameters_t, 721
pReRegTraffic
pack_wds_SetMobileIPParameters_t, 721
pReadAcknowledgementReq
cdmaMsgDecodingParams, 167
cdmaMsgDecodingParamsExt, 170
pReadResult
UIMReadTransparentResp, 1011
unpack_uim_ReadTransparent_t, 1214
pReason
voiceSUPSInfo, 1382
pReasonMask
unpack_dms_GetOfflineReason_t, 1052
pRecurrenceType
LOCStartReq, 393
pack_loc_Start_t, 628
pRedirNumInfo
voiceInfoRec, 1364
pRefData
FirmwareUpdatStat, 249
pRefString
FirmwareUpdatStat, 249
pRefStringLen
FirmwareUpdatStat, 249
pReferenceNum
cdmaMsgDecodingParamsExt, 170
wcdmaLongMsgDecodingParams, 1388
pRefreshEvent
UIMRefreshGetLastEventResp, 1014
unpack_uim_SLQSUIMRefreshGetLastEvent_t, 1220
pRegCallStatInfoEvt
_getIndicationRegResp, 64
_setIndicationRegReq, 81
sms_getIndicationReg, 900
sms_setIndicationReg, 909
pRegDTMFEvents
pack_voice_SLQSVoiceIndicationRegister_t, 704
voiceIndicationRegisterInfo, 1362
pRegInd
_getTransLayerInfoResp, 66
sms_getTransLayerInfo, 901
pRegMgrConfigEvents
imsCfgIndRegisterInfo, 323
pRegStatus
_getTransNWRegInfoResp, 67
sms_getTransNWRegInfo, 902
pRegStatusConfig
IMSAIndRegisterInfo, 315
pRegStatusErrorCode
imsaRegStatusInfo, 318
pRegTransLayerInfoEvt
_getIndicationRegResp, 64
_setIndicationRegReq, 81
sms_getIndicationReg, 900
sms_setIndicationReg, 909
pRegTransNWRegInfoEvt
_getIndicationRegResp, 64
_setIndicationRegReq, 81
sms_getIndicationReg, 900
sms_setIndicationReg, 909
pRegVoicePrivacyEvents
pack_voice_SLQSVoiceIndicationRegister_t, 704
voiceIndicationRegisterInfo, 1362
pRejectReason
pack_swioma_SLQSOMADMSendSelection_t, 680
pRelValidity
cdmaMsgEncodingParams, 172
pRelativeValidity
cdmaMsgDecodingParams, 167
cdmaMsgDecodingParamsExt, 170
pRemainingRetries
UIMDepersonalizationResp, 1003
UIMPinResp, 1008
unpack_uim_ChangePin_t, 1213
unpack_uim_SetPinProtection_t, 1215
unpack_uim_SLQSUIMDepersonalization_t, 1217
unpack_uim_UnblockPin_t, 1221
unpack_uim_VerifyPin_t, 1222
pRemotePartyName
unpack_voice_SLQSVoiceGetCallInfo_t, 1236
voiceCallInfoResp, 1338
pRemotePartyNum
unpack_voice_SLQSVoiceGetCallInfo_t, 1236
voiceCallInfoResp, 1338
pReportChannelRate
getDUNCallInfoReq, 266
pack_wds_SLQSGetDUNCallInfo_t, 726
pReportConnStatus
getDUNCallInfoReq, 266
pack_wds_SLQSGetDUNCallInfo_t, 726
pReportDataBearerTech
getDUNCallInfoReq, 266
pack_wds_SLQSGetDUNCallInfo_t, 726
pReportDormStatus
getDUNCallInfoReq, 266
pack_wds_SLQSGetDUNCallInfo_t, 726
pReqFieldsList
IMSASupportedFieldsResp, 321
pReqMitigationLvl
TmdGetMitigationLvlResp, 968
pReqSettings

pack_wds_SLQSGetRuntimeSettings_t, 728
pRequestOptionList
pack_wds_SLQSSSetDHCPv4ClientConfig_t, 733
unpack_wds_SLQSSGetDHCPv4ClientConfig_t, 1284
WdsDHCPv4Config, 1416
pRequestedTag
pack_sms_SLQSGetSMSList_t, 672
pRespData
USSDRespFNetwork, 1293
pRespFieldsList
IMSASupportedFieldsResp, 321
pRetryCount
_SLQSOMADMSessionInfo, 91
pRetryInterval
pack_wds_SetMobileIPParameters_t, 721
pRetryLimit
pack_wds_SetMobileIPParameters_t, 721
pRetryMessage
slqssendasyncsmsparams_s, 888
sms_sendAsyncsmsParams, 908
pRetryMessageId
slqssendasyncsmsparams_s, 888
sms_sendAsyncsmsParams, 908
pRevInUse
CDMASysInfo, 177
nas_CDMASysInfo, 430
pRevInUseValid
CDMASysInfo, 177
nas_CDMASysInfo, 430
pRevTunneling
pack_wds_SetMobileIPProfile_t, 723
pRingBackTimer
GetIMSVoIPConfigResp, 276
imsVoIPConfigInfo, 328
SetIMSVoIPConfigReq, 860
pRingingTimer
GetIMSVoIPConfigResp, 276
imsVoIPConfigInfo, 329
SetIMSVoIPConfigReq, 860
pRoamPref
_sysSelectPrefInfo, 104
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 652
unpack_nas_SLQSGetSysSelectionPref_t, 1140
pRoamTimer
pack_voice_SLQSVoiceGetConfig_t, 703
voiceGetConfigReq, 1359
pRoamTimerCnt
unpack_voice_SLQSVoiceGetConfig_t, 1247
voiceGetConfigResp, 1361
pRoamTimerConfig
pack_voice_SLQSVoiceSetConfig_t, 709
voiceSetConfigReq, 1374
pRoamTimerStatus
unpack_voice_SLQSVoiceSetConfig_t, 1251
voiceSetConfigResp, 1375
pRoaming
pack_nas_SetCDMANetworkParameters_t, 631
QmiNas3GppNetworkInfo, 804
pRscp
nasSigInfo, 570
unpack_nas_SLQSNasSigInfoCallback_ind_t, 1147
pRxFilter
swiQosModifyReq, 951
swiQosReq, 952
pRxFlow
swiQosGranted, 950
swiQosModifyReq, 951
swiQosReq, 952
pRxQFilter
QosFlowInfo, 814
pRxQFlowGranted
QosFlowInfo, 814
PSDetachReq, 773
pDetachAction, 774
PSDomain
unpack_nas_GetServingNetwork_t, 1114
pSID
unpack_nas_GetHomeNetwork3GPP2_t, 1109
pSIPConfigEvents
imsCfgIndRegisterInfo, 323
pSIPLocalPort
GetSIPConfigResp, 284
imsSIPConfigInfo, 325
SetSIPConfigReq, 873
pSMSAttemptedFlag
getDyingGaspStatistics, 270
packgetDyingGaspStatistics, 739
pSMSCAddress
pack_sms_SetSMSCAddress_t, 670
unpack_sms_GetSMSCAddress_t, 1196
pSMSCAddressInfo
SMSEventInfo_s, 918
pSMSCType
pack_sms_SetSMSCAddress_t, 670
unpack_sms_GetSMSCAddress_t, 1196
pSMSConfigEvents
imsCfgIndRegisterInfo, 323
pSMSFormat
GetIMSSMSConfigParams, 273
imsSMSConfigInfo, 326
SetIMSSMSConfigReq, 856
pSMSOnIMSInfo
SMSEventInfo_s, 918
pSMSOverIPNwInd
GetIMSSMSConfigParams, 273
imsSMSConfigInfo, 326
SetIMSSMSConfigReq, 856
pSMSSupport
custFeaturesInfo, 208
pSMSSvcRAT
imsaSvcStatusInfo, 322
pSMSSvcStatus
imsaSvcStatusInfo, 322

pSPC
pack_nas_SetCDMANetworkParameters_t, 631
pack_wds_SetMobileIPParameters_t, 721
pSV
BdsSVInfo, 141
loc_BdsSVInfo, 367
loc_SVInfo, 378
SVInfo, 935
pSVInfo
LocDelAssDataReq, 382
pack_loc_Delete_Assist_Data_t, 614
pSWVerString
_SLQSSwiGetHostDevInfoParams, 96
_SLQSSwiSetHostDevInfoParams, 98
pSarRFState
unpack_sar_SLQSGetRfSarState_t, 1196
pSatelliteInfo
gnssSvInfoNotification, 287
unpack_loc_GnssSvInfo_Ind_t, 1094
pScAddr
wcdmaLongMsgDecodingParams, 1388
wcdmaMsgDecodingParams, 1389
pScAddrLength
wcdmaLongMsgDecodingParams, 1388
wcdmaMsgDecodingParams, 1389
pScanResult
_slqsNetworkScanInfo, 88
unpack_nas_PerformNetworkScan_t, 1117
pSccRxInfo
LteSccRxInfoResp, 405
unpack_nas_SLQSSwiGetLteSccRxInfo_t, 1153
pScrAmrEnable
GetIMSVoIPConfigResp, 276
imsVoIPConfigInfo, 329
SetIMSVoIPConfigReq, 860
pScrAmrWbEnable
GetIMSVoIPConfigResp, 276
imsVoIPConfigInfo, 329
SetIMSVoIPConfigReq, 860
pSeDNSIPv4Address
swiPDPRuntimeSettingsResp, 944
pSeDNSIPv6Address
swiPDPRuntimeSettingsResp, 944
pSePCSCFIPv4Address
swiPDPRuntimeSettingsResp, 944
pSePCSCFIPv6Address
swiPDPRuntimeSettingsResp, 944
pSecDNSIPv4AddPref
LibpackProfile3GPP, 342
LibPackprofile_3GPP, 354
Profile3GPP, 766
pSecDNSIPv6addpref
LibpackProfile3GPP, 342
LibPackprofile_3GPP, 354
Profile3GPP, 766
pSecV6DnsAddress
LibpackProfile3GPP2, 348
LibPackprofile_3GPP2, 359
Profile3GPP2, 771
pSecondaryDNSV4
qmiWdsRunTimeSettings, 811
pSecondaryDNSV6
qmiWdsRunTimeSettings, 811
pSecondaryFlag
LibpackProfile3GPP, 342
LibPackprofile_3GPP, 354
Profile3GPP, 766
pSecondaryHA
pack_wds_SetMobileIPProfile_t, 723
pSecondaryV4DnsAddress
LibpackProfile3GPP2, 348
LibPackprofile_3GPP2, 359
Profile3GPP2, 771
pSectorID
nas_NetworkStatEVDO, 483
NetworkStatEVDO, 583
pSendSmsParams
pack_sms_SLQSSendAsyncSMS_t, 674
pSenderAddr
cdmaMsgDecodingParams, 167
cdmaMsgDecodingParamsExt, 170
wcdmaLongMsgDecodingParams, 1388
wcdmaMsgDecodingParams, 1389
pSenderAddrLength
cdmaMsgDecodingParams, 167
cdmaMsgDecodingParamsExt, 170
wcdmaLongMsgDecodingParams, 1388
wcdmaMsgDecodingParams, 1389
pSensorDataUsage
QmiCbkLocBestAvailPosInd, 785
QmiCbkLocPositionReportInd, 797
unpack_loc_BestAvailPos_Ind_t, 1089
unpack_loc_PositionRpt_Ind_t, 1100
pServerAddrList
qmiWdsRunTimeSettings, 811
pServerAddrTypeMask
LocGetServerReq, 385
pServerAddress
pack_pds_SLQSSetAGPSConfig_t, 662
unpack_pds_SLQSGetAGPSConfig_t, 1169
pServerPort
pack_pds_SLQSSetAGPSConfig_t, 662
unpack_pds_SLQSGetAGPSConfig_t, 1169
pServerURL
pack_pds_SLQSSetAGPSConfig_t, 662
unpack_pds_SLQSGetAGPSConfig_t, 1169
pServerURLLength
pack_pds_SLQSSetAGPSConfig_t, 662
unpack_pds_SLQSGetAGPSConfig_t, 1169
pServiceClass
pack_voice_SLQSVoiceSetSUPSService_t, 711
voiceSetSUPSServiceReq, 1378
pServiceOption
slqssendasyncsmsparams_s, 888
sms_sendAsyncsmsParams, 908
pServiceStatusConfig

IMSAIndRegisterInfo, 315
pServingSystemInd
nasIndicationRegisterReq, 549
pack_nas_SLQSNasIndicationRegisterExt_t, 643
pSessionExpiryTimer
GetIMSVoIPConfigResp, 276
imsVoIPConfigInfo, 329
SetIMSVoIPConfigReq, 860
pSessionIDv4
GetSessionIDResp, 283
pSessionIDv6
GetSessionIDResp, 283
pSessionState
_SLQSOMADMSessionInfo, 91
pSessionType
_SLQSOMADMSessionInfo, 91
pSetIndicationRegReq
pack_sms_SLQSSetIndicationRegister_t, 674
pSetRoutesReq
pack_sms_SLQSSmsSetRoutes_t, 677
pSettingResp
GetIMSSMSConfigParams, 273
GetIMSUserConfigParams, 274
GetIMSVoIPConfigResp, 276
GetRegMgrConfigParams, 283
GetSIPConfigResp, 284
SetIMSSMSConfigResp, 857
SetIMSUserConfigResp, 858
SetIMSVoIPConfigResp, 861
SetRegMgrConfigResp, 867
SetSIPConfigResp, 874
pSeverity
_SLQSOMADMSessionInfo, 91
pSigCompEnabled
GetSIPConfigResp, 284
imsSIPConfigInfo, 325
SetSIPConfigReq, 873
pSigIndReq
pack_nas_SLQSSetSignalStrengthsCallback_t, 647
pSignalInfo
voiceInfoRec, 1364
pSignalStrengthInd
nasIndicationRegisterReq, 549
pack_nas_SLQSNasIndicationRegisterExt_t, 643
pSmsOnIms
slqssendasyncsmsparams_s, 888
slqssendsmsparams_s, 889
sms_sendAsyncsmsParams, 908
pSmsServiceRat
IMSAServiceStatus, 320
pSmsServiceStatus
IMSAServiceStatus, 320
pSmsStorage
unpack_sms_SLQSSwiGetSMSStorage_t, 1204
pSource
_SLQSOMADMSessionInfo, 91
pSourceIP
LibPackTFTIDParams, 362
TFTIDParams, 964
pSourceLength
_SLQSOMADMSessionInfo, 91
pSpeedHorizontal
QmiCbkLocBestAvailPosInd, 785
QmiCbkLocPositionReportInd, 797
unpack_loc_BestAvailPos_Ind_t, 1089
unpack_loc_PositionRpt_Ind_t, 1100
pSpeedUnc
QmiCbkLocBestAvailPosInd, 785
QmiCbkLocPositionReportInd, 797
unpack_loc_BestAvailPos_Ind_t, 1089
unpack_loc_PositionRpt_Ind_t, 1100
pSpeedVertical
QmiCbkLocBestAvailPosInd, 785
QmiCbkLocPositionReportInd, 797
unpack_loc_BestAvailPos_Ind_t, 1089
unpack_loc_PositionRpt_Ind_t, 1100
pSpeedVerticalUnc
QmiCbkLocBestAvailPosInd, 785
unpack_loc_BestAvailPos_Ind_t, 1089
pSrcRAT
imsaRatStatusInfo, 316
pSrvDomainPref
_sysSelectPrefInfo, 104
_sysSelectPrefParams, 111
pack_nas_SLQSSetSysSelectionPref_t, 652
unpack_nas_SLQSGetSysSelectionPref_t, 1140
pSrvOpt
unpack_voice_SLQSVoiceGetCallInfo_t, 1236
voiceCallInfoResp, 1338
pSrvRegRestriction
_sysSelectPrefParams, 111
pack_nas_SLQSSetSysSelectionPref_t, 652
pSrvcProviderName
nasOperatorNameResp, 557
unpack_nas_SLQSGetOperatorNameData_t, 1123
pStage0Val
RXPCMIIRFltr, 833
TXPCMIIRFltr, 976
pStage1Val
RXPCMIIRFltr, 833
TXPCMIIRFltr, 976
pStage2Val
RXPCMIIRFltr, 833
TXPCMIIRFltr, 976
pStage3Val
RXPCMIIRFltr, 833
TXPCMIIRFltr, 976
pStage4Val
RXPCMIIRFltr, 833
TXPCMIIRFltr, 976
pStageCnt
RXPCMIIRFltr, 833
TXPCMIIRFltr, 976
pStatMask
pack_wds_GetPacketStatistics_t, 715

WdsPktStatisticsReq, 1424
pStatus
_SLQSOMADMSessionInfo, 91
pSubnetMaskV4
qmiWdsRunTimeSettings, 811
pSubscribeTimer
GetSIPConfigResp, 284
imsSIPConfigInfo, 325
SetSIPConfigReq, 873
pSubscriptionInfoInd
nasIndicationRegisterReq, 549
pack_nas_SLQSNasIndicationRegisterExt_t, 643
pSupUSBComps
USBCompParams, 1292
pSupportedMsgList
IMSASupportedMsgInfo, 321
pSuppressSysInfoInd
nasIndicationRegisterReq, 549
pack_nas_SLQSNasIndicationRegisterExt_t, 643
pSuppsNotifEvents
pack_voice_SLQSVoiceIndicationRegister_t, 704
voiceIndicationRegisterInfo, 1362
pSvUsedforFix
QmiCbkLocBestAvailPosInd, 785
QmiCbkLocPositionReportInd, 797
unpack_loc_BestAvailPos_Ind_t, 1089
unpack_loc_PositionRpt_Ind_t, 1100
pSvcClass
pack_voice_SLQSVoiceGetCallBarring_t, 700
pack_voice_SLQSVoiceGetCallForwardingStatus-
_t, 701
pack_voice_SLQSVoiceGetCallWaiting_t, 702
unpack_voice_SLQSVoiceGetCallBarring_t, 1232
unpack_voice_SLQSVoiceGetCallWaiting_t, 1238
voiceGetCallBarringReq, 1346
voiceGetCallBarringResp, 1348
voiceGetCallFWReq, 1348
voiceGetCallWaitInfo, 1351
voiceSUPSInfo, 1382
pSvcType
pack_voice_SLQSVoiceDialCall_t, 699
voiceCallRequestParams, 1340
pSwiGetResetInd
dmsIndicationRegisterReq, 232
pSysInfoInd
nasIndicationRegisterReq, 549
pack_nas_SLQSNasIndicationRegisterExt_t, 643
pSysInfoNoChange
nasSysInfo, 576
unpack_nas_SLQSSysInfoCallback_ind_t, 1157
pSystemSelectionInd
nasIndicationRegisterReq, 549
pack_nas_SLQSNasIndicationRegisterExt_t, 643
pTCPDstPort
swiQosFilter, 946
pTCPSrcPort
swiQosFilter, 946
pTDSCDMAECIODelta
pack_nas_SLQSNasConfigSigInfo2_t, 639
setSignalStrengthInfo, 872
pTDSCDMAECIOThresh
pack_nas_SLQSNasConfigSigInfo2_t, 639
setSignalStrengthInfo, 872
pTDSCDMAECIOThreshList
nas_TDSCDMAECIOThresh, 518
TDSCDMAECIOThresh, 956
pTDSCDMARSCPDelta
pack_nas_SLQSNasConfigSigInfo2_t, 639
setSignalStrengthInfo, 872
pTDSCDMARSCPThresh
pack_nas_SLQSNasConfigSigInfo2_t, 639
setSignalStrengthInfo, 872
pTDSCDMARSCPThreshList
nas_TDSCDMARSCPThresh, 518
TDSCDMARSCPThresh, 957
pTDSCDMARSSIDelta
pack_nas_SLQSNasConfigSigInfo2_t, 639
setSignalStrengthInfo, 872
pTDSCDMARSSIThresh
pack_nas_SLQSNasConfigSigInfo2_t, 639
setSignalStrengthInfo, 872
pTDSCDMARSSIThreshList
nas_TDSCDMARSSIThresh, 519
TDSCDMARSSIThresh, 957
pTDSCDMASINRCONFThresh
sigInfo, 876
pTDSCDMASINRDelta
pack_nas_SLQSNasConfigSigInfo2_t, 639
setSignalStrengthInfo, 872
pTDSCDMASINRThresh
pack_nas_SLQSNasConfigSigInfo2_t, 639
setSignalStrengthInfo, 872
pTDSCDMASINRThreshList
nas_TDSCDMASINRThresh, 520
TDSCDMASINRThresh, 960
pTDSCDMASigInfoExt
nasGetSigInfoResp, 541
nasSigInfo, 570
unpack_nas_SLQSNasSigInfoCallback_ind_t,
1147
pTDSCDMASigInfoRscp
nasGetSigInfoResp, 541
pTFTID1Params
LibpackProfile3GPP, 342
LibPackprofile_3GPP, 354
Profile3GPP, 766
pTFTID2Params
LibpackProfile3GPP, 342
LibPackprofile_3GPP, 354
Profile3GPP, 766
pTTYConfigStatus
unpack_voice_SLQSVoiceSetConfig_t, 1251
voiceSetConfigResp, 1375
pTTYMode
pack_voice_SLQSVoiceGetConfig_t, 703
pack_voice_SLQSVoiceSetConfig_t, 709

voiceGetConfigReq, 1359
voiceSetConfigReq, 1374
pTXAGCList
GetAudioPathConfigResp, 260
SetAudioPathConfigReq, 852
pTXAIG
TXAGCList, 974
pTXAVCSwitch
GetAudioPathConfigResp, 260
SetAudioPathConfigReq, 852
pTXComprSlope
TXAGCList, 974
pTXComprThres
TXAGCList, 974
pTXDroppedCount
unpack_wds_GetPacketStatistics_t, 1264
WdsPktStatisticsElmnts, 1424
pTXExpSlope
TXAGCList, 974
pTXExpThres
TXAGCList, 974
pTXGain
GetAudioPathConfigResp, 260
SetAudioPathConfigReq, 852
pTXInfo
nasGetTxRxInfoResp, 545
unpack_nas_SLQSNasGetTxRxInfo_t, 1145
pTXOKBytesCount
getDUNCallInfoResp, 269
pTXOKBytesLastCall
unpack_wds_GetPacketStatistics_t, 1264
WdsPktStatisticsElmnts, 1424
pTXOkBytesCount
unpack_wds_GetPacketStatistics_t, 1264
WdsPktStatisticsElmnts, 1424
pTXPCMIIRFltr
GetAudioPathConfigResp, 260
SetAudioPathConfigReq, 852
pTXPacketErrors
unpack_wds_GetPacketStatistics_t, 1264
WdsPktStatisticsElmnts, 1424
pTXPacketOverflows
unpack_wds_GetPacketStatistics_t, 1264
WdsPktStatisticsElmnts, 1424
pTXPacketSuccesses
unpack_wds_GetPacketStatistics_t, 1264
WdsPktStatisticsElmnts, 1424
pTXStaticGain
TXAGCList, 974
pTXTotalBytes
unpack_wds_GetByteTotals_t, 1254
WdsByteTotalsElmnts, 1413
pTdsBandCapability
BandCapabilityResp, 140
pTdscdmaBandPref
_sysSelectPrefParams, 111
pack_nas_SLQSSetSysSelectionPref_t, 653
pTech
pack_wds_SLQSStartDataSession_t, 734
pTechnology
qmiWdsRunTimeSettings, 811
ssdatasession_params, 932
pTechnologyMask
QmiCbkLocBestAvailPosInd, 785
QmiCbkLocPositionReportInd, 797
unpack_loc_BestAvailPos_Ind_t, 1089
unpack_loc_PositionRpt_Ind_t, 1100
pTextMsg
cdmaMsgDecodingParams, 167
cdmaMsgDecodingParamsExt, 170
cdmaMsgEncodingParams, 172
wcdmaLongMsgDecodingParams, 1388
wcdmaMsgDecodingParams, 1389
wcdmaMsgEncodingParams, 1390
pTextMsgLength
cdmaMsgDecodingParams, 167
cdmaMsgDecodingParamsExt, 170
wcdmaLongMsgDecodingParams, 1388
wcdmaMsgDecodingParams, 1389
pTgtRAT
imsaRatStatusInfo, 316
pTime
_SLQSOMADMSessionInfo, 91
SwiOTAMsg_s, 941
pTimeLength
_SLQSOMADMSessionInfo, 91
pTimeSrc
QmiCbkLocBestAvailPosInd, 785
QmiCbkLocPositionReportInd, 797
unpack_loc_BestAvailPos_Ind_t, 1089
unpack_loc_PositionRpt_Ind_t, 1100
pTimeStamp
getDyingGaspStatistics, 271
pack_pds_SLQSPDSInjectPositionData_t, 661
packgetDyingGaspStatistics, 739
PDSPositionData, 743
pTimeType
pack_pds_SLQSPDSInjectPositionData_t, 661
PDSPositionData, 743
pTimeUnc
QmiCbkLocBestAvailPosInd, 785
QmiCbkLocPositionReportInd, 797
unpack_loc_BestAvailPos_Ind_t, 1089
unpack_loc_PositionRpt_Ind_t, 1101
pTimeZone
nasNetworkTime, 556
unpack_nas_SLQSNasNetworkTimeCallBack_ind-
_t, 1146
pTimeout
unpack_pds_GetPDSDefaults_t, 1162
pTimer
NasSwiIndReg, 572
pack_nas_SLQSNasSwiIndicationRegister_t, 644
pTimerSIPReg
GetSIPConfigResp, 284
imsSIPConfigInfo, 325

SetSIPConfigReq, 873
pTimerT1
GetSIPConfigResp, 284
imsSIPConfigInfo, 325
SetSIPConfigReq, 873
pTimerT2
GetSIPConfigResp, 284
imsSIPConfigInfo, 325
SetSIPConfigReq, 873
pTimerTf
GetSIPConfigResp, 284
imsSIPConfigInfo, 325
SetSIPConfigReq, 873
pTimerVal
pack_voice_SLQSVoiceSetSUPSService_t, 711
voiceSetSUPSServiceReq, 1378
pTimestampAge
LocInjectPositionReq, 389
pTimestampUtc
LocInjectPositionReq, 389
QmiCbkLocBestAvailPosInd, 785
QmiCbkLocPositionReportInd, 797
unpack_loc_BestAvailPos_Ind_t, 1089
unpack_loc_PositionRpt_Ind_t, 1100
pTokenBucket
swiQosFlow, 950
pTos
swiQosFilter, 946
pTotalBytesRX
QosEventInfo, 813
slqsWdsEventInfo, 898
pTotalBytesTX
QosEventInfo, 813
slqsWdsEventInfo, 898
pTotalNum
cdmaMsgDecodingParamsExt, 170
wcdmaLongMsgDecodingParams, 1388
pTrackingStatus
unpack_pds_GetPDSState_t, 1163
pTrafficClass
swiQosFlow, 950
pTranDstPort
swiQosFilter, 946
pTranSrcPort
swiQosFilter, 946
pTransLayerInfo
_getTransLayerInfoResp, 66
_transLayerInfoNotification, 112
sms_getTransLayerInfo, 901
pTransferRouteMTMessageInfo
SMSEventInfo_s, 918
pTransferStatInd
getDUNCallInfoReq, 266
pack_wds_SLQSGetDUNCallInfo_t, 726
pack_wds_SLQSWdsSetEventReport_t, 737
wdsSetEventReportReq, 1428
pTransferStatusReport
sms_setRoutesReq, 910
smsSetRoutesReq, 925
pTrueIMSI
nasGet3GPP2SubscriptionInfoResp, 538
unpack_nas_SLQSNasGet3GPP2Subscription_t, 1141
pTxFilter
swiQosModifyReq, 951
swiQosReq, 952
pTxFlow
swiQosGranted, 950
swiQosModifyReq, 951
swiQosReq, 952
pTxQFilter
QosFlowInfo, 814
pTxQFlowGranted
QosFlowInfo, 814
pTypeCode
USSDRespFNetwork, 1293
pUATI
GetHRPDStatsResp, 272
unpack_nas_SLQSSwiGetHRPDStats_t, 1152
pUDPDstPort
swiQosFilter, 946
pUDPSrcPort
swiQosFilter, 946
pUMTSCellID
nasCellLocationInfoResp, 536
unpack_nas_SLQSNasGetCellLocationInfo_t, 1143
pUMTSGrantedQoS
qmiWdsRunTimeSettings, 811
pUMTSInfo
nasCellLocationInfoResp, 536
unpack_nas_SLQSNasGetCellLocationInfo_t, 1143
pUMTSMinQoS
LibpackProfile3GPP, 343
LibPackprofile_3GPP, 354
Profile3GPP, 766
pUMTSMinQosSigInd
LibpackProfile3GPP, 343
LibPackprofile_3GPP, 354
Profile3GPP, 766
pUMTSReqQoS
LibpackProfile3GPP, 343
LibPackprofile_3GPP, 354
Profile3GPP, 766
pUMTSReqQoSSigInd
LibpackProfile3GPP, 343
LibPackprofile_3GPP, 354
Profile3GPP, 766
pURL
QmiCbkLocGetServerInd, 788
unpack_loc_GetServer_Ind_t, 1093
pUSBComp
USBCompConfig, 1290
USBCompParams, 1292
pUSSDData

USSDNoWaitIndicationInfo, 1293
pUSSDInfo
unpack_voice_SLQSOriginateUSSD_t, 1225
USSResp, 1295
pUSSInfo
voiceSUPSInfo, 1382
pUTSvcRAT
imsaSvcStatusInfo, 322
pUTSvcStatus
imsaSvcStatusInfo, 322
pUUSInFo
pack_voice_SLQSVoiceDialCall_t, 699
voiceCallRequestParams, 1340
pUUSInfo
unpack_voice_SLQSVoiceGetCallInfo_t, 1236
voiceCallInfoResp, 1338
pUimSlotsStatus
UIMGetSlotsStatusResp, 1007
unpack_uim_SLQSUIMGetSlotsStatus_t, 1219
pUname
_SLQSAVMSSettingsAPNInfo, 86
pUpdateCompleteStatus
_SLQSOMADMSessionInfo, 91
pUser
pack_wds_SLQSStartDataSession_t, 734
pUserAcknowledgementReq
cdmaMsgDecodingParams, 167
cdmaMsgDecodingParamsExt, 170
pUserConfigEvents
imsCfgIndRegisterInfo, 323
pUserData
slqssendasyncsmsparams_s, 888
sms_sendAsyncsmsParams, 908
pUserId
LibpackProfile3GPP2, 348
LibPackprofile_3GPP2, 359
Profile3GPP2, 771
pUserIdSize
LibpackProfile3GPP2, 348
LibPackprofile_3GPP2, 359
Profile3GPP2, 771
pUsername
LibpackProfile3GPP, 343
LibPackprofile_3GPP, 354
pack_wds_SetDefaultProfile_t, 719
Profile3GPP, 766
qmiWdsRunTimeSettings, 811
ssdatasession_params, 932
pUsernameSize
LibpackProfile3GPP, 343
LibPackprofile_3GPP, 354
Profile3GPP, 766
pUtServiceRat
IMSAServiceStatus, 320
pUtServiceStatus
IMSAServiceStatus, 320
pV4sessionId
WdsByteTotals, 1412
WdsConnectionRate, 1414
WdsPktStatisticsResp, 1425
pV6sessionId
WdsByteTotals, 1412
WdsConnectionRate, 1414
WdsPktStatisticsResp, 1425
pVOIPSvcRAT
imsaSvcStatusInfo, 322
pVOIPSvcStatus
imsaSvcStatusInfo, 322
pVTSvcRAT
imsaSvcStatusInfo, 322
pVTSvcStatus
imsaSvcStatusInfo, 322
pVerboseFailReasonType
unpack_wds_SLQSStartDataSession_t, 1285
pVerboseFailureReason
unpack_wds_SLQSStartDataSession_t, 1285
pVersionString
_SLQSSwiGetOSInfoParams, 97
_SLQSSwiSetOSInfoParams, 99
pVertConfidence
LocInjectPositionReq, 389
QmiCbkLocBestAvailPosInd, 785
QmiCbkLocPositionReportInd, 797
unpack_loc_BestAvailPos_Ind_t, 1089
unpack_loc_PositionRpt_Ind_t, 1101
pVertReliability
LocInjectPositionReq, 389
QmiCbkLocBestAvailPosInd, 785
QmiCbkLocPositionReportInd, 797
unpack_loc_BestAvailPos_Ind_t, 1089
unpack_loc_PositionRpt_Ind_t, 1101
pVertUnc
LocInjectPositionReq, 389
QmiCbkLocBestAvailPosInd, 785
QmiCbkLocPositionReportInd, 797
unpack_loc_BestAvailPos_Ind_t, 1089
unpack_loc_PositionRpt_Ind_t, 1101
pVerticalConfidence
pack_pds_SLQSPDSInjectPositionData_t, 661
PDSPositionData, 743
pVerticalUnc
pack_pds_SLQSPDSInjectPositionData_t, 661
PDSPositionData, 743
pVoIPConfigEvents
imsCfgIndRegisterInfo, 323
pVoiceDomainPref
pack_voice_SLQSVoiceGetConfig_t, 703
voiceGetConfigReq, 1359
pVoiceDomainPrefStatus
unpack_voice_SLQSVoiceSetConfig_t, 1251
voiceSetConfigResp, 1375
pVoicePrivacy
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1230
unpack_voice_SLQSVoiceGetCallInfo_t, 1236
voiceCallInfoResp, 1338
voiceGetAllCallInfo, 1346

pVoipServiceRat
IMSAServiceStatus, 320
pVoipServiceStatus
IMSAServiceStatus, 320
pVolume
SetM2MAudioProfileReq, 863
pVsServiceRat
IMSAServiceStatus, 320
pVsServiceStatus
IMSAServiceStatus, 320
pVtServiceRat
IMSAServiceStatus, 320
pVtServiceStatus
IMSAServiceStatus, 320
pWCDMABER
GetErrRateResp, 271
unpack_nas_SLQSGetErrorRate_t, 1121
pWCDMACallBarringSysInfo
nasGetSysInfoResp, 544
nasSysInfo, 576
unpack_nas_SLQSGetSysInfo_t, 1135
unpack_nas_SLQSSysInfoCallback_ind_t, 1157
pWCDMACipherDomainSysInfo
nasGetSysInfoResp, 544
nasSysInfo, 576
unpack_nas_SLQSGetSysInfo_t, 1135
unpack_nas_SLQSSysInfoCallback_ind_t, 1157
pWCDMAECIODelta
pack_nas_SLQSNasConfigSigInfo2_t, 639
setSignalStrengthInfo, 872
pWCDMAECIOThresh
pack_nas_SLQSNasConfigSigInfo2_t, 639
setSignalStrengthInfo, 872
pWCDMAECIOThreshList
nas_WCDMAECIOThresh, 529
WCDMAECIOThresh, 1385
pWCDMAInfoLTENeighborCell
nasCellLocationInfoResp, 536
unpack_nas_SLQSNasGetCellLocationInfo_t, 1143
pWCDMARSSIDelta
pack_nas_SLQSNasConfigSigInfo2_t, 639
setSignalStrengthInfo, 872
pWCDMARSSIThresh
pack_nas_SLQSNasConfigSigInfo2_t, 639
setSignalStrengthInfo, 872
pWCDMARSSIThreshList
nas_WCDMARSSIThresh, 531
WCDMARSSIThresh, 1391
pWCDMASSInfo
nasGetSigInfoResp, 541
pWCDMASigInfo
nasSigInfo, 570
unpack_nas_SLQSNasSigInfoCallback_ind_t, 1147
pWCDMASrvStatusInfo
nasGetSysInfoResp, 544
nasSysInfo, 576
unpack_nas_SLQSGetSysInfo_t, 1136
unpack_nas_SLQSSysInfoCallback_ind_t, 1157
pWCDMASysInfo
nasGetSysInfoResp, 544
nasSysInfo, 576
unpack_nas_SLQSGetSysInfo_t, 1136
unpack_nas_SLQSSysInfoCallback_ind_t, 1157
pWcdmaUARFCN
nasSwiGetChannelLockResp, 571
nasSwiSetChannelLockReq, 573
pack_nas_SLQSNASSwiSetChannelLock_t, 645
unpack_nas_SLQSNASSwiGetChannelLock_t, 1148
pWifiState
pack_pds_SLQSSetPositionMethodState_t, 663
PDSPosMethodStateReq, 744
pXid
QmiCbkLocBestAvailPosInd, 785
unpack_loc_BestAvailPos_Ind_t, 1089
pXtraDataState
pack_pds_SLQSSetPositionMethodState_t, 663
PDSPosMethodStateReq, 744
pXtraTimeState
pack_pds_SLQSSetPositionMethodState_t, 663
PDSPosMethodStateReq, 744
pack_dms_ActivateAutomatic
dms.h, 1444
pack_dms_ActivateAutomatic_t, 597
actCode, 597
pack_dms_GetActivationState
dms.h, 1444
pack_dms_GetBandCapability
dms.h, 1444
pack_dms_GetCrashAction
dms.h, 1445
pack_dms_GetCustFeature
dms.h, 1445
pack_dms_GetCustFeaturesV2
dms.h, 1445
pack_dms_GetCustFeaturesV2_t, 597
cust_id, 598
list_type, 598
Tlvresult, 598
pack_dms_GetDeviceCap
dms.h, 1446
pack_dms_GetDeviceCapabilities
dms.h, 1446
pack_dms_GetDeviceHardwareRev
dms.h, 1446
pack_dms_GetDeviceMfr
dms.h, 1447
pack_dms_GetDeviceSerialNumbers
dms.h, 1447
pack_dms_GetFSN
dms.h, 1449
pack_dms_GetFirmwareInfo
dms.h, 1448
pack_dms_GetFirmwareRevision

dms.h, 1448
pack_dms_GetFirmwareRevisions
dms.h, 1448
pack_dms_GetHardwareRevision
dms.h, 1449
pack_dms_GetIMSI
dms.h, 1449
pack_dms_GetManufacturer
dms.h, 1450
pack_dms_GetModelID
dms.h, 1450
pack_dms_GetNetworkTime
dms.h, 1450
pack_dms_GetOfflineReason
dms.h, 1451
pack_dms_GetPRLVersion
dms.h, 1451
pack_dms_GetPower
dms.h, 1451
pack_dms_GetSerialNumbers
dms.h, 1452
pack_dms_GetUSBComp
dms.h, 1452
pack_dms_GetVoiceNumber
dms.h, 1452
pack_dms_ResetToFactoryDefaults
dms.h, 1453
pack_dms_ResetToFactoryDefaults_t, 598
spc, 598
pack_dms_SLQSDmsSwiGetResetInfo
dms.h, 1456
pack_dms_SLQSDmsSwiIndicationRegister
dms.h, 1456
pack_dms_SLQSDmsSwiIndicationRegister_t, 604
resetInfoInd, 605
pack_dms_SLQSGetBandCapability
dms.h, 1457
pack_dms_SLQSGetERIFile
dms.h, 1457
pack_dms_SLQSSwiClearDyingGaspStatistics
dms.h, 1458
pack_dms_SLQSSwiGetCrashInfo
dms.h, 1458
pack_dms_SLQSSwiGetCrashInfo_t, 605
clear, 605
pack_dms_SLQSSwiGetDyingGaspCfg
dms.h, 1458
pack_dms_SLQSSwiGetDyingGaspStatistics
dms.h, 1459
pack_dms_SLQSSwiGetFirmwareCurr
dms.h, 1459
pack_dms_SLQSSwiGetFwUpdateStatus
dms.h, 1459
pack_dms_SLQSSwiGetHostDevInfo
dms.h, 1460
pack_dms_SLQSSwiGetOSInfo
dms.h, 1460
pack_dms_SLQSSwiGetSerialNoExt
dms.h, 1460
pack_dms_SLQSSwiSetDyingGaspCfg
dms.h, 1461
pack_dms_SLQSSwiSetDyingGaspCfg_t, 605
pDestSMSContent, 605
pDestSMSNum, 605
pack_dms_SLQSSwiSetHostDevInfo
dms.h, 1461
pack_dms_SLQSSwiSetHostDevInfo_t, 606
manString, 606
modelString, 606
plasmaIDString, 606
swVerString, 606
pack_dms_SLQSSwiSetOSInfo
dms.h, 1461
pack_dms_SLQSSwiSetOSInfo_t, 606
nameString, 607
versionString, 607
pack_dms_SLQSUIMGetState
dms.h, 1462
pack_dms_SetActivationStatusCallback
dms.h, 1453
pack_dms_SetActivationStatusCallback_t, 598
activationState, 599
pack_dms_SetCrashAction
dms.h, 1454
pack_dms_SetCrashAction_t, 599
crashAction, 599
pack_dms_SetCustFeature
dms.h, 1454
pack_dms_SetCustFeature_t, 599
DHCPRelayEnabled, 601
DisableIMSI, 601
GPSLPM, 601
GPSSel, 601
GpsEnable, 601
IPFamSupport, 601
IsVoiceEnabled, 601
RMAutoConnect, 601
SMSSupport, 601
pack_dms_SetCustFeaturesV2
dms.h, 1454
pack_dms_SetCustFeaturesV2_t, 601
cust_id, 602
cust_value, 602
Tlvresult, 602
value_length, 602
pack_dms_SetEventReport
dms.h, 1455
pack_dms_SetEventReport_t, 602
mode, 602
pack_dms_SetFirmwarePreference
dms.h, 1455
pack_dms_SetPower
dms.h, 1455
pack_dms_SetPower_t, 603
mode, 603
Tlvresult, 603

pack_dms_SetUSBComp
dms.h, 1456
pack_dms_SetUSBComp_t, 603
Tlvresult, 604
USBComp, 604
pack_dms_UIMChangePIN
dms.h, 1462
pack_dms_UIMChangePIN_t, 607
id, 607
newValue, 607
oldValue, 607
pack_dms_UIMGetControlKeyStatus
dms.h, 1462
pack_dms_UIMGetControlKeyStatus_t, 608
facility, 608
pack_dms_UIMGetICCID
dms.h, 1463
pack_dms_UIMGetICCID_t, 608
Tlvresult, 608
pack_dms_UIMGetPINStatus
dms.h, 1463
pack_dms_UIMSetControlKeyProtection
dms.h, 1464
pack_dms_UIMSetControlKeyProtection_t, 608
facility, 609
facilityCk, 609
facilityState, 609
pack_dms_UIMSetPINProtection
dms.h, 1464
pack_dms_UIMSetPINProtection_t, 609
bEnable, 610
id, 610
value, 610
pack_dms_UIMUnblockControlKey
dms.h, 1464
pack_dms_UIMUnblockControlKey_t, 610
facility, 610
facilityCk, 610
pack_dms_UIMUnblockPIN
dms.h, 1465
pack_dms_UIMUnblockPIN_t, 610
id, 611
newPin, 611
pukValue, 611
pack_dms_UIMVerifyPIN
dms.h, 1465
pack_dms_UIMVerifyPIN_t, 611
id, 611
value, 611
pack_dms_ValidateSPC
dms.h, 1465
pack_fms_GetImagesPreference
fms.h, 1490
pack_fms_GetImagesPreference_t, 612
Tlvresult, 612
pack_fms_GetStoredImages
fms.h, 1490
pack_fms_GetStoredImages_t, 612
Tlvresult, 612
pack_fms_SetImagesPreference
fms.h, 1490
pack_fms_SetImagesPreference_t, 612
bForceDownload, 613
imageListSize, 613
modemindex, 613
pImageList, 613
Tlvresult, 613
pack_loc_Delete_Assist_Data_t, 613
pBdsSVInfo, 614
pCellDb, 614
pClkInfo, 614
pGnssData, 614
pSVInfo, 614
Tlvresult, 614
pack_loc_DeleteAssistData
loc.h, 1497
pack_loc_EventRegister
loc.h, 1497
pack_loc_EventRegister_t, 614
eventRegister, 616
Tlvresult, 616
pack_loc_SLQSLOCGetBestAvailPos
loc.h, 1498
pack_loc_SLQSLOCGetBestAvailPos_t, 617
Tlvresult, 617
xid, 617
pack_loc_SLQSLOCGetServer
loc.h, 1498
pack_loc_SLQSLOCGetServer_t, 618
has_serverAddrTypeMask, 618
serverAddrTypeMask, 618
serverType, 618
pack_loc_SLQSLOCInjectPosition
loc.h, 1499
pack_loc_SLQSLOCInjectPosition_t, 618
altitudeSrcInfo, 622
altitudeWrtEllipsoid, 622
altitudeWrtMeanSeaLevel, 622
has_altitudeSrcInfo, 622
has_altitudeWrtEllipsoid, 622
has_horConfidence, 622
has_horReliability, 622
has_horUncCircular, 622
has_latitude, 622
has_longitude, 622
has_positionSrc, 622
has_rawHorConfidence, 622
has_rawHorUncCircular, 622
has_timestampAge, 622
has_timestampUtc, 622
has_vertConfidence, 622
has_vertRelicability, 622
has_vertUnc, 623
horConfidence, 623
horReliability, 623
horUncCircular, 623

latitude, 623
longitude, 623
positionSrc, 623
rawHorConfidence, 623
rawHorUncCircular, 623
timestampAge, 623
timestampUtc, 623
vertConfidence, 623
vertReliability, 623
vertUnc, 623
pack_loc_SLQSLOCInjectSensorData
loc.h, 1499
pack_loc_SLQSLOCInjectSensorData_t, 623
accelTemp, 624
acceleroData, 624
acceleroTimeSrc, 624
gyroData, 624
gyroTemp, 624
gyroTimeSrc, 625
has_accelTemp, 625
has_acceleroTimeSrc, 625
has_accleroData, 625
has_gyroData, 625
has_gyroTemp, 625
has_gyroTimeSrc, 625
has_opaqueId, 625
opaqueId, 625
pack_loc_SLQSLOCInjectUTCTime
loc.h, 1500
pack_loc_SLQSLOCInjectUTCTime_t, 625
timeMsec, 625
timeUncMsec, 625
pack_loc_SLQSLOCSetCradleMountConfig
loc.h, 1500
pack_loc_SLQSLOCSetCradleMountConfig_t, 625
confidence, 626
has_confidence, 626
state, 626
pack_loc_SetExtPowerState
loc.h, 1497
pack_loc_SetExtPowerState_t, 616
extPowerState, 616
Tlvresult, 616
pack_loc_SetOperationMode
loc.h, 1498
pack_loc_SetOperationMode_t, 616
mode, 617
Tlvresult, 617
pack_loc_Start
loc.h, 1500
pack_loc_Start_t, 626
pApplicationInfo, 627
pConfigAltitudeAssumed, 627
pHorizontalAccuracyLvl, 628
pIntermediateReportState, 628
pMinIntervalTime, 628
pRecurrenceType, 628
SessionId, 628
Tlvresult, 628
pack_loc_Stop
loc.h, 1501
pack_loc_Stop_t, 628
SessionId, 628
Tlvresult, 628
pack_nas_GetACCOLC
nas.h, 1517
pack_nas_GetANAAAAuthenticationStatus
nas.h, 1518
pack_nas_GetCDMANetworkParameters
nas.h, 1518
pack_nas_GetHomeNetwork
nas.h, 1518
pack_nas_GetHomeNetwork3GPP2
nas.h, 1519
pack_nas_GetNetworkPreference
nas.h, 1519
pack_nas_GetRFInfo
nas.h, 1519
pack_nas_GetServingNetwork
nas.h, 1519
pack_nas_GetServingNetworkCapabilities
nas.h, 1520
pack_nas_GetSignalStrengths
nas.h, 1520
pack_nas_InitiateDomainAttach
nas.h, 1520
pack_nas_InitiateDomainAttach_t, 628
action, 629
pack_nas_PerformNetworkScan
nas.h, 1521
pack_nas_SLQSConfigSigInfo
nas.h, 1523
pack_nas_SLQSConfigSigInfo_t, 632
pECIOThresh, 633
pIOThresh, 633
pLTESNRThresh, 633
pLTESigRptCfg, 633
pRSRPThresh, 633
pRSRQThresh, 633
pRSSIThresh, 633
pack_nas_SLQSGetErrorRate
nas.h, 1523
pack_nas_SLQSGetNetworkTime
nas.h, 1524
pack_nas_SLQSGetOperatorNameData
nas.h, 1524
pack_nas_SLQSGetPLMNName
nas.h, 1524
pack_nas_SLQSGetPLMNName_t, 633
mcc, 634
mnc, 634
pMncPcsStatus, 634
pack_nas_SLQSGetServingSystem
nas.h, 1525
pack_nas_SLQSGetSignalStrength
nas.h, 1525

pack_nas_SLQSGetSysInfo
nas.h, 1525
pack_nas_SLQSGetSysSelectionPref
nas.h, 1526
pack_nas_SLQSInitiateNetworkRegistration
nas.h, 1526
pack_nas_SLQSInitiateNetworkRegistration_t, 634
pChangeDuration, 635
pMNRInfo, 635
pMncPcsDigitStatus, 635
regAction, 635
pack_nas_SLQSNASSwiGetChannelLock
nas.h, 1529
pack_nas_SLQSNASSwiSetChannelLock
nas.h, 1530
pack_nas_SLQSNASSwiSetChannelLock_t, 645
pack_nas_SLQSNasConfigSigInfo2
nas.h, 1526
pack_nas_SLQSNasConfigSigInfo2_t, 635
pHDRIODelta, 639
pHDRIOThresh, 639
pLTESigRptConfig, 639
pack_nas_SLQSNasGet3GPP2Subscription
nas.h, 1527
pack_nas_SLQSNasGet3GPP2Subscription_t, 640
namID, 640
pack_nas_SLQSNasGetCellLocationInfo
nas.h, 1527
pack_nas_SLQSNasGetHDRColorCode
nas.h, 1528
pack_nas_SLQSNasGetSigInfo
nas.h, 1528
pack_nas_SLQSNasGetTxRxInfo
nas.h, 1528
pack_nas_SLQSNasGetTxRxInfo_t, 640
radio_if, 640
pack_nas_SLQSNasIndicationRegisterExt
nas.h, 1529
pack_nas_SLQSNasIndicationRegisterExt_t, 640
pDDTMInd, 643
pDualStandByPrefInd, 643
pErrorRateInd, 643
pHDRSessionCloseInd, 643
pLTECphyCa, 643
pManagedRoamingInd, 643
pNetworkRejectInd, 643
pNetworkTimeInd, 643
pServingSystemInd, 643
pSignalStrengthInd, 643
pSubscriptionInfoInd, 643
pSuppressSysInfoInd, 643
pSysInfoInd, 643
pSystemSelectionInd, 643
pack_nas_SLQSNasSwiIndicationRegister
nas.h, 1529
pack_nas_SLQSNasSwiIndicationRegister_t, 643
gsmUmtsDl, 644
gsmUmtsUl, 644
lteEmmDl, 644
lteEmmUl, 644
lteEsmDl, 644
lteEsmUl, 644
pRankIndicatorInd, 644
pTimer, 644
pack_nas_SLQSNasSwiModemStatus
nas.h, 1530
pack_nas_SLQSSetBandPreference
nas.h, 1531
pack_nas_SLQSSetBandPreference_t, 645
bandPref, 646
pack_nas_SLQSSetSignalStrengthsCallback
nas.h, 1531
pack_nas_SLQSSetSignalStrengthsCallback_t, 647
bEnable, 647
pSigIndReq, 647
pack_nas_SLQSSetSysSelectionPref
nas.h, 1531
pack_nas_SLQSSetSysSelectionPref_t, 647
pAcqOrderPref, 652
pBandPref, 652
pCSGID, 652
pChgDuration, 652
pEmerMode, 652
pGWAcqOrderPref, 652
pLTEBandPref, 652
pModePref, 652
pNetSelPref, 652
pPRLPref, 652
pRAT, 652
pRoamPref, 652
pSrvDomainPref, 652
pSrvRegRestriction, 652
pTdscdmaBandPref, 653
pack_nas_SLQSSwiGetHDRPersonality
nas.h, 1532
pack_nas_SLQSSwiGetHDRProtSubtype
nas.h, 1532
pack_nas_SLQSSwiGetHRPDStats
nas.h, 1532
pack_nas_SLQSSwiGetLteCQI
nas.h, 1533
pack_nas_SLQSSwiGetLteSccRxInfo
nas.h, 1533
pack_nas_SLQSSwiNetworkDebug
nas.h, 1533
pack_nas_SLQSSwiPSDetach
nas.h, 1534
pack_nas_SLQSSwiPSDetach_t, 653
pDetachAction, 653
pack_nas_SetACCOLC
nas.h, 1521
pack_nas_SetACCOLC_t, 629
accolc, 629
spc, 629
pack_nas_SetCDMANetworkParameters
nas.h, 1522

pack_nas_SetCDMANetworkParameters_t, 629
pApplication, 630
pBroadcast, 630
pCustomSCP, 630
pForceRev0, 631
pProtocol, 631
pRoaming, 631
pSPC, 631
pack_nas_SetLURejectCallback
nas.h, 1522
pack_nas_SetNetworkPreference
nas.h, 1522
pack_nas_SetNetworkPreference_t, 631
Duration, 631
TechnologyPref, 631
Tlvresult, 632
pack_nas_SetRFInfoCallback
nas.h, 1523
pack_nas_SlqsGetLTECphyCAInfo
nas.h, 1524
pack_pds_ForceXTRADownload
pds.h, 1559
pack_pds_GetPDSDefaults
pds.h, 1559
pack_pds_GetPDSState
pds.h, 1560
pack_pds_GetPortAutomaticTracking
pds.h, 1560
pack_pds_GetServiceAutomaticTracking
pds.h, 1560
pack_pds_GetXTRAAutomaticDownload
pds.h, 1561
pack_pds_GetXTRANetwork
pds.h, 1561
pack_pds_GetXTRAValidity
pds.h, 1561
pack_pds_PDSInjectTimeReference
pds.h, 1562
pack_pds_PDSInjectTimeReference_t, 653
systemDiscontinuities, 653
systemTime, 653
pack_pds_ResetPDSData
pds.h, 1562
pack_pds_ResetPDSData_t, 654
pCellDataMask, 654
pGPSDataMask, 654
pack_pds_SLQSGetAGPSConfig
pds.h, 1565
pack_pds_SLQSGetAGPSConfig_t, 658
pNetworkMode, 659
pack_pds_SLQSGetGPSStateInfo
pds.h, 1565
pack_pds_SLQSPDSDeterminePosition
pds.h, 1566
pack_pds_SLQSPDSInjectAbsoluteTimeReference
pds.h, 1566
pack_pds_SLQSPDSInjectAbsoluteTimeReference_t, 659
forceFlag, 659
timeBase, 659
timeMsec, 659
timeUncMsec, 659
pack_pds_SLQSPDSInjectPositionData
pds.h, 1566
pack_pds_SLQSPDSInjectPositionData_t, 659
pAltitudeWrtEllipsoid, 661
pAltitudeWrtSealevel, 661
pHorizontalConfidence, 661
pHorizontalUncCircular, 661
pLatitude, 661
pLongitude, 661
pPositionSource, 661
pTimeStamp, 661
pTimeType, 661
pVerticalConfidence, 661
pVerticalUnc, 661
pack_pds_SLQSSetAGPSConfig
pds.h, 1567
pack_pds_SLQSSetAGPSConfig_t, 661
pNetworkMode, 662
pServerAddress, 662
pServerPort, 662
pServerURL, 662
pack_pds_SLQSSetPositionMethodState
pds.h, 1567
pack_pds_SLQSSetPositionMethodState_t, 662
pWifiState, 663
pXtraDataState, 663
pXtraTimeState, 663
pack_pds_SetEventReportCallback
pds.h, 1563
pack_pds_SetEventReportCallback_t, 654
posDataNmea, 655
rptPosData, 655
pack_pds_SetPDSDefaults
pds.h, 1563
pack_pds_SetPDSDefaults_t, 655
accuracy, 656
interval, 656
operation, 656
timeout, 656
pack_pds_SetPDSState
pds.h, 1563
pack_pds_SetPDSState_t, 656
enable, 656
pack_pds_SetPortAutomaticTracking
pds.h, 1564
pack_pds_SetPortAutomaticTracking_t, 656
bAuto, 657
pack_pds_SetServiceAutomaticTracking
pds.h, 1564
pack_pds_SetServiceAutomaticTracking_t, 657
bAuto, 657
pack_pds_SetXTRAAutomaticDownload
pds.h, 1564
pack_pds_SetXTRAAutomaticDownload_t, 657

bEnabled, 658
interval, 658
pack_pds_SetXTRANetwork
pds.h, 1565
pack_pds_SetXTRANetwork_t, 658
preference, 658
pack_pds_StartPDSTrackingSessionExt
pds.h, 1568
pack_pds_StartPDSTrackingSessionExt_t, 663
fixAccuracy, 664
fixCount, 664
fixInterval, 664
fixTimeout, 664
sessionControl, 664
sessionOperation, 664
sessionServerOption, 664
sessionType, 664
pack_pds_StopPDSTrackingSession
pds.h, 1568
pack_qmi_t, 664
msgid, 665
svc, 665
timeout, 665
xid, 665
pack_qos_SLQSQosGetNetworkStatus
qos.h, 2004
pack_qos_SLQSQosSwiReadApnExtraParams
qos.h, 2004
pack_qos_SLQSQosSwiReadApnExtraParams_t, 665
apnId, 665
pack_qos_SLQSQosSwiReadDataStats
qos.h, 2005
pack_qos_SLQSQosSwiReadDataStats_t, 665
apnId, 666
pack_qos_SLQSSetQosEventCallback
qos.h, 2006
pack_qos_SLQSSetQosEventCallback_t, 666
enable, 666
pack_rms_GetSMSWake
rms.h, 2011
pack_rms_SetSMSWake
rms.h, 2011
pack_rms_SetSMSWake_t, 666
enabled, 667
wake_mask, 667
pack_sar_SLQSGetRfSarState
sar.h, 2012
pack_sar_SLQSSetRfSarState
sar.h, 2013
pack_sar_SLQSSetRfSarState_t, 667
RfSarState, 667
pack_sms_GetSMSCAddress
sms.h, 2018
pack_sms_SLQSDeleteSMS
sms.h, 2019
pack_sms_SLQSDeleteSMS_t, 670
pMessageIndex, 670
pMessageMode, 671
pMessageTag, 671
storageType, 671
pack_sms_SLQSGetIndicationRegister
sms.h, 2020
pack_sms_SLQSGetMessageWaiting
sms.h, 2020
pack_sms_SLQSGetSMS
sms.h, 2020
pack_sms_SLQSGetSMS_t, 671
messageIndex, 671
pMessageMode, 671
storageType, 671
pack_sms_SLQSGetSMSList
sms.h, 2021
pack_sms_SLQSGetSMSList_t, 672
pMessageMode, 672
pRequestedTag, 672
storageType, 672
pack_sms_SLQSGetSmsBroadcastConfig
sms.h, 2021
pack_sms_SLQSGetSmsBroadcastConfig_t, 671
mode, 672
pack_sms_SLQSGetTransLayerInfo
sms.h, 2021
pack_sms_SLQSGetTransNWRegInfo
sms.h, 2022
pack_sms_SLQSModifySMSStatus
sms.h, 2022
pack_sms_SLQSModifySMSStatus_t, 673
messageIndex, 673
messageTag, 673
pMessageMode, 673
storageType, 673
pack_sms_SLQSSendAsyncSMS
sms.h, 2022
pack_sms_SLQSSendAsyncSMS_t, 673
pSendSmsParams, 674
pack_sms_SLQSSetIndicationRegister
sms.h, 2023
pack_sms_SLQSSetIndicationRegister_t, 674
pSetIndicationRegReq, 674
pack_sms_SLQSSetSmsBroadcastActivation
sms.h, 2023
pack_sms_SLQSSetSmsBroadcastActivation_t, 674
broadcastActivate, 675
mode, 675
pack_sms_SLQSSetSmsBroadcastConfig
sms.h, 2024
pack_sms_SLQSSetSmsBroadcastConfig_t, 675
mode, 675
pBroadcastConfig, 675
pCDMABroadcastConfig, 675
pack_sms_SLQSSetSmsStorage
sms.h, 2024
pack_sms_SLQSSetSmsStorage_t, 675
smsStorage, 676
pack_sms_SLQSSmsGetMaxStorageSize
sms.h, 2024

pack_sms_SLQSSmsGetMaxStorageSize_t, 676
pMaxStorageSizeReq, 676
pack_sms_SLQSSmsGetMessageProtocol
sms.h, 2025
pack_sms_SLQSSmsSetRoutes
sms.h, 2025
pack_sms_SLQSSmsSetRoutes_t, 676
pSetRoutesReq, 677
pack_sms_SLQSSwiGetSMSStorage
sms.h, 2025
pack_sms_SaveSMS
sms.h, 2018
pack_sms_SaveSMS_t, 667
messageFormat, 668
messageSize, 668
pMessage, 668
storageType, 668
pack_sms_SendSMS
sms.h, 2018
pack_sms_SendSMS_t, 668
messageFormat, 669
messageSize, 669
pLinktimer, 669
pMessage, 669
pack_sms_SetNewSMSCallback
sms.h, 2019
pack_sms_SetNewSMSCallback_t, 669
status, 669
pack_sms_SetSMSCAddress
sms.h, 2019
pack_sms_SetSMSCAddress_t, 669
pSMSCAddress, 670
pSMSCType, 670
pack_swiloc_SwiLocGetAutoStart
swiloc.h, 2036
pack_swiloc_SwiLocSetAutoStart
swiloc.h, 2036
pack_swiloc_SwiLocSetAutoStart_t, 677
fix_rate, 678
fix_type, 678
function, 678
max_dist, 678
max_time, 678
set_fix_rate, 678
set_fix_type, 678
set_function, 678
set_max_dist, 678
set_max_time, 678
pack_swioma_SLQSOMADMAlertCallback
swioma.h, 2038
pack_swioma_SLQSOMADMCancelSession
swioma.h, 2039
pack_swioma_SLQSOMADMCancelSession_t, 678
sessionType, 679
pack_swioma_SLQSOMADMGetSessionInfo
swioma.h, 2040
pack_swioma_SLQSOMADMGetSessionInfo_t, 679
SessionType, 679
pack_swioma_SLQSOMADMGetSettings
swioma.h, 2040
pack_swioma_SLQSOMADMSendSelection
swioma.h, 2041
pack_swioma_SLQSOMADMSendSelection_t, 679
pDeferTime, 680
pRejectReason, 680
selection, 680
pack_swioma_SLQSOMADMSetSettings
swioma.h, 2041
pack_swioma_SLQSOMADMSetSettings_t, 680
FOTAdownload, 681
pAutosdm, 681
pFwAutoCheck, 681
pack_swioma_SLQSOMADMStartSession
swioma.h, 2042
pack_swioma_SLQSOMADMStartSession_t, 681
sessionType, 681
pack_uim_ChangePin
uim.h, 2050
pack_uim_ChangePin_t, 681
changePIN, 682
EncryptedPIN1, 682
pIndicationToken, 682
pKeyReferenceID, 682
sessionInfo, 682
Tlvresult, 682
pack_uim_GetCardStatus
uim.h, 2050
pack_uim_ReadTransparent
uim.h, 2050
pack_uim_ReadTransparent_t, 682
fileIndex, 683
pEncryptData, 683
pIndicationToken, 683
readTransparent, 683
sessionInfo, 683
Tlvresult, 683
pack_uim_SLQSUIMAuthenticate
uim.h, 2051
pack_uim_SLQSUIMAuthenticate_t, 685
authData, 685
pIndicationToken, 685
sessionInfo, 685
pack_uim_SLQSUIMDepersonalization
uim.h, 2051
pack_uim_SLQSUIMDepersonalization_t, 685
depersonilisationInfo, 686
pack_uim_SLQSUIMEventRegister
uim.h, 2052
pack_uim_SLQSUIMEventRegister_t, 686
eventMask, 686
pack_uim_SLQSUIMGetConfiguration
uim.h, 2052
pack_uim_SLQSUIMGetConfiguration_t, 686
pConfigurationMask, 686
pack_uim_SLQSUIMGetFileAttributes
uim.h, 2052

pack_uim_SLQSUIMGetFileAttributes_t, 687
fileIndex, 687
pIndicationToken, 687
sessionInfo, 687
pack_uim_SLQSUIMGetSlotsStatus
uim.h, 2053
pack_uim_SLQSUIMPowerDown
uim.h, 2053
pack_uim_SLQSUIMPowerDown_t, 687
slot, 688
pack_uim_SLQSUIMPowerUp
uim.h, 2054
pack_uim_SLQSUIMPowerUp_t, 688
pIgnoreHotSwapSwitch, 688
slot, 688
pack_uim_SLQSUIMRefreshComplete
uim.h, 2054
pack_uim_SLQSUIMRefreshComplete_t, 688
refreshComplete, 689
sessionInfo, 689
pack_uim_SLQSUIMRefreshGetLastEvent
uim.h, 2054
pack_uim_SLQSUIMRefreshGetLastEvent_t, 689
sessionInfo, 689
pack_uim_SLQSUIMRefreshOK
uim.h, 2055
pack_uim_SLQSUIMRefreshOK_t, 689
OKtoRefresh, 690
sessionInfo, 690
pack_uim_SLQSUIMRefreshRegister
uim.h, 2055
pack_uim_SLQSUIMRefreshRegister_t, 690
regRefresh, 690
sessionInfo, 690
pack_uim_SLQSUIMReset
uim.h, 2055
pack_uim_SLQSUIMSwitchSlot
uim.h, 2056
pack_uim_SLQSUIMSwitchSlot_t, 690
bLogicalSlot, 691
ulPhysicalSlot, 691
pack_uim_SetPinProtection
uim.h, 2051
pack_uim_SetPinProtection_t, 684
EncryptedPIN1, 684
pIndicationToken, 684
pKeyReferenceID, 684
pinProtection, 684
sessionInfo, 684
Tlvresult, 684
pack_uim_UnblockPin
uim.h, 2056
pack_uim_UnblockPin_t, 691
EncryptedPIN1, 692
pIndicationToken, 692
pKeyReferenceID, 692
pinProtection, 692
sessionInfo, 692
Tlvresult, 692
pack_uim_VerifyPin
uim.h, 2056
pack_uim_VerifyPin_t, 692
pEncryptedPIN1, 693
pIndicationToken, 693
pKeyReferenceID, 693
sessionInfo, 693
Tlvresult, 693
verifyPIN, 693
pack_voice_AnswerUSSD
voice.h, 2070
pack_voice_AnswerUSSD_t, 693
pInfo, 694
pack_voice_CancelUSSD
voice.h, 2070
pack_voice_OriginateUSSD
voice.h, 2070
pack_voice_OriginateUSSD_t, 694
pInfo, 694
pack_voice_SLQSOriginateUSSD
voice.h, 2071
pack_voice_SLQSOriginateUSSD_t, 694
ussDCS, 695
ussData, 695
ussLen, 695
pack_voice_SLQSVoiceALSSelectLine
voice.h, 2071
pack_voice_SLQSVoiceALSSelectLine_t, 695
lineValue, 695
pack_voice_SLQSVoiceALSSetLineSwitching
voice.h, 2071
pack_voice_SLQSVoiceALSSetLineSwitching_t, 695
switchOption, 695
pack_voice_SLQSVoiceAnswerCall
voice.h, 2072
pack_voice_SLQSVoiceAnswerCall_t, 696
pCallId, 696
pack_voice_SLQSVoiceBindSubscription
voice.h, 2072
pack_voice_SLQSVoiceBindSubscription_t, 696
subsType, 696
pack_voice_SLQSVoiceBurstDTMF
voice.h, 2073
pack_voice_SLQSVoiceBurstDTMF_t, 696
pack_voice_SLQSVoiceDialCall
voice.h, 2073
pack_voice_SLQSVoiceDialCall_t, 697
callNumber, 698
pCLIRType, 698
pCUGInfo, 698
pCallPartySubAdd, 698
pCallType, 698
pEmergencyCategory, 698
pSvcType, 699
pUUSInFo, 699
pack_voice_SLQSVoiceEndCall
voice.h, 2073

pack_voice_SLQSVoiceEndCall_t, 699
pCallId, 699
pack_voice_SLQSVoiceGetAllCallInfo
voice.h, 2074
pack_voice_SLQSVoiceGetCLIP
voice.h, 2076
pack_voice_SLQSVoiceGetCLIR
voice.h, 2076
pack_voice_SLQSVoiceGetCNAP
voice.h, 2076
pack_voice_SLQSVoiceGetCOLP
voice.h, 2077
pack_voice_SLQSVoiceGetCOLR
voice.h, 2077
pack_voice_SLQSVoiceGetCallBarring
voice.h, 2074
pack_voice_SLQSVoiceGetCallBarring_t, 699
pSvcClass, 700
reason, 700
pack_voice_SLQSVoiceGetCallForwardingStatus
voice.h, 2074
pack_voice_SLQSVoiceGetCallForwardingStatus_t, 700
pSvcClass, 701
Reason, 701
pack_voice_SLQSVoiceGetCallInfo
voice.h, 2075
pack_voice_SLQSVoiceGetCallInfo_t, 701
callID, 701
pack_voice_SLQSVoiceGetCallWaiting
voice.h, 2075
pack_voice_SLQSVoiceGetCallWaiting_t, 701
pSvcClass, 702
pack_voice_SLQSVoiceGetConfig
voice.h, 2077
pack_voice_SLQSVoiceGetConfig_t, 702
pAMRStatus, 703
pAirTimer, 703
pAutoAnswer, 703
pNamID, 703
pPrefVoicePrivacy, 703
pPrefVoiceSO, 703
pRoamTimer, 703
pTTYMode, 703
pVoiceDomainPref, 703
pack_voice_SLQSVoiceIndicationRegister
voice.h, 2078
pack_voice_SLQSVoiceIndicationRegister_t, 704
pRegDTMFEvents, 704
pRegVoicePrivacyEvents, 704
pSuppsNotifEvents, 704
pack_voice_SLQSVoiceManageCalls
voice.h, 2078
pack_voice_SLQSVoiceManageCalls_t, 704
pCallID, 705
SUPSType, 705
pack_voice_SLQSVoiceOrigUSSDNoWait
voice.h, 2078
pack_voice_SLQSVoiceOrigUSSDNoWait_t, 705
pack_voice_SLQSVoiceSendFlash
voice.h, 2079
pack_voice_SLQSVoiceSendFlash_t, 706
pCallID, 706
pFlashPayLd, 706
pFlashType, 706
pack_voice_SLQSVoiceSetCallBarringPassword
voice.h, 2079
pack_voice_SLQSVoiceSetCallBarringPassword_t, 706
newPasswd, 707
newPasswdAgain, 707
oldPasswd, 707
Reason, 707
pack_voice_SLQSVoiceSetConfig
voice.h, 2079
pack_voice_SLQSVoiceSetConfig_t, 707
pAirTimerConfig, 709
pAutoAnswer, 709
pPrefVoiceDomain, 709
pPrefVoiceSO, 709
pRoamTimerConfig, 709
pTTYMode, 709
pack_voice_SLQSVoiceSetPreferredPrivacy
voice.h, 2080
pack_voice_SLQSVoiceSetPreferredPrivacy_t, 709
privacyPref, 709
pack_voice_SLQSVoiceSetSUPSService
voice.h, 2080
pack_voice_SLQSVoiceSetSUPSService_t, 709
pCallBarringPasswd, 711
pCallForwardingNumber, 711
pServiceClass, 711
pTimerVal, 711
reason, 711
voiceSvc, 711
pack_voice_SLQSVoiceStartContDTMF
voice.h, 2081
pack_voice_SLQSVoiceStartContDTMF_t, 711
DTMFdigit, 712
pCallID, 712
pack_voice_SLQSVoiceStopContDTMF
voice.h, 2081
pack_voice_SLQSVoiceStopContDTMF_t, 712
callID, 712
pack_wds_DHCPv4ClientLeaseChange
wds.h, 2099
pack_wds_DHCPv4ClientLeaseChange_t, 712
pEnableNotification, 713
pack_wds_GetAutoconnect
wds.h, 2099
pack_wds_GetByteTotals
wds.h, 2099
pack_wds_GetConnectionRate
wds.h, 2100
pack_wds_GetDataBearerTechnology
wds.h, 2100
pack_wds_GetDefaultProfile
wds.h, 2100

pack_wds_GetDefaultProfile_t, 713
profiletype, 713
pack_wds_GetDefaultProfileNum
wds.h, 2101
pack_wds_GetDefaultProfileNum_t, 713
family, 713
type, 713
pack_wds_GetDormancyState
wds.h, 2101
pack_wds_GetDormancyState_t, 714
pack_wds_GetLastMobileIPError
wds.h, 2102
pack_wds_GetLastMobileIPError_t, 714
pack_wds_GetMobileIP
wds.h, 2102
pack_wds_GetMobileIP_t, 714
pack_wds_GetMobileIPProfile
wds.h, 2103
pack_wds_GetMobileIPProfile_t, 714
index, 715
pack_wds_GetPacketStatistics
wds.h, 2103
pack_wds_GetPacketStatistics_t, 715
pStatMask, 715
pack_wds_GetPacketStatus
wds.h, 2103
pack_wds_GetPacketStatus_t, 715
statmask, 716
pack_wds_GetSessionDuration
wds.h, 2104
pack_wds_GetSessionDuration_t, 716
pack_wds_GetSessionState
wds.h, 2104
pack_wds_RMSetTransferStatistics
wds.h, 2105
pack_wds_RMSetTransferStatistics_t, 716
RmTrasnferStaticsReq, 716
pack_wds_SLQSCreateProfile
wds.h, 2108
pack_wds_SLQSCreateProfile_t, 723
pCurProfile, 724
pProfileId, 724
pProfileType, 724
pack_wds_SLQSDeleteProfile
wds.h, 2108
pack_wds_SLQSDeleteProfile_t, 724
profileIndex, 724
profileType, 724
pack_wds_SLQSGet3GPPConfigItem
wds.h, 2109
pack_wds_SLQSGetCurrDataSystemStat
wds.h, 2109
pack_wds_SLQSGetCurrDataSystemStat_t, 724
pack_wds_SLQSGetCurrentChannelRate
wds.h, 2110
pack_wds_SLQSGetDUNCallInfo
wds.h, 2111
pack_wds_SLQSGetDUNCallInfo_t, 725
Mask, 726
pReportChannelRate, 726
pReportConnStatus, 726
pReportDataBearerTech, 726
pReportDormStatus, 726
pTransferStatInd, 726
pack_wds_SLQSGetDataBearerTechnology
wds.h, 2110
pack_wds_SLQSGetDataBearerTechnology_t, 725
pack_wds_SLQSGetProfileSettings
wds.h, 2111
pack_wds_SLQSGetProfileSettings_t, 726
ProfileId, 727
ProfileType, 727
pack_wds_SLQSGetRuntimeSettings
wds.h, 2111
pack_wds_SLQSGetRuntimeSettings_t, 727
pReqSettings, 728
pack_wds_SLQSModifyProfile
wds.h, 2112
pack_wds_SLQSModifyProfile_t, 728
curProfile, 729
pProfileId, 729
pProfileType, 729
pack_wds_SLQSResetPacketStatics
wds.h, 2112
pack_wds_SLQSSGetDHCPv4ClientConfig
wds.h, 2114
pack_wds_SLQSSGetDHCPv4ClientConfig_t, 732
pProfileId, 732
pack_wds_SLQSSGetLoopback
wds.h, 2114
pack_wds_SLQSSSetDHCPv4ClientConfig
wds.h, 2115
pack_wds_SLQSSSetDHCPv4ClientConfig_t, 732
pHwConfig, 733
pProfileId, 733
pack_wds_SLQSSSetLoopback
wds.h, 2115
pack_wds_SLQSSSetLoopback_t, 733
loopbackMode, 733
loopbackMultiplier, 733
pack_wds_SLQSSet3GPPConfigItem
wds.h, 2113
pack_wds_SLQSSet3GPPConfigItem_t, 729
p3gppRelease, 730
pProfileList, 730
pack_wds_SLQSSetIPFamilyPreference
wds.h, 2113
pack_wds_SLQSSetIPFamilyPreference_t, 731
IPFamilyPreference, 731
pack_wds_SLQSSetWdsEventCallback
wds.h, 2113
pack_wds_SLQSSetWdsEventCallback_t, 731
currentDataBearer, 732
dataBearer, 732
dataBearerTechExt, 732
dataSystemStatus, 732

dormancyStatus, 732
interval, 732
mobileIP, 732
transferStats, 732
pack_wds_SLQSStartDataSession
wds.h, 2115
pack_wds_SLQSStartDataSession_t, 733
pAuth, 734
pPass, 734
pTech, 734
pUser, 734
pprofileid3gpp, 734
pprofileid3gpp2, 734
pack_wds_SLQSStopDataSession
wds.h, 2116
pack_wds_SLQSStopDataSession_t, 735
psid, 735
pack_wds_SLQSWdsGoActive
wds.h, 2116
pack_wds_SLQSWdsGoDormant
wds.h, 2117
pack_wds_SLQSWdsSetEventReport
wds.h, 2117
pack_wds_SLQSWdsSetEventReport_t, 735
pCurrChannelRateInd, 736
pCurrDataBearerTechInd, 736
pCurrPrefDataSysInd, 736
pDataBearerTechInd, 736
pDataCallStatusChangeInd, 736
pDataSystemStatusChangeInd, 736
pDormancyStatusInd, 737
pMIPStatusInd, 737
pTransferStatInd, 737
pack_wds_SLQSWdsSwiPDPRuntimeSettings
wds.h, 2118
pack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 737
contextId, 737
contextType, 737
pack_wds_SetAutoconnect
wds.h, 2105
pack_wds_SetAutoconnect_t, 717
acroamsetting, 717
acsetting, 717
pack_wds_SetDefaultProfile
wds.h, 2105
pack_wds_SetDefaultProfile_t, 717
authentication, 718
ipAddress, 719
pApnname, 719
pName, 719
pPassword, 719
pUsername, 719
pdpType, 719
primaryDNS, 719
profileType, 719
secondaryDNS, 719
pack_wds_SetDefaultProfileNum
wds.h, 2106
pack_wds_SetDefaultProfileNum_t, 719
family, 719
index, 719
type, 719
pack_wds_SetMobileIP
wds.h, 2106
pack_wds_SetMobileIP_t, 720
mode, 720
pack_wds_SetMobileIPParameters
wds.h, 2107
pack_wds_SetMobileIPParameters_t, 720
pHA2002bis, 721
pHAAuthenticator, 721
pMode, 721
pReRegPeriod, 721
pReRegTraffic, 721
pRetryInterval, 721
pRetryLimit, 721
pSPC, 721
pack_wds_SetMobileIPProfile
wds.h, 2107
pack_wds_SetMobileIPProfile_t, 721
index, 722
pAAASPI, 723
pAddress, 723
pEnabled, 723
pHASPI, 723
pMNAAA, 723
pMNHA, 723
pNAI, 723
pPrimaryHA, 723
pRevTunneling, 723
pSecondaryHA, 723
spc, 723
pack_wds_SetMuxID
wds.h, 2108
PackCreateProfileOut, 737
ExtErrorCode, 738
ProfileIndex, 738
ProfileType, 738
package_name
omaDmFotaTlv, 592
omaDmFotaTlvExt, 594
unpack_omaDmFotaTlv_t, 1160
PackageID
_SwiAvmsEventInd, 100
packageSize
omaDmFotaTlvExt, 594
packageid_str
slqsfwinfo_s, 884
unpack_dms_GetFirmwareInfo_t, 1046
packetSrvStatus
qaGobiApiCbk.h, 1598
packetZone
CDMASysInfo, 177
nas_CDMASysInfo, 430
packetZoneValid
CDMASysInfo, 177

nas_CDMASysInfo, 430
packgetDyingGaspCfg, 738
pDestSMSContent, 738
pDestSMSNum, 738
packgetDyingGaspStatistics, 738
pSMSAttemptedFlag, 739
pTimeStamp, 739
path
fileInfo, 247
uim_fileInfo, 989
pathLen
fileInfo, 247
uim_fileInfo, 989
pbAuto
unpack_pds_GetPortAutomaticTracking_t, 1163
unpack_pds_GetServiceAutomaticTracking_t, 1164
pbEnabled
unpack_pds_GetXTRAAutomaticDownload_t, 1164
pbIMSRegistered
imsaRegStatusInfo, 318
pbPlatform
unpack_dms_GetOfflineReason_t, 1052
pci
cellParams, 179
ltePCI, 403
nas_cellParams, 431
nas_ltePCI, 467
nas_PhyCaAggPcellInfo, 487
nas_PhyCaAggScellIndType, 488
nas_PhyCaAggScellInfo, 491
nas_umtsLTENbrCell, 527
NASPhyCaAggPcellInfo, 558
NASPhyCaAggScellIndType, 560
NASPhyCaAggScellInfo, 561
PhyCaAggPcellInfo, 747
PhyCaAggScellIndType, 749
PhyCaAggScellInfo, 752
umtsLTENbrCell, 1025
pcsDigit
PlmnID, 754
pcsfFQDNAddress
PCSCFFQDNAddressList, 740
wds_PCSCFFQDNAddressList, 1406
pdpType
pack_wds_SetDefaultProfile_t, 719
pdptype
unpack_wds_GetDefaultProfile_t, 1258
pds.h, 1557
pack_pds_ForceXTRADownload, 1559
pack_pds_GetPDSDefaults, 1559
pack_pds_GetPDSState, 1560
pack_pds_GetPortAutomaticTracking, 1560
pack_pds_GetServiceAutomaticTracking, 1560
pack_pds_GetXTRAAutomaticDownload, 1561
pack_pds_GetXTRANetwork, 1561
pack_pds_GetXTRAValidity, 1561
pack_pds_PDSInjectTimeReference, 1562
pack_pds_ResetPDSData, 1562
pack_pds_SLQSGetAGPSConfig, 1565
pack_pds_SLQSGetGPSStateInfo, 1565
pack_pds_SLQSPDSDeterminePosition, 1566
pack_pds_SLQSPDSInjectAbsoluteTimeReference, 1566
pack_pds_SLQSPDSInjectPositionData, 1566
pack_pds_SLQSSetAGPSConfig, 1567
pack_pds_SLQSSetPositionMethodState, 1567
pack_pds_SetEventReportCallback, 1563
pack_pds_SetPDSDefaults, 1563
pack_pds_SetPDSState, 1563
pack_pds_SetPortAutomaticTracking, 1564
pack_pds_SetServiceAutomaticTracking, 1564
pack_pds_SetXTRAAutomaticDownload, 1564
pack_pds_SetXTRANetwork, 1565
pack_pds_StartPDSTrackingSessionExt, 1568
pack_pds_StopPDSTrackingSession, 1568
unpack_pds_ForceXTRADownload, 1568
unpack_pds_GetPDSDefaults, 1569
unpack_pds_GetPDSState, 1569
unpack_pds_GetPortAutomaticTracking, 1569
unpack_pds_GetServiceAutomaticTracking, 1570
unpack_pds_GetXTRAAutomaticDownload, 1570
unpack_pds_GetXTRANetwork, 1570
unpack_pds_GetXTRAValidity, 1571
unpack_pds_PDSInjectTimeReference, 1571
unpack_pds_ResetPDSData, 1571
unpack_pds_SLQSGetAGPSConfig, 1575
unpack_pds_SLQSGetGPSStateInfo, 1575
unpack_pds_SLQSPDSDeterminePosition, 1575
unpack_pds_SLQSPDSInjectAbsoluteTime-Reference, 1576
unpack_pds_SLQSPDSInjectPositionData, 1576
unpack_pds_SLQSSetAGPSConfig, 1576
unpack_pds_SLQSSetPositionMethodState, 1577
unpack_pds_SetEventReport_Ind, 1572
unpack_pds_SetEventReportCallback, 1572
unpack_pds_SetPDSDefaults, 1572
unpack_pds_SetPDSState, 1573
unpack_pds_SetPdsState_Ind, 1573
unpack_pds_SetPortAutomaticTracking, 1573
unpack_pds_SetServiceAutomaticTracking, 1574
unpack_pds_SetXTRAAutomaticDownload, 1574
unpack_pds_SetXTRANetwork, 1574
unpack_pds_StartPDSTrackingSessionExt, 1577
unpack_pds_StopPDSTrackingSession, 1577
peakRate
tokenBucket, 969
unpack_qos_tokenBucket_t, 1192
peakThroughputClass
GPRSQoS, 288
GPRSRequestedQoS, 289
LibPackGPRSRequestedQoS, 336
wds_GPRSQoS, 1403
peerNumberInfo, 744
callID, 745

numLen, 745
numPI, 745
numPlan, 745
numSI, 745
numType, 746
number, 745
PerformNetworkScan
qaGobiApiNas.h, 1775
PersistentTechPref
unpack_nas_GetNetworkPreference_t, 1111
persoFeature
appStats, 124
appStatus, 127
uim_appStatus, 979
persoRetries
appStats, 124
appStatus, 127
uim_appStatus, 979
persoState
appStats, 124
appStatus, 127
uim_appStatus, 979
persoUnblockRetries
appStats, 124
appStatus, 127
uim_appStatus, 979
personalizationStatus, 746
feature, 746
numFeatures, 746
unblockLeft, 746
verifyLeft, 746
pfailureCause
unpack_voice_SLQSOriginateUSSD_t, 1225
USSResp, 1294
phase
nas_rxInfo, 505
rxInfo, 831
PhyCaAggPcellInfo, 747
dl_bw_value, 747
freq, 747
iLTEbandValue, 747
NasGetLTECphyCaInfo, 540
pci, 747
TlvPresent, 747
PhyCaAggScellDlBw, 747
dl_bw_value, 748
NasGetLTECphyCaInfo, 540
TlvPresent, 748
PhyCaAggScellIndType, 748
freq, 749
NasGetLTECphyCaInfo, 540
pci, 749
scell_state, 749
TlvPresent, 749
PhyCaAggScellIndex, 748
NasGetLTECphyCaInfo, 540
scell_idx, 748
TlvPresent, 748
PhyCaAggScellInfo, 749
dl_bw_value, 752
freq, 752
iLTEbandValue, 752
NasGetLTECphyCaInfo, 540
pci, 752
scell_state, 752
TlvPresent, 752
PhysicalLayer
nas_protocolSubtypeElement, 496
protocolSubtypeElement, 773
PilotEnergy
nas_NetworkStatEVDO, 483
NetworkStatEVDO, 583
PilotPN
nas_PilotSetParams, 492
PilotSetParams, 753
PilotSetData, 752
NumPilots, 752
pPilotSetInfo, 752
PilotSetParams, 753
PilotPN, 753
PilotStrength, 753
PilotType, 753
PilotStrength
nas_PilotSetParams, 492
PilotSetParams, 753
PilotType
nas_PilotSetParams, 492
PilotSetParams, 753
pin1Len
encryptedPIN1, 240
uim_encryptedPIN1, 985
pin1Retries
appStats, 124
appStatus, 127
uim_appStatus, 979
pin1State
appStats, 124
appStatus, 127
uim_appStatus, 979
pin1Val
encryptedPIN1, 240
uim_encryptedPIN1, 985
pin2Retries
appStats, 124
appStatus, 127
uim_appStatus, 979
pin2State
appStats, 124
appStatus, 127
uim_appStatus, 979
pinID
changeUIMPIN, 180
setPINProtection, 866
uim_changeUIMPIN, 983
uim_setPINProtection, 996
uim_unblockUIMPIN, 999

uim_verifyUIMPIN, 1000
unblockUIMPIN, 1032
verifyUIMPIN, 1296
pinLen
changeUIMPIN, 180
uim_changeUIMPIN, 983
uim_verifyUIMPIN, 1000
verifyUIMPIN, 1296
pinLength
setPINProtection, 866
uim_setPINProtection, 996
pinOperation
setPINProtection, 866
uim_setPINProtection, 996
pinProtection
pack_uim_SetPinProtection_t, 684
pack_uim_UnblockPin_t, 692
UIMSetPinProtectionReq, 1017
pinVal
changeUIMPIN, 180
uim_changeUIMPIN, 983
uim_verifyUIMPIN, 1000
verifyUIMPIN, 1296
pinValue
setPINProtection, 866
uim_setPINProtection, 996
PkQmiNasGetRFBandInfo
qaNasGetRFBandInfo.h, 1994
PkQmiNasPerformNetworkScan
qaNasPerformNetworkScan.h, 1995
PkgDescLength
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1209
PkgDescription
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1209
PkgName
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1209
PkgNameLength
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1209
pkgver
CurrentImgList, 201
unpack_dms_SLQSSwiGetFirmwareCurr_t, 1071
PktErrRate
unpack_qos_swiQosFlow_t, 1191
pktErrRate, 753
exponent, 753
multiplier, 753
PktStatElmntsV4
WdsPktStatisticsResp, 1425
PktStatElmntsV6
WdsPktStatisticsResp, 1425
plasmaIDString
pack_dms_SLQSSwiSetHostDevInfo_t, 606
unpack_dms_SLQSSwiGetHostDevInfo_t, 1074
plmn
GERANInfo, 255
LTEInfoIntrafreq, 400
nas_GERANInfo, 445
nas_LTEInfoIntrafreq, 465
nas_UMTSInfo, 525
UMTSInfo, 1023
PlmnID, 754
mcc, 754
mnc, 754
pcsDigit, 754
TlvPresent, 754
polarityIncluded
lineCtrlInfo, 366
Port, 757
port, 757
range, 757
port
IPv4Info, 332
IPv6Info, 335
loc_IPv4Info, 372
loc_IPv6Info, 372
Port, 757
unpack_qos_Port_t, 1176
posDataNmea
pack_pds_SetEventReportCallback_t, 655
posSrc
unpack_pds_SetEventReport_Ind_t, 1167
Position Determination Service (PDS), 43
PositionDataNMEA
unpack_pds_SetEventReport_Ind_t, 1167
positionSrc
pack_loc_SLQSLOCInjectPosition_t, 623
pprofileid3gpp
pack_wds_SLQSStartDataSession_t, 734
pprofileid3gpp2
pack_wds_SLQSStartDataSession_t, 734
prDNSIPv4Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 1287
prDNSIPv6Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 1288
prPCSCFIPv4Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 1288
prPCSCFIPv6Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 1288
Precedence
unpack_qos_swiQosFilter_t, 1187
precedenceClass
GPRSQoS, 288
GPRSRequestedQoS, 289
LibPackGPRSRequestedQoS, 336
wds_GPRSQoS, 1403
precisionDilution
qaGobiApiCbk.h, 1600
precisionDilution_s, 757

HDOP, 758
PDOP, 758
VDOP, 758
PrefImageList, 758
listEntries, 758
listSize, 758
prefNetwork
unpack_wds_SLQSGetCurrDataSystemStat_t, 1271
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1283
prefVoiceSO, 758
evrcCapability, 760
homeOrigVoiceSO, 760
homePageVoiceSO, 760
namID, 760
roamOrigVoiceSO, 760
preference
pack_pds_SetXTRANetwork_t, 658
Preferred
nas_QmiNas3GppNetworkInfo, 498
SlqsNas3GppNetworkInfo, 885
prefixLen
IPv6Addr, 332
unpack_qos_IPv6Addr_t, 1175
presentationInd
ECTNum, 239
remotePartyNum, 822
voice_remotePartyNum, 1330
priChA
CDMAChannel, 162
nas_CDMAChannel, 424
priChB
CDMAChannel, 162
nas_CDMAChannel, 424
priSize
unpack_dms_GetFirmwareRevisions_t, 1048
pridns
unpack_wds_GetDefaultProfile_t, 1258
pridnsv6
unpack_wds_GetDefaultProfile_t, 1258
primaryDNS
pack_wds_SetDefaultProfile_t, 719
PrimaryDNSV4
unpack_wds_SLQSGetRuntimeSettings_t, 1279
PrimaryDNSV6
unpack_wds_SLQSGetRuntimeSettings_t, 1279
primaryHA
unpack_wds_GetMobileIPProfile_t, 1262
privacyPref
pack_voice_SLQSVoiceSetPreferredPrivacy_t, 709
voiceSetPrefPrivacy, 1376
priver
CurrentImgList, 201
unpack_dms_SLQSSwiGetFirmwareCurr_t, 1071
priversion_str
slqsfwinfo_s, 884
unpack_dms_GetFirmwareInfo_t, 1046
Profile
GetAudioPathConfigReq, 258
GetAudioProfileResp, 261
GetAudioVolTLBConfigReq, 262
GetM2MAudioProfileResp, 278
GetM2MAudioVolumeReq, 279
GetM2MAVMuteReq, 279
GetM2MSpkrGainReq, 281
SetAudioPathConfigReq, 852
SetAudioProfileReq, 853
SetAudioVolTLBConfigReq, 854
SetM2MAudioAVCFGReq, 861
SetM2MAudioProfileReq, 863
SetM2MAudioVolumeReq, 864
SetM2MAVMuteReq, 865
SetM2MSpkrGainReq, 865
Profile3GPP, 760
pAPNClass, 765
pAPNDisabledFlag, 765
pAPNName, 765
pAPNnameSize, 765
pAddrAllocPref, 765
pAuthenticationPref, 765
pGPRSMinimumQoS, 765
pGPRSRequestedQos, 765
pIPv4AddrPref, 765
pIPv6AddPref, 765
pImCnFlag, 765
pPDNInactivTimeout, 765
pPDPtype, 765
pPassword, 765
pPasswordSize, 765
pPcscfAddrUsingDhcp, 765
pPcscfAddrUsingPCO, 765
pPdpAccessConFlag, 765
pPdpContext, 765
pPdpDataCompType, 765
pPdpHdrCompType, 765
pPriDNSIPv4AddPref, 765
pPriDNSIPv6addpref, 765
pPrimaryID, 765
pProfilename, 766
pProfilenameSize, 766
pQosClassID, 766
pSecDNSIPv4AddPref, 766
pSecDNSIPv6addpref, 766
pSecondaryFlag, 766
pTFTID1Params, 766
pTFTID2Params, 766
pUMTSMinQoS, 766
pUMTSMinQosSigInd, 766
pUMTSReqQoS, 766
pUMTSReqQoSSigInd, 766
pUsername, 766
pUsernameSize, 766
Profile3GPP2, 766
pAPNClass3GPP2, 770

pAPNEnabled3GPP2, 770
pAllowLinger, 770
pApnString, 770
pApnStringSize, 770
pAppPriority, 770
pAppType, 770
pAuthPassword, 770
pAuthPasswordSize, 770
pAuthProtocol, 770
pAuthRetryCount, 771
pAuthTimeout, 771
pDataMode, 771
pDataRate, 771
pIpcpAckTimeout, 771
pIpcpCreqRetryCount, 771
pIsPcscfAddressNedded, 771
pLcpAckTimeout, 771
pLcpCreqRetryCount, 771
pNegoDnsSrvrPref, 771
pPDNInactivTimeout3GPP2, 771
pPdnType, 771
pPppSessCloseTimer1x, 771
pPppSessCloseTimerDO, 771
pPriV6DnsAddress, 771
pPrimaryV4DnsAddress, 771
pRATType, 771
pSecV6DnsAddress, 771
pSecondaryV4DnsAddress, 771
pUserId, 771
pUserIdSize, 771
ProfileID
_GetProfileSettingIn, 64
unpack_wds_SLQSGetRuntimeSettings_t, 1279
ProfileId
pack_wds_SLQSGetProfileSettings_t, 727
profileId
wds_DHCPProfileIdTlv, 1399
wds_DHCPv4ProfileId, 1401
WdsDHCPv4ProfileId, 1420
wdsDhcpv4ProfileId, 1421
ProfileId3GPP2
unpack_qos_swiQosFlow_t, 1191
ProfileIdTlv
unpack_wds_DHCPv4ClientLease_ind_t, 1253
ProfileIdentifier, 771
profileIndex, 772
profileType, 772
ProfileIndex
PackCreateProfileOut, 738
profileIndex
pack_wds_SLQSDeleteProfile_t, 724
ProfileIdentifier, 772
SLQSDeleteProfileParams, 883
wds_ProfileIdentifier, 1407
profileList
unpack_wds_SLQSGet3GPPConfigItem_t, 1270
ProfileName
unpack_wds_SLQSGetRuntimeSettings_t, 1279
ProfileType
_GetProfileSettingIn, 64
pack_wds_SLQSGetProfileSettings_t, 727
PackCreateProfileOut, 738
unpack_wds_SLQSGetProfileSettings_t, 1276
profileType
pack_wds_SetDefaultProfile_t, 719
pack_wds_SLQSDeleteProfile_t, 724
ProfileIdentifier, 772
SLQSDeleteProfileParams, 883
wds_DHCPProfileIdTlv, 1399
wds_DHCPv4ProfileId, 1402
wds_ProfileIdentifier, 1407
WdsDHCPv4ProfileId, 1420
wdsDhcpv4ProfileId, 1421
profiletype
pack_wds_GetDefaultProfile_t, 713
Protocol
unpack_nas_GetCDMANetworkParameters_t, 1106
protocolSubtypeElement, 772
AccessMac, 773
AuthProt, 773
ControlMac, 773
EncryptProt, 773
ForwardMac, 773
IdleState, 773
KeyExchange, 773
MultDisc, 773
PhysicalLayer, 773
ReverseMac, 773
SecProt, 773
VirtStream, 773
ProvisionStatus
CLIPResp, 182
CLIRResp, 183
CNAPResp, 185
COLPResp, 186
COLRResp, 186
voice_CLIPResp, 1315
voice_CLIRResp, 1316
voice_CNAPResp, 1316
voice_COLPResp, 1317
voice_COLRResp, 1318
psAttachState
nas_servSystem, 510
NASServingSystemInfo, 569
ServingSystemInfo, 846
servSystem, 848
psBarStatus
CallBarringSysInfo, 143
callBarStatus, 144
nas_CallBarringSysInfo, 422
nas_callBarStatus, 423
psState
CommInfo, 188
nas_CommInfo, 433
psc

nas_UMTSInfo, 525
nas_wcdmaCellInfo, 529
nas_WCDMASysInfo, 534
UMTSInfo, 1024
wcdmaCellInfo, 1384
WCDMASysInfo, 1394
pscValid
nas_WCDMASysInfo, 534
WCDMASysInfo, 1394
pscsfIPv4Addr
PCSCFIPv4ServerAddressList, 741
wds_PCSCFIPv4ServerAddressList, 1407
psetting
unpack_wds_GetAutoconnect_t, 1254
psid
pack_wds_SLQSStopDataSession_t, 735
unpack_wds_SLQSStartDataSession_t, 1285
puk1Retries
appStats, 124
appStatus, 127
uim_appStatus, 979
puk2Retries
appStats, 124
appStatus, 127
uim_appStatus, 979
pukLen
uim_unblockUIMPIN, 999
unblockUIMPIN, 1032
pukVal
uim_unblockUIMPIN, 1000
unblockUIMPIN, 1032
pukValue
pack_dms_UIMUnblockPIN_t, 611
pv4sessionId
WdsIpAddressInfoReq, 1422
pv6sessionId
WdsIpAddressInfoReq, 1422
pwrDenialTime
lineCtrlInfo, 366

QMI_LITE_WDS_CURRENT_CALL_DB_MASK
wds.h, 2098
QMI_LITE_WDS_LAST_CALL_DB_MASK
wds.h, 2098
QMI_SAR_RF_STATE_1
qaGobiApiSar.h, 1821
QMI_SAR_RF_STATE_2
qaGobiApiSar.h, 1821
QMI_SAR_RF_STATE_3
qaGobiApiSar.h, 1821
QMI_SAR_RF_STATE_4
qaGobiApiSar.h, 1821
QMI_SAR_RF_STATE_5
qaGobiApiSar.h, 1821
QMI_SAR_RF_STATE_6
qaGobiApiSar.h, 1821
QMI_SAR_RF_STATE_7
qaGobiApiSar.h, 1821
QMI_SAR_RF_STATE_8
qaGobiApiSar.h, 1821
QMI_SAR_RF_STATE_DEFAULT
qaGobiApiSar.h, 1821
QMI_WDS_CURRENT_CALL_DB_MASK
qaGobiApiWds.h, 1957
QMI_WDS_LAST_CALL_DB_MASK
qaGobiApiWds.h, 1957
QCI
LibPackQosClassID, 360
QosClassID, 812
QCWWAN2kConnect
qaGobiApiDcs.h, 1674
QCWWAN2kEnumerateDevices
qaGobiApiDcs.h, 1675
QCWWAN2kGetConnectedDeviceID
qaGobiApiDcs.h, 1675
QCWWANConnect
qaGobiApiDcs.h, 1676
QCWWANDisconnect
qaGobiApiDcs.h, 1676
QCWWANEnumerateDevices
qaGobiApiDcs.h, 1677
QFlowState
unpack_qos_QosFlowInfo_t, 1177
QLIC
DeviceConfigDetail, 225
nas_DeviceConfigDetail, 440
qaCbkCatEventReportInd.h
eTLV_CBK_ALPHA_IDENTIFIER, 1579
eTLV_CBK_DISPLAY_TEXT, 1579
eTLV_CBK_END_PROACTIVE_SESSION, 1579
eTLV_CBK_GET_IN_KEY, 1579
eTLV_CBK_GET_INPUT, 1579
eTLV_CBK_LANGUAGE_NOTIFICATION, 1579
eTLV_CBK_REFRESH, 1579
eTLV_CBK_SELECT_ITEM, 1579
eTLV_CBK_SETUP_EVENT_LIST, 1579
eTLV_CBK_SETUP_IDLE_MODE_TEXT, 1579
eTLV_CBK_SETUP_MENU, 1579
eTLV_END_PROACTIVE_SESSION_LENGTH, 1579
eTLV_REFRESH_LENGTH, 1579
eTLV_SETUP_EVENT_LIST_LENGTH, 1579
qaCbkSwiOmaDmEventReportInd.h
eTLV_IND_OMA_DM_CONFIG, 1580
eTLV_IND_OMA_DM_FOTA, 1580
eTLV_IND_OMA_DM_NOT, 1580
qaGobiApiCbk.h
DEVICE_STATE_BOOT, 1635
DEVICE_STATE_DISCONNECTED, 1635
DEVICE_STATE_READY, 1635
DEVICE_STATE_SERIAL_ONLY, 1635
eQA_QMI_SVC_NA, 1635
eQA_QMI_SVC_NAS, 1635
eQA_QMI_SVC_WDS, 1635
SMS_EVENT_ETWS, 1635
SMS_EVENT_ETWS_PLMN, 1635
SMS_EVENT_MESSAGE_MODE, 1635

SMS_EVENT_MT_MESSAGE, 1635
SMS_EVENT_SMS_ON_IMS, 1635
SMS_EVENT_SMSC_ADDRESS, 1635
SMS_EVENT_TRANSFER_ROUTE_MT_MESSAGE, 1635
SYS_SRV_DOMAIN_CAMPED, 1635
SYS_SRV_DOMAIN_CS_ONLY, 1635
SYS_SRV_DOMAIN_CS_PS, 1635
SYS_SRV_DOMAIN_NO_SRV, 1635
SYS_SRV_DOMAIN_PS_ONLY, 1635
qaGobiApiFms.h
eGOBI_DEV_SERIES_9X07_9X50, 1716
eGOBI_DEV_SERIES_9X15, 1716
eGOBI_DEV_SERIES_9X30, 1716
eGOBI_DEV_SERIES_G3K, 1716
eGOBI_DEV_SERIES_NON_GOBI, 1716
eGOBI_DEV_SERIES_SIERRA_GOBI, 1716
eGOBI_DEV_SERIES_UNKNOWN, 1716
eGOBI_IMG_CAR_3, 1717
eGOBI_IMG_CAR_AERIS, 1717
eGOBI_IMG_CAR_ALLTEL, 1716
eGOBI_IMG_CAR_AMX_TELCEL, 1717
eGOBI_IMG_CAR_ATT, 1716
eGOBI_IMG_CAR_BELL, 1716
eGOBI_IMG_CAR_BHARTI, 1717
eGOBI_IMG_CAR_BRASIL_VIVO, 1717
eGOBI_IMG_CAR_CHINA_MOBILE, 1717
eGOBI_IMG_CAR_CHINA_TELECOM, 1716
eGOBI_IMG_CAR_CHINA_UNICOM, 1716
eGOBI_IMG_CAR_EMOBILE, 1717
eGOBI_IMG_CAR_FACTORY, 1716
eGOBI_IMG_CAR_GENERIC, 1716
eGOBI_IMG_CAR_GENERIC_CDMA, 1716
eGOBI_IMG_CAR_IUSACELL, 1716
eGOBI_IMG_CAR_KDDI, 1716
eGOBI_IMG_CAR_KT_FREETEL, 1717
eGOBI_IMG_CAR_LEAP, 1716
eGOBI_IMG_CAR_METROPCS, 1716
eGOBI_IMG_CAR_NETCOM, 1717
eGOBI_IMG_CAR_NORF, 1716
eGOBI_IMG_CAR_NTT_DOCOMO, 1717
eGOBI_IMG_CAR_O2, 1717
eGOBI_IMG_CAR_OMH, 1716
eGOBI_IMG_CAR_ORANGE, 1717
eGOBI_IMG_CAR_RELIANCE1, 1716
eGOBI_IMG_CAR_RELIANCE2, 1717
eGOBI_IMG_CAR_ROGERS, 1717
eGOBI_IMG_CAR_SFR, 1717
eGOBI_IMG_CAR_SINGTEL_OPTUS, 1717
eGOBI_IMG_CAR_SK_TELCOM1, 1716
eGOBI_IMG_CAR_SK_TELCOM2, 1717
eGOBI_IMG_CAR_SOFTBANK, 1717
eGOBI_IMG_CAR_SPRINT, 1716
eGOBI_IMG_CAR_SWISSCOM, 1717
eGOBI_IMG_CAR_TATA, 1716
eGOBI_IMG_CAR_TELCOM_ITALIA, 1717
eGOBI_IMG_CAR_TELCOM_NZ, 1716
eGOBI_IMG_CAR_TELEFONICA, 1717
eGOBI_IMG_CAR_TELENOR, 1717
eGOBI_IMG_CAR_TELIASONERA, 1717
eGOBI_IMG_CAR_TELSTRA1, 1716
eGOBI_IMG_CAR_TELSTRA2, 1717
eGOBI_IMG_CAR_TELUS, 1716
eGOBI_IMG_CAR_TMOBILE, 1716
eGOBI_IMG_CAR_US, 1716
eGOBI_IMG_CAR_VERIZON, 1716
eGOBI_IMG_CAR_VODAFONE, 1716
eGOBI_IMG_GPS_ASSISTED, 1717
eGOBI_IMG_GPS_NO_XTRA, 1717
eGOBI_IMG_GPS_NONE, 1717
eGOBI_IMG_GPS_STAND_ALONE, 1717
eGOBI_IMG_REG_ASIA, 1717
eGOBI_IMG_REG_AUS, 1717
eGOBI_IMG_REG_EU, 1717
eGOBI_IMG_REG_GLOBAL, 1717
eGOBI_IMG_REG_LA, 1717
eGOBI_IMG_REG_NA, 1717
eGOBI_IMG_TECH_CDMA, 1718
eGOBI_IMG_TECH_UMTS, 1718
eGobi_DEV_SERIES_MC83, 1716
qaGobiApiNas.h
eNAS_LTE_CPHY_CA_BW_NRB_100, 1765
eNAS_LTE_CPHY_CA_BW_NRB_15, 1765
eNAS_LTE_CPHY_CA_BW_NRB_25, 1765
eNAS_LTE_CPHY_CA_BW_NRB_50, 1765
eNAS_LTE_CPHY_CA_BW_NRB_6, 1765
eNAS_LTE_CPHY_CA_BW_NRB_75, 1765
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED, 1765
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED, 1765
eNAS_LTE_CPHY_SCELL_STATE_DECONFIGURED, 1765
eNAS_RADIO_IF_GSM, 1764
eNAS_RADIO_IF_LTE, 1764
eNAS_RADIO_IF_TDSCDMA, 1764
eNAS_RADIO_IF_UMTS, 1764
eSYS_SRV_DOMAIN_CAMPED, 1764
eSYS_SRV_DOMAIN_CS_ONLY, 1764
eSYS_SRV_DOMAIN_CS_PS, 1764
eSYS_SRV_DOMAIN_NO_SRV, 1764
eSYS_SRV_DOMAIN_PS_ONLY, 1764
eSYS_SRV_DOMAIN_UNKNOWN, 1764
qaGobiApiPds.h
eSetServiceAutomaticTrackingDisable, 1798
eSetServiceAutomaticTrackingEnable, 1798
qaGobiApiSar.h
QMI_SAR_RF_STATE_1, 1821
QMI_SAR_RF_STATE_2, 1821
QMI_SAR_RF_STATE_3, 1821
QMI_SAR_RF_STATE_4, 1821
QMI_SAR_RF_STATE_5, 1821
QMI_SAR_RF_STATE_6, 1821
QMI_SAR_RF_STATE_7, 1821
QMI_SAR_RF_STATE_8, 1821
QMI_SAR_RF_STATE_DEFAULT, 1821

qaGobiApiVoice.h
VOICE_SUPS_SRV_CLASS_DATA, 1933
VOICE_SUPS_SRV_CLASS_DATACIRCUITASY-NC, 1933
VOICE_SUPS_SRV_CLASS_DATACIRCUITSYN-C, 1933
VOICE_SUPS_SRV_CLASS_FAX, 1933
VOICE_SUPS_SRV_CLASS_NONE, 1933
VOICE_SUPS_SRV_CLASS_PACKETACCESS, 1933
VOICE_SUPS_SRV_CLASS_PADACCESS, 1933
VOICE_SUPS_SRV_CLASS_SMS, 1933
VOICE_SUPS_SRV_CLASS_VOICE, 1933
qaGobiApiWds.h
QMI_WDS_CURRENT_CALL_DB_MASK, 1957
QMI_WDS_LAST_CALL_DB_MASK, 1957
qaNasGetRFBandInfo.h
eTLV_RF_BAND_INFO, 1993
qaNasPerformNetworkScan.h
eTLV_3GPP_NETWORK_INFO, 1995
qaCbkCatEventReportInd.h, 1578
UpkQmiCbkCatEventReportInd, 1579
qaCbkSwiOmaDmEventReportInd.h, 1579
UpkQmiCbkSwiOmaDmEventReportInd, 1580
UpkQmiCbkSwiOmaDmEventReportIndExt, 1580
qaGobiApiAudio.h, 1580
SLQSGetAudioPathConfig, 1581
SLQSGetAudioProfile, 1581
SLQSGetAudioVolTLBConfig, 1582
SLQSSetAudioPathConfig, 1582
SLQSSetAudioProfile, 1583
SLQSSetAudioVolTLBConfig, 1583
qaGobiApiCat.h, 1584
CATSendEnvelopeCommand, 1584
CATSendTerminalResponse, 1585
qaGobiApiCbk.h, 1585
accelAcceptReady, 1595
accelTempAcceptReady, 1596
CBK_ENABLE_EVENT, 1594
CBK_NOCHANGE, 1594
DEREGISTER_EVENT, 1594
DEREGISTER_SRV, 1594
device_state_enum, 1635
eDevState, 1596
eQaQMIService, 1635
eSMSEventType, 1596
EVENT_MASK_CARD, 1594
FIRST_INSTANCE, 1594
gpsTime, 1597
gyroAcceptReady, 1597
gyroTempAcceptReady, 1597
INVALID_INSTACNE, 1594
IPV4, 1594
IPV4V6, 1594
IPV6, 1594
iSLQSSetDUNCallInfoCallback, 1636
iSLQSSetSignalStrengthsCallback, 1636
iSLQSSetWdsFirstInstEventCallback, 1636
iSLQSSetWdsSecondInstEventCallback, 1636
iSLQSSetWdsThirdInstEventCallback, 1636
iSLQSSetWdsXferStatsFirstInstCallback, 1636
iSLQSSetWdsXferStatsSecondInstCallback, 1636
iSetCATEventCallback, 1636
iSetSignalStrengthCallback, 1636
LteNasReleaseInfo, 1598
MAX_NO_OF_CALLS, 1594
MAX_NO_OF_FILES, 1595
MAX_NO_OF_SLOTS, 1595
MAX_PATH_LENGTH, 1595
MAXUSSDLENGTH, 1595
modemTempNotification, 1598
NAS_SRV, 1595
NUM_OF_SET, 1595
PDS_SRV, 1595
packetSrvStatus, 1598
precisionDilution, 1600
REGISTER_EVENT, 1595
REGISTER_SRV, 1595
ResetInfoNotification, 1600
SECOND_INSTANCE, 1595
SLQSNasNetworkRejectCallback, 1651
SLQSNasNetworkTimeCallBack, 1651
SLQSNasSigInfo2CallBack, 1651
SLQSNasSigInfoCallBack, 1652
SLQSNasSwiOTAMessageCallback, 1652
SLQSNasSysInfoCallBack, 1653
SLQSNasTimerCallback, 1653
SLQSSetBandPreferenceCbk, 1653
SLQSSetDHCPv4ClientLeaseStatusCallback, 1654
SLQSSetDUNCallInfoCallback, 1655
SLQSSetDataSystemStatusCallback, 1654
SLQSSetIMSAPdpStatusCallback, 1655
SLQSSetIMSARatStatusCallback, 1655
SLQSSetIMSARegStatusCallback, 1656
SLQSSetIMSASvcStatusCallback, 1656
SLQSSetIMSSMSConfigCallback, 1657
SLQSSetIMSUserConfigCallback, 1657
SLQSSetIMSVoIPConfigCallback, 1658
SLQSSetLocInjectPositionCallback, 1658
SLQSSetLocInjectUTCTimeCallback, 1658
SLQSSetModemTempCallback, 1658
SLQSSetPacketSrvStatusCallback, 1659
SLQSSetQosEventCallback, 1659
SLQSSetQosNWStatusCallback, 1660
SLQSSetQosPriEventCallback, 1660
SLQSSetQosStatusCallback, 1660
SLQSSetRegMgrConfigCallback, 1661
SLQSSetSDKTerminatedCallback, 1661
SLQSSetSIPConfigCallback, 1663
SLQSSetSMSEventCallback, 1664
SLQSSetServingSystemCallback, 1662
SLQSSetSessionStateCallback, 1662
SLQSSetSignalStrengthsCallback, 1663
SLQSSetSwiGetResetInfoCallback, 1664
SLQSSetSwiHDRPersCallback, 1664

SLQSSetSysSelectionPrefCallBack, 1665
SLQSSetTransLayerInfoCallback, 1665
SLQSSetTransNWRegInfoCallback, 1665
SLQSSetWdsEventCallback, 1666
SLQSSetWdsTransferStatisticCallback, 1667
SLQSSwiAVMSEventReportCallback, 1667
SLQSTmdMitigationLvlRptCallback, 1667
SLQSUIMSetRefreshCallBack, 1668
SLQSUIMSetStatusChangeCallBack, 1668
SLQSVoiceInfoRecCallback, 1668
SLQSVoiceSetAllCallStatusCallBack, 1669
SLQSVoiceSetDTMFEventCallBack, 1669
SLQSVoiceSetOTASPStatusCallBack, 1670
SLQSVoiceSetPrivacyChangeCallBack, 1670
SLQSVoiceSetSUPSCallBack, 1671
SLQSVoiceSetSUPSNotificationCallback, 1671
SLQSWmsAsyncRawSendCallBack, 1671
SLQSWmsMemoryFullCallBack, 1672
SLQSWmsMessageWaitingCallBack, 1672
SMSAsyncRawSend, 1602
SMSCAddressInfo, 1603
SMSEtwsMessageInfo, 1603
SMSEtwsPlmnInfo, 1603
SMSEventInfo, 1603
SMSEventType, 1635
SMSMTMessageInfo, 1604
SMSMessageModeInfo, 1604
SMSOnIMSInfo, 1605
SMSTransferRouteMTMessageInfo, 1605
sensorDataUsage, 1601
sessionInformation, 1601
sessionInformationExt, 1601
SetActivationStatusCallback, 1636
SetCATEventCallback, 1636
SetDataCapabilitiesCallback, 1637
SetDeviceStateChangeCbk, 1638
SetFwDldCompletionCbk, 1638
SetGPSCallback, 1639
SetLURejectCallback, 1643
SetLocBestAvailPosCallback, 1639
SetLocCradleMountCallback, 1639
SetLocDeleteAssistDataCallback, 1640
SetLocEngineStateCallback, 1640
SetLocEventMaskNMEACallback, 1640
SetLocEventPositionCallback, 1640
SetLocEventTimeSyncCallback, 1641
SetLocGetServerCallback, 1641
SetLocGnssSvInfoCallback, 1641
SetLocInjectSensorDataCallback, 1642
SetLocInjectTimeCallback, 1642
SetLocOpModeCallback, 1642
SetLocSensorStreamingCallback, 1642
SetLocSetExtPowerConfigCallback, 1643
SetMobileIPStatusCallback, 1643
SetNMEACallback, 1645
SetNasLTECphyCaIndCallback, 1644
SetNetChangeCbk, 1644
SetNewSMSCallback, 1645
SetOMADMStateCallback, 1646
SetPDSStateCallback, 1646
SetPowerCallback, 1646
SetRFInfoCallback, 1647
SetRMTransferStatisticsCallback, 1647
SetRankIndicatorCallback, 1647
SetRoamingIndicatorCallback, 1647
SetSLQSOMADMAlertCallback, 1649
SetSLQSOMADMAlertCallbackExt, 1649
SetSignalStrengthCallback, 1648
SetUSSDNoWaitIndicationCallback, 1650
SetUSSDNotificationCallback, 1650
SetUSSDReleaseCallback, 1650
SetUimSlotStatusChangeCallback, 1649
svUsedforFix, 1605
SwiAvmsEventInd, 1606
SwiOTAMsg, 1606
tFNASwiLTECphyCaInfo, 1607
tFNASwiOTAMsg, 1607
tFNActivationStatus, 1606
tFNAllCallStatus, 1607
tFNAsyncRawSend, 1607
tFNBandPreference, 1608
tFNBestAvailPos, 1609
tFNCATEvent, 1610
tFNCbkUimSlotStatusChangeInd, 1610
tFNDHCPv4ClientLeaseStatus, 1612
tFNDTMFEvent, 1612
tFNDUNCallInfo, 1612
tFNDataCapabilities, 1610
tFNDataSysStatus, 1611
tFNDelAssistData, 1611
tFNDeviceStateChange, 1611
tFNEventNmea, 1612
tFNEventPosition, 1612
tFNFwDldCompletion, 1613
tFNGetServer, 1613
tFNGnssSvInfo, 1613
tFNHDRPersonaity, 1613
tFNImsRegMgrConfig, 1614
tFNImsSIPConfig, 1615
tFNImsSMSConfig, 1615
tFNImsUserConfig, 1615
tFNImsVoIPConfig, 1615
tFNImsaPdpStatus, 1614
tFNImsaRatStatus, 1614
tFNImsaRegStatus, 1614
tFNImsaSvcStatus, 1614
tFNInfoRec, 1615
tFNInjectPosition, 1616
tFNInjectSensorData, 1616
tFNInjectTimeStatus, 1616
tFNInjectUTCTime, 1616
tFNLUReject, 1617
tFNMemoryFull, 1619
tFNMessageWaiting, 1619
tFNMitiLvlRpt, 1619
tFNMobileIPStatus, 1619

tFNModemTempInfo, 1619
tFNNasTimer, 1620
tFNNet, 1620
tFNNetworkReject, 1620
tFNNetworkTime, 1620
tFNNewGPS, 1621
tFNNewNMEA, 1621
tFNNewRMTransferStatistics, 1621
tFNNewSMS, 1622
tFNOMADMState, 1622
tFNOTASPStatus, 1624
tFNOpMode, 1623
tFNPDSState, 1624
tFNPacketSrvState, 1624
tFNPower, 1625
tFNPrivacyChange, 1625
tFNQosNWStatus, 1625
tFNQosPriEvent, 1626
tFNQosStatus, 1626
tFNRFInfo, 1627
tFNRankIndicator, 1627
tFNResetInfo, 1627
tFNRoamingIndicator, 1628
tFNSDKTerminated, 1628
tFNSLQSOMADMAlert, 1630
tFNSLQSQOSEvent, 1630
tFNSLQSSessionState, 1630
tFNSLQSSignalStrengths, 1631
tFNSLQSWDSEvent, 1631
tFNSMSEvents, 1631
tFNSUPSInfo, 1631
tFNSUPSNotification, 1631
tFNSensorStreaming, 1628
tFNServingSystem, 1628
tFNSetCradleMount, 1629
tFNSetEngineState, 1629
tFNSetEventTimeSync, 1629
tFNSetExtPowerConfig, 1629
tFNSigInfo, 1629
tFNSignalStrength, 1630
tFNSwiAvmEventReport, 1632
tFNSysInfo, 1632
tFNSysSelectionPref, 1632
tFNUIMRefresh, 1633
tFNUIMStatusChangeInfo, 1633
tFNUSSDNoWaitIndication, 1633
tFNUSSDNotification, 1633
tFNUSSDRelease, 1634
tFNtransLayerInfo, 1632
tFNtransNWRegInfo, 1632
THIRD_INSTANCE, 1595
transLayerNotification, 1634
transNWRegInfoNotification, 1634
USSD_DCS_8BIT, 1595
USSD_DCS_ASCII, 1595
USSD_DCS_UCS2, 1595
VOICE_SRV, 1595
WDS_SRV, 1595
qaGobiApiDcs.h, 1673
GetQmapIsSupported, 1674
LEN, 1674
PORTNAM_LEN, 1674
QCWWAN2kConnect, 1674
QCWWAN2kEnumerateDevices, 1675
QCWWAN2kGetConnectedDeviceID, 1675
QCWWANConnect, 1676
QCWWANDisconnect, 1676
QCWWANEnumerateDevices, 1677
SLQSGetDeviceMode, 1677
SLQSGetNetStatistic, 1678
SLQSGetUsbPortNames, 1678
SLQSKillSDKProcess, 1679
SLQSSetLoggingMask, 1679
SLQSStart, 1679
SLQSStart_AVAgent, 1680
SLQSStartSrv, 1680
SetSDKImagePath, 1677
qaGobiApiDms.h, 1681
ActivateAutomatic, 1690
custFeaturesInfo, 1683
custFeaturesSetting, 1685
dmsCurrentPRLInfo, 1686
ERIFileparams, 1687
GetActivationState, 1691
GetDeviceCapabilities, 1691
GetFirmwareRevision, 1692
GetFirmwareRevisions, 1693
GetHardwareRevision, 1693
GetManufacturer, 1694
GetModelID, 1694
GetNetworkTime, 1695
GetOfflineReason, 1695
GetPRLVersion, 1696
GetPower, 1696
GetSerialNumbers, 1697
GetVoiceNumber, 1698
MAX_CUST_ID_LEN, 1683
MAX_FSN_LENGTH, 1683
ResetToFactoryDefaults, 1698
SLQSDmsSwiGetResetInfo, 1699
SLQSDmsSwiIndicationRegister, 1699
SLQSGetBandCapabilities, 1700
SLQSGetBandCapability, 1700
SLQSGetCurrentPRLInfo, 1702
SLQSGetCustFeatures, 1702
SLQSGetCustFeaturesV2, 1702
SLQSGetERIFile, 1703
SLQSGetSerialNumbers, 1703
SLQSSetCustFeatures, 1704
SLQSSetCustFeaturesV2, 1704
SLQSSwiClearDyingGaspStatistics, 1704
SLQSSwiGetCrashAction, 1705
SLQSSwiGetCrashInfo, 1705
SLQSSwiGetDyingGaspCfg, 1706
SLQSSwiGetDyingGaspStatistics, 1706
SLQSSwiGetFSN, 1706

SLQSSwiGetFwUpdateStatus, 1707
SLQSSwiGetHostDevInfo, 1707
SLQSSwiGetHostDevInfoParams, 1688
SLQSSwiGetOSInfo, 1707
SLQSSwiGetOSInfoParams, 1688
SLQSSwiGetSerialNoExt, 1708
SLQSSwiGetSerialNoExtParams, 1689
SLQSSwiGetUSBComp, 1708
SLQSSwiSetCrashAction, 1709
SLQSSwiSetDyingGaspCfg, 1709
SLQSSwiSetHostDevInfo, 1709
SLQSSwiSetHostDevInfoParams, 1689
SLQSSwiSetOSInfo, 1710
SLQSSwiSetOSInfoParams, 1690
SLQSSwiSetUSBComp, 1710
serialNumbersInfo, 1687
SetPower, 1699
ValidateSPC, 1711
qaGobiApiFms.h, 1711
BUILD_ID_LEN, 1715
DEVICE_OFFLINE, 1715
DEVICE_RESET, 1715
DEVICE_SHUTDOWN, 1715
DeleteStoredImage, 1718
DownloadToSlot, 1718
eGetDeviceSeries, 1719
eGobiDeviceSeries, 1716
eGobiImageCarrier, 1716
eGobiImageGPS, 1717
eGobiImageRegion, 1717
eGobiImageTech, 1717
GetImageStore, 1720
GetImagesPreference, 1719
GetStoredImages, 1720
IMG_ID_LEN, 1715
IMGDETAILS_LEN, 1715
PRI_UPDATE_FAIL, 1715
SLQSDownloadFirmwareToSlot, 1721
SLQSGetBootVersionNumber, 1722
SLQSGetFirmwareInfo, 1723
SLQSGetImageInfo, 1723
SLQSGetImageInfo_9x15, 1724
SLQSGetImageInfoMC77xx, 1724
SLQSGetImageInfoMC83xx, 1725
SLQSGetValidFwPriCombinations, 1725
SLQSIsSpkgFormatRequired, 1726
SLQSSetCrashStateCheckIgnore, 1726
SLQSSetSIMBasedImageSwitching, 1727
SLQSSetSpkgFormatRequired, 1727
SLQSSwiGetAllCarrierImages, 1727
SLQSSwiGetFirmwareCurr, 1728
SLQSUpgradeFirmware9x07, 1728
SLQSUpgradeFirmware9x15, 1729
SetImagesPreference, 1720
UNIQUE_ID_LEN, 1715
upgrade_mc77xx_fw, 1730
UpgradeFirmware2k, 1730
qaGobiApiIms.h, 1730
SLQSGetIMSSMSConfig, 1731
SLQSGetIMSUserConfig, 1732
SLQSGetIMSVoIPConfig, 1732
SLQSGetRegMgrConfig, 1733
SLQSGetSIPConfig, 1733
SLQSImsConfigIndicationRegister, 1734
SLQSSetIMSSMSConfig, 1734
SLQSSetIMSUserConfig, 1735
SLQSSetIMSVoIPConfig, 1735
SLQSSetRegMgrConfig, 1736
SLQSSetSIPConfig, 1736
qaGobiApiImsa.h, 1737
SLQSGetIMSARegStatus, 1737
SLQSGetIMSAServiceStatus, 1738
SLQSGetIMSASupportedFields, 1738
SLQSGetIMSASupportedMsg, 1739
SLQSRegisterIMSAIndication, 1739
qaGobiApiLoc.h, 1740
SLQSLOCDelAssData, 1741
SLQSLOCEventRegister, 1741
SLQSLOCGetBestAvailPos, 1742
SLQSLOCGetServer, 1742
SLQSLOCInjectPosition, 1743
SLQSLOCInjectSensorData, 1743
SLQSLOCInjectUTCTime, 1743
SLQSLOCSetCradleMountConfig, 1744
SLQSLOCSetExtPowerState, 1744
SLQSLOCSetOpMode, 1745
SLQSLOCStart, 1745
SLQSLOCStop, 1746
SwiLocGetAutoStart, 1746
SwiLocSetAutoStart, 1746
qaGobiApiNas.h, 1747
eSYS_SRV_DOMAIN, 1764
GetACCOLC, 1765
GetANAAAAuthenticationStatus, 1765
GetCDMANetworkParameters, 1766
GetHomeNetwork, 1767
GetHomeNetwork3GPP2, 1768
GetNetworkPreference, 1770
GetRFInfo, 1771
GetServingNetwork, 1771
GetServingNetworkCapabilities, 1772
GetSignalStrengths, 1773
IMSI_M_S1_LENGTH, 1752
IMSI_M_S2_LENGTH, 1752
InitiateDomainAttach, 1774
InitiateNetworkRegistration, 1774
MAX_PILOT_SETS, 1752
NAM_NAME_LENGTH, 1752
PLMN_LENGTH, 1752
PerformNetworkScan, 1775
SLQSConfigSigInfo, 1778
SLQSGetErrorRate, 1779
SLQSGetNetworkTime, 1779
SLQSGetOperatorNameData, 1779
SLQSGetPLMNName, 1780
SLQSGetServingSystem, 1780

SLQSGetSignalStrength, 1781
SLQSGetSysSelectionPref, 1781
SLQSInitiateNetworkRegistration, 1782
SLQSNASGetLTECPHYCaInfo, 1784
SLQSNASSwiGetChannelLock, 1787
SLQSNASSwiSetChannelLock, 1789
SLQSNasConfigSigInfo2, 1782
SLQSNasGet3GPP2Subscription, 1783
SLQSNasGetCellLocationInfo, 1783
SLQSNasGetHDRColorCode, 1784
SLQSNasGetSigInfo, 1784
SLQSNasGetSysInfo, 1785
SLQSNasGetTxRxInfo, 1785
SLQSNasIndicationRegister, 1786
SLQSNasIndicationRegisterExt, 1787
SLQSNasIndicationRegisterLTECphyCa, 1787
SLQSNasSwiIndicationRegister, 1788
SLQSNasSwiModemStatus, 1788
SLQSPerformNetworkScan, 1789
SLQSSetBandPreference, 1789
SLQSSetSysSelectionPref, 1791
SLQSSwiGetHDRPersonality, 1791
SLQSSwiGetHDRProtSubtype, 1792
SLQSSwiGetHRPDStats, 1792
SLQSSwiGetLteCQI, 1792
SLQSSwiGetLteSccRxInfo, 1793
SLQSSwiNetworkDebug, 1793
SLQSSwiPSDetach, 1794
SetACCOLC, 1775
SetCDMANetworkParameters, 1776
SetNetworkPreference, 1777
SlqsNas3GppNetworkRAT, 1753
slqsNetworkScanInfo, 1753
sysSelectPrefInfo, 1754
sysSelectPrefParams, 1758
UATISIZE, 1753
qaGobiApiOmadm.h, 1794
OMADMCancelSession, 1794
OMADMGetPendingNIA, 1795
OMADMGetSessionInfo, 1795
OMADMStartSession, 1797
qaGobiApiPds.h, 1797
DEFAULTBYTEVALUE, 1798
DEFAULTLONGVALUE, 1798
DEFAULTWORDVALUE, 1798
ForceXTRADownload, 1799
GetPDSDefaults, 1799
GetPDSState, 1800
GetPortAutomaticTracking, 1800
GetServiceAutomaticTracking, 1801
GetXTRAAutomaticDownload, 1801
GetXTRANetwork, 1802
GetXTRAValidity, 1802
PDSInjectTimeReference, 1803
ResetPDSData, 1803
SLQSGetAGPSConfig, 1807
SLQSGetGPSStateInfo, 1808
SLQSPDSDeterminePosition, 1808
SLQSPDSInjectAbsoluteTimeReference, 1809
SLQSPDSInjectPositionData, 1809
SLQSSetAGPSConfig, 1810
SLQSSetPositionMethodState, 1810
SetPDSDefaults, 1804
SetPDSState, 1805
SetPortAutomaticTracking, 1805
SetServiceAutomaticTracking, 1806
SetXTRAAutomaticDownload, 1806
SetXTRANetwork, 1807
StartPDSTrackingSessionExt, 1811
StopPDSTrackingSession, 1812
qaGobiApiQos.h, 1812
SLQSQosGetFlowStatus, 1813
SLQSQosGetGranted, 1813
SLQSQosGetNWProf, 1814
SLQSQosGetNetworkStatus, 1814
SLQSQosModify, 1815
SLQSQosRel, 1815
SLQSQosReq, 1816
SLQSQosReset, 1816
SLQSQosResume, 1817
SLQSQosSuspend, 1817
SLQSQosSwiReadApnExtraParams, 1817
SLQSQosSwiReadDataStats, 1818
qaGobiApiRms.h, 1818
GetSMSWake, 1819
SetSMSWake, 1819
qaGobiApiSar.h, 1820
eQMISARRFState, 1820
SLQSGetRfSarState, 1821
SLQSSetRfSarState, 1822
qaGobiApiSms.h, 1822
ABSOLUTE_VALIDITY, 1824
CONFIG_LEN, 1824
getIndicationRegResp, 1825
GetSMSCAddress, 1828
getTransLayerInfoResp, 1825
getTransNWRegInfoResp, 1826
MAX_SMS_ROUTES, 1824
NUM_OF_SET, 1824
qaQmi3GPP2BroadcastCfgInfo, 1826
qaQmi3GPPBroadcastCfgInfo, 1827
SLQSCDMADecodeMTTextMsg, 1831
SLQSCDMADecodeMTTextMsgExt, 1831
SLQSCDMAEncodeMOTextMsg, 1832
SLQSDeleteSMS, 1832
SLQSGetIndicationRegister, 1833
SLQSGetMessageWaiting, 1834
SLQSGetSMS, 1834
SLQSGetSMSList, 1836
SLQSGetSmsBroadcastConfig, 1835
SLQSGetTransLayerInfo, 1836
SLQSGetTransNWRegInfo, 1837
SLQSModifySMSStatus, 1837
SLQSSendAsyncSMS, 1838
SLQSSendLongSMS, 1838
SLQSSendSMS, 1839

SLQSSetIndicationRegister, 1840
SLQSSetSmsBroadcastActivation, 1840
SLQSSetSmsBroadcastConfig, 1841
SLQSSetSmsStorage, 1841
SLQSSmsGetMaxStorageSize, 1842
SLQSSmsGetMessageProtocol, 1842
SLQSSmsSetRoutes, 1843
SLQSSwiGetSMSStorage, 1843
SLQSWCDMADecodeLongTextMsg, 1844
SLQSWCDMADecodeMTTextMsg, 1844
SLQSWCDMAEncodeMOTextMsg, 1844
SaveSMS, 1829
SendSMS, 1829
setIndicationRegReq, 1827
SetSMSCAddress, 1830
TIME_DATE_BUF, 1824
TIME_STAMP_BUF, 1824
transLayerInfo, 1828
qaGobiApiSwi.h, 1845
SLQSGetPidof, 1845
SLQSGetSdkVersion, 1846
SLQSSendRawQMI, 1846
qaGobiApiSwiAudio.h, 1846
SLQSGetM2MAVMute, 1848
SLQSGetM2MAudioProfile, 1847
SLQSGetM2MAudioVolume, 1848
SLQSGetM2MSpkrGain, 1848
SLQSSetM2MAVMute, 1851
SLQSSetM2MAudioAVCFG, 1849
SLQSSetM2MAudioLPBK, 1849
SLQSSetM2MAudioNVDef, 1850
SLQSSetM2MAudioProfile, 1850
SLQSSetM2MAudioVolume, 1850
SLQSSetM2MSpkrGain, 1851
qaGobiApiSwiAvms.h, 1852
QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 1854
QmiSwiAvmsEventReportConfigResp, 1855
QmiSwiAvmsEventReportConnectionRequestResp, 1855
QmiSwiAvmsEventReportDataSessionStatusResp, 1856
QmiSwiAvmsEventReportHTTPStatusResp, 1856
QmiSwiAvmsEventReportNotificationResp, 1857
QmiSwiAvmsEventReportPackageIDResp, 1858
QmiSwiAvmsEventReportRegStatusResp, 1858
QmiSwiAvmsEventReportSessionTypeResp, 1858
QmiSwiAvmsEventReportWAMSParamChangeResp, 1859
QmiSwiAvmsSetSettingsAPNInfo, 1859
QmiSwiAvmsSetSettingsConnectionRetryTimers, 1859
QmiSwiAvmsSetSettingsPeriodInfo, 1859
SLQSAVMSGetSessionInfo, 1861
SLQSAVMSGetSettings, 1861
SLQSAVMSSelection, 1861
SLQSAVMSSetEventReport, 1862
SLQSAVMSSetSettings, 1862
SLQSAVMSSetSettingsIgnoreAutoReboot, 1863
SLQSAVMSSetSettingsReq, 1859
SLQSAVMSSettings, 1860
SLQSAVMSSettingsAPNInfo, 1860
SLQSAVMSSettingsConnectionRetryTimers, 1860
SLQSAVMSSettingsPeriodsInfo, 1860
SLQSAVMSStartSession, 1863
SLQSAVMSStopSession, 1864
qaGobiApiSwiOmadms.h, 1864
SLQSOMADMCancelSession, 1870
SLQSOMADMGetSessionInfo, 1870
SLQSOMADMGetSettings, 1871
SLQSOMADMGetSettings2, 1871
SLQSOMADMSendSelection, 1872
SLQSOMADMSendSelection2, 1872
SLQSOMADMSessionInfo, 1865
SLQSOMADMSetSettings, 1873
SLQSOMADMSetSettings2, 1874
SLQSOMADMSetSettings3, 1874
SLQSOMADMSettings, 1867
SLQSOMADMSettingsReqParams, 1868
SLQSOMADMSettingsReqParams3, 1869
SLQSOMADMStartSession, 1874
SLQSOMADMStartSession2, 1875
qaGobiApiTableBandClasses.h, 1875
qaGobiApiTableCallControlReturnReasons.h, 1878
qaGobiApiTableCallEndReasons.h, 1879
qaGobiApiTableCarrierCodes.h, 1895
qaGobiApiTableCodingScheme.h, 1897
qaGobiApiTableGpsCapabilityCodes.h, 1900
qaGobiApiTablePowerModes.h, 1901
qaGobiApiTableRadioInterfaces.h, 1901
qaGobiApiTableRegionCodes.h, 1902
qaGobiApiTableServiceOptions.h, 1902
qaGobiApiTableSupServiceInfoClasses.h, 1905
qaGobiApiTableSwiAudio.h, 1905
qaGobiApiTableSwiOMADMSessionStatus.h, 1906
qaGobiApiTableSwiOMADMUpdateCompleteStatus.h, 1907
qaGobiApiTableVoiceCallEndReasons.h, 1908
qaGobiApiTmd.h, 1915
SLQSTmdDeRegNotMitigationLvl, 1915
SLQSTmdGetMitigationDevList, 1916
SLQSTmdGetMitigationLvl, 1916
SLQSTmdRegNotMitigationLvl, 1916
qaGobiApiUim.h, 1917
MAX_ICCID_LENGTH, 1919
MAX_NO_OF_SLOTS, 1919
MAX_PATH_LENGTH, 1919
MAX_PUK_LENGTH, 1919
MAX_SLOTS_STATUS, 1919
SLQSUIMAuthenticate, 1919
SLQSUIMChangePin, 1920
SLQSUIMDepersonalization, 1920
SLQSUIMEventRegister, 1921
SLQSUIMGetCardStatus, 1921
SLQSUIMGetConfiguration, 1922
SLQSUIMGetFileAttributes, 1922

SLQSUIMGetSlotsStatus, 1923
SLQSUIMPowerDown, 1923
SLQSUIMPowerUp, 1924
SLQSUIMReadTransparent, 1924
SLQSUIMRefreshComplete, 1925
SLQSUIMRefreshGetLastEvent, 1925
SLQSUIMRefreshOK, 1926
SLQSUIMRefreshRegister, 1926
SLQSUIMReset, 1927
SLQSUIMSetPinProtection, 1927
SLQSUIMSwitchSlot, 1928
SLQSUIMUnblockPin, 1928
SLQSUIMVerifyPin, 1929
qaGobiApiVoice.h, 1929
AnswerUSSD, 1933
CancelUSSD, 1933
MAX_CALL_NO_LEN, 1932
MAX_NO_OF_CALLS, 1932
MAXUSSDLENGTH, 1932
OriginateUSSD, 1934
PASSWORD_LENGTH, 1933
SLQSOriginateUSSD, 1934
SLQSVoiceALSSelectLine, 1935
SLQSVoiceALSSetLineSwitching, 1935
SLQSVoiceAnswerCall, 1936
SLQSVoiceBindSubscription, 1936
SLQSVoiceBurstDTMF, 1937
SLQSVoiceDialCall, 1937
SLQSVoiceEndCall, 1938
SLQSVoiceGetAllCallInfo, 1938
SLQSVoiceGetCLIP, 1941
SLQSVoiceGetCLIR, 1942
SLQSVoiceGetCNAP, 1942
SLQSVoiceGetCOLP, 1943
SLQSVoiceGetCOLR, 1943
SLQSVoiceGetCallBarring, 1939
SLQSVoiceGetCallForwardingStatus, 1939
SLQSVoiceGetCallInfo, 1940
SLQSVoiceGetCallWaiting, 1940
SLQSVoiceGetConfig, 1944
SLQSVoiceIndicationRegister, 1944
SLQSVoiceManageCalls, 1945
SLQSVoiceOrigUSSDNoWait, 1945
SLQSVoiceSendFlash, 1946
SLQSVoiceSetCallBarringPassword, 1946
SLQSVoiceSetConfig, 1947
SLQSVoiceSetPreferredPrivacy, 1947
SLQSVoiceSetSUPSService, 1948
SLQSVoiceStartContDTMF, 1949
SLQSVoiceStopContDTMF, 1949
serviceClassInformation, 1933
qaGobiApiWds.h, 1950
GetAutoconnect, 1957
GetByteTotals, 1957
GetConnectionRate, 1958
GetDataBearerTechnology, 1959
GetDefaultProfile, 1959
GetDefaultProfileLTE, 1961
GetDefaultProfileNum, 1963
GetDormancyState, 1963
GetIPAddressLTE, 1964
GetLastMobileIPError, 1964
GetMobileIP, 1965
GetMobileIPProfile, 1965
GetPacketStatistics, 1967
GetPacketStatus, 1967
GetProfileSettingIn, 1953
GetProfileSettingOut, 1954
GetSessionDuration, 1968
GetSessionState, 1968
iGetByteTotals, 1969
iGetConnectionRate, 1969
iGetPacketStatistics, 1969
iSLQSMISetIPFamilyPreference, 1969
qmiDataBearerMasks, 1957
QmiProfileInfo, 1954
QmiWDSDataBearerTechnology, 1955
QmiWDSDataBearers, 1954
RMSetTransferStatistics, 1969
SLQSAutoConnect, 1978
SLQSCreateProfile, 1978
SLQSDeleteProfile, 1979
SLQSGet3GPPConfigItem, 1980
SLQSGetByteTotals, 1980
SLQSGetConnectionRate, 1980
SLQSGetCurrDataSystemStat, 1981
SLQSGetCurrentChannelRate, 1981
SLQSGetDUNCallInfo, 1983
SLQSGetDataBearerTechnology, 1982
SLQSGetDataBearerTechnologyExt, 1982
SLQSGetPacketStatistics, 1983
SLQSGetProfile, 1984
SLQSGetProfileSettings, 1985
SLQSGetRuntimeSettings, 1986
SLQSGetSessionState, 1986
SLQSModifyProfile, 1987
SLQSResetPacketStatics, 1987
SLQSSGetDHCPv4ClientConfig, 1989
SLQSSGetLoopback, 1990
SLQSSSetDHCPv4ClientConfig, 1990
SLQSSSetLoopback, 1990
SLQSSet3GPPConfigItem, 1987
SLQSSetProfile, 1988
SLQSStartStopDataSession, 1991
SLQSWdsGoActive, 1991
SLQSWdsGoDormant, 1992
SLQSWdsSetEventReport, 1992
SLQSWdsSwiPDPRuntimeSettings, 1992
SetActiveMobileIPProfile, 1970
SetAutoconnect, 1970
SetDefaultProfile, 1970
SetDefaultProfileLTE, 1972
SetDefaultProfileLTEV2, 1973
SetDefaultProfileNum, 1975
SetMobileIP, 1975
SetMobileIPParameters, 1976

SetMobileIPProfile, 1977
slqs3GPPConfigItem, 1955
WDS_IsGobiDevice, 1993
qaNasGetRFBandInfo.h, 1993
PkQmiNasGetRFBandInfo, 1994
UpkQmiNasGetRFBandInfo, 1994
qaNasPerformNetworkScan.h, 1994
FORBIDDEN_INDEX, 1994
INDEX_ZERO, 1994
PREFERRED_INDEX, 1994
PkQmiNasPerformNetworkScan, 1995
ROAMING_INDEX, 1994
UpkQmiNasPerformNetworkScan, 1995
qaQmi3GPP2BroadcastCfgInfo
qaGobiApiSms.h, 1826
qaQmi3GPPBroadcastCfgInfo
qaGobiApiSms.h, 1827
qaQmi3Gpp2TimeZone, 774
daylightSavings, 774
leapSeconds, 774
localTimeOffset, 774
qaQmiInterfaceInfo, 775
qaQmiinstanceid, 775
qaQmisvctype, 775
v4sessionId, 775
v6sessionId, 775
qaQmiServingSystemParam, 775
BasestationID, 778
BasestationLatitude, 778
BasestationLongitude, 778
CDMA_P_Rev, 778
CDMASystemInfoExt, 778
CallBarStatus, 778
CellID, 778
concSvcInfo, 778
CurrentPLMN, 778
DTMInd, 779
DataSrvCapabilities, 779
defaultRoamInd, 779
DetailedSvcInfo, 779
Gpp2TimeZone, 779
GppNetworkDSTAdjustment, 779
GppTimeZone, 779
hdrPersonality, 779
Lac, 779
NetworkID, 779
PRLInd, 779
roamIndicatorVal, 779
RoamingIndicatorList, 779
ServingSystem, 779
SystemID, 779
trackAreaCode, 779
qaQmiinstanceid
qaQmiInterfaceInfo, 775
qaQmisvctype
qaQmiInterfaceInfo, 775
qm_wds_ds_profile_extended_err_codes
qmerrno.h, 2001
qmerrno.h
eQCWWAN_ERR_API_MUTEX_TIMEOUT, 1998
eQCWWAN_ERR_BUFFER_SZ, 1997
eQCWWAN_ERR_CANCEL_OP, 1998
eQCWWAN_ERR_DRIVER, 1998
eQCWWAN_ERR_ENUM_BEGIN, 1997
eQCWWAN_ERR_ENUM_END, 1998
eQCWWAN_ERR_FILE_COPY, 1997
eQCWWAN_ERR_FILE_OPEN, 1997
eQCWWAN_ERR_GENERAL, 1997
eQCWWAN_ERR_INTERNAL, 1997
eQCWWAN_ERR_INVALID_ARG, 1997
eQCWWAN_ERR_INVALID_DEVID, 1997
eQCWWAN_ERR_INVALID_FILE, 1997
eQCWWAN_ERR_INVALID_QMI_RSP, 1997
eQCWWAN_ERR_INVALID_XID, 1998
eQCWWAN_ERR_MALFORMED_QMI_RSP, 1997
eQCWWAN_ERR_MEMORY, 1997
eQCWWAN_ERR_MULTIPLE_DEVICES, 1998
eQCWWAN_ERR_MULTIPLE_SMS_UNSUPPORTED, 1998
eQCWWAN_ERR_NO_CANCELABLE_OP, 1998
eQCWWAN_ERR_NO_CONNECTION, 1997
eQCWWAN_ERR_NO_DEVICE, 1997
eQCWWAN_ERR_NO_SIGNAL, 1998
eQCWWAN_ERR_NONE, 1997
eQCWWAN_ERR_NULL_TLV, 2001
eQCWWAN_ERR_OFFLINE, 1998
eQCWWAN_ERR_PDU_GENERATION, 1998
eQCWWAN_ERR_QMI_ABORTED, 1998
eQCWWAN_ERR_QMI_ACCESS_DENIED, 2000
eQCWWAN_ERR_QMI_ACK_NOT_SENT, 2000
eQCWWAN_ERR_QMI_ARG_TOO_LONG, 1998
eQCWWAN_ERR_QMI_AUTHENTICATION_FAILED, 1999
eQCWWAN_ERR_QMI_AUTHENTICATION_LOCK, 1999
eQCWWAN_ERR_QMI_BUNDLING_NOT_SUPPORTED, 2000
eQCWWAN_ERR_QMI_CALL_FAILED, 1998
eQCWWAN_ERR_QMI_CARD_BUSY_RSP, 2001
eQCWWAN_ERR_QMI_CARD_CALL_CONTROL_FAILED, 2000
eQCWWAN_ERR_QMI_CAT_END, 2001
eQCWWAN_ERR_QMI_CAT_START, 2001
eQCWWAN_ERR_QMI_CAUSE_CODE, 1999
eQCWWAN_ERR_QMI_CLIENT_IDS_EXHAUSTED, 1998
eQCWWAN_ERR_QMI_CONNECT, 1997
eQCWWAN_ERR_QMI_DEVICE_IN_USE, 1998
eQCWWAN_ERR_QMI_DEVICE_NOT_READY, 1999
eQCWWAN_ERR_QMI_DEVICE_STORAGE_FULL, 1999
eQCWWAN_ERR_QMI_DISABLED, 1999
eQCWWAN_ERR_QMI_ENCODING, 1999

eQCWWAN_ERR_QMI_ENVELOPE_CMD_FAILURE, 2001
eQCWWAN_ERR_QMI_EVENT_REG_FAILED, 2001
eQCWWAN_ERR_QMI_EXTENDED_INTERNAL, 2000
eQCWWAN_ERR_QMI_FDN_RESTRICT, 2000
eQCWWAN_ERR_QMI_FLOW_SUSPENDED, 1999
eQCWWAN_ERR_QMI_GENERAL, 1999
eQCWWAN_ERR_QMI_HARDWARE_RESTRICTED, 2000
eQCWWAN_ERR_QMI_IFACE, 1997
eQCWWAN_ERR_QMI_INCOMPATIBLE_STATE, 2000
eQCWWAN_ERR_QMI_INCORRECT_FLOW_FILTER, 1999
eQCWWAN_ERR_QMI_INCORRECT_PIN, 1998
eQCWWAN_ERR_QMI_INFO_UNAVAILABLE, 2000
eQCWWAN_ERR_QMI_INJECT_TIMEOUT, 2000
eQCWWAN_ERR_QMI_INSUFFICIENT_RESOURCES, 1999
eQCWWAN_ERR_QMI_INTERFACE_NOT_FOUND, 1999
eQCWWAN_ERR_QMI_INTERNAL, 1998
eQCWWAN_ERR_QMI_INVALID_ARG, 1999
eQCWWAN_ERR_QMI_INVALID_CLIENT_ID, 1998
eQCWWAN_ERR_QMI_INVALID_DATA_FORMAT, 1999
eQCWWAN_ERR_QMI_INVALID_ENVELOPE_CMD, 2001
eQCWWAN_ERR_QMI_INVALID_HANDLE, 1998
eQCWWAN_ERR_QMI_INVALID_ID, 1999
eQCWWAN_ERR_QMI_INVALID_INDEX, 1999
eQCWWAN_ERR_QMI_INVALID_IP_FAMILY_PREF, 1999
eQCWWAN_ERR_QMI_INVALID_MCAST_HANDLE, 1999
eQCWWAN_ERR_QMI_INVALID_MESSAGE_ID, 1999
eQCWWAN_ERR_QMI_INVALID_OPERATION, 2000
eQCWWAN_ERR_QMI_INVALID_PDP_TYPE, 1998
eQCWWAN_ERR_QMI_INVALID_PINID, 1998
eQCWWAN_ERR_QMI_INVALID_PROFILE, 1998
eQCWWAN_ERR_QMI_INVALID_PROFILE_TYPE, 1998
eQCWWAN_ERR_QMI_INVALID_PS_ATTACH_ACTION, 1999
eQCWWAN_ERR_QMI_INVALID_QMI_CMD, 2000
eQCWWAN_ERR_QMI_INVALID_QOS_ID, 1999
eQCWWAN_ERR_QMI_INVALID_REGISTER_ACTION, 1999
eQCWWAN_ERR_QMI_INVALID_SERVICE_TYPE, 1998
eQCWWAN_ERR_QMI_INVALID_TECH_PREF, 1998
eQCWWAN_ERR_QMI_INVALID_TERMINAL_RSP, 2001
eQCWWAN_ERR_QMI_INVALID_TRANSITION, 1999
eQCWWAN_ERR_QMI_INVALID_TX_ID, 1998
eQCWWAN_ERR_QMI_MALFORMED_MSG, 1998
eQCWWAN_ERR_QMI_MAX, 2000
eQCWWAN_ERR_QMI_MAX_MCAST_REQUESTS_IN_USE, 1999
eQCWWAN_ERR_QMI_MAX_QOS_REQUESTS_IN_USE, 1999
eQCWWAN_ERR_QMI_MESSAGE_DELIVERY_FAILURE, 1999
eQCWWAN_ERR_QMI_MESSAGE_NOT_SENT, 1999
eQCWWAN_ERR_QMI_MISSING_ARG, 1998
eQCWWAN_ERR_QMI_MSG_BLOCKED, 2000
eQCWWAN_ERR_QMI_NETWORK_ABORTED, 2000
eQCWWAN_ERR_QMI_NETWORK_NOT_READY, 1999
eQCWWAN_ERR_QMI_NETWORK_QOS_UNAWARE, 1999
eQCWWAN_ERR_QMI_NO_EFFECT, 1998
eQCWWAN_ERR_QMI_NO_ENTRY, 1999
eQCWWAN_ERR_QMI_NO_FREE_PROFILE, 1998
eQCWWAN_ERR_QMI_NO_MEMORY, 1998
eQCWWAN_ERR_QMI_NO_NETWORK_FOUND, 1998
eQCWWAN_ERR_QMI_NO_RADIO, 2000
eQCWWAN_ERR_QMI_NO_SUBSCRIPTION, 2000
eQCWWAN_ERR_QMI_NO_THRESHOLDS, 1998
eQCWWAN_ERR_QMI_NOT_A_MCAST_IFACE, 1999
eQCWWAN_ERR_QMI_NOT_PROVISIONED, 1998
eQCWWAN_ERR_QMI_NOT_SUPPORTED, 2000
eQCWWAN_ERR_QMI_OFFSET, 1998
eQCWWAN_ERR_QMI_OP_DEVICE_UNSUPPORTED, 1998
eQCWWAN_ERR_QMI_OP_NETWORK_UNSUPPORTED, 1998
eQCWWAN_ERR_QMI_OP_PARTIAL_FAILURE, 2000
eQCWWAN_ERR_QMI_OUT_OF_CALL, 1998
eQCWWAN_ERR_QMI_PIN_BLOCKED, 1999
eQCWWAN_ERR_QMI_PIN_PERM_BLOCKED, 1999

eQCWWAN_ERR_QMI_POLICY_MISMATCH, 2000
eQCWWAN_ERR_QMI_REQ, 1997
eQCWWAN_ERR_QMI_REQ_SCH, 1997
eQCWWAN_ERR_QMI_REQ_TO, 1997
eQCWWAN_ERR_QMI_REQUESTED_NUM_UNSUPPORTED, 1999
eQCWWAN_ERR_QMI_RSP, 1997
eQCWWAN_ERR_QMI_RSP_TO, 1997
eQCWWAN_ERR_QMI_SEGMENT_ORDER, 2000
eQCWWAN_ERR_QMI_SEGMENT_TOO_LONG, 2000
eQCWWAN_ERR_QMI_SESSION_INACTIVE, 1999
eQCWWAN_ERR_QMI_SESSION_INVALID, 1999
eQCWWAN_ERR_QMI_SESSION_OWNERSHIP, 1999
eQCWWAN_ERR_QMI_SIM_FILE_NOT_FOUND, 2000
eQCWWAN_ERR_QMI_SIM_NOT_INITIALIZED, 1999
eQCWWAN_ERR_QMI_SMSC_ADDR, 2000
eQCWWAN_ERR_QMI_SUPS_FAILURE_CAUSE, 2000
eQCWWAN_ERR_QMI_TPDU_TYPE, 2000
eQCWWAN_ERR_QMI_UNABORTABLE_TRANSACTION, 1998
eQCWWAN_ERR_QMI_UNKNOWN, 1999
eQCWWAN_ERR_QMI_WIDTH, 2001
eQCWWAN_ERR_RESET, 1998
eQCWWAN_ERR_SWICM_AM_VERS_ERROR, 2000
eQCWWAN_ERR_SWICM_CALL_IN_PROGRESS, 2000
eQCWWAN_ERR_SWICM_END, 2001
eQCWWAN_ERR_SWICM_FAILED_TO_KILL_SDK_PROCESS, 2000
eQCWWAN_ERR_SWICM_INVALID_SESSION_ID, 2000
eQCWWAN_ERR_SWICM_INVALID_V4_SESSION_ID, 2000
eQCWWAN_ERR_SWICM_INVALID_V6_SESSION_ID, 2000
eQCWWAN_ERR_SWICM_NOT_IMPLEMENTED, 2000
eQCWWAN_ERR_SWICM_QMI_CLNT_NOT_SUPPORTED, 2000
eQCWWAN_ERR_SWICM_QMI_SVC_NOT_SUPPORTED, 2000
eQCWWAN_ERR_SWICM_SM_NO_AVAILABLE_SESSIONS, 2000
eQCWWAN_ERR_SWICM_SOCKET_IN_USE, 2000
eQCWWAN_ERR_SWICM_START, 2000
eQCWWAN_ERR_SWICM_TIMEOUT, 2000
eQCWWAN_ERR_SWICM_V4DWN_V6DWN, 2000
eQCWWAN_ERR_SWICM_V4DWN_V6UP, 2000
eQCWWAN_ERR_SWICM_V4UP_V6DWN, 2000
eQCWWAN_ERR_SWICM_V4UP_V6UP, 2000
eQCWWAN_ERR_SWIDCS_APP_DISCONNECTED, 2001
eQCWWAN_ERR_SWIDCS_DEVNODE_NOT_FOUND, 2001
eQCWWAN_ERR_SWIDCS_END, 2001
eQCWWAN_ERR_SWIDCS_FILEIO_ERR, 2001
eQCWWAN_ERR_SWIDCS_IOCTL_ERR, 2001
eQCWWAN_ERR_SWIDCS_START, 2001
eQCWWAN_ERR_SWIIM_CORRUPTED_FW_IMAGE, 2001
eQCWWAN_ERR_SWIIM_END, 2001
eQCWWAN_ERR_SWIIM_FILE_NOT_FOUND, 2001
eQCWWAN_ERR_SWIIM_FIRMWARE_NOT_DOWNLOADED, 2001
eQCWWAN_ERR_SWIIM_FW_ENTER_DOWNLOAD_MODE, 2001
eQCWWAN_ERR_SWIIM_FW_FLASH_COMPLETE, 2001
eQCWWAN_ERR_SWIIM_FW_INVALID_SLOT_INDEX, 2001
eQCWWAN_ERR_SWIIM_FW_PREFERENCE_MISMATCH, 2001
eQCWWAN_ERR_SWIIM_FW_SAME_AS_CURRENT_ACTIVE_IMAGE, 2001
eQCWWAN_ERR_SWIIM_FW_UPDATE_FAIL, 2001
eQCWWAN_ERR_SWIIM_FW_UPDATE_SUCCESS, 2001
eQCWWAN_ERR_SWIIM_FW_WAIT_FOR_REBOOT, 2001
eQCWWAN_ERR_SWIIM_INVALID_CRASH_STATE, 2001
eQCWWAN_ERR_SWIIM_INVALID_PATH, 2001
eQCWWAN_ERR_SWIIM_OPENING_DIR, 2001
eQCWWAN_ERR_SWIIM_OPENING_FILE, 2001
eQCWWAN_ERR_SWIIM_START, 2001
eQCWWAN_ERR_SWISM_END, 2001
eQCWWAN_ERR_SWISMS_BEARER_DATA_NOT_FOUND, 2001
eQCWWAN_ERR_SWISMS_MSG_CORRUPTED, 2001
eQCWWAN_ERR_SWISMS_MSG_LEN_TOO_LONG, 2001
eQCWWAN_ERR_SWISMS_SMSC_NUM_CORRUPTED, 2001
eQCWWAN_ERR_SWISMS_START, 2001
eWDS_ERR_PROFILE_REG_3GPP2_ERR_INVALID_IDENT_FOR_PROFILE, 2002
eWDS_ERR_PROFILE_REG_3GPP_ACCESS_ERR, 2002
eWDS_ERR_PROFILE_REG_3GPP_CONTEXT_NOT_DEFINED, 2002

eWDS_ERR_PROFILE_REG_3GPP_ERR_OUT-_OF_PROFILES, 2002
eWDS_ERR_PROFILE_REG_3GPP_INVAL_PR-OFILE_FAMILY, 2002
eWDS_ERR_PROFILE_REG_3GPP_READ_ON-LY_FLAG_SET, 2002
eWDS_ERR_PROFILE_REG_3GPP_VALID_FLA-G_NOT_SET, 2002
eWDS_ERR_PROFILE_REG_END, 2002
eWDS_ERR_PROFILE_REG_INVAL_PROFILE_-FAMILY, 2002
eWDS_ERR_PROFILE_REG_RESULT_ERR_IN-VAL, 2002
eWDS_ERR_PROFILE_REG_RESULT_ERR_IN-VAL_HNDL, 2002
eWDS_ERR_PROFILE_REG_RESULT_ERR_IN-VAL_IDENT, 2002
eWDS_ERR_PROFILE_REG_RESULT_ERR_IN-VAL_OP, 2002
eWDS_ERR_PROFILE_REG_RESULT_ERR_IN-VAL_PROFILE_NUM, 2002
eWDS_ERR_PROFILE_REG_RESULT_ERR_IN-VAL_PROFILE_TYPE, 2002
eWDS_ERR_PROFILE_REG_RESULT_ERR_IN-VAL_SUBS_ID, 2002
eWDS_ERR_PROFILE_REG_RESULT_ERR_LE-N_INVALID, 2002
eWDS_ERR_PROFILE_REG_RESULT_ERR_LI-B_NOT_INITED, 2002
eWDS_ERR_PROFILE_REG_RESULT_FAIL, 2002
eWDS_ERR_PROFILE_REG_RESULT_LIST_E-ND, 2002
qmerrno.h, 1995
eQCWWANError, 1997
qm_wds_ds_profile_extended_err_codes, 2001
QmiCbkCatEventStatusReportInd, 779
CCETlv, 779
event_Index, 779
QmiCbkLocBestAvailPosInd, 779
pAltitudeWrtEllipsoid, 784
pAltitudeWrtMeanSeaLevel, 784
pGpsTime, 784
pHeading, 784
pHeadingUnc, 784
pHorCirConf, 784
pHorEllpConf, 784
pHorReliability, 784
pHorUncCircular, 784
pHorUncEllipseOrientAzimuth, 784
pHorUncEllipseSemiMajor, 784
pHorUncEllipseSemiMinor, 784
pLatitude, 785
pLongitude, 785
pMagneticDeviation, 785
pPrecisionDilution, 785
pSensorDataUsage, 785
pSpeedHorizontal, 785
pSpeedUnc, 785
pSpeedVertical, 785
pSpeedVerticalUnc, 785
pSvUsedforFix, 785
pTechnologyMask, 785
pTimeSrc, 785
pTimeUnc, 785
pTimestampUtc, 785
pVertConfidence, 785
pVertReliability, 785
pVertUnc, 785
pXid, 785
status, 785
QmiCbkLocCradleMountInd, 785
cradleMountConfigStatus, 786
QmiCbkLocEngineStateInd, 786
engineState, 786
QmiCbkLocEventTimeSyncInd, 786
timeSyncRefCounter, 787
QmiCbkLocGetServerInd, 787
pIPv4AddrInfo, 788
pIPv6AddrInfo, 788
pURL, 788
serverStatus, 788
serverType, 788
QmiCbkLocInjectPositionInd, 788
status, 789
QmiCbkLocInjectSensorDataInd, 789
injectSensorDataStatus, 790
pAccelSamplesAccepted, 790
pAccelTempSamplesAccepted, 790
pGyroSamplesAccepted, 790
pGyroTempSamplesAccepted, 790
pOpaqueIdentifier, 790
QmiCbkLocInjectTimeInd, 790
injectTimeSyncStatus, 791
QmiCbkLocInjectUTCTimeInd, 791
status, 792
QmiCbkLocPositionReportInd, 792
pAltitudeAssumed, 796
pAltitudeWrtEllipsoid, 796
pAltitudeWrtMeanSeaLevel, 796
pFixId, 796
pGpsTime, 796
pHeading, 796
pHeadingUnc, 797
pHorConfidence, 797
pHorReliability, 797
pHorUncCircular, 797
pHorUncEllipseOrientAzimuth, 797
pHorUncEllipseSemiMajor, 797
pHorUncEllipseSemiMinor, 797
pLatitude, 797
pLeapSeconds, 797
pLongitude, 797
pMagneticDeviation, 797
pPrecisionDilution, 797
pSensorDataUsage, 797

pSpeedHorizontal, 797
pSpeedUnc, 797
pSpeedVertical, 797
pSvUsedforFix, 797
pTechnologyMask, 797
pTimeSrc, 797
pTimeUnc, 797
pTimestampUtc, 797
pVertConfidence, 797
pVertReliability, 797
pVertUnc, 797
sessionId, 797
sessionStatus, 797
QmiCbkLocSensorStreamingInd, 797
pAccelAcceptReady, 798
pAccelTempAcceptReady, 798
pGyroAcceptReady, 798
pGyroTempAcceptReady, 798
QmiCbkLocSetExtPowerConfigInd, 798
status, 799
QmiCbkNasLTECphyCaInfo, 799
sPhyCaAggPcellInfo, 800
sPhyCaAggScellDlBw, 800
sPhyCaAggScellIndType, 800
sPhyCaAggScellIndex, 800
sPhyCaAggScellInfo, 800
QmiCbkSwiOmaDmEventStatusReportInd, 800
SITlv, 800
QmiCbkSwiOmaDmEventStatusReportIndExt, 800
SITlv, 800
QmiCbkTmdMitiLvlRptInd, 800
currentMitigationLvl, 801
MitigationDevInfo, 801
QmiCbkWdsStatisticsIndState, 801
RxDropConutTlv, 801
RxOkByteCountTlv, 801
RxOkConutTlv, 801
TxDropConutTlv, 801
TxOkByteCountTlv, 801
TxOkConutTlv, 801
qmiDataBearerMasks
qaGobiApiWds.h, 1957
QmiNas3GppNetworkInfo, 802
pDesription, 803
pForbidden, 804
pInUse, 804
pMCC, 804
pMNC, 804
pPreferred, 804
pRoaming, 804
QmiNasGetRFBandInfoResp, 804
pInstancesSize, 804
pRFBandInfoElements, 804
results, 804
QmiNasPerformNetworkScanResp, 804
pInstanceSize, 804
pInstances, 804
results, 804
QmiProfileInfo
qaGobiApiWds.h, 1954
qmiSmsMessageList, 804
messageIndex, 805
messageTag, 805
QmiSwiAvmsEventReportBinaryUpdateSessionInfo-Resp
qaGobiApiSwiAvms.h, 1854
QmiSwiAvmsEventReportConfigResp
qaGobiApiSwiAvms.h, 1855
QmiSwiAvmsEventReportConnectionRequestResp
qaGobiApiSwiAvms.h, 1855
QmiSwiAvmsEventReportDataSessionStatusResp
qaGobiApiSwiAvms.h, 1856
QmiSwiAvmsEventReportHTTPStatusResp
qaGobiApiSwiAvms.h, 1856
QmiSwiAvmsEventReportNotificationResp
qaGobiApiSwiAvms.h, 1857
QmiSwiAvmsEventReportPackageIDResp
qaGobiApiSwiAvms.h, 1858
QmiSwiAvmsEventReportRegStatusResp
qaGobiApiSwiAvms.h, 1858
QmiSwiAvmsEventReportSessionTypeResp
qaGobiApiSwiAvms.h, 1858
QmiSwiAvmsEventReportWAMSParamChangeResp
qaGobiApiSwiAvms.h, 1859
QmiSwiAvmsSetSettingsAPNInfo
qaGobiApiSwiAvms.h, 1859
QmiSwiAvmsSetSettingsConnectionRetryTimers
qaGobiApiSwiAvms.h, 1859
QmiSwiAvmsSetSettingsPeriodInfo
qaGobiApiSwiAvms.h, 1859
QmiWDSDataBearerTechnology
qaGobiApiWds.h, 1955
qmiWDSDataBearerTechnology, 805
currentNetwork, 806
ratMask, 806
soMask, 806
QmiWDSDataBearers
qaGobiApiWds.h, 1954
QmiWdsIpAddressInfo, 806
pIPAddressV4, 807
pIPAddressV6, 807
pIPv6prefixlen, 807
qmiWdsRunTimeSettings, 807
pAPNName, 810
pAuthentication, 810
pDomainList, 810
pGPRSGrantedQoS, 810
pGWAddressV4, 810
pIMCNflag, 810
pIPAddressV4, 810
pIPFamilyPreference, 810
pIPV6AddrInfo, 810
pIPV6GWAddrInfo, 810
pMtu, 810
pPCSCFAddrPCO, 810
pPCSCFFQDNAddrList, 810

pPDPType, 810
pPrimaryDNSV4, 810
pPrimaryDNSV6, 810
pProfileID, 810
pProfileName, 810
pSecondaryDNSV4, 811
pSecondaryDNSV6, 811
pServerAddrList, 811
pSubnetMaskV4, 811
pTechnology, 811
pUMTSGrantedQoS, 811
pUsername, 811
qmifwinfo_s, 802
dev, 802
g, 802
s, 802
qmuxIPTable, 811
ipAddress, 811
muxID, 811
qos.h, 2002
pack_qos_SLQSQosGetNetworkStatus, 2004
pack_qos_SLQSQosSwiReadApnExtraParams, 2004
pack_qos_SLQSQosSwiReadDataStats, 2005
pack_qos_SLQSSetQosEventCallback, 2006
unpack_qos_SLQSQosGetNetworkStatus, 2006
unpack_qos_SLQSQosSwiReadApnExtraParams, 2007
unpack_qos_SLQSQosSwiReadDataStats, 2007
unpack_qos_SLQSSetQosEventCallback, 2008
unpack_qos_SLQSSetQosEventCallback_ind, 2008
unpack_qos_SLQSSetQosNWStatusCallback_ind, 2008
unpack_qos_SLQSSetQosPriEventCallback_ind, 2009
unpack_qos_SLQSSetQosStatusCallback_ind, 2010
qos_id
QosMap, 816
QosClassID, 811
gDlBitRate, 812
gUlBitRate, 812
maxDlBitRate, 812
maxUlBitRate, 812
QCI, 812
qosDeliveryOrder
LibPackUMTSQoS, 364
UMTSMinQoS, 1028
UMTSQoS, 1030
wds_UMTSMinQoS, 1411
QosEventInfo, 812
pDataBearer, 813
pPacketsCountRX, 813
pPacketsCountTX, 813
pTotalBytesRX, 813
pTotalBytesTX, 813
qosFlow
sQosStat, 928
unpack_qos_SLQSQosSwiReadDataStats_t, 1181
QosFlowInfo, 814
pBearerID, 814
pQFlowState, 814
pRxQFilter, 814
pRxQFlowGranted, 814
pTxQFilter, 814
pTxQFlowGranted, 814
unpack_qos_SLQSSetQosEventCallback_ind_t, 1182
QosFlowInfoState, 815
id, 815
isNewFlow, 815
state, 815
QosMap, 815
dscp, 816
qos_id, 816
state, 816
Quality of Service (QOS), 54

RAN
unpack_nas_GetServingNetwork_t, 1114
RAT
SlqsNas3GppNetworkRAT, 87
nas_QmiNas3GppNetworkRAT, 499
RATMask
CurrNetworkInfo, 205
currNetworkInfo, 206
wds_currNetworkInfo, 1396
REGISTER_EVENT
qaGobiApiCbk.h, 1595
REGISTER_SRV
qaGobiApiCbk.h, 1595
RFBandInfoElements, 823
activeBandClass, 824
activeChannel, 824
radioInterface, 824
unpack_nas_GetRFInfo_t, 1112
RFTlv
unpack_nas_SetEventReportInd_t, 1118
RMAutoConnect
pack_dms_SetCustFeature_t, 601
unpack_dms_GetCustFeature_t, 1039
RMSetTransferStatistics
qaGobiApiWds.h, 1969
ROAMING_INDEX
qaNasPerformNetworkScan.h, 1994
RPCause
SMSAsyncRawSend_s, 912
RPTlv
NASQmiCbkNasSystemSelPrefInd, 567
RRTlv
unpack_nas_SetEventReportInd_t, 1118
RSRPThresListLen
nas_RSRPThresh, 502
RSRPThresh, 827
RSRPThresh, 827
pRSRPThresList, 827

RSRPThresListLen, 827
RSRQThresListLen
nas_RSRQThresh, 503
RSRQThresh, 828
RSRQThresh, 828
pRSRQThresList, 828
RSRQThresListLen, 828
RSSIThresListLen
nas_RSSIThresh, 504
RSSIThresh, 829
RSSIThresh, 828
pRSSIThresList, 829
RSSIThresListLen, 829
RX_EC_IO
nas_NetworkStat1x, 481
NetworkStat1x, 582
RX_PWR
nas_NetworkStat1x, 481
nas_NetworkStatEVDO, 483
NetworkStat1x, 582
NetworkStatEVDO, 583
RXAGCList, 829
pRXAIG, 830
pRXComprSlope, 830
pRXComprThres, 830
pRXExpSlope, 830
pRXExpThres, 830
pRXStaticGain, 830
RXAVCList, 830
pAVRXAVCHeadroom, 830
pAVRXAVCSens, 831
RXChan
LTEInfo, 398
nas_LTEInfo, 462
rXDroppedCount
unpack_wds_GetPacketStatus_t, 1265
RXOKBytesCount
DUNCallInfoInd, 236
rXOKBytesLastCall
unpack_wds_GetPacketStatus_t, 1266
rXOkBytesCount
unpack_wds_GetPacketStatus_t, 1265
RXPCMIIRFltr, 832
pFlag, 833
pStage0Val, 833
pStage1Val, 833
pStage2Val, 833
pStage3Val, 833
pStage4Val, 833
pStageCnt, 833
rXPacketErrors
unpack_wds_GetPacketStatus_t, 1266
rXPacketOverflows
unpack_wds_GetPacketStatus_t, 1266
rXPacketSuccesses
unpack_wds_GetPacketStatus_t, 1266
radio
unpack_nas_GetSignalStrengths_t, 1116
radio_if
nasGetTxRxInfoReq, 545
pack_nas_SLQSNasGetTxRxInfo_t, 640
radioIf
ecioListElement, 238
errorRateListElement, 242
nas_ecioListElement, 442
nas_errorRateListElement, 443
nas_rsrqInformation, 503
nas_rxSignalStrengthListElement, 507
nasNetworkReject, 555
rsrqInformation, 828
rxSignalStrengthListElement, 835
RadioIfaces
unpack_dms_GetDeviceCap_t, 1040
unpack_dms_GetDeviceCapabilities_t, 1042
unpack_nas_GetServingNetwork_t, 1114
RadioIfacesSize
unpack_dms_GetDeviceCap_t, 1040
unpack_nas_GetServingNetwork_t, 1114
radioIfacesSize
unpack_dms_GetDeviceCapabilities_t, 1042
radioInterface
nas_RFInfoTlv, 501
nas_roamIndList, 502
nas_servSystem, 510
nas_SignalStrengthTlv, 511
nas_timeInfo, 522
RFBandInfoElements, 824
roamIndList, 826
servSystem, 848
timeInfo, 966
radioInterfaceList
NASServingSystemInfo, 569
ServingSystemInfo, 846
radioInterfaceNo
NASServingSystemInfo, 569
ServingSystemInfo, 847
radioInterfaceSize
nas_RFInfoTlv, 501
range
Port, 757
unpack_qos_Port_t, 1176
RankIndicatorInd, 816
Count1, 816
Count2, 816
rat
CSGID, 197
MNRInfo, 417
nas_CSGID, 434
nas_MNRInfo, 478
ratMask
dataBearerTechnology, 216
qmiWDSDataBearerTechnology, 806
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1283
ratValue
DataBearerTech, 214

rawHorConfidence
pack_loc_SLQSLOCInjectPosition_t, 623
rawHorUncCircular
pack_loc_SLQSLOCInjectPosition_t, 623
rawLen
fileAttributes, 246
uim_fileAttributes, 988
rawValue
fileAttributes, 246
uim_fileAttributes, 988
rcv4
ssdatasession_params, 932
rcv6
ssdatasession_params, 932
readResult, 816
content, 817
contentLen, 817
readTransparent
pack_uim_ReadTransparent_t, 683
UIMReadTransparentReq, 1010
readTransparentInfo, 817
length, 817
offset, 817
Reason
pack_voice_SLQSVoiceGetCallForwardingStatus-_t, 701
pack_voice_SLQSVoiceSetCallBarringPassword_t, 707
voiceGetCallFWReq, 1348
voiceSetCallBarringPwdInfo, 1371
reason
ccSUPSType, 161
pack_voice_SLQSVoiceGetCallBarring_t, 700
pack_voice_SLQSVoiceSetSUPSService_t, 711
redirNumInfo, 819
unpack_qos_SLQSSetQosStatusCallback_ind_t, 1184
voice_ccSUPSType, 1314
voiceGetCallBarringReq, 1346
voiceSetSUPSServiceReq, 1378
receiptAction
sms_routeEntry, 906
smsRouteEntry, 924
receivedBytes
omaDmFotaTlvExt, 594
reconfigReqd
_packetSrvStatus, 70
unpack_wds_SLQSSetPacketSrvStatusCallback_-t, 1281
reconfiguration_required
slqsSessionStateInfo, 890
recordCount
fileAttributes, 246
uim_fileAttributes, 988
recordSize
fileAttributes, 246
uim_fileAttributes, 988
redirNumInfo, 817
numLen, 819
numPlan, 819
numType, 819
number, 819
PI, 819
reason, 819
SI, 819
RedirPartyNum
arrRedirPartyNum, 133
voice_arrRedirPartyNum, 1305
refData
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 1073
refString
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 1073
ReferenceID
CatAlPhaIdentifierTlv, 158
CatEventIDDataTlv, 159
refpn
CDMAInfo, 164
nas_CDMAInfo, 426
refreshComplete
pack_uim_SLQSUIMRefreshComplete_t, 689
UIMRefreshCompleteReq, 1012
RefreshMode
CatRefreshTlv, 160
RefreshStage
CatRefreshTlv, 160
regAction
nasInitNetworkReg, 550
pack_nas_SLQSInitiateNetworkRegistration_t, 635
RegForeignNID
unpack_nas_GetCDMANetworkParameters_t, 1106
RegForeignSID
unpack_nas_GetCDMANetworkParameters_t, 1106
RegHomeSID
unpack_nas_GetCDMANetworkParameters_t, 1106
regInd
_transLayerInfoNotification, 112
regPrd
AddCDMASysInfo, 116
nas_AddCDMASysInfo, 421
regRefresh
pack_uim_SLQSUIMRefreshRegister_t, 690
UIMRefreshRegisterReq, 1015
regRejectInfoValid
GSMSysInfo, 299
LTESysInfo, 413
nas_GSMSysInfo, 451
nas_LTESysInfo, 475
nas_WCDMASysInfo, 534
WCDMASysInfo, 1394
regState
nas_servSystem, 510

servSystem, 848
RegStatus
_SwiAvmsEventInd, 100
Region
fwinfo_s, 253
registerFlag
registerRefresh, 820
uim_registerRefresh, 994
registerRefresh, 819
arrfileInfo, 820
numFiles, 820
registerFlag, 820
voteForInit, 820
RegistrationState
unpack_nas_GetServingNetwork_t, 1114
registrationState
NASServingSystemInfo, 569
ServingSystemInfo, 847
rejCause
GSMSysInfo, 299
LTESysInfo, 413
nas_GSMSysInfo, 451
nas_LTESysInfo, 475
nas_WCDMASysInfo, 534
WCDMASysInfo, 1394
rejectCause
nas_RejectReasonTlv, 500
nasNetworkReject, 555
rejectSrvDomain
GSMSysInfo, 299
LTESysInfo, 413
nas_GSMSysInfo, 451
nas_LTESysInfo, 475
nas_WCDMASysInfo, 534
WCDMASysInfo, 1394
reliabilityClass
GPRSQoS, 288
GPRSRequestedQoS, 289
LibPackGPRSRequestedQoS, 336
wds_GPRSQoS, 1403
remPartyNumber
remotePartyNum, 822
voice_remotePartyNum, 1330
remainingRetries, 820
unblockLeft, 820
verifyLeft, 820
Remote Management Service (RMS), 45
RemotePartyName
getAllCallRmtPtyName, 257
voice_getAllCallRmtPtyName, 1323
remotePartyName, 820
callerName, 821
codingScheme, 821
nameLen, 821
namePI, 821
RemotePartyNum
getAllCallRmtPtyNum, 257
voice_getAllCallRmtPtyNum, 1324
remotePartyNum, 821
numLen, 822
presentationInd, 822
remPartyNumber, 822
ReqFieldsList, 822
requestFields, 822
requestFieldsLen, 822
requestFields
ReqFieldsList, 822
requestFieldsLen
ReqFieldsList, 822
resBerRatio
LibPackUMTSQoS, 365
UMTSMinQoS, 1028
UMTSQoS, 1030
wds_UMTSMinQoS, 1411
ResCode
FirmwareUpdatStat, 249
GetAudioVolTLBConfigResp, 263
SetAudioVolTLBConfigResp, 854
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 1073
reserved
omaDmFotaTlvExt, 594
resetInfoInd
pack_dms_SLQSDmsSwiIndicationRegister_t, 605
ResetInfoNotification
qaGobiApiCbk.h, 1600
ResetPDSData
qaGobiApiPds.h, 1803
ResetToFactoryDefaults
qaGobiApiDms.h, 1698
RespFieldsList, 823
responseFields, 823
responseFieldsLen, 823
responseFields
RespFieldsList, 823
responseFieldsLen
RespFieldsList, 823
results
QmiNasGetRFBandInfoResp, 804
QmiNasPerformNetworkScanResp, 804
RetryCount
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1209
revPolarity
lineCtrlInfo, 366
revTunneling
unpack_wds_GetMobileIPProfile_t, 1262
ReverseMac
nas_protocolSubtypeElement, 496
protocolSubtypeElement, 773
RfSarState
pack_sar_SLQSSetRfSarState_t, 667
RmTrasnferStaticsReq
pack_wds_RMSetTransferStatistics_t, 716
rmTrasnferStaticsReq, 824
bResetStatistics, 825

ulMask, 825
rms.h, 2010
pack_rms_GetSMSWake, 2011
pack_rms_SetSMSWake, 2011
unpack_rms_GetSMSWake, 2011
unpack_rms_SetSMSWake, 2012
RmtPtyNum
arrRemotePartyNum, 134
voice_arrRemotePartyNum, 1306
roamIndList, 825
numInstances, 826
radioInterface, 826
roamIndicator, 826
roamIndicator
nas_roamIndList, 502
roamIndList, 826
RoamIndicatorVal
unpack_nas_SLQSGetServingSystem_t, 1130
roamIndicatorVal
qaQmiServingSystemParam, 779
roamOrigVoiceSO
prefVoiceSO, 760
voice_prefVoiceSO, 1329
RoamPref
NASRoamPreferenceTlv, 567
roamStatus
nas_sysInfoCommon, 517
sysInfoCommon, 955
roamStatusValid
nas_sysInfoCommon, 517
sysInfoCommon, 955
roamTimer, 826
namID, 827
roamTimerValue, 827
roamTimerValue
roamTimer, 827
voice_roamTimer, 1331
Roaming
nas_QmiNas3GppNetworkInfo, 498
SlqsNas3GppNetworkInfo, 886
unpack_nas_GetCDMANetworkParameters_t, 1106
unpack_nas_GetServingNetwork_t, 1114
roaming
unpack_nas_SetRoamingIndicatorCallback_ind_t, 1119
roaming_ind
RoamingInfo, 826
RoamingIndicatorList
qaQmiServingSystemParam, 779
unpack_nas_SLQSGetServingSystem_t, 1130
RoamingInfo, 826
roaming_ind, 826
TlvPresent, 826
routeList
sms_setRoutesReq, 910
smsSetRoutesReq, 925
routeStorage
sms_routeEntry, 906
smsRouteEntry, 924
rptPosData
pack_pds_SetEventReportCallback_t, 655
rptRate
LTESigRptCfg, 406
LTESigRptConfig, 407
nas_LTESigRptCfg, 470
nas_LTESigRptConfig, 471
rscp
nas_rxInfo, 505
nas_UMTSInfo, 525
rxInfo, 832
TDSCDMASigInfoExt, 958
tdscdmaSigInfoExt, 959
UMTSInfo, 1024
rsrp
cellParams, 179
LTESSInfo, 410
lteSSInfo, 410
nas_cellParams, 431
nas_rxInfo, 505
nas_RxSigInfo, 506
nas_umtsLTENbrCell, 527
rxInfo, 832
RxSigInfo, 834
umtsLTENbrCell, 1025
rsrplevel
lteRsrpinformation, 404
nas_lteRsrpinformation, 468
rsrq
cellParams, 179
LTESSInfo, 410
lteSSInfo, 410
nas_cellParams, 431
nas_rsrqInformation, 503
nas_SccRxInfo, 508
nas_umtsLTENbrCell, 527
rsrqInformation, 828
SccRxInfo, 839
umtsLTENbrCell, 1025
rsrqDelta
nas_SLQSSignalStrengthsIndReq, 513
SLQSSignalStrengthsIndReq, 894
rsrqInfo
nas_SLQSSignalStrengthsInformation, 514
slqsSignalStrengthInfo, 892
SLQSSignalStrengthsInformation, 896
unpack_nas_SLQSGetSignalStrength_t, 1132
rsrqInformation, 827
radioIf, 828
rsrq, 828
rssi
CDMASSInfo, 173
cdmaSSInfo, 173
cellParams, 179
gsmCellInfo, 294
HDRSSInfo, 306

hdrSSInfo, 307
LTESSInfo, 410
lteSSInfo, 410
nas_cellParams, 431
nas_gsmCellInfo, 447
TDSCDMASigInfoExt, 958
tdscdmaSigInfoExt, 959
unpack_nas_GetSignalStrengths_t, 1116
rts
WdsRunTimeSettings, 1426
rx_bytes
NetStats, 578
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1283
rx_errors
NetStats, 578
rx_overflows
NetStats, 579
rx_packets
NetStats, 579
rx_pkts
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1283
rxChainIndex
nas_RxSigInfo, 506
RxSigInfo, 834
RxDropConutTlv
QmiCbkWdsStatisticsIndState, 801
unpack_RMTransferStatistics_ind_t, 1195
rxInfo, 831
ecio, 831
isRadioTuned, 831
phase, 831
rscp, 832
rsrp, 832
rxPower, 832
rxLev
GERANInfo, 255
nas_GERANInfo, 445
rxOKBytesCount
unpack_wds_SLQSGetDUNCallInfo_t, 1275
RxOkByteCountTlv
QmiCbkWdsStatisticsIndState, 801
unpack_RMTransferStatistics_ind_t, 1195
RxOkConutTlv
QmiCbkWdsStatisticsIndState, 801
unpack_RMTransferStatistics_ind_t, 1195
rxPower
nas_rxInfo, 505
nas_RxSigInfo, 506
rxInfo, 832
RxSigInfo, 834
RxQFilter
unpack_qos_QosFlowInfo_t, 1177
RxQFlowGranted
unpack_qos_QosFlowInfo_t, 1177
RxSigInfo, 833
isRadioTuned, 834
rsrp, 834
rxChainIndex, 834
rxPower, 834
rxSignalStrength
nas_rxSignalStrengthListElement, 507
rxSignalStrengthListElement, 835
rxSignalStrengthDelta
nas_SLQSSignalStrengthsIndReq, 513
SLQSSignalStrengthsIndReq, 894
rxSignalStrengthInfo
nas_SLQSSignalStrengthsInformation, 514
SLQSSignalStrengthsInformation, 896
rxSignalStrengthList
slqsSignalStrengthInfo, 892
unpack_nas_SLQSGetSignalStrength_t, 1132
rxSignalStrengthListElement, 834
radioIf, 835
rxSignalStrength, 835
rxSignalStrengthListLen
slqsSignalStrengthInfo, 892
unpack_nas_SLQSGetSignalStrength_t, 1132

s
qmifwinfo_s, 802
SMS_EVENT_ETWS
qaGobiApiCbk.h, 1635
SMS_EVENT_ETWS_PLMN
qaGobiApiCbk.h, 1635
SMS_EVENT_MESSAGE_MODE
qaGobiApiCbk.h, 1635
SMS_EVENT_MT_MESSAGE
qaGobiApiCbk.h, 1635
SMS_EVENT_SMS_ON_IMS
qaGobiApiCbk.h, 1635
SMS_EVENT_SMSC_ADDRESS
qaGobiApiCbk.h, 1635
SMS_EVENT_TRANSFER_ROUTE_MT_MESSAGE
qaGobiApiCbk.h, 1635
SYS_SRV_DOMAIN_CAMPED
qaGobiApiCbk.h, 1635
SYS_SRV_DOMAIN_CS_ONLY
qaGobiApiCbk.h, 1635
SYS_SRV_DOMAIN_CS_PS
qaGobiApiCbk.h, 1635
SYS_SRV_DOMAIN_NO_SRV
qaGobiApiCbk.h, 1635
SYS_SRV_DOMAIN_PS_ONLY
qaGobiApiCbk.h, 1635
sApnExtraParams, 835
ambr_dl, 836
ambr_dl_ext, 836
ambr_dl_ext2, 836
ambr_ul, 836
ambr_ul_ext, 836
ambr_ul_ext2, 836
apnId, 836
SCI
unpack_nas_GetCDMANetworkParameters_t, 1106

SCM
unpack_nas_GetCDMANetworkParameters_t, 1107
SDPTlv
NASQmiCbkNasSystemSelPrefInd, 567
SDU_HDR_LEN
common.h, 1433
SECOND_INSTANCE
qaGobiApiCbk.h, 1595
sGetDeviceSeriesResult, 874
eDevice, 874
uResult, 874
SHORT
SwiDataTypes.h, 2035
SI
calledPartyInfo, 146
callFWExtInfo, 150
callingPartyInfo, 154
redirNumInfo, 819
voice_callFWExtInfo, 1311
SITlv
QmiCbkSwiOmaDmEventStatusReportInd, 800
QmiCbkSwiOmaDmEventStatusReportIndExt, 800
sIntraSearch
LTEInfoIntrafreq, 400
nas_LTEInfoIntrafreq, 465
SLQSAVMSGetSessionInfo
qaGobiApiSwiAvms.h, 1861
SLQSAVMSGetSettings
qaGobiApiSwiAvms.h, 1861
SLQSAVMSSelection
qaGobiApiSwiAvms.h, 1861
SLQSAVMSSetEventReport
qaGobiApiSwiAvms.h, 1862
SLQSAVMSSetSettings
qaGobiApiSwiAvms.h, 1862
SLQSAVMSSetSettingsIgnoreAutoReboot
qaGobiApiSwiAvms.h, 1863
SLQSAVMSSetSettingsReq
qaGobiApiSwiAvms.h, 1859
SLQSAVMSSettings
qaGobiApiSwiAvms.h, 1860
SLQSAVMSSettingsAPNInfo
qaGobiApiSwiAvms.h, 1860
SLQSAVMSSettingsConnectionRetryTimers
qaGobiApiSwiAvms.h, 1860
SLQSAVMSSettingsPeriodsInfo
qaGobiApiSwiAvms.h, 1860
SLQSAVMSStartSession
qaGobiApiSwiAvms.h, 1863
SLQSAVMSStopSession
qaGobiApiSwiAvms.h, 1864
SLQSAutoConnect
qaGobiApiWds.h, 1978
SLQSCDMADecodeMTTextMsg
qaGobiApiSms.h, 1831
SLQSCDMADecodeMTTextMsgExt
qaGobiApiSms.h, 1831
SLQSCDMAEncodeMOTextMsg
qaGobiApiSms.h, 1832
SLQSConfigSigInfo
qaGobiApiNas.h, 1778
SLQSCreateProfile
qaGobiApiWds.h, 1978
SLQSDeleteProfile
qaGobiApiWds.h, 1979
SLQSDeleteProfileParams, 882
profileIndex, 883
profileType, 883
SLQSDeleteSMS
qaGobiApiSms.h, 1832
SLQSDmsSwiGetResetInfo
qaGobiApiDms.h, 1699
SLQSDmsSwiIndicationRegister
qaGobiApiDms.h, 1699
SLQSDownloadFirmwareToSlot
qaGobiApiFms.h, 1721
SLQSFWINFO_SKU_SZ
qaGobiApiFms.h, 1715
SLQSGet3GPPConfigItem
qaGobiApiWds.h, 1980
SLQSGetAGPSConfig
qaGobiApiPds.h, 1807
SLQSGetAudioPathConfig
qaGobiApiAudio.h, 1581
SLQSGetAudioProfile
qaGobiApiAudio.h, 1581
SLQSGetAudioVolTLBConfig
qaGobiApiAudio.h, 1582
SLQSGetBandCapabilities
qaGobiApiDms.h, 1700
SLQSGetBandCapability
qaGobiApiDms.h, 1700
SLQSGetBootVersionNumber
qaGobiApiFms.h, 1722
SLQSGetByteTotals
qaGobiApiWds.h, 1980
SLQSGetConnectionRate
qaGobiApiWds.h, 1980
SLQSGetCurrDataSystemStat
qaGobiApiWds.h, 1981
SLQSGetCurrentChannelRate
qaGobiApiWds.h, 1981
SLQSGetCurrentPRLInfo
qaGobiApiDms.h, 1702
SLQSGetCustFeatures
qaGobiApiDms.h, 1702
SLQSGetCustFeaturesV2
qaGobiApiDms.h, 1702
SLQSGetDUNCallInfo
qaGobiApiWds.h, 1983
SLQSGetDataBearerTechnology
qaGobiApiWds.h, 1982
SLQSGetDataBearerTechnologyExt
qaGobiApiWds.h, 1982
SLQSGetDeviceMode

qaGobiApiDcs.h, 1677
SLQSGetERIFile
qaGobiApiDms.h, 1703
SLQSGetErrorRate
qaGobiApiNas.h, 1779
SLQSGetFirmwareInfo
qaGobiApiFms.h, 1723
SLQSGetGPSStateInfo
qaGobiApiPds.h, 1808
SLQSGetIMSARegStatus
qaGobiApiImsa.h, 1737
SLQSGetIMSAServiceStatus
qaGobiApiImsa.h, 1738
SLQSGetIMSASupportedFields
qaGobiApiImsa.h, 1738
SLQSGetIMSASupportedMsg
qaGobiApiImsa.h, 1739
SLQSGetIMSSMSConfig
qaGobiApiIms.h, 1731
SLQSGetIMSUserConfig
qaGobiApiIms.h, 1732
SLQSGetIMSVoIPConfig
qaGobiApiIms.h, 1732
SLQSGetImageInfo
qaGobiApiFms.h, 1723
SLQSGetImageInfo_9x15
qaGobiApiFms.h, 1724
SLQSGetImageInfoMC77xx
qaGobiApiFms.h, 1724
SLQSGetImageInfoMC83xx
qaGobiApiFms.h, 1725
SLQSGetIndicationRegister
qaGobiApiSms.h, 1833
SLQSGetM2MAVMute
qaGobiApiSwiAudio.h, 1848
SLQSGetM2MAudioProfile
qaGobiApiSwiAudio.h, 1847
SLQSGetM2MAudioVolume
qaGobiApiSwiAudio.h, 1848
SLQSGetM2MSpkrGain
qaGobiApiSwiAudio.h, 1848
SLQSGetMessageWaiting
qaGobiApiSms.h, 1834
SLQSGetNetStatistic
qaGobiApiDcs.h, 1678
SLQSGetNetworkTime
qaGobiApiNas.h, 1779
SLQSGetOperatorNameData
qaGobiApiNas.h, 1779
SLQSGetPLMNName
qaGobiApiNas.h, 1780
SLQSGetPacketStatistics
qaGobiApiWds.h, 1983
SLQSGetPidof
qaGobiApiSwi.h, 1845
SLQSGetProfile
qaGobiApiWds.h, 1984
SLQSGetProfileSettings
qaGobiApiWds.h, 1985
SLQSGetRegMgrConfig
qaGobiApiIms.h, 1733
SLQSGetRfSarState
qaGobiApiSar.h, 1821
SLQSGetRuntimeSettings
qaGobiApiWds.h, 1986
SLQSGetSIPConfig
qaGobiApiIms.h, 1733
SLQSGetSMS
qaGobiApiSms.h, 1834
SLQSGetSMSList
qaGobiApiSms.h, 1836
SLQSGetSdkVersion
qaGobiApiSwi.h, 1846
SLQSGetSerialNumbers
qaGobiApiDms.h, 1703
SLQSGetServingSystem
qaGobiApiNas.h, 1780
SLQSGetSessionState
qaGobiApiWds.h, 1986
SLQSGetSignalStrength
qaGobiApiNas.h, 1781
SLQSGetSmsBroadcastConfig
qaGobiApiSms.h, 1835
SLQSGetSysSelectionPref
qaGobiApiNas.h, 1781
SLQSGetTransLayerInfo
qaGobiApiSms.h, 1836
SLQSGetTransNWRegInfo
qaGobiApiSms.h, 1837
SLQSGetUsbPortNames
qaGobiApiDcs.h, 1678
SLQSGetValidFwPriCombinations
qaGobiApiFms.h, 1725
SLQSImsConfigIndicationRegister
qaGobiApiIms.h, 1734
SLQSInitiateNetworkRegistration
qaGobiApiNas.h, 1782
SLQSIsSpkgFormatRequired
qaGobiApiFms.h, 1726
SLQSKillSDKProcess
qaGobiApiDcs.h, 1679
SLQSLOCDelAssData
qaGobiApiLoc.h, 1741
SLQSLOCEventRegister
qaGobiApiLoc.h, 1741
SLQSLOCGetBestAvailPos
qaGobiApiLoc.h, 1742
SLQSLOCGetServer
qaGobiApiLoc.h, 1742
SLQSLOCInjectPosition
qaGobiApiLoc.h, 1743
SLQSLOCInjectSensorData
qaGobiApiLoc.h, 1743
SLQSLOCInjectUTCTime
qaGobiApiLoc.h, 1743
SLQSLOCSetCradleMountConfig

qaGobiApiLoc.h, 1744
SLQSLOCSetExtPowerState
qaGobiApiLoc.h, 1744
SLQSLOCSetOpMode
qaGobiApiLoc.h, 1745
SLQSLOCStart
qaGobiApiLoc.h, 1745
SLQSLOCStop
qaGobiApiLoc.h, 1746
SLQSModifyProfile
qaGobiApiWds.h, 1987
SLQSModifySMSStatus
qaGobiApiSms.h, 1837
SLQSNASGetLTECPHYCaInfo
qaGobiApiNas.h, 1784
SLQSNASSwiGetChannelLock
qaGobiApiNas.h, 1787
SLQSNASSwiSetChannelLock
qaGobiApiNas.h, 1789
SLQSNasConfigSigInfo2
qaGobiApiNas.h, 1782
SLQSNasGet3GPP2Subscription
qaGobiApiNas.h, 1783
SLQSNasGetCellLocationInfo
qaGobiApiNas.h, 1783
SLQSNasGetHDRColorCode
qaGobiApiNas.h, 1784
SLQSNasGetSigInfo
qaGobiApiNas.h, 1784
SLQSNasGetSysInfo
qaGobiApiNas.h, 1785
SLQSNasGetTxRxInfo
qaGobiApiNas.h, 1785
SLQSNasIndicationRegister
qaGobiApiNas.h, 1786
SLQSNasIndicationRegisterExt
qaGobiApiNas.h, 1787
SLQSNasIndicationRegisterLTECphyCa
qaGobiApiNas.h, 1787
SLQSNasNetworkRejectCallback
qaGobiApiCbk.h, 1651
SLQSNasNetworkTimeCallBack
qaGobiApiCbk.h, 1651
SLQSNasSigInfo2CallBack
qaGobiApiCbk.h, 1651
SLQSNasSigInfoCallBack
qaGobiApiCbk.h, 1652
SLQSNasSwiIndicationRegister
qaGobiApiNas.h, 1788
SLQSNasSwiModemStatus
qaGobiApiNas.h, 1788
SLQSNasSwiOTAMessageCallback
qaGobiApiCbk.h, 1652
SLQSNasSysInfoCallBack
qaGobiApiCbk.h, 1653
SLQSNasTimerCallback
qaGobiApiCbk.h, 1653
SLQSOMADMCancelSession
qaGobiApiSwiOmadms.h, 1870
SLQSOMADMGetSessionInfo
qaGobiApiSwiOmadms.h, 1870
SLQSOMADMGetSettings
qaGobiApiSwiOmadms.h, 1871
SLQSOMADMGetSettings2
qaGobiApiSwiOmadms.h, 1871
SLQSOMADMSendSelection
qaGobiApiSwiOmadms.h, 1872
SLQSOMADMSendSelection2
qaGobiApiSwiOmadms.h, 1872
SLQSOMADMSessionInfo
qaGobiApiSwiOmadms.h, 1865
SLQSOMADMSetSettings
qaGobiApiSwiOmadms.h, 1873
SLQSOMADMSetSettings2
qaGobiApiSwiOmadms.h, 1874
SLQSOMADMSetSettings3
qaGobiApiSwiOmadms.h, 1874
SLQSOMADMSettings
qaGobiApiSwiOmadms.h, 1867
SLQSOMADMSettingsReqParams
qaGobiApiSwiOmadms.h, 1868
SLQSOMADMSettingsReqParams3
qaGobiApiSwiOmadms.h, 1869
SLQSOMADMStartSession
qaGobiApiSwiOmadms.h, 1874
SLQSOMADMStartSession2
qaGobiApiSwiOmadms.h, 1875
SLQSOriginateUSSD
qaGobiApiVoice.h, 1934
SLQSPDSDeterminePosition
qaGobiApiPds.h, 1808
SLQSPDSInjectAbsoluteTimeReference
qaGobiApiPds.h, 1809
SLQSPDSInjectPositionData
qaGobiApiPds.h, 1809
SLQSPerformNetworkScan
qaGobiApiNas.h, 1789
SLQSQosGetFlowStatus
qaGobiApiQos.h, 1813
SLQSQosGetGranted
qaGobiApiQos.h, 1813
SLQSQosGetNWProf
qaGobiApiQos.h, 1814
SLQSQosGetNetworkStatus
qaGobiApiQos.h, 1814
SLQSQosModify
qaGobiApiQos.h, 1815
SLQSQosRel
qaGobiApiQos.h, 1815
SLQSQosReq
qaGobiApiQos.h, 1816
SLQSQosReset
qaGobiApiQos.h, 1816
SLQSQosResume
qaGobiApiQos.h, 1817
SLQSQosSuspend

qaGobiApiQos.h, 1817
SLQSQosSwiReadApnExtraParams
qaGobiApiQos.h, 1817
SLQSQosSwiReadDataStats
qaGobiApiQos.h, 1818
SLQSRegisterIMSAIndication
qaGobiApiImsa.h, 1739
SLQSResetPacketStatics
qaGobiApiWds.h, 1987
SLQSSGetDHCPv4ClientConfig
qaGobiApiWds.h, 1989
SLQSSGetLoopback
qaGobiApiWds.h, 1990
SLQSSSTlv
unpack_nas_SetEventReportInd_t, 1118
SLQSSSetDHCPv4ClientConfig
qaGobiApiWds.h, 1990
SLQSSSetLoopback
qaGobiApiWds.h, 1990
SLQSSendAsyncSMS
qaGobiApiSms.h, 1838
SLQSSendLongSMS
qaGobiApiSms.h, 1838
SLQSSendRawQMI
qaGobiApiSwi.h, 1846
SLQSSendSMS
qaGobiApiSms.h, 1839
SLQSSet3GPPConfigItem
qaGobiApiWds.h, 1987
SLQSSetAGPSConfig
qaGobiApiPds.h, 1810
SLQSSetAudioPathConfig
qaGobiApiAudio.h, 1582
SLQSSetAudioProfile
qaGobiApiAudio.h, 1583
SLQSSetAudioVolTLBConfig
qaGobiApiAudio.h, 1583
SLQSSetBandPreference
qaGobiApiNas.h, 1789
SLQSSetBandPreferenceCbk
qaGobiApiCbk.h, 1653
SLQSSetCrashStateCheckIgnore
qaGobiApiFms.h, 1726
SLQSSetCustFeatures
qaGobiApiDms.h, 1704
SLQSSetCustFeaturesV2
qaGobiApiDms.h, 1704
SLQSSetDHCPv4ClientLeaseStatusCallback
qaGobiApiCbk.h, 1654
SLQSSetDUNCallInfoCallback
qaGobiApiCbk.h, 1655
SLQSSetDataSystemStatusCallback
qaGobiApiCbk.h, 1654
SLQSSetIMSAPdpStatusCallback
qaGobiApiCbk.h, 1655
SLQSSetIMSARatStatusCallback
qaGobiApiCbk.h, 1655
SLQSSetIMSARegStatusCallback
qaGobiApiCbk.h, 1656
SLQSSetIMSASvcStatusCallback
qaGobiApiCbk.h, 1656
SLQSSetIMSSMSConfig
qaGobiApiIms.h, 1734
SLQSSetIMSSMSConfigCallback
qaGobiApiCbk.h, 1657
SLQSSetIMSUserConfig
qaGobiApiIms.h, 1735
SLQSSetIMSUserConfigCallback
qaGobiApiCbk.h, 1657
SLQSSetIMSVoIPConfig
qaGobiApiIms.h, 1735
SLQSSetIMSVoIPConfigCallback
qaGobiApiCbk.h, 1658
SLQSSetIndicationRegister
qaGobiApiSms.h, 1840
SLQSSetLocInjectPositionCallback
qaGobiApiCbk.h, 1658
SLQSSetLocInjectUTCTimeCallback
qaGobiApiCbk.h, 1658
SLQSSetLoggingMask
qaGobiApiDcs.h, 1679
SLQSSetM2MAVMute
qaGobiApiSwiAudio.h, 1851
SLQSSetM2MAudioAVCFG
qaGobiApiSwiAudio.h, 1849
SLQSSetM2MAudioLPBK
qaGobiApiSwiAudio.h, 1849
SLQSSetM2MAudioNVDef
qaGobiApiSwiAudio.h, 1850
SLQSSetM2MAudioProfile
qaGobiApiSwiAudio.h, 1850
SLQSSetM2MAudioVolume
qaGobiApiSwiAudio.h, 1850
SLQSSetM2MSpkrGain
qaGobiApiSwiAudio.h, 1851
SLQSSetModemTempCallback
qaGobiApiCbk.h, 1658
SLQSSetPacketSrvStatusCallback
qaGobiApiCbk.h, 1659
SLQSSetPositionMethodState
qaGobiApiPds.h, 1810
SLQSSetProfile
qaGobiApiWds.h, 1988
SLQSSetQosEventCallback
qaGobiApiCbk.h, 1659
SLQSSetQosNWStatusCallback
qaGobiApiCbk.h, 1660
SLQSSetQosPriEventCallback
qaGobiApiCbk.h, 1660
SLQSSetQosStatusCallback
qaGobiApiCbk.h, 1660
SLQSSetRegMgrConfig
qaGobiApiIms.h, 1736
SLQSSetRegMgrConfigCallback
qaGobiApiCbk.h, 1661
SLQSSetRfSarState

qaGobiApiSar.h, 1822
SLQSSetSDKTerminatedCallback
qaGobiApiCbk.h, 1661
SLQSSetSIMBasedImageSwitching
qaGobiApiFms.h, 1727
SLQSSetSIPConfig
qaGobiApiIms.h, 1736
SLQSSetSIPConfigCallback
qaGobiApiCbk.h, 1663
SLQSSetSMSEventCallback
qaGobiApiCbk.h, 1664
SLQSSetServingSystemCallback
qaGobiApiCbk.h, 1662
SLQSSetSessionStateCallback
qaGobiApiCbk.h, 1662
SLQSSetSignalStrengthsCallback
qaGobiApiCbk.h, 1663
SLQSSetSmsBroadcastActivation
qaGobiApiSms.h, 1840
SLQSSetSmsBroadcastConfig
qaGobiApiSms.h, 1841
SLQSSetSmsStorage
qaGobiApiSms.h, 1841
SLQSSetSpkgFormatRequired
qaGobiApiFms.h, 1727
SLQSSetSwiGetResetInfoCallback
qaGobiApiCbk.h, 1664
SLQSSetSwiHDRPersCallback
qaGobiApiCbk.h, 1664
SLQSSetSysSelectionPref
qaGobiApiNas.h, 1791
SLQSSetSysSelectionPrefCallBack
qaGobiApiCbk.h, 1665
SLQSSetTransLayerInfoCallback
qaGobiApiCbk.h, 1665
SLQSSetTransNWRegInfoCallback
qaGobiApiCbk.h, 1665
SLQSSetWdsEventCallback
qaGobiApiCbk.h, 1666
SLQSSetWdsTransferStatisticCallback
qaGobiApiCbk.h, 1667
SLQSSignalStrengthsIndReq, 893
ecioDelta, 894
ecioThresholdList, 894
ecioThresholdListLen, 894
ioDelta, 894
lteRsrpDelta, 894
lteSnrDelta, 894
rsrqDelta, 894
rxSignalStrengthDelta, 894
sinrDelta, 894
sinrThresholdList, 894
sinrThresholdListLen, 894
SLQSSignalStrengthsInformation, 895
ecioInfo, 895
errorRateInfo, 896
io, 896
lteRsrpinfo, 896
lteSnrinfo, 896
rsrqInfo, 896
rxSignalStrengthInfo, 896
sinr, 896
SLQSSmsGetMaxStorageSize
qaGobiApiSms.h, 1842
SLQSSmsGetMessageProtocol
qaGobiApiSms.h, 1842
SLQSSmsSetRoutes
qaGobiApiSms.h, 1843
SLQSStart
qaGobiApiDcs.h, 1679
SLQSStart_AVAgent
qaGobiApiDcs.h, 1680
SLQSStartSrv
qaGobiApiDcs.h, 1680
SLQSStartStopDataSession
qaGobiApiWds.h, 1991
SLQSSwiAVMSEventReportCallback
qaGobiApiCbk.h, 1667
SLQSSwiClearDyingGaspStatistics
qaGobiApiDms.h, 1704
SLQSSwiGetAllCarrierImages
qaGobiApiFms.h, 1727
SLQSSwiGetCrashAction
qaGobiApiDms.h, 1705
SLQSSwiGetCrashInfo
qaGobiApiDms.h, 1705
SLQSSwiGetDyingGaspCfg
qaGobiApiDms.h, 1706
SLQSSwiGetDyingGaspStatistics
qaGobiApiDms.h, 1706
SLQSSwiGetFSN
qaGobiApiDms.h, 1706
SLQSSwiGetFirmwareCurr
qaGobiApiFms.h, 1728
SLQSSwiGetFwUpdateStatus
qaGobiApiDms.h, 1707
SLQSSwiGetHDRPersonality
qaGobiApiNas.h, 1791
SLQSSwiGetHDRProtSubtype
qaGobiApiNas.h, 1792
SLQSSwiGetHRPDStats
qaGobiApiNas.h, 1792
SLQSSwiGetHostDevInfo
qaGobiApiDms.h, 1707
SLQSSwiGetHostDevInfoParams
qaGobiApiDms.h, 1688
SLQSSwiGetLteCQI
qaGobiApiNas.h, 1792
SLQSSwiGetLteSccRxInfo
qaGobiApiNas.h, 1793
SLQSSwiGetOSInfo
qaGobiApiDms.h, 1707
SLQSSwiGetOSInfoParams
qaGobiApiDms.h, 1688
SLQSSwiGetSMSStorage
qaGobiApiSms.h, 1843

SLQSSwiGetSerialNoExt
qaGobiApiDms.h, 1708
SLQSSwiGetSerialNoExtParams
qaGobiApiDms.h, 1689
SLQSSwiGetUSBComp
qaGobiApiDms.h, 1708
SLQSSwiNetworkDebug
qaGobiApiNas.h, 1793
SLQSSwiPSDetach
qaGobiApiNas.h, 1794
SLQSSwiSetCrashAction
qaGobiApiDms.h, 1709
SLQSSwiSetDyingGaspCfg
qaGobiApiDms.h, 1709
SLQSSwiSetHostDevInfo
qaGobiApiDms.h, 1709
SLQSSwiSetHostDevInfoParams
qaGobiApiDms.h, 1689
SLQSSwiSetOSInfo
qaGobiApiDms.h, 1710
SLQSSwiSetOSInfoParams
qaGobiApiDms.h, 1690
SLQSSwiSetUSBComp
qaGobiApiDms.h, 1710
SLQSTmdDeRegNotMitigationLvl
qaGobiApiTmd.h, 1915
SLQSTmdGetMitigationDevList
qaGobiApiTmd.h, 1916
SLQSTmdGetMitigationLvl
qaGobiApiTmd.h, 1916
SLQSTmdMitigationLvlRptCallback
qaGobiApiCbk.h, 1667
SLQSTmdRegNotMitigationLvl
qaGobiApiTmd.h, 1916
SLQSUIMAuthenticate
qaGobiApiUim.h, 1919
SLQSUIMChangePin
qaGobiApiUim.h, 1920
SLQSUIMDepersonalization
qaGobiApiUim.h, 1920
SLQSUIMEventRegister
qaGobiApiUim.h, 1921
SLQSUIMGetCardStatus
qaGobiApiUim.h, 1921
SLQSUIMGetConfiguration
qaGobiApiUim.h, 1922
SLQSUIMGetFileAttributes
qaGobiApiUim.h, 1922
SLQSUIMGetSlotsStatus
qaGobiApiUim.h, 1923
SLQSUIMPowerDown
qaGobiApiUim.h, 1923
SLQSUIMPowerUp
qaGobiApiUim.h, 1924
SLQSUIMReadTransparent
qaGobiApiUim.h, 1924
SLQSUIMRefreshComplete
qaGobiApiUim.h, 1925
SLQSUIMRefreshGetLastEvent
qaGobiApiUim.h, 1925
SLQSUIMRefreshOK
qaGobiApiUim.h, 1926
SLQSUIMRefreshRegister
qaGobiApiUim.h, 1926
SLQSUIMReset
qaGobiApiUim.h, 1927
SLQSUIMSetPinProtection
qaGobiApiUim.h, 1927
SLQSUIMSetRefreshCallBack
qaGobiApiCbk.h, 1668
SLQSUIMSetStatusChangeCallBack
qaGobiApiCbk.h, 1668
SLQSUIMSwitchSlot
qaGobiApiUim.h, 1928
SLQSUIMUnblockPin
qaGobiApiUim.h, 1928
SLQSUIMVerifyPin
qaGobiApiUim.h, 1929
SLQSUpgradeFirmware9x07
qaGobiApiFms.h, 1728
SLQSUpgradeFirmware9x15
qaGobiApiFms.h, 1729
SLQSVoiceALSSelectLine
qaGobiApiVoice.h, 1935
SLQSVoiceALSSetLineSwitching
qaGobiApiVoice.h, 1935
SLQSVoiceAnswerCall
qaGobiApiVoice.h, 1936
SLQSVoiceBindSubscription
qaGobiApiVoice.h, 1936
SLQSVoiceBurstDTMF
qaGobiApiVoice.h, 1937
SLQSVoiceDialCall
qaGobiApiVoice.h, 1937
SLQSVoiceEndCall
qaGobiApiVoice.h, 1938
SLQSVoiceGetAllCallInfo
qaGobiApiVoice.h, 1938
SLQSVoiceGetCLIP
qaGobiApiVoice.h, 1941
SLQSVoiceGetCLIR
qaGobiApiVoice.h, 1942
SLQSVoiceGetCNAP
qaGobiApiVoice.h, 1942
SLQSVoiceGetCOLP
qaGobiApiVoice.h, 1943
SLQSVoiceGetCOLR
qaGobiApiVoice.h, 1943
SLQSVoiceGetCallBarring
qaGobiApiVoice.h, 1939
SLQSVoiceGetCallForwardingStatus
qaGobiApiVoice.h, 1939
SLQSVoiceGetCallInfo
qaGobiApiVoice.h, 1940
SLQSVoiceGetCallWaiting
qaGobiApiVoice.h, 1940

SLQSVoiceGetConfig
qaGobiApiVoice.h, 1944
SLQSVoiceIndicationRegister
qaGobiApiVoice.h, 1944
SLQSVoiceInfoRecCallback
qaGobiApiCbk.h, 1668
SLQSVoiceManageCalls
qaGobiApiVoice.h, 1945
SLQSVoiceOrigUSSDNoWait
qaGobiApiVoice.h, 1945
SLQSVoiceSendFlash
qaGobiApiVoice.h, 1946
SLQSVoiceSetAllCallStatusCallBack
qaGobiApiCbk.h, 1669
SLQSVoiceSetCallBarringPassword
qaGobiApiVoice.h, 1946
SLQSVoiceSetConfig
qaGobiApiVoice.h, 1947
SLQSVoiceSetDTMFEventCallBack
qaGobiApiCbk.h, 1669
SLQSVoiceSetOTASPStatusCallBack
qaGobiApiCbk.h, 1670
SLQSVoiceSetPreferredPrivacy
qaGobiApiVoice.h, 1947
SLQSVoiceSetPrivacyChangeCallBack
qaGobiApiCbk.h, 1670
SLQSVoiceSetSUPSCallBack
qaGobiApiCbk.h, 1671
SLQSVoiceSetSUPSNotificationCallback
qaGobiApiCbk.h, 1671
SLQSVoiceSetSUPSService
qaGobiApiVoice.h, 1948
SLQSVoiceStartContDTMF
qaGobiApiVoice.h, 1949
SLQSVoiceStopContDTMF
qaGobiApiVoice.h, 1949
SLQSWCDMADecodeLongTextMsg
qaGobiApiSms.h, 1844
SLQSWCDMADecodeMTTextMsg
qaGobiApiSms.h, 1844
SLQSWCDMAEncodeMOTextMsg
qaGobiApiSms.h, 1844
SLQSWdsGoActive
qaGobiApiWds.h, 1991
SLQSWdsGoDormant
qaGobiApiWds.h, 1992
SLQSWdsSetEventReport
qaGobiApiWds.h, 1992
SLQSWdsSwiPDPRuntimeSettings
qaGobiApiWds.h, 1992
SLQSWmsAsyncRawSendCallBack
qaGobiApiCbk.h, 1671
SLQSWmsMemoryFullCallBack
qaGobiApiCbk.h, 1672
SLQSWmsMessageWaitingCallBack
qaGobiApiCbk.h, 1672
SMS_CONFIG_LEN
sms.h, 2017
SMS_MAX_SMS_ROUTES
sms.h, 2017
SMS_NUM_OF_SET
sms.h, 2017
SMSAsyncRawSend
qaGobiApiCbk.h, 1602
SMSAsyncRawSend_s, 911
alphaIDLen, 912
causeCode, 912
errorClass, 912
messageID, 912
msgDelFailureCause, 912
msgDelFailureType, 912
pAlphaID, 912
RPCause, 912
sendStatus, 912
TPCause, 912
userData, 912
SMSC_TYPE_LEN
sms.h, 2017
SMSCAddress, 912
data, 913
length, 913
SMSCAddressInfo
qaGobiApiCbk.h, 1603
sMSCAddressInfo, 913
data, 913
length, 913
sMSCAddressTlv, 913
SMSCInfo, 914
TlvPresent, 914
SMSCInfo
sMSCAddressTlv, 914
SMSCTlv
unpack_sms_SetNewSMSCallback_ind_t, 1198
SMSEtwsMessage, 914
data, 914
length, 914
notificationType, 914
SMSEtwsMessageInfo
qaGobiApiCbk.h, 1603
sMSEtwsMessageInfo, 915
data, 915
length, 915
notificationType, 915
sMSEtwsMessageTlv, 915
EtwsMessageInfo, 916
TlvPresent, 916
SMSEtwsPlmn, 916
mobileCountryCode, 916
mobileNetworkCode, 916
SMSEtwsPlmnInfo
qaGobiApiCbk.h, 1603
sMSEtwsPlmnInfo, 916
mobileCountryCode, 917
mobileNetworkCode, 917
SMSEventInfo
qaGobiApiCbk.h, 1603

SMSEventInfo_s, 917
pEtwsMessageInfo, 918
pEtwsPlmnInfo, 918
pMTMessageInfo, 918
pMessageModeInfo, 918
pSMSCAddressInfo, 918
pSMSOnIMSInfo, 918
pTransferRouteMTMessageInfo, 918
smsEventType, 918
SMSEventType
qaGobiApiCbk.h, 1635
SMSMTMessage, 921
messageIndex, 921
storageType, 921
SMSMTMessageInfo
qaGobiApiCbk.h, 1604
sMSMTMessageInfo, 921
messageIndex, 922
storageType, 922
SMSMemoryInfo, 919
messageMode, 920
storageType, 920
SMSMessageMode, 920
messageMode, 920
SMSMessageModeInfo
qaGobiApiCbk.h, 1604
sMSMessageModeInfo, 920
messageMode, 920
SMSOnIMS, 922
smsOnIMS, 922
SMSOnIMSInfo
qaGobiApiCbk.h, 1605
sMSOnIMSInfo, 922
smsOnIMS, 923
sMSOnIMSTlv, 923
IMSInfo, 923
TlvPresent, 923
SMSSupport
pack_dms_SetCustFeature_t, 601
unpack_dms_GetCustFeature_t, 1039
SMSTransferRouteMTMessage, 925
ackIndicator, 926
data, 926
format, 926
length, 926
transactionID, 926
SMSTransferRouteMTMessageInfo
qaGobiApiCbk.h, 1605
sMSTransferRouteMTMessageInfo, 926
ackIndicator, 926
data, 926
format, 926
length, 926
transactionID, 927
SNR
nas_NetworkStatEVDO, 483
NetworkStatEVDO, 583
sNonIntraSearch
LTEInfoIntrafreq, 400
nas_LTEInfoIntrafreq, 465
SO
nas_NetworkStat1x, 481
NetworkStat1x, 582
SOMask
CurrNetworkInfo, 205
currNetworkInfo, 206
wds_currNetworkInfo, 1396
SPC_SIZE
dms.h, 1443
sPhyCaAggPcellInfo
nasGetLTECphyCa, 539
QmiCbkNasLTECphyCaInfo, 800
unpack_nas_SetNasLTECphyCaIndCallback_ind_t, 1118
sPhyCaAggScellDlBw
nasGetLTECphyCa, 539
QmiCbkNasLTECphyCaInfo, 800
unpack_nas_SetNasLTECphyCaIndCallback_ind_t, 1118
sPhyCaAggScellIndType
nasGetLTECphyCa, 539
QmiCbkNasLTECphyCaInfo, 800
unpack_nas_SetNasLTECphyCaIndCallback_ind_t, 1119
sPhyCaAggScellIndex
nasGetLTECphyCa, 539
QmiCbkNasLTECphyCaInfo, 800
unpack_nas_SetNasLTECphyCaIndCallback_ind_t, 1119
sPhyCaAggScellInfo
nasGetLTECphyCa, 539
QmiCbkNasLTECphyCaInfo, 800
unpack_nas_SetNasLTECphyCaIndCallback_ind_t, 1119
sQosFlowStat, 927
bearerId, 927
tx_bytes, 927
tx_bytes_drp, 927
tx_pkt, 927
tx_pkt_drp, 927
sQosStat, 927
apnId, 928
numQosFlow, 928
qosFlow, 928
total_rx_bytes, 928
total_rx_pkt, 929
total_tx_bytes, 929
total_tx_bytes_drp, 929
total_tx_pkt, 929
total_tx_pkt_drp, 929
SSInfo
unpack_nas_SetServingSystemCallback_ind_t, 1120
sSLQSSignalStrengthsInfo
nas_SLQSSignalStrengthsTlv, 514
SSTlv

unpack_nas_SetEventReportInd_t, 1118
SUPSInfo, 933
isModByCC, 933
svcType, 933
SUPSInformation
voiceSUPSInfo, 1382
SUPSType
pack_voice_SLQSVoiceManageCalls_t, 705
voiceManageCallsReq, 1365
SV, 933
id, 934
mask, 934
system, 934
SVInfo, 934
len, 935
pSV, 935
SWI Audio Service(SWIAUDIO), 56
SWI Open Mobile Alliance Service (SWIOMA), 49
SWI_API
SwiDataTypes.h, 2035
SWI_STRUCT_CarrierImage, 935
m_FwBuildId, 936
m_FwImageId, 936
m_PriBuildId, 936
m_PriImageId, 937
m_nCarrierId, 936
m_nFolderId, 936
m_nStorage, 936
SWIWWANCMAPI.h, 2046
samplesPerBatch
accelAcceptReady_s, 113
accelTempAcceptReady_s, 114
gyroAcceptReady_s, 300
gyroTempAcceptReady_s, 300
sar.h, 2012
pack_sar_SLQSGetRfSarState, 2012
pack_sar_SLQSSetRfSarState, 2013
unpack_sar_SLQSGetRfSarState, 2013
unpack_sar_SLQSSetRfSarState, 2013
satelliteInfo, 836
azimuth, 838
elevation, 838
gnssSvId, 838
healthStatus, 838
snr, 838
svInfoMask, 838
svListLen, 838
svStatus, 838
system, 838
validMask, 838
SaveSMS
qaGobiApiSms.h, 1829
sbas_almanac_sv_msk
GPSStateInfo, 292
unpack_pds_SLQSGetGPSStateInfo_t, 1172
sbas_ephemeris_sv_msk
GPSStateInfo, 292
unpack_pds_SLQSGetGPSStateInfo_t, 1172
sbas_health_sv_msk
GPSStateInfo, 292
unpack_pds_SLQSGetGPSStateInfo_t, 1172
sbas_visible_sv_msk
GPSStateInfo, 293
unpack_pds_SLQSGetGPSStateInfo_t, 1172
SccRxInfo, 839
numInstances, 839
rsrq, 839
sigInfo, 839
snr, 839
TlvPresent, 839
scell_idx
nas_PhyCaAggScellIndex, 488
NASPhyCaAggScellIndex, 559
PhyCaAggScellIndex, 748
scell_state
nas_PhyCaAggScellIndType, 488
nas_PhyCaAggScellInfo, 491
NASPhyCaAggScellIndType, 560
NASPhyCaAggScellInfo, 561
PhyCaAggScellIndType, 749
PhyCaAggScellInfo, 752
screeningInd
connectNumInfo, 191
voice_connectNumInfo, 1320
sduErrorRatio
LibPackUMTSQoS, 365
UMTSMinQoS, 1028
UMTSQoS, 1030
wds_UMTSMinQoS, 1411
seDNSIPv4Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1288
seDNSIPv6Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1288
sePCSCFIPv4Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1288
sePCSCFIPv6Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1288
secActivate
fileAttributes, 246
uim_fileAttributes, 988
secActivateMask
fileAttributes, 246
uim_fileAttributes, 988
secChA
CDMAChannel, 162
nas_CDMAChannel, 424
secChB
CDMAChannel, 163
nas_CDMAChannel, 424
secDeactivate
fileAttributes, 246
uim_fileAttributes, 988

secDeactivateMask
fileAttributes, 246
uim_fileAttributes, 988
secIncrease
fileAttributes, 247
uim_fileAttributes, 988
secIncreaseMask
fileAttributes, 247
uim_fileAttributes, 988
SecProt
nas_protocolSubtypeElement, 496
protocolSubtypeElement, 773
secRead
fileAttributes, 247
uim_fileAttributes, 988
secReadMask
fileAttributes, 247
uim_fileAttributes, 988
secWrite
fileAttributes, 247
uim_fileAttributes, 988
secWriteMask
fileAttributes, 247
uim_fileAttributes, 988
secdns
unpack_wds_GetDefaultProfile_t, 1258
secdnsv6
unpack_wds_GetDefaultProfile_t, 1258
second
nas_timeInfo, 522
nas_UniversalTime, 528
timeInfo, 966
UniversalTime, 1033
secondaryDNS
pack_wds_SetDefaultProfile_t, 719
SecondaryDNSV4
unpack_wds_SLQSGetRuntimeSettings_t, 1279
SecondaryDNSV6
unpack_wds_SLQSGetRuntimeSettings_t, 1279
secondaryHA
unpack_wds_GetMobileIPProfile_t, 1262
SectorIDLen
nas_NetworkStatEVDO, 483
NetworkStatEVDO, 583
selNetwork
nas_servSystem, 510
servSystem, 848
selected
BroadcastConfig, 142
CDMABroadcastConfig, 162
sms_BroadcastConfig, 898
sms_CDMABroadcastConfig, 899
selectedNetwork
NASServingSystemInfo, 569
ServingSystemInfo, 847
selection
pack_swioma_SLQSOMADMSendSelection_t, 680
SendSMS
qaGobiApiSms.h, 1829
sendStatus
SMSAsyncRawSend_s, 912
sensorData, 839
flags, 841
sensorDataLen, 841
timeOfFirstSample, 841
timeOffset, 841
xAxis, 841
yAxis, 841
zAxis, 841
sensorData_t, 841
flags, 842
sensorDataLen, 842
timeOfFirstSample, 842
timeOffset, 842
xAxis, 842
yAxis, 842
zAxis, 842
sensorDataLen
sensorData, 841
sensorData_t, 842
sensorDataUsage
qaGobiApiCbk.h, 1601
sensorDataUsage_s, 843
aidingIndicatorMask, 843
usageMask, 843
serialNumbersInfo, 843
esnSize, 844
imeiSize, 844
imeiSvnSize, 844
meidSize, 844
pESNString, 844
pIMEIString, 844
pImeiSvnString, 844
pMEIDString, 844
qaGobiApiDms.h, 1687
servSystem, 847
csAttachState, 848
numRadioInterfaces, 848
psAttachState, 848
radioInterface, 848
regState, 848
selNetwork, 848
ServerAddrList
unpack_wds_SLQSGetRuntimeSettings_t, 1279
serverAddrTypeMask
pack_loc_SLQSLOCGetServer_t, 618
serverStatus
QmiCbkLocGetServerInd, 788
unpack_loc_GetServer_Ind_t, 1093
serverType
LocGetServerReq, 385
pack_loc_SLQSLOCGetServer_t, 618
QmiCbkLocGetServerInd, 788
unpack_loc_GetServer_Ind_t, 1093
serviceCategory

CDMABroadcastConfig, 162
sms_CDMABroadcastConfig, 899
serviceClassInformation
qaGobiApiVoice.h, 1933
serviceDomain
nas_RejectReasonTlv, 500
nasNetworkReject, 555
serviceProviderName, 845
displayCondition, 845
spn, 845
spnLength, 845
servingCellId
LTEInfoIntrafreq, 400
nas_LTEInfoIntrafreq, 465
ServingSystem
qaQmiServingSystemParam, 779
unpack_nas_SLQSGetServingSystem_t, 1130
ServingSystemInfo, 845
csAttachState, 846
hdrPersonality, 846
psAttachState, 846
radioInterfaceList, 846
radioInterfaceNo, 847
registrationState, 847
selectedNetwork, 847
sessionControl
pack_pds_StartPDSTrackingSessionExt_t, 664
sessionEndReason
_packetSrvStatus, 70
slqsSessionStateInfo, 890
unpack_wds_SLQSSetPacketSrvStatusCallback_t, 1281
SessionId
LOCStartReq, 393
pack_loc_Start_t, 628
pack_loc_Stop_t, 628
sessionId
LOCStopReq, 393
QmiCbkLocPositionReportInd, 797
ssdatasession_params, 932
unpack_loc_PositionRpt_Ind_t, 1101
sessionInfo, 848
omaDmConfig, 849
omaDmFota, 849
omaDmNotifications, 849
pack_uim_ChangePin_t, 682
pack_uim_ReadTransparent_t, 683
pack_uim_SetPinProtection_t, 684
pack_uim_SLQSUIMAuthenticate_t, 685
pack_uim_SLQSUIMGetFileAttributes_t, 687
pack_uim_SLQSUIMRefreshComplete_t, 689
pack_uim_SLQSUIMRefreshGetLastEvent_t, 689
pack_uim_SLQSUIMRefreshOK_t, 690
pack_uim_SLQSUIMRefreshRegister_t, 690
pack_uim_UnblockPin_t, 692
pack_uim_VerifyPin_t, 693
sessionInfoTlv, 849
sessionInfoTlvExt, 850
UIMAuthenticateReq, 1001
UIMChangePinReq, 1002
UIMGetFileAttributesReq, 1006
UIMReadTransparentReq, 1010
UIMRefreshCompleteReq, 1012
UIMRefreshGetLastEventReq, 1014
UIMRefreshOKReq, 1015
UIMRefreshRegisterReq, 1015
UIMSetPinProtectionReq, 1017
UIMUnblockPinReq, 1021
UIMVerifyPinReq, 1022
SessionInfoConfig
unpack_swioma_SLQSOMADMAlertCallback_ind_t, 1207
sessionInfoExt, 849
omaDmConfig, 849
omaDmFota, 849
SessionInfoFota
unpack_swioma_SLQSOMADMAlertCallback_ind_t, 1207
SessionInfoNotification
unpack_swioma_SLQSOMADMAlertCallback_ind_t, 1207
sessionInfoTlv, 849
sessionInfo, 849
sessionType, 849
TlvPresent, 849
sessionInfoTlvExt, 850
sessionInfo, 850
sessionType, 850
TlvPresent, 850
sessionInformation
qaGobiApiCbk.h, 1601
sessionInformationExt
qaGobiApiCbk.h, 1601
sessionOperation
pack_pds_StartPDSTrackingSessionExt_t, 664
sessionServerOption
pack_pds_StartPDSTrackingSessionExt_t, 664
SessionState
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1209
SessionStatus
unpack_pds_SetEventReport_Ind_t, 1167
sessionStatus
omaDmNotificationsTlv, 595
QmiCbkLocPositionReportInd, 797
unpack_loc_PositionRpt_Ind_t, 1101
unpack_omaDmNotificationsTlv_t, 1161
SessionType
_SwiAvmsEventInd, 100
pack_swioma_SLQSOMADMGetSessionInfo_t, 679
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1209
sessionType
omaDmFotaTlv, 592
pack_pds_StartPDSTrackingSessionExt_t, 664

pack_swioma_SLQSOMADMCancelSession_t, 679
pack_swioma_SLQSOMADMStartSession_t, 681
sessionInfoTlv, 849
sessionInfoTlvExt, 850
uim_refreshevent, 993
uim_sessionInformation, 995
uim_UIMSessionInformation, 999
UIMRefreshEvent, 1013
UIMSessionInformation, 1016
unpack_omaDmFotaTlv_t, 1160
set_fix_rate
pack_swiloc_SwiLocSetAutoStart_t, 678
SwiLocSetAutoStartReq, 940
set_fix_type
pack_swiloc_SwiLocSetAutoStart_t, 678
SwiLocSetAutoStartReq, 940
set_function
pack_swiloc_SwiLocSetAutoStart_t, 678
SwiLocSetAutoStartReq, 940
set_max_dist
pack_swiloc_SwiLocSetAutoStart_t, 678
SwiLocSetAutoStartReq, 940
set_max_time
pack_swiloc_SwiLocSetAutoStart_t, 678
SwiLocSetAutoStartReq, 940
SetACCOLC
qaGobiApiNas.h, 1775
SetActivationStatusCallback
qaGobiApiCbk.h, 1636
SetActiveMobileIPProfile
qaGobiApiWds.h, 1970
SetAudioPathConfigReq, 850
pCodecSTGain, 851
pDTMFTXGain, 851
pECMode, 852
pNSEnable, 852
pRXAGCList, 852
pRXAVCAGCSwitch, 852
pRXAVCList, 852
pRXPCMIIRFltr, 852
pTXAGCList, 852
pTXAVCSwitch, 852
pTXGain, 852
pTXPCMIIRFltr, 852
Profile, 852
SetAudioProfileReq, 852
EarMute, 853
Generator, 853
MicMute, 853
Profile, 853
Volume, 853
SetAudioVolTLBConfigReq, 853
Generator, 854
Item, 854
Profile, 854
VolValue, 854
Volume, 854
SetAudioVolTLBConfigResp, 854
ResCode, 854
SetAutoconnect
qaGobiApiWds.h, 1970
SetCATEventCallback
qaGobiApiCbk.h, 1636
SetCDMANetworkParameters
qaGobiApiNas.h, 1776
setCustomSettingV2, 855
cust_id, 855
cust_value, 855
value_length, 855
SetDataCapabilitiesCallback
qaGobiApiCbk.h, 1637
SetDefaultProfile
qaGobiApiWds.h, 1970
SetDefaultProfileLTE
qaGobiApiWds.h, 1972
SetDefaultProfileLTEV2
qaGobiApiWds.h, 1973
SetDefaultProfileNum
qaGobiApiWds.h, 1975
SetDeviceStateChangeCbk
qaGobiApiCbk.h, 1638
setDyingGaspCfg, 855
pDestSMSContent, 856
pDestSMSNum, 856
SetFwDldCompletionCbk
qaGobiApiCbk.h, 1638
SetGPSCallback
qaGobiApiCbk.h, 1639
SetIMSSMSConfigReq, 856
pPhoneCtxtURI, 856
pPhoneCtxtURILen, 856
pSMSFormat, 856
pSMSOverIPNwInd, 856
SetIMSSMSConfigResp, 856
pSettingResp, 857
SetIMSUserConfigReq, 857
pIMSDomain, 857
pIMSDomainLen, 857
SetIMSUserConfigResp, 857
pSettingResp, 858
SetIMSVoIPConfigReq, 858
pAmrMode, 860
pAmrOctetAligned, 860
pAmrWBMode, 860
pAmrWBOctetAligned, 860
pAmrWbEnable, 860
pMinSessionExpiryTimer, 860
pRTPRTCPInactTimer, 860
pRingBackTimer, 860
pRingingTimer, 860
pScrAmrEnable, 860
pScrAmrWbEnable, 860
pSessionExpiryTimer, 860
SetIMSVoIPConfigResp, 860
pSettingResp, 861

SetImagesPreference
qaGobiApiFms.h, 1720
setIndicationRegReq
qaGobiApiSms.h, 1827
SetLURejectCallback
qaGobiApiCbk.h, 1643
SetLocBestAvailPosCallback
qaGobiApiCbk.h, 1639
SetLocCradleMountCallback
qaGobiApiCbk.h, 1639
SetLocDeleteAssistDataCallback
qaGobiApiCbk.h, 1640
SetLocEngineStateCallback
qaGobiApiCbk.h, 1640
SetLocEventMaskNMEACallback
qaGobiApiCbk.h, 1640
SetLocEventPositionCallback
qaGobiApiCbk.h, 1640
SetLocEventTimeSyncCallback
qaGobiApiCbk.h, 1641
SetLocGetServerCallback
qaGobiApiCbk.h, 1641
SetLocGnssSvInfoCallback
qaGobiApiCbk.h, 1641
SetLocInjectSensorDataCallback
qaGobiApiCbk.h, 1642
SetLocInjectTimeCallback
qaGobiApiCbk.h, 1642
SetLocOpModeCallback
qaGobiApiCbk.h, 1642
SetLocSensorStreamingCallback
qaGobiApiCbk.h, 1642
SetLocSetExtPowerConfigCallback
qaGobiApiCbk.h, 1643
SetM2MAVMuteReq, 864
EarMute, 864
MicMute, 864
pCwtMute, 864
Profile, 865
SetM2MAudioAVCFGReq, 861
Device, 861
PIFACEId, 861
pPCMParams, 861
Profile, 861
SetM2MAudioLPBKReq, 862
Enable, 862
SetM2MAudioProfileReq, 862
pCwtMute, 863
pEarMute, 863
pGenerator, 863
pMicMute, 863
pVolume, 863
Profile, 863
SetM2MAudioVolumeReq, 863
Generator, 864
Level, 864
Profile, 864
SetM2MSpkrGainReq, 865
Profile, 865
Value, 865
SetMobileIP
qaGobiApiWds.h, 1975
SetMobileIPParameters
qaGobiApiWds.h, 1976
SetMobileIPProfile
qaGobiApiWds.h, 1977
SetMobileIPStatusCallback
qaGobiApiCbk.h, 1643
SetNMEACallback
qaGobiApiCbk.h, 1645
SetNasLTECphyCaIndCallback
qaGobiApiCbk.h, 1644
SetNetChangeCbk
qaGobiApiCbk.h, 1644
SetNetworkPreference
qaGobiApiNas.h, 1777
SetNewSMSCallback
qaGobiApiCbk.h, 1645
SetOMADMStateCallback
qaGobiApiCbk.h, 1646
SetPDSDefaults
qaGobiApiPds.h, 1804
SetPDSState
qaGobiApiPds.h, 1805
SetPDSStateCallback
qaGobiApiCbk.h, 1646
setPINProtection, 865
pinID, 866
pinLength, 866
pinOperation, 866
pinValue, 866
SetPortAutomaticTracking
qaGobiApiPds.h, 1805
SetPower
qaGobiApiDms.h, 1699
SetPowerCallback
qaGobiApiCbk.h, 1646
SetRFInfoCallback
qaGobiApiCbk.h, 1647
SetRMTransferStatisticsCallback
qaGobiApiCbk.h, 1647
SetRankIndicatorCallback
qaGobiApiCbk.h, 1647
SetRegMgrConfigReq, 866
pCSCFPortName, 867
pCSCFPortNameLen, 867
pIMSTestMode, 867
pPriCSCFPort, 867
SetRegMgrConfigResp, 867
pSettingResp, 867
SetRoamingIndicatorCallback
qaGobiApiCbk.h, 1647
SetSDKImagePath
qaGobiApiDcs.h, 1677
SetSIPConfigReq, 872
pSIPLocalPort, 873

pSigCompEnabled, 873
pSubscribeTimer, 873
pTimerSIPReg, 873
pTimerT1, 873
pTimerT2, 873
pTimerTf, 873
SetSIPConfigResp, 873
pSettingResp, 874
SetSLQSOMADMAlertCallback
qaGobiApiCbk.h, 1649
SetSLQSOMADMAlertCallbackExt
qaGobiApiCbk.h, 1649
SetSMSCAddress
qaGobiApiSms.h, 1830
SetSMSWake
qaGobiApiRms.h, 1819
SetServiceAutomaticTracking
qaGobiApiPds.h, 1806
SetSignalStrengthCallback
qaGobiApiCbk.h, 1648
setSignalStrengthInfo, 867
pCDMAECIODelta, 871
pCDMAECIOThresh, 871
pCDMARSSIDelta, 871
pCDMARSSIThresh, 871
pGSMRSSIDelta, 871
pGSMRSSIThresh, 871
pHDRECIODelta, 871
pHDRECIOThresh, 871
pHDRIODelta, 871
pHDRIOThresh, 871
pHDRRSSIDelta, 871
pHDRRSSIThresh, 871
pHDRSINRDelta, 871
pHDRSINRThresh, 871
pLTERSRPDelta, 871
pLTERSRPThresh, 871
pLTERSRQDelta, 871
pLTERSRQThresh, 871
pLTERSSIDelta, 872
pLTERSSIThresh, 872
pLTESNRDelta, 872
pLTESNRThresh, 872
pLTESigRptConfig, 872
pTDSCDMAECIODelta, 872
pTDSCDMAECIOThresh, 872
pTDSCDMARSCPDelta, 872
pTDSCDMARSCPThresh, 872
pTDSCDMARSSIDelta, 872
pTDSCDMARSSIThresh, 872
pTDSCDMASINRDelta, 872
pTDSCDMASINRThresh, 872
pWCDMAECIODelta, 872
pWCDMAECIOThresh, 872
pWCDMARSSIDelta, 872
pWCDMARSSIThresh, 872
SetUSSDNoWaitIndicationCallback
qaGobiApiCbk.h, 1650
SetUSSDNotificationCallback
qaGobiApiCbk.h, 1650
SetUSSDReleaseCallback
qaGobiApiCbk.h, 1650
SetUimSlotStatusChangeCallback
qaGobiApiCbk.h, 1649
SetXTRAAutomaticDownload
qaGobiApiPds.h, 1806
SetXTRANetwork
qaGobiApiPds.h, 1807
SetupEventList
CatEventListTlv, 160
Severity
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1209
severity
omaDmFotaTlv, 592
unpack_omaDmFotaTlv_t, 1160
Short Message Service (SMS), 42
shortName
nas_PLMNNetworkNameData, 494
nasPLMNNameResp, 564
PLMNNetworkNameData, 756
unpack_nas_SLQSGetPLMNName_t, 1125
shortNameCI
nasPLMNNameResp, 564
unpack_nas_SLQSGetPLMNName_t, 1125
shortNameEn
nasPLMNNameResp, 564
unpack_nas_SLQSGetPLMNName_t, 1126
shortNameLen
nas_PLMNNetworkNameData, 494
nasPLMNNameResp, 564
PLMNNetworkNameData, 756
unpack_nas_SLQSGetPLMNName_t, 1126
shortNameSB
nasPLMNNameResp, 564
unpack_nas_SLQSGetPLMNName_t, 1126
shortNameSpareBits
nas_PLMNNetworkNameData, 494
PLMNNetworkNameData, 757
sid
CDMAInfo, 164
nas_CDMAInfo, 426
nas_sidNid, 511
sidNid, 875
unpack_nas_GetHomeNetwork_t, 1110
SidNid
homeSIDNID, 310
nas_homeSIDNID, 456
sidNid, 874
nid, 875
sid, 875
SigInd
LibPackUMTSReqQoSSigInd, 365
UMTSReqQoSSigInd, 1031
sigInfo, 875
nas_SccRxInfo, 508

pECIOThresh, 876
pHDRSINRThresh, 876
pIOThresh, 876
pLTESNRThresh, 876
pLTESigRptCfg, 876
pRSRPThresh, 876
pRSRQThresh, 876
pRSSIThresh, 876
pTDSCDMASINRCONFThresh, 876
SccRxInfo, 839
signal
signalInfo, 877
signalInfo, 876
alertPitch, 877
signal, 877
signalType, 877
signalStrength
nas_SignalStrengthTlv, 511
SignalStrengthDataType, 877
thresholds, 877
thresholdsSize, 877
signalStrengthReqMask
slqsSignalStrengthInfo, 892
unpack_nas_SLQSGetSignalStrength_t, 1132
signalType
signalInfo, 877
SimCapability
unpack_dms_GetDeviceCap_t, 1040
simCapability
unpack_dms_GetDeviceCapabilities_t, 1042
sinr
HDRSSInfo, 306
hdrSSInfo, 307
nas_SLQSSignalStrengthsInformation, 514
slqsSignalStrengthInfo, 893
SLQSSignalStrengthsInformation, 896
TDSCDMASigInfoExt, 958
tdscdmaSigInfoExt, 959
unpack_nas_SLQSGetSignalStrength_t, 1132
sinrDelta
nas_SLQSSignalStrengthsIndReq, 513
SLQSSignalStrengthsIndReq, 894
sinrThresholdList
nas_SLQSSignalStrengthsIndReq, 513
SLQSSignalStrengthsIndReq, 894
sinrThresholdListLen
nas_SLQSSignalStrengthsIndReq, 513
SLQSSignalStrengthsIndReq, 894
sku_str
slqsfwinfo_s, 884
unpack_dms_GetFirmwareInfo_t, 1046
slot
pack_uim_SLQSUIMPowerDown_t, 688
pack_uim_SLQSUIMPowerUp_t, 688
UIMPowerDownReq, 1009
UIMPowerUpReq, 1009
slot_t, 877
bICCID, 878
bICCIDLength, 878
bLogicalSlot, 878
uPhyCardStatus, 878
uPhySlotStatus, 878
slotInf, 878
AppStatus, 879
cardState, 879
errorState, 879
numApp, 879
upinRetries, 879
upinState, 880
upukRetries, 880
SlotInfo
cardStatus, 156
uim_cardStatus, 982
slotInfo, 880
AppStatus, 881
cardState, 881
errorState, 881
numApp, 881
upinRetries, 881
upinState, 881
upukRetries, 881
slots_t, 881
uimSlotStatus, 882
slotsstatusChange
UIMSlotStatusChangeInfo, 1019
unpack_uim_SetUimSlotStatusChangeCallback_-ind_t, 1216
slqs3GPPConfigItem
qaGobiApiWds.h, 1955
SlqsNas3GppNetworkInfo, 884
Description, 885
Forbidden, 885
InUse, 885
MCC, 885
MNC, 885
Preferred, 885
Roaming, 886
SlqsNas3GppNetworkRAT
qaGobiApiNas.h, 1753
SlqsNasPcsDigit, 886
includes_pcs_digit, 886
MCC, 886
MNC, 886
slqsNetworkScanInfo
qaGobiApiNas.h, 1753
SlqsProfile3GPP
unpackWdsProfileParam, 1289
wds_profileInfo, 1408
WdsProfileParam, 1425
SlqsProfile3GPP2
unpackWdsProfileParam, 1289
wds_profileInfo, 1408
WdsProfileParam, 1426
slqsSessionStateInfo, 890
pQmiInterfaceInfo, 890
reconfiguration_required, 890

sessionEndReason, 890
state, 890
slqsSignalStrengthInfo, 890
ecioList, 892
ecioListLen, 892
errorRateList, 892
errorRateListLen, 892
Io, 892
ltersrp, 892
ltesnr, 892
rsrqInfo, 892
rxSignalStrengthList, 892
rxSignalStrengthListLen, 892
signalStrengthReqMask, 892
sinr, 893
slqsWdsEventInfo, 896
pDataBearer, 897
pDormancyStatus, 897
pPacketsCountRX, 897
pPacketsCountTX, 897
pQmiInterfaceInfo, 897
pTotalBytesRX, 898
pTotalBytesTX, 898
slqsautoconnect, 882
acroamsetting, 882
acsetting, 882
action, 882
slqsfwinfo_s, 883
appversion_str, 884
bootversion_str, 884
carrier_str, 884
cur_carr_name, 884
cur_carr_rev, 884
modelid_str, 884
packageid_str, 884
priversion_str, 884
sku_str, 884
slqssendasyncsmsparams_s, 886
messageFormat, 888
messageSize, 888
pFollowOnDC, 888
pForceOnDC, 888
pLinktimer, 888
pMessage, 888
pRetryMessage, 888
pRetryMessageId, 888
pServiceOption, 888
pSmsOnIms, 888
pUserData, 888
slqssendsmsparams_s, 888
messageFailureCode, 889
messageFormat, 889
messageID, 889
messageSize, 889
pLinktimer, 889
pMessage, 889
pSmsOnIms, 889
sms.h
LITEQMI_QMI_CBK_PARAM_NOCHANGE, 2017
LITEQMI_QMI_CBK_PARAM_RESET, 2017
LITEQMI_QMI_CBK_PARAM_SET, 2017
sms.h, 2014
eqmiCbkSetStatus, 2017
MAX_MSE_TWS_MSG, 2017
MAX_SMS_LIST_SIZE, 2017
pack_sms_GetSMSCAddress, 2018
pack_sms_SLQSDeleteSMS, 2019
pack_sms_SLQSGetIndicationRegister, 2020
pack_sms_SLQSGetMessageWaiting, 2020
pack_sms_SLQSGetSMS, 2020
pack_sms_SLQSGetSMSList, 2021
pack_sms_SLQSGetSmsBroadcastConfig, 2021
pack_sms_SLQSGetTransLayerInfo, 2021
pack_sms_SLQSGetTransNWRegInfo, 2022
pack_sms_SLQSModifySMSStatus, 2022
pack_sms_SLQSSendAsyncSMS, 2022
pack_sms_SLQSSetIndicationRegister, 2023
pack_sms_SLQSSetSmsBroadcastActivation, 2023
pack_sms_SLQSSetSmsBroadcastConfig, 2024
pack_sms_SLQSSetSmsStorage, 2024
pack_sms_SLQSSmsGetMaxStorageSize, 2024
pack_sms_SLQSSmsGetMessageProtocol, 2025
pack_sms_SLQSSmsSetRoutes, 2025
pack_sms_SLQSSwiGetSMSStorage, 2025
pack_sms_SaveSMS, 2018
pack_sms_SendSMS, 2018
pack_sms_SetNewSMSCallback, 2019
pack_sms_SetSMSCAddress, 2019
SMS_CONFIG_LEN, 2017
SMS_MAX_SMS_ROUTES, 2017
SMS_NUM_OF_SET, 2017
SMSC_TYPE_LEN, 2017
unpack_sms_GetSMSCAddress, 2026
unpack_sms_SLQSDeleteSMS, 2028
unpack_sms_SLQSGetIndicationRegister, 2028
unpack_sms_SLQSGetMessageWaiting, 2028
unpack_sms_SLQSGetSMS, 2029
unpack_sms_SLQSGetSMSList, 2029
unpack_sms_SLQSGetSmsBroadcastConfig, 2029
unpack_sms_SLQSGetTransLayerInfo, 2030
unpack_sms_SLQSGetTransNWRegInfo, 2030
unpack_sms_SLQSModifySMSStatus, 2031
unpack_sms_SLQSSendAsyncSMS, 2031
unpack_sms_SLQSSetIndicationRegister, 2031
unpack_sms_SLQSSetSmsBroadcastActivation, 2032
unpack_sms_SLQSSetSmsBroadcastConfig, 2032
unpack_sms_SLQSSetSmsStorage, 2032
unpack_sms_SLQSSmsGetMaxStorageSize, 2032
unpack_sms_SLQSSmsGetMessageProtocol, 2033
unpack_sms_SLQSSmsSetRoutes, 2033
unpack_sms_SLQSSwiGetSMSStorage, 2034

unpack_sms_SLQSWmsMemoryFullCallBack_ind, 2034
unpack_sms_SaveSMS, 2026
unpack_sms_SendSMS, 2026
unpack_sms_SetNewSMSCallback, 2027
unpack_sms_SetNewSMSCallback_ind, 2027
unpack_sms_SetSMSCAddress, 2027
sms_BroadcastConfig, 898
fromServiceId, 898
selected, 898
toServiceId, 898
sms_CDMABroadcastConfig, 898
language, 899
selected, 899
serviceCategory, 899
sms_getIndicationReg, 899
pRegCallStatInfoEvt, 900
pRegTransLayerInfoEvt, 900
pRegTransNWRegInfoEvt, 900
sms_getMsgWaitingInfo, 900
msgWaitInfo, 900
numInstances, 900
sms_getTransLayerInfo, 900
pRegInd, 901
pTransLayerInfo, 901
sms_getTransNWRegInfo, 901
pRegStatus, 902
sms_maxStorageSizeReq, 902
pMessageMode, 902
storageType, 902
sms_maxStorageSizeResp, 902
freeSlots, 903
maxStorageSize, 903
sms_messageWaitingInfoContent, 903
activeInd, 903
msgCount, 903
msgType, 903
sms_msgProtocolResp, 904
msgProtocol, 904
sms_qaQmi3GPP2BroadcastCfgInfo, 904
activated_ind, 905
CDMABroadcastConfig, 905
num_instances, 905
sms_qaQmi3GPPBroadcastCfgInfo, 905
activated_ind, 905
broadcastConfig, 905
num_instances, 905
sms_routeEntry, 905
messageClass, 906
messageType, 906
receiptAction, 906
routeStorage, 906
sms_sendAsyncsmsParams, 906
messageFormat, 908
messageSize, 908
pFollowOnDC, 908
pForceOnDC, 908
pLinktimer, 908
pMessage, 908
pRetryMessage, 908
pRetryMessageId, 908
pServiceOption, 908
pSmsOnIms, 908
pUserData, 908
sms_setIndicationReg, 909
pRegCallStatInfoEvt, 909
pRegTransLayerInfoEvt, 909
pRegTransNWRegInfoEvt, 909
sms_setRoutesReq, 909
numOfRoutes, 910
pTransferStatusReport, 910
routeList, 910
sms_transLayerInfo, 910
TransCap, 910
TransType, 911
smsEventType
SMSEventInfo_s, 918
smsMaxStorageSizeReq, 918
pMessageMode, 918
storageType, 919
smsMaxStorageSizeResp, 919
freeSlots, 919
maxStorageSize, 919
smsMsgprotocolResp, 920
msgProtocol, 921
smsOnIMS
SMSOnIMS, 922
sMSOnIMSInfo, 923
smsRouteEntry, 923
messageClass, 924
messageType, 924
receiptAction, 924
routeStorage, 924
smsSetRoutesReq, 924
numOfRoutes, 925
pTransferStatusReport, 925
routeList, 925
smsStorage
pack_sms_SLQSSetSmsStorage_t, 676
snr
loc_satelliteInfo, 376
LTESSInfo, 410
lteSSInfo, 410
nas_SccRxInfo, 508
satelliteInfo, 838
SccRxInfo, 839
snrlevel
lteSnrinformation, 408
nas_lteSnrinformation, 471
soMask
DataBearerTech, 214
dataBearerTechnology, 216
qmiWDSDataBearerTechnology, 806
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1283
Source

unpack_swioma_SLQSOMADMGetSessionInfo_t, 1209
source
_getResetInfoNotification, 66
altitudeSrcInfo, 120
altSrcInfo_t, 121
dmsSwiGetResetInfo, 233
unpack_dms_GetNetworkTime_t, 1051
unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, 1063
unpack_dms_SLQSDmsSwiGetResetInfo_t, 1064
sourceIPMask
LibPackTFTIDParams, 362
TFTIDParams, 964
SourceLength
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1209
spc
pack_dms_ResetToFactoryDefaults_t, 598
pack_nas_SetACCOLC_t, 629
pack_wds_SetMobileIPProfile_t, 723
Specific Absorption Rate (SAR), 48
spn
nas_serviceProviderName, 509
nasPLMNNameResp, 564
serviceProviderName, 845
unpack_nas_SLQSGetPLMNName_t, 1126
spnEncoding
nasPLMNNameResp, 564
unpack_nas_SLQSGetPLMNName_t, 1126
spnLength
nas_serviceProviderName, 509
nasPLMNNameResp, 565
serviceProviderName, 845
unpack_nas_SLQSGetPLMNName_t, 1126
srcPortRangeEnd
LibPackTFTIDParams, 362
TFTIDParams, 964
srcPortRangeStart
LibPackTFTIDParams, 362
TFTIDParams, 964
srvCapability
detailSvcInfo, 223
nas_detailSvcInfo, 439
nas_sysInfoCommon, 517
sysInfoCommon, 955
srvCapabilityValid
nas_sysInfoCommon, 517
sysInfoCommon, 955
srvDomain
nas_sysInfoCommon, 517
sysInfoCommon, 955
SrvDomainPref
NASServDomainPrefTlv, 568
srvDomainValid
nas_sysInfoCommon, 517
sysInfoCommon, 955
srvOption
arrSvcOption, 134
voice_arrSvcOption, 1306
srvStatus
detailSvcInfo, 224
GSMSrvStatusInfo, 296
nas_detailSvcInfo, 439
nas_GSMSrvStatusInfo, 448
nas_SrvStatusInfo, 515
SrvStatusInfo, 929
SrvStatusInfo, 929
isPrefDataPath, 929
srvStatus, 929
srxlev
cellParams, 179
gsmCellInfo, 294
nas_cellParams, 431
nas_gsmCellInfo, 447
nas_umtsLTENbrCell, 527
nas_wcdmaCellInfo, 529
umtsLTENbrCell, 1025
wcdmaCellInfo, 1385
ssdatasession_params, 929
action, 931
failureReason, 931
failureReasonv4, 932
failureReasonv6, 932
instanceId, 932
ipfamily, 932
pAuthentication, 932
pPassword, 932
pProfileId3GPP, 932
pProfileId3GPP2, 932
pTechnology, 932
pUsername, 932
rcv4, 932
rcv6, 932
sessionId, 932
v4sessionId, 932
v6sessionId, 932
verbFailReason, 932
verbFailReasonType, 932
stage
uim_refreshevent, 993
UIMRefreshEvent, 1013
StartPDSTrackingSessionExt
qaGobiApiPds.h, 1811
State
nas_NetworkStat1x, 481
nas_NetworkStatEVDO, 483
NetworkStat1x, 582
NetworkStatEVDO, 583
state
LocSetCradleMountReq, 391
omaDmConfigTlv, 587
omaDmConfigTlvExt, 590
omaDmFotaTlv, 592
omaDmFotaTlvExt, 594
pack_loc_SLQSLOCSetCradleMountConfig_t, 626

QosFlowInfoState, 815
QosMap, 816
slqsSessionStateInfo, 890
unpack_dms_GetActivationState_t, 1034
unpack_dms_SLQSUIMGetState_t, 1077
unpack_omaDmConfigTlv_t, 1158
unpack_omaDmFotaTlv_t, 1160
unpack_qos_QosFlowInfoState_t, 1178
statmask
pack_wds_GetPacketStatus_t, 716
StatsMask
TransferStatInd, 971
wds_transferStatInd, 1408
statsMask
TrStatInd, 972
wds_TrStatInd, 1409
StatsPeriod
TransferStatInd, 971
wds_transferStatInd, 1408
statsPeriod
TrStatInd, 972
wds_TrStatInd, 1409
Status
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1209
status
delAssistDataStatus, 221
lteEARFCN, 395
ltePCI, 403
nas_lteEARFCN, 459
nas_ltePCI, 467
nas_wcdmaUARFCN, 535
pack_sms_SetNewSMSCallback_t, 669
QmiCbkLocBestAvailPosInd, 785
QmiCbkLocInjectPositionInd, 789
QmiCbkLocInjectUTCTimeInd, 792
QmiCbkLocSetExtPowerConfigInd, 799
unpack_loc_BestAvailPos_Ind_t, 1089
unpack_loc_DeleteAssistData_Ind_t, 1090
unpack_loc_SetExtPowerConfig_Ind_t, 1101
unpack_loc_SetOperationMode_Ind_t, 1103
unpack_qos_SLQSSetQosNWStatusCallback_ind_t, 1182
unpack_qos_SLQSSetQosStatusCallback_ind_t, 1184
wcdmaUARFCN, 1395
statusChange
UIMStatusChangeInfo, 1019
StopPDSTrackingSession
qaGobiApiPds.h, 1812
storageIndex
FMSImageIdElement, 250
ImageIdElement, 313
storageType
pack_sms_SaveSMS_t, 668
pack_sms_SLQSDeleteSMS_t, 671
pack_sms_SLQSGetSMS_t, 671
pack_sms_SLQSGetSMSList_t, 672
pack_sms_SLQSModifySMSStatus_t, 673
sms_maxStorageSizeReq, 902
smsMaxStorageSizeReq, 919
SMSMemoryInfo, 920
SMSMTMessage, 921
sMSMTMessageInfo, 922
unpack_sms_SLQSWmsMemoryFullCallBack_ind_t, 1204
String
unpack_dms_GetDeviceHardwareRev_t, 1042
unpack_dms_GetDeviceMfr_t, 1043
unpack_dms_GetFSN_t, 1049
unpack_dms_UIMGetICCID_t, 1078
stringSize
unpack_dms_GetDeviceHardwareRev_t, 1042
unpack_dms_GetDeviceMfr_t, 1043
unpack_dms_UIMGetICCID_t, 1078
subAddr
calledPartySubAdd, 146
voice_calledPartySubAdd, 1308
subAddrLen
calledPartySubAdd, 146
voice_calledPartySubAdd, 1308
subAddrType
calledPartySubAdd, 146
voice_calledPartySubAdd, 1308
subnetMask
IPv4Addr, 331
unpack_qos_IPv4Addr_t, 1174
SubnetMaskV4
unpack_wds_SLQSGetRuntimeSettings_t, 1279
subsType
pack_voice_SLQSVoiceBindSubscription_t, 696
voiceBindSubscriptionInfo, 1334
SupUSBComps
unpack_dms_GetUSBComp_t, 1057
SuppOA
CUGInfo, 198
voice_CUGInfo, 1320
SuppPrefCUG
CUGInfo, 198
voice_CUGInfo, 1320
supportedMsgLen
SupportedMsgList, 933
SupportedMsgList, 932
supportedMsgLen, 933
supportedMsgs, 933
supportedMsgs
SupportedMsgList, 933
svInfoMask
loc_satelliteInfo, 376
satelliteInfo, 838
svListLen
loc_satelliteInfo, 376
satelliteInfo, 838
svStatus
loc_satelliteInfo, 376
satelliteInfo, 838

svUsedforFix
qaGobiApiCbk.h, 1605
svUsedforFix_s, 935
gnssSvUsedList, 935
gnssSvUsedList_len, 935
svc
pack_qmi_t, 665
SvcClass
callFWExtInfo, 150
callFWInfo, 151
voice_callFWExtInfo, 1311
voice_callFWInfo, 1312
SvcStatus
callFWExtInfo, 150
callFWInfo, 151
voice_callFWExtInfo, 1311
voice_callFWInfo, 1312
svcType
ccSUPSType, 161
SUPSInfo, 933
voice_ccSUPSType, 1314
sw1
cardResult, 155
uim_cardResult, 981
sw2
cardResult, 155
uim_cardResult, 981
swVerString
pack_dms_SLQSSwiSetHostDevInfo_t, 606
unpack_dms_SLQSSwiGetHostDevInfo_t, 1074
SwiAvmsEventInd
qaGobiApiCbk.h, 1606
SwiDataTypes.h, 2034
BOOL, 2035
BYTE, 2035
CHAR, 2035
FLOAT, 2035
INT32, 2035
INT8, 2035
LPCSTR, 2035
SHORT, 2035
SWI_API, 2035
ULONG, 2035
ULONGLONG, 2035
UNUSEDPARAM, 2035
USHORT, 2035
WORD, 2036
SwiLocGetAutoStart
qaGobiApiLoc.h, 1746
SwiLocGetAutoStartResp, 937
fix_rate, 938
fix_rate_reported, 938
fix_type, 938
fix_type_reported, 938
function, 938
function_reported, 938
max_dist, 938
max_dist_reported, 938
max_time, 938
max_time_reported, 938
SwiLocSetAutoStart
qaGobiApiLoc.h, 1746
SwiLocSetAutoStartReq, 938
fix_rate, 939
fix_type, 940
function, 940
max_dist, 940
max_time, 940
set_fix_rate, 940
set_fix_type, 940
set_function, 940
set_max_dist, 940
set_max_time, 940
swiModemStatusResp, 940
commonInfo, 940
pLTEInfo, 940
SwiOTAMsg
qaGobiApiCbk.h, 1606
SwiOTAMsg_s, 940
data, 941
data_len, 941
pLteNasRelInfo, 941
pTime, 941
type, 941
swiPDPRuntimeSettingsReq, 941
contextId, 942
contextType, 942
swiPDPRuntimeSettingsResp, 942
pAPNName, 944
pBearerId, 944
pContextId, 944
pIPv4Address, 944
pIPv4GWAddress, 944
pIPv6Address, 944
pIPv6GWAddress, 944
pPrDNSIPv4Address, 944
pPrDNSIPv6Address, 944
pPrPCSCFIPv4Address, 944
pPrPCSCFIPv6Address, 944
pSeDNSIPv4Address, 944
pSeDNSIPv6Address, 944
pSePCSCFIPv4Address, 944
pSePCSCFIPv6Address, 944
swiQosFilter, 944
index, 946
pEspSpi, 946
pIPv4DstAddr, 946
pIPv4SrcAddr, 946
pIPv6DstAddr, 946
pIPv6Label, 946
pIPv6SrcAddr, 946
pIPv6TrafCls, 946
pId, 946
pNxtHdrProto, 946
pPrecedence, 946
pTCPDstPort, 946

pTCPSrcPort, 946
pTos, 946
pTranDstPort, 946
pTranSrcPort, 946
pUDPDstPort, 946
pUDPSrcPort, 946
version, 947
swiQosFlow, 947
index, 949
p3GPP2Pri, 949
p3GPPImCn, 949
p3GPPResResidualBER, 949
p3GPPSigInd, 949
p3GPPTraHdlPri, 949
pDataRate, 949
pJitter, 949
pLatency, 949
pLteQci, 949
pMaxAllowedPktSz, 949
pMinPolicedPktSz, 950
pPktErrRate, 950
pProfileId3GPP2, 950
pTokenBucket, 950
pTrafficClass, 950
swiQosGranted, 950
pRxFlow, 950
pTxFlow, 950
swiQosIds, 950
pIds, 950
sz, 951
swiQosModifyReq, 951
id, 951
pRxFilter, 951
pRxFlow, 951
pTxFilter, 951
pTxFlow, 951
swiQosReq, 951
index, 952
pRxFilter, 952
pRxFlow, 952
pTxFilter, 952
pTxFlow, 952
swiRMTrasnferStaticsReq, 952
bResetStatistics, 953
ulMask, 953
swiloc.h, 2036
pack_swiloc_SwiLocGetAutoStart, 2036
pack_swiloc_SwiLocSetAutoStart, 2036
unpack_swiloc_SwiLocGetAutoStart, 2037
unpack_swiloc_SwiLocSetAutoStart, 2037
swioma.h, 2037
pack_swioma_SLQSOMADMAlertCallback, 2038
pack_swioma_SLQSOMADMCancelSession, 2039
pack_swioma_SLQSOMADMGetSessionInfo, 2040
pack_swioma_SLQSOMADMGetSettings, 2040
pack_swioma_SLQSOMADMSendSelection, 2041
pack_swioma_SLQSOMADMSetSettings, 2041
pack_swioma_SLQSOMADMStartSession, 2042
unpack_swioma_SLQSOMADMAlertCallback, 2043
unpack_swioma_SLQSOMADMAlertCallback_ind, 2043
unpack_swioma_SLQSOMADMCancelSession, 2044
unpack_swioma_SLQSOMADMGetSessionInfo, 2044
unpack_swioma_SLQSOMADMGetSettings, 2044
unpack_swioma_SLQSOMADMSendSelection, 2045
unpack_swioma_SLQSOMADMSetSettings, 2045
unpack_swioma_SLQSOMADMStartSession, 2046
switchOption
pack_voice_SLQSVoiceALSSetLineSwitching_t, 695
voiceALSSetLineSwitchInfo, 1333
sysInfoCDMA
CDMASysInfo, 177
nas_CDMASysInfo, 430
sysInfoCommon, 953
isSysForbidden, 955
isSysForbiddenValid, 955
roamStatus, 955
roamStatusValid, 955
srvCapability, 955
srvCapabilityValid, 955
srvDomain, 955
srvDomainValid, 955
sysInfoGSM
GSMSysInfo, 299
nas_GSMSysInfo, 451
sysInfoHDR
HDRSysInfo, 309
nas_HDRSysInfo, 456
sysInfoLTE
LTESysInfo, 413
nas_LTESysInfo, 475
sysInfoWCDMA
nas_WCDMASysInfo, 534
WCDMASysInfo, 1394
sysSelectPrefInfo
qaGobiApiNas.h, 1754
sysSelectPrefParams
qaGobiApiNas.h, 1758
system
loc_satelliteInfo, 376
loc_SV, 378
satelliteInfo, 838
SV, 934
systemDiscontinuities
pack_pds_PDSInjectTimeReference_t, 653
SystemID
qaQmiServingSystemParam, 779
unpack_nas_SLQSGetServingSystem_t, 1130
systemID

CDMASysInfo, 177
nas_CDMASysInfo, 430
systemMode
CommInfo, 188
nas_CommInfo, 433
systemTime
pack_pds_PDSInjectTimeReference_t, 653
sz
swiQosIds, 951
szAPN
_QmiSwiAvmsSetSettingsAPNInfo, 80
szAlertMsg
_QmiSwiAvmsEventReportConfigResp, 74
szDescription
_QmiSwiAvmsEventReportBinaryUpdateSession-InfoResp, 73
szName
_QmiSwiAvmsEventReportBinaryUpdateSession-InfoResp, 73
szPWD
_QmiSwiAvmsSetSettingsAPNInfo, 80
szUname
_QmiSwiAvmsSetSettingsAPNInfo, 80
szVersion
_QmiSwiAvmsEventReportBinaryUpdateSession-InfoResp, 73

t3396_apn
nasTimers, 577
unpack_nas_SLQSNasTimerCallback_ind_t, 1149
t3396_plmn_id
nasTimers, 577
unpack_nas_SLQSNasTimerCallback_ind_t, 1149
t3396_val
nasTimers, 577
unpack_nas_SLQSNasTimerCallback_ind_t, 1149
t_Sv, 956
entries, 956
len, 956
t_gpsTime, 955
gpsTimeOfWeekMs, 955
gpsWeek, 955
t_sensor, 955
aidingIndicatorMask, 955
usageMask, 955
TCPDstPort
unpack_qos_swiQosFilter_t, 1187
TCPSrcPort
unpack_qos_swiQosFilter_t, 1187
TDSCDMAECIOThresh, 956
TDSCDMAECIOThreshListLen
nas_TDSCDMAECIOThresh, 518
TDSCDMAECIOThresh, 956
TDSCDMARSCPThresh, 956
TDSCDMARSCPThreshListLen
nas_TDSCDMARSCPThresh, 518
TDSCDMARSCPThresh, 957
TDSCDMARSSIThresh, 957
TDSCDMARSSIThreshListLen
nas_TDSCDMARSSIThresh, 519
TDSCDMARSSIThresh, 957
TDSCDMASINRCONFThresh, 959
TDSCDMASINRThresh, 959
TDSCDMASINRThreshListLen
nas_TDSCDMASINRThresh, 520
TDSCDMASINRThresh, 960
TDSCDMASigInfoExt, 957
ecio, 958
rscp, 958
rssi, 958
sinr, 958
tFNASwiLTECphyCaIInfo
qaGobiApiCbk.h, 1607
tFNASwiOTAMsg
qaGobiApiCbk.h, 1607
tFNActivationStatus
qaGobiApiCbk.h, 1606
tFNAllCallStatus
qaGobiApiCbk.h, 1607
tFNAsyncRawSend
qaGobiApiCbk.h, 1607
tFNBandPreference
qaGobiApiCbk.h, 1608
tFNBestAvailPos
qaGobiApiCbk.h, 1609
tFNCATEvent
qaGobiApiCbk.h, 1610
tFNCbkUimSlotStatusChangeInd
qaGobiApiCbk.h, 1610
tFNDHCPv4ClientLeaseStatus
qaGobiApiCbk.h, 1612
tFNDTMFEvent
qaGobiApiCbk.h, 1612
tFNDUNCallInfo
qaGobiApiCbk.h, 1612
tFNDataCapabilities
qaGobiApiCbk.h, 1610
tFNDataSysStatus
qaGobiApiCbk.h, 1611
tFNDelAssistData
qaGobiApiCbk.h, 1611
tFNDeviceStateChange
qaGobiApiCbk.h, 1611
tFNEventNmea
qaGobiApiCbk.h, 1612
tFNEventPosition
qaGobiApiCbk.h, 1612
tFNFwDldCompletion
qaGobiApiCbk.h, 1613
tFNGetServer
qaGobiApiCbk.h, 1613
tFNGnssSvInfo
qaGobiApiCbk.h, 1613
tFNHDRPersonaity
qaGobiApiCbk.h, 1613
tFNImsRegMgrConfig
qaGobiApiCbk.h, 1614

tFNImsSIPConfig
qaGobiApiCbk.h, 1615
tFNImsSMSConfig
qaGobiApiCbk.h, 1615
tFNImsUserConfig
qaGobiApiCbk.h, 1615
tFNImsVoIPConfig
qaGobiApiCbk.h, 1615
tFNImsaPdpStatus
qaGobiApiCbk.h, 1614
tFNImsaRatStatus
qaGobiApiCbk.h, 1614
tFNImsaRegStatus
qaGobiApiCbk.h, 1614
tFNImsaSvcStatus
qaGobiApiCbk.h, 1614
tFNInfoRec
qaGobiApiCbk.h, 1615
tFNInjectPosition
qaGobiApiCbk.h, 1616
tFNInjectSensorData
qaGobiApiCbk.h, 1616
tFNInjectTimeStatus
qaGobiApiCbk.h, 1616
tFNInjectUTCTime
qaGobiApiCbk.h, 1616
tFNLUReject
qaGobiApiCbk.h, 1617
tFNMemoryFull
qaGobiApiCbk.h, 1619
tFNMessageWaiting
qaGobiApiCbk.h, 1619
tFNMitiLvlRpt
qaGobiApiCbk.h, 1619
tFNMobileIPStatus
qaGobiApiCbk.h, 1619
tFNModemTempInfo
qaGobiApiCbk.h, 1619
tFNNasTimer
qaGobiApiCbk.h, 1620
tFNNet
qaGobiApiCbk.h, 1620
tFNNetworkReject
qaGobiApiCbk.h, 1620
tFNNetworkTime
qaGobiApiCbk.h, 1620
tFNNewGPS
qaGobiApiCbk.h, 1621
tFNNewNMEA
qaGobiApiCbk.h, 1621
tFNNewRMTransferStatistics
qaGobiApiCbk.h, 1621
tFNNewSMS
qaGobiApiCbk.h, 1622
tFNOMADMState
qaGobiApiCbk.h, 1622
tFNOTASPStatus
qaGobiApiCbk.h, 1624
tFNOpMode
qaGobiApiCbk.h, 1623
tFNPDSState
qaGobiApiCbk.h, 1624
tFNPacketSrvState
qaGobiApiCbk.h, 1624
tFNPower
qaGobiApiCbk.h, 1625
tFNPrivacyChange
qaGobiApiCbk.h, 1625
tFNQosNWStatus
qaGobiApiCbk.h, 1625
tFNQosPriEvent
qaGobiApiCbk.h, 1626
tFNQosStatus
qaGobiApiCbk.h, 1626
tFNRFInfo
qaGobiApiCbk.h, 1627
tFNRankIndicator
qaGobiApiCbk.h, 1627
tFNResetInfo
qaGobiApiCbk.h, 1627
tFNRoamingIndicator
qaGobiApiCbk.h, 1628
tFNSDKTerminated
qaGobiApiCbk.h, 1628
tFNSLQSOMADMAlert
qaGobiApiCbk.h, 1630
tFNSLQSQOSEvent
qaGobiApiCbk.h, 1630
tFNSLQSSessionState
qaGobiApiCbk.h, 1630
tFNSLQSSignalStrengths
qaGobiApiCbk.h, 1631
tFNSLQSWDSEvent
qaGobiApiCbk.h, 1631
tFNSMSEvents
qaGobiApiCbk.h, 1631
tFNSUPSInfo
qaGobiApiCbk.h, 1631
tFNSUPSNotification
qaGobiApiCbk.h, 1631
tFNSensorStreaming
qaGobiApiCbk.h, 1628
tFNServingSystem
qaGobiApiCbk.h, 1628
tFNSetCradleMount
qaGobiApiCbk.h, 1629
tFNSetEngineState
qaGobiApiCbk.h, 1629
tFNSetEventTimeSync
qaGobiApiCbk.h, 1629
tFNSetExtPowerConfig
qaGobiApiCbk.h, 1629
tFNSigInfo
qaGobiApiCbk.h, 1629
tFNSignalStrength
qaGobiApiCbk.h, 1630

tFNSwiAvmEventReport
qaGobiApiCbk.h, 1632
tFNSysInfo
qaGobiApiCbk.h, 1632
tFNSysSelectionPref
qaGobiApiCbk.h, 1632
tFNUIMRefresh
qaGobiApiCbk.h, 1633
tFNUIMStatusChangeInfo
qaGobiApiCbk.h, 1633
tFNUSSDNoWaitIndication
qaGobiApiCbk.h, 1633
tFNUSSDNotification
qaGobiApiCbk.h, 1633
tFNUSSDRelease
qaGobiApiCbk.h, 1634
tFNtransLayerInfo
qaGobiApiCbk.h, 1632
tFNtransNWRegInfo
qaGobiApiCbk.h, 1632
TFTIDParams, 962
destPortRangeEnd, 963
destPortRangeStart, 963
eValid, 963
filterId, 963
flowLabel, 964
IPSECSPI, 964
ipVersion, 964
nextHeader, 964
pSourceIP, 964
sourceIPMask, 964
srcPortRangeEnd, 964
srcPortRangeStart, 964
tosMask, 964
THIRD_INSTANCE
qaGobiApiCbk.h, 1595
TIME_DATE_BUF
qaGobiApiSms.h, 1824
TIME_STAMP_BUF
qaGobiApiSms.h, 1824
TPCause
SMSAsyncRawSend_s, 912
TRMessageTlv
unpack_sms_SetNewSMSCallback_ind_t, 1198
TX_PWR
nas_NetworkStat1x, 481
NetworkStat1x, 582
TXAGCList, 973
pTXAIG, 974
pTXComprSlope, 974
pTXComprThres, 974
pTXExpSlope, 974
pTXExpThres, 974
pTXStaticGain, 974
TXChan
LTEInfo, 398
nas_LTEInfo, 462
tXDroppedCount
unpack_wds_GetPacketStatus_t, 1266
TXOKBytesCount
DUNCallInfoInd, 236
tXOKBytesLastCall
unpack_wds_GetPacketStatus_t, 1266
tXOkBytesCount
unpack_wds_GetPacketStatus_t, 1266
TXPCMIIRFltr, 975
pFlag, 976
pStage0Val, 976
pStage1Val, 976
pStage2Val, 976
pStage3Val, 976
pStage4Val, 976
pStageCnt, 976
tXPacketErrors
unpack_wds_GetPacketStatus_t, 1266
tXPacketOverflows
unpack_wds_GetPacketStatus_t, 1266
tXPacketSuccesses
unpack_wds_GetPacketStatus_t, 1266
Tables, 61
tac
LTEInfoIntrafreq, 400
LTESysInfo, 413
nas_LTEInfoIntrafreq, 465
nas_LTESysInfo, 475
tacValid
LTESysInfo, 413
nas_LTESysInfo, 475
TdsBandCapability
unpack_dms_SLQSGetBandCapability_t, 1068
tdscdmaSigInfoExt, 958
ecio, 959
rscp, 959
rssi, 959
sinr, 959
tech
NWProfile, 587
techName
_packetSrvStatus, 70
unpack_wds_SLQSSetPacketSrvStatusCallback_-
t, 1281
techType
DataBearerTech, 214
Technology
DeviceConfigDetail, 225
fwinfo_s, 253
nas_DeviceConfigDetail, 440
unpack_wds_SLQSGetRuntimeSettings_t, 1279
TechnologyPref
pack_nas_SetNetworkPreference_t, 631
tempData_t, 960
temperature, 961
temperatureDataLen, 961
timeOfFirstSample, 961
timeOffset, 961
timeSource, 961

temperature
CommInfo, 188
nas_CommInfo, 433
tempData_t, 961
tempratureData, 962
temperatureDataLen
tempData_t, 961
tempratureData, 962
tempratureData, 961
temperature, 962
temperatureDataLen, 962
timeOfFirstSample, 962
timeOffset, 962
timeSource, 962
textMsgLength
cdmaMsgEncodingParams, 172
Thermal Mitigation Device(TMD), 58
threshGsmHigh
lteGsmCellInfo, 396
nas_lteGsmCellInfo, 460
threshGsmLow
lteGsmCellInfo, 396
nas_lteGsmCellInfo, 460
threshServingLow
LTEInfoIntrafreq, 400
nas_LTEInfoIntrafreq, 465
threshXHigh
infoInterFreq, 330
nas_infoInterFreq, 457
threshXLow
infoInterFreq, 330
nas_infoInterFreq, 458
threshXhigh
lteWcdmaCellInfo, 414
nas_lteWcdmaCellInfo, 476
threshXlow
lteWcdmaCellInfo, 414
nas_lteWcdmaCellInfo, 476
thresholds
SignalStrengthDataType, 877
thresholdsSize
SignalStrengthDataType, 877
Time
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1210
wcdmaLongMsgDecodingParams, 1388
wcdmaMsgDecodingParams, 1389
time
NASTimeInfoTlv, 576
Time_uncert_ms
GPSStateInfo, 293
unpack_pds_SLQSGetGPSStateInfo_t, 1172
timeBase
pack_pds_SLQSPDSInjectAbsoluteTimeReference_t, 659
timeInfo, 964
day, 965
dayLtSavingAdj, 965
dayOfWeek, 965
hour, 965
minute, 965
month, 965
radioInterface, 966
second, 966
timeZone, 966
TlvPresent, 966
year, 966
TimeLength
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1210
timeMsec
pack_loc_SLQSLOCInjectUTCTime_t, 625
pack_pds_SLQSPDSInjectAbsoluteTimeReference_t, 659
timeOfFirstSample
sensorData, 841
sensorData_t, 842
tempData_t, 961
tempratureData, 962
timeOffset
sensorData, 841
sensorData_t, 842
tempData_t, 961
tempratureData, 962
timeSource
tempData_t, 961
tempratureData, 962
TimeStmp_gps_week
GPSStateInfo, 293
unpack_pds_SLQSGetGPSStateInfo_t, 1172
TimeStmp_tow_ms
GPSStateInfo, 293
unpack_pds_SLQSGetGPSStateInfo_t, 1172
timeSyncRefCounter
QmiCbkLocEventTimeSyncInd, 787
timeTlv
NASQmiCbkNasSwiOTAMessageInd, 566
timeUncMsec
pack_loc_SLQSLOCInjectUTCTime_t, 625
pack_pds_SLQSPDSInjectAbsoluteTimeReference_t, 659
timeZone
nas_timeInfo, 522
timeInfo, 966
timeout
pack_pds_SetPDSDefaults_t, 656
pack_qmi_t, 665
Timers
_QmiSwiAvmsSetSettingsConnectionRetryTimers, 80
_SLQSAVMSSettingsConnectionRetryTimers, 86
timestamp
unpack_dms_GetNetworkTime_t, 1051
timestampAge
pack_loc_SLQSLOCInjectPosition_t, 623
timestampUtc

pack_loc_SLQSLOCInjectPosition_t, 623
timingAdvance
GERANInfo, 255
nas_GERANInfo, 445
TlvPresent
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 73
_QmiSwiAvmsEventReportConfigResp, 74
_QmiSwiAvmsEventReportConnectionRequestResp, 75
_QmiSwiAvmsEventReportDataSessionStatusResp, 75
_QmiSwiAvmsEventReportHTTPStatusResp, 76
_QmiSwiAvmsEventReportNotificationResp, 77
_QmiSwiAvmsEventReportPackageIDResp, 78
_QmiSwiAvmsEventReportRegStatusResp, 78
_QmiSwiAvmsEventReportSessionTypeResp, 79
_QmiSwiAvmsEventReportWAMSParamChangeResp, 79
CatCommonEventTlv, 158
CsgId, 196
DataULongLongTlv, 220
DataULongTlv, 220
dms_ActivationStatusTlv, 228
dms_OperatingModeTlv, 228
eTWSPLMNInfoTlv, 242
IPv4Info, 332
IPv6Info, 335
loc_IPv4Info, 372
loc_IPv6Info, 372
loc_urlAddr, 379
messageModeTlv, 414
nas_PhyCaAggPcellInfo, 487
nas_PhyCaAggScellDlBw, 487
nas_PhyCaAggScellIndex, 488
nas_PhyCaAggScellIndType, 488
nas_PhyCaAggScellInfo, 491
nas_RejectReasonTlv, 500
nas_RFInfoTlv, 501
nas_SccRxInfo, 508
nas_SignalStrengthTlv, 511
nas_SLQSSignalStrengthsTlv, 514
nas_timeInfo, 522
NASBandPreferenceTlv, 535
NASEmergencyModeTlv, 537
NASGWAcqOrderPrefTlv, 546
NASLTEBandPreferenceTlv, 550
NASLteNasReleaseInfoTlv, 551
NASModePreferenceTlv, 552
NASNetSelPreferenceTlv, 552
NASOTAMessageTlv, 558
NASPhyCaAggPcellInfo, 559
NASPhyCaAggScellDlBw, 559
NASPhyCaAggScellIndex, 560
NASPhyCaAggScellIndType, 560
NASPhyCaAggScellInfo, 561
NASPRLPreferenceTlv, 565
NASRoamPreferenceTlv, 567
NASServDomainPrefTlv, 568
NASTimeInfoTlv, 576
newMTMessageTlv, 584
PhyCaAggPcellInfo, 747
PhyCaAggScellDlBw, 748
PhyCaAggScellIndex, 748
PhyCaAggScellIndType, 749
PhyCaAggScellInfo, 752
PlmnID, 754
RoamingInfo, 826
SccRxInfo, 839
sessionInfoTlv, 849
sessionInfoTlvExt, 850
sMSCAddressTlv, 914
sMSEtwsMessageTlv, 916
sMSOnIMSTlv, 923
timeInfo, 966
transferRouteMessageTlv, 970
unpack_pds_SetPdsState_Ind_t, 1168
urlAddr, 1289
wds_DataULongLongTlv, 1396
wds_DataULongTlv, 1397
wds_DHCPLeaseOptTlv, 1398
wds_DHCPLeaseStateTlv, 1398
wds_DHCPProfileIdTlv, 1399
wds_IPv4AdTlv, 1404
Tlvresult
pack_dms_GetCustFeaturesV2_t, 598
pack_dms_SetCustFeaturesV2_t, 602
pack_dms_SetPower_t, 603
pack_dms_SetUSBComp_t, 604
pack_dms_UIMGetICCID_t, 608
pack_fms_GetImagesPreference_t, 612
pack_fms_GetStoredImages_t, 612
pack_fms_SetImagesPreference_t, 613
pack_loc_Delete_Assist_Data_t, 614
pack_loc_EventRegister_t, 616
pack_loc_SetExtPowerState_t, 616
pack_loc_SetOperationMode_t, 617
pack_loc_SLQSLOCGetBestAvailPos_t, 617
pack_loc_Start_t, 628
pack_loc_Stop_t, 628
pack_nas_SetNetworkPreference_t, 632
pack_uim_ChangePin_t, 682
pack_uim_ReadTransparent_t, 683
pack_uim_SetPinProtection_t, 684
pack_uim_UnblockPin_t, 692
pack_uim_VerifyPin_t, 693
unpack_dms_GetBandCapability_t, 1036
unpack_dms_GetCrashAction_t, 1036
unpack_dms_GetCustFeature_t, 1039
unpack_dms_GetCustFeaturesV2_t, 1039
unpack_dms_GetDeviceCap_t, 1040
unpack_dms_GetDeviceHardwareRev_t, 1042
unpack_dms_GetDeviceMfr_t, 1043
unpack_dms_GetDeviceSerialNumbers_t, 1044
unpack_dms_GetFirmwareInfo_t, 1046
unpack_dms_GetFirmwareRevision_t, 1047

unpack_dms_GetFirmwareRevisions_t, 1048
unpack_dms_GetFSN_t, 1049
unpack_dms_GetIMSI_t, 1049
unpack_dms_GetManufacturer_t, 1050
unpack_dms_GetModelID_t, 1050
unpack_dms_GetNetworkTime_t, 1051
unpack_dms_GetOfflineReason_t, 1052
unpack_dms_GetPower_t, 1053
unpack_dms_GetPRLVersion_t, 1054
unpack_dms_GetUSBComp_t, 1057
unpack_dms_GetVoiceNumber_t, 1058
unpack_dms_ResetToFactoryDefaults_t, 1059
unpack_dms_SetActivationStatusCallback_t, 1059
unpack_dms_SetCustFeature_t, 1060
unpack_dms_SetCustFeaturesV2_t, 1060
unpack_dms_SetEventReport_ind_t, 1061
unpack_dms_SetEventReport_t, 1061
unpack_dms_SetFirmwarePreference_t, 1061
unpack_dms_SetPower_t, 1062
unpack_dms_SetUSBComp_t, 1062
unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, 1063
unpack_dms_SLQSDmsSwiGetResetInfo_t, 1064
unpack_dms_SLQSDmsSwiIndicationRegister_t, 1065
unpack_dms_SLQSGetERIFile_t, 1068
unpack_dms_SLQSSwiClearDyingGaspStatistics_t, 1069
unpack_dms_SLQSSwiGetCrashInfo_t, 1069
unpack_dms_SLQSSwiGetDyingGaspCfg_t, 1070
unpack_dms_SLQSSwiGetDyingGaspStatistics_t, 1070
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 1073
unpack_dms_SLQSSwiGetHostDevInfo_t, 1074
unpack_dms_SLQSSwiGetOSInfo_t, 1075
unpack_dms_SLQSSwiGetSerialNoExt_t, 1075
unpack_dms_SLQSSwiSetDyingGaspCfg_t, 1075
unpack_dms_SLQSSwiSetHostDevInfo_t, 1076
unpack_dms_SLQSSwiSetOSInfo_t, 1076
unpack_dms_SLQSUIMGetState_t, 1077
unpack_dms_UIMGetControlKeyStatus_t, 1078
unpack_dms_UIMGetICCID_t, 1078
unpack_dms_UIMGetPINStatus_t, 1080
unpack_dms_UIMSetControlKeyProtection_t, 1080
unpack_dms_UIMSetPINProtection_t, 1081
unpack_dms_UIMUnblockControlKey_t, 1082
unpack_fms_GetImagesPreference_t, 1082
unpack_fms_GetStoredImages_t, 1083
unpack_fms_SetImagesPreference_t, 1083
unpack_loc_BestAvailPos_Ind_t, 1089
unpack_loc_Delete_Assist_Data_t, 1090
unpack_loc_DeleteAssistData_Ind_t, 1090
unpack_loc_EngineState_Ind_t, 1091
unpack_loc_EventNMEA_Ind_t, 1092
unpack_loc_EventRegister_t, 1092
unpack_loc_GetServer_Ind_t, 1093
unpack_loc_GnssSvInfo_Ind_t, 1094
unpack_loc_PositionRpt_Ind_t, 1101
unpack_loc_SetExtPowerConfig_Ind_t, 1101
unpack_loc_SetExtPowerState_t, 1102
unpack_loc_SetOperationMode_Ind_t, 1103
unpack_loc_SetOperationMode_t, 1103
unpack_loc_SLQSLOCGetBestAvailPos_t, 1103
unpack_loc_Start_t, 1104
unpack_loc_Stop_t, 1104
unpack_nas_GetNetworkPreference_t, 1111
unpack_nas_SetNetworkPreference_t, 1119
unpack_nas_SetServingSystemCallback_ind_t, 1120
unpack_nas_SlqsGetLTECphyCAInfo_t, 1122
unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t, 1149
unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t, 1150
unpack_uim_ChangePin_t, 1213
unpack_uim_GetCardStatus_t, 1213
unpack_uim_ReadTransparent_t, 1214
unpack_uim_SetPinProtection_t, 1215
unpack_uim_UnblockPin_t, 1221
unpack_uim_VerifyPin_t, 1222
unpack_wds_SLQSCreateProfile_t, 1268
unpack_wds_SLQSGetProfileSettings_t, 1276
unpack_wds_SLQSSetIPFamilyPreference_t, 1280
TmdDeRegNotMitigationLvlReq, 966
mitigationDevID, 966
mitigationDevIDLen, 966
TmdGetMitigationDevListResp, 966
pMitigationDevList, 967
pMitigationDevListLen, 967
TmdGetMitigationLvlReq, 967
mitigationDevID, 967
mitigationDevIDLen, 967
TmdGetMitigationLvlResp, 967
pCurrentmitigationLvl, 968
pReqMitigationLvl, 968
TmdMitigationLvlIndReq, 968
mitigationDevID, 968
mitigationDevIDLen, 968
TmdRegNotMitigationLvlReq, 968
mitigationDevID, 969
mitigationDevIDLen, 969
toServiceId
BroadcastConfig, 142
sms_BroadcastConfig, 898
toggleMode
lineCtrlInfo, 366
TokenBucket
unpack_qos_swiQosFlow_t, 1191
tokenBucket, 969
bucketSz, 969
peakRate, 969
tokenRate, 969
tokenRate
tokenBucket, 969

unpack_qos_tokenBucket_t, 1192
Tos, 969
mask, 970
val, 970
tosMask
LibPackTFTIDParams, 362
TFTIDParams, 964
total_rx_bytes
sQosStat, 928
unpack_qos_SLQSQosSwiReadDataStats_t, 1181
total_rx_pkt
sQosStat, 929
unpack_qos_SLQSQosSwiReadDataStats_t, 1181
total_tx_bytes
sQosStat, 929
unpack_qos_SLQSQosSwiReadDataStats_t, 1181
total_tx_bytes_drp
sQosStat, 929
unpack_qos_SLQSQosSwiReadDataStats_t, 1181
total_tx_pkt
sQosStat, 929
unpack_qos_SLQSQosSwiReadDataStats_t, 1181
total_tx_pkt_drp
sQosStat, 929
unpack_qos_SLQSQosSwiReadDataStats_t, 1181
TrStatInd, 971
statsMask, 972
statsPeriod, 972
TrackAreaCode
unpack_nas_SLQSGetServingSystem_t, 1130
trackAreaCode
qaQmiServingSystemParam, 779
TrackingStatus
unpack_pds_SetPdsState_Ind_t, 1168
TrafficClass
unpack_qos_swiQosFlow_t, 1191
trafficClass
LibPackUMTSQoS, 365
UMTSMinQoS, 1028
UMTSQoS, 1030
wds_UMTSMinQoS, 1411
trafficPriority
LibPackUMTSQoS, 365
UMTSMinQoS, 1028
UMTSQoS, 1030
wds_UMTSMinQoS, 1411
TranDstPort
unpack_qos_swiQosFilter_t, 1187
TranSrcPort
unpack_qos_swiQosFilter_t, 1187
TransCap
_transLayerinfo, 111
sms_transLayerInfo, 910
transLayerInfo
qaGobiApiSms.h, 1828
transLayerNotification
qaGobiApiCbk.h, 1634
transNWRegInfoNotification
qaGobiApiCbk.h, 1634
TransType
_transLayerinfo, 111
sms_transLayerInfo, 911
transactionID
SMSTransferRouteMTMessage, 926
sMSTransferRouteMTMessageInfo, 927
transferDelay
LibPackUMTSQoS, 365
UMTSMinQoS, 1028
UMTSQoS, 1030
wds_UMTSMinQoS, 1411
TransferRouteMTMessageInfo
transferRouteMessageTlv, 970
transferRouteMessageTlv, 970
TlvPresent, 970
TransferRouteMTMessageInfo, 970
TransferStatInd, 971
StatsMask, 971
StatsPeriod, 971
transferStats
pack_wds_SLQSSetWdsEventCallback_t, 732
TransferStatsDataType, 971
interval, 971
trueIMSI, 972
imsiT1112, 973
imsiTS1, 973
imsiTS2, 973
imsiTaddrNum, 973
mccT, 973
trueSrvStatus
GSMSrvStatusInfo, 296
nas_GSMSrvStatusInfo, 448
tx_bytes
NetStats, 579
sQosFlowStat, 927
unpack_QosFlowStat_t, 1193
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1283
tx_bytes_drp
sQosFlowStat, 927
unpack_QosFlowStat_t, 1193
tx_errors
NetStats, 579
tx_overflows
NetStats, 579
tx_packets
NetStats, 579
tx_pkt
sQosFlowStat, 927
unpack_QosFlowStat_t, 1193
tx_pkt_drp
sQosFlowStat, 927
unpack_QosFlowStat_t, 1193
tx_pkts
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1283
TxDropConutTlv

QmiCbkWdsStatisticsIndState, 801
unpack_RMTransferStatistics_ind_t, 1195
txInfo, 974
isInTraffic, 974
txPower, 974
txOKBytesCount
unpack_wds_SLQSGetDUNCallInfo_t, 1275
TxOkByteCountTlv
QmiCbkWdsStatisticsIndState, 801
unpack_RMTransferStatistics_ind_t, 1195
TxOkConutTlv
QmiCbkWdsStatisticsIndState, 801
unpack_RMTransferStatistics_ind_t, 1195
txPower
nas_txInfo, 523
txInfo, 974
TxQFilter
unpack_qos_QosFlowInfo_t, 1177
TxQFlowGranted
unpack_qos_QosFlowInfo_t, 1177
type
_getResetInfoNotification, 66
dmsSwiGetResetInfo, 233
pack_wds_GetDefaultProfileNum_t, 713
pack_wds_SetDefaultProfileNum_t, 719
SwiOTAMsg_s, 941
unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, 1063
unpack_dms_SLQSDmsSwiGetResetInfo_t, 1064
unpack_qmi_t, 1173
typeSize
unpack_sms_GetSMSCAddress_t, 1196

u16PRLVersion
unpack_dms_GetPRLVersion_t, 1054
u8PRLPreference
unpack_dms_GetPRLVersion_t, 1054
UATISIZE
qaGobiApiNas.h, 1753
UDPDstPort
unpack_qos_swiQosFilter_t, 1187
UDPSrcPort
unpack_qos_swiQosFilter_t, 1187
UIMAuthenticateReq, 1000
authData, 1001
pIndicationToken, 1001
sessionInfo, 1001
UIMAuthenticateResp, 1001
pAuthenticateResult, 1002
pCardResult, 1002
pIndicationToken, 1002
UIMChangePinReq, 1002
changePIN, 1002
pIndicationToken, 1002
pKeyReferenceID, 1002
sessionInfo, 1002
UIMDepersonalizationReq, 1003
depersonilisationInfo, 1003
UIMDepersonalizationResp, 1003
pRemainingRetries, 1003
UIMEventRegisterReqResp, 1003
eventMask, 1004
UIMGetCardStatusResp, 1004
pCardStatus, 1004
pHotSwapStatus, 1004
UIMGetConfigurationReq, 1004
pConfigurationMask, 1005
UIMGetConfigurationResp, 1005
pAutoSelection, 1006
pHaltSubscription, 1006
pPersonalizationStatus, 1006
UIMGetFileAttributesReq, 1006
fileIndex, 1006
pIndicationToken, 1006
sessionInfo, 1006
UIMGetFileAttributesResp, 1006
pCardResult, 1007
pFileAttributes, 1007
pIndicationToken, 1007
UIMGetSlotsStatusResp, 1007
pNumberOfPhySlot, 1007
pUimSlotsStatus, 1007
UIMPinResp, 1008
pEncryptedPIN1, 1008
pIndicationToken, 1008
pRemainingRetries, 1008
UIMPowerDownReq, 1008
slot, 1009
UIMPowerUpReq, 1009
pIgnoreHotSwapSwitch, 1009
slot, 1009
UIMReadTransparentReq, 1009
fileIndex, 1010
pEncryptData, 1010
pIndicationToken, 1010
readTransparent, 1010
sessionInfo, 1010
UIMReadTransparentResp, 1010
pCardResult, 1011
pEncryptedData, 1011
pIndicationToken, 1011
pReadResult, 1011
UIMRefreshCompleteReq, 1011
refreshComplete, 1012
sessionInfo, 1012
UIMRefreshEvent, 1012
aid, 1013
aidLength, 1013
arrfileInfo, 1013
mode, 1013
numOfFiles, 1013
sessionType, 1013
stage, 1013
UIMRefreshGetLastEventReq, 1013
sessionInfo, 1014
UIMRefreshGetLastEventResp, 1014
pRefreshEvent, 1014

UIMRefreshOKReq, 1014
OKtoRefresh, 1015
sessionInfo, 1015
UIMRefreshRegisterReq, 1015
regRefresh, 1015
sessionInfo, 1015
UIMSessionInformation, 1015
aid, 1016
aidLength, 1016
sessionType, 1016
UIMSetPinProtectionReq, 1016
pIndicationToken, 1017
pKeyReferenceID, 1017
pinProtection, 1017
sessionInfo, 1017
UIMSlotStatus, 1017
bICCID, 1018
bICCIDLength, 1018
bLogicalSlot, 1018
uPhyCardStatus, 1018
uPhySlotStatus, 1018
UIMSlotStatusChangeInfo, 1019
bNumberOfPhySlots, 1019
slotsstatusChange, 1019
UIMSlotsStatus, 1017
uimSlotStatus, 1017
UIMStatusChangeInfo, 1019
statusChange, 1019
UIMSwitchSlotReq, 1019
bLogicalSlot, 1020
ulPhysicalSlot, 1020
UIMUnblockPinReq, 1020
pIndicationToken, 1021
pKeyReferenceID, 1021
sessionInfo, 1021
unblockPIN, 1021
UIMVerifyPinReq, 1021
pEncryptedPIN1, 1022
pIndicationToken, 1022
pKeyReferenceID, 1022
sessionInfo, 1022
verifyPIN, 1022
ULONG
SwiDataTypes.h, 2035
ULONGLONG
SwiDataTypes.h, 2035
UMTSGrantedQoS
unpack_wds_SLQSGetRuntimeSettings_t, 1279
UMTSInfo, 1022
cellID, 1023
ecio, 1023
geranInst, 1023
GeranInstInfo, 1023
lac, 1023
plmn, 1023
psc, 1024
rscp, 1024
UMTSInstInfo, 1024
uarfcn, 1024
umtsInst, 1024
UMTSInstInfo
nas_UMTSInfo, 525
UMTSInfo, 1024
UMTSLTENbrCell
nas_WCDMAInfoLTENeighborCell, 530
WCDMAInfoLTENeighborCell, 1386
UMTSMinQoS, 1025
deliveryErrSDU, 1027
grntDownlinkBitrate, 1027
grntUplinkBitrate, 1028
maxDownlinkBitrate, 1028
maxSDUSize, 1028
maxUplinkBitrate, 1028
qosDeliveryOrder, 1028
resBerRatio, 1028
sduErrorRatio, 1028
trafficClass, 1028
trafficPriority, 1028
transferDelay, 1028
UMTSQoS, 1028
deliveryErrSDU, 1030
grntDownlinkBitrate, 1030
grntUplinkBitrate, 1030
maxDownlinkBitrate, 1030
maxSDUSize, 1030
maxUplinkBitrate, 1030
qosDeliveryOrder, 1030
resBerRatio, 1030
sduErrorRatio, 1030
trafficClass, 1030
trafficPriority, 1030
transferDelay, 1030
UMTSReqQoS
LibPackUMTSReqQoSSigInd, 365
UMTSReqQoSSigInd, 1031
UMTSReqQoSSigInd, 1030
SigInd, 1031
UMTSReqQoS, 1031
UMTSinstInfo, 1024
umtsEcio, 1024
umtsPsc, 1024
umtsRscp, 1024
umtsUarfcn, 1024
UNIQUE_ID_LEN
dms.h, 1443
qaGobiApiFms.h, 1715
UNUSEDPARAM
common.h, 1433
SwiDataTypes.h, 2035
uPhyCardStatus
slot_t, 878
UIMSlotStatus, 1018
uPhySlotStatus
slot_t, 878
UIMSlotStatus, 1018
uResult

sGetDeviceSeriesResult, 874
USBComp
pack_dms_SetUSBComp_t, 604
unpack_dms_GetUSBComp_t, 1057
USBCompConfig, 1289
pUSBComp, 1290
USBCompParams, 1290
pNumSupUSBComps, 1292
pSupUSBComps, 1292
pUSBComp, 1292
USHORT
SwiDataTypes.h, 2035
USSD_DCS_8BIT
qaGobiApiCbk.h, 1595
USSD_DCS_ASCII
qaGobiApiCbk.h, 1595
USSD_DCS_UCS2
qaGobiApiCbk.h, 1595
USSDNoWaitIndicationInfo, 1292
pAlphaIdentifier, 1293
pError, 1293
pFailureCause, 1293
pUSSDData, 1293
USSDRespFNetwork, 1293
pRespData, 1293
pTypeCode, 1293
USSInfo, 1293
ussDCS, 1294
ussData, 1294
ussLen, 1294
USSInformation
pack_voice_SLQSVoiceOrigUSSDNoWait_t, 706
voiceOrigUSSDNoWaitInfo, 1366
USSResp, 1294
pAlphaIDInfo, 1294
pCCSuppsType, 1294
pCallId, 1294
pCcResultType, 1294
pUSSDInfo, 1295
pfailureCause, 1294
UUSData
UUSInfo, 1296
voice_UUSInfo, 1333
UUSDatalen
UUSInfo, 1296
voice_UUSInfo, 1333
UUSDcs
UUSInfo, 1296
voice_UUSInfo, 1333
UUSInfo, 1295
UUSData, 1296
UUSDatalen, 1296
UUSDcs, 1296
UUSType, 1296
UUSType
UUSInfo, 1296
voice_UUSInfo, 1333
uarfcn
lteWcdmaCellInfo, 414
nas_lteWcdmaCellInfo, 476
nas_UMTSInfo, 525
nas_wcdmaUARFCN, 535
UMTSInfo, 1024
wcdmaUARFCN, 1395
ueInIdle
LTEInfoInterfreq, 398
LTEInfoIntrafreq, 400
LTEInfoNeighboringGSM, 401
LTEInfoNeighboringWCDMA, 402
nas_LTEInfoInterfreq, 463
nas_LTEInfoIntrafreq, 465
nas_LTEInfoNeighboringGSM, 466
nas_LTEInfoNeighboringWCDMA, 467
uim.h, 2046
MAX_ICCID_LENGTH, 2049
MAX_NO_OF_SLOTS, 2049
MAX_SLOTS_STATUS, 2049
pack_uim_ChangePin, 2050
pack_uim_GetCardStatus, 2050
pack_uim_ReadTransparent, 2050
pack_uim_SLQSUIMAuthenticate, 2051
pack_uim_SLQSUIMDepersonalization, 2051
pack_uim_SLQSUIMEventRegister, 2052
pack_uim_SLQSUIMGetConfiguration, 2052
pack_uim_SLQSUIMGetFileAttributes, 2052
pack_uim_SLQSUIMGetSlotsStatus, 2053
pack_uim_SLQSUIMPowerDown, 2053
pack_uim_SLQSUIMPowerUp, 2054
pack_uim_SLQSUIMRefreshComplete, 2054
pack_uim_SLQSUIMRefreshGetLastEvent, 2054
pack_uim_SLQSUIMRefreshOK, 2055
pack_uim_SLQSUIMRefreshRegister, 2055
pack_uim_SLQSUIMReset, 2055
pack_uim_SLQSUIMSwitchSlot, 2056
pack_uim_SetPinProtection, 2051
pack_uim_UnblockPin, 2056
pack_uim_VerifyPin, 2056
unpack_uim_ChangePin, 2057
unpack_uim_GetCardStatus, 2057
unpack_uim_ReadTransparent, 2058
unpack_uim_SLQSUIMAuthenticate, 2059
unpack_uim_SLQSUIMDepersonalization, 2059
unpack_uim_SLQSUIMEventRegister, 2059
unpack_uim_SLQSUIMGetConfiguration, 2060
unpack_uim_SLQSUIMGetFileAttributes, 2060
unpack_uim_SLQSUIMGetSlotsStatus, 2060
unpack_uim_SLQSUIMPowerDown, 2061
unpack_uim_SLQSUIMPowerUp, 2061
unpack_uim_SLQSUIMRefreshComplete, 2061
unpack_uim_SLQSUIMRefreshGetLastEvent, 2062
unpack_uim_SLQSUIMRefreshOK, 2062
unpack_uim_SLQSUIMRefreshRegister, 2062
unpack_uim_SLQSUIMReset, 2063
unpack_uim_SLQSUIMSetStatusChangeCallBack_ind, 2063

unpack_uim_SLQSUIMSwitchSlot, 2064
unpack_uim_SetPinProtection, 2058
unpack_uim_SetUimSlotStatusChangeCallback_ind, 2058
unpack_uim_UnblockPin, 2064
unpack_uim_VerifyPin, 2064
uim_UIMSessionInformation, 998
aid, 999
aidLength, 999
sessionType, 999
uim_appStatus, 976
aidLength, 978
aidVal, 979
appState, 979
appType, 979
persoFeature, 979
persoRetries, 979
persoState, 979
persoUnblockRetries, 979
pin1Retries, 979
pin1State, 979
pin2Retries, 979
pin2State, 979
puk1Retries, 979
puk2Retries, 979
univPin, 979
uim_authenticateResult, 979
content, 979
contentLen, 979
uim_authenticationData, 980
context, 980
data, 980
dataLen, 980
uim_cardResult, 981
sw1, 981
sw2, 981
uim_cardStatus, 981
index1xPri, 982
index1xSec, 982
indexGwPri, 982
indexGwSec, 982
numSlot, 982
SlotInfo, 982
uim_changeUIMPIN, 982
oldPINLen, 983
oldPINVal, 983
pinID, 983
pinLen, 983
pinVal, 983
uim_depersonalizationInformation, 983
ckLen, 984
ckVal, 984
feature, 984
operation, 984
uim_encryptedPIN1, 984
pin1Len, 985
pin1Val, 985
uim_fileAttributes, 985
fileID, 988
fileSize, 988
fileType, 988
rawLen, 988
rawValue, 988
recordCount, 988
recordSize, 988
secActivate, 988
secActivateMask, 988
secDeactivate, 988
secDeactivateMask, 988
secIncrease, 988
secIncreaseMask, 988
secRead, 988
secReadMask, 988
secWrite, 988
secWriteMask, 988
uim_fileInfo, 988
fileID, 989
path, 989
pathLen, 989
uim_hotSwapStatus, 989
hotSwap, 989
hotSwapLength, 989
uim_personalizationStatus, 989
feature, 990
numFeatures, 990
unblockLeft, 990
verifyLeft, 990
uim_readResult, 990
content, 991
contentLen, 991
uim_readTransparentInfo, 991
length, 991
offset, 991
uim_refreshevent, 991
aid, 993
aidLength, 993
arrfileInfo, 993
mode, 993
numOfFiles, 993
sessionType, 993
stage, 993
uim_registerRefresh, 993
arrfileInfo, 994
numFiles, 994
registerFlag, 994
voteForInit, 994
uim_remainingRetries, 994
unblockLeft, 994
verifyLeft, 994
uim_sessionInformation, 995
aid, 995
aidLength, 995
sessionType, 995
uim_setPINProtection, 995
pinID, 996
pinLength, 996

pinOperation, 996
pinValue, 996
uim_slotInfo, 996
AppStatus, 998
cardState, 998
errorState, 998
numApp, 998
upinRetries, 998
upinState, 998
upukRetries, 998
uim_unblockUIMPIN, 999
newPINLen, 999
newPINVal, 999
pinID, 999
pukLen, 999
pukVal, 1000
uim_verifyUIMPIN, 1000
pinID, 1000
pinLen, 1000
pinVal, 1000
uimSlotStatus
slots_t, 882
UIMSlotsStatus, 1017
ulData
DataULongTlv, 220
wds_DataULongTlv, 1397
ulMask
rmTrasnferStaticsReq, 825
swiRMTrasnferStaticsReq, 953
ulMax
_QmiSwiAvmsSetSettingsPeriodInfo, 80
ulMin
_QmiSwiAvmsSetSettingsPeriodInfo, 80
ulPhysicalSlot
pack_uim_SLQSUIMSwitchSlot_t, 691
UIMSwitchSlotReq, 1020
ulPkgDownloadComplete
_QmiSwiAvmsEventReportBinaryUpdateSession-InfoResp, 73
ulPkgDownloadSize
_QmiSwiAvmsEventReportBinaryUpdateSession-InfoResp, 73
ullData
DataULongLongTlv, 220
wds_DataULongLongTlv, 1397
umtsEcio
nas_UMTSinstInfo, 526
UMTSinstInfo, 1024
umtsInst
nas_UMTSInfo, 525
UMTSInfo, 1024
umtsLTENbrCell, 1024
cellIsTDD, 1025
earfcn, 1025
pci, 1025
rsrp, 1025
rsrq, 1025
srxlev, 1025
umtsLTENbrCellLen
nas_WCDMAInfoLTENeighborCell, 530
WCDMAInfoLTENeighborCell, 1386
umtsPsc
nas_UMTSinstInfo, 526
UMTSinstInfo, 1024
umtsRscp
nas_UMTSinstInfo, 526
UMTSinstInfo, 1024
umtsUarfcn
nas_UMTSinstInfo, 526
UMTSinstInfo, 1024
UnPackGetProfileSettingOut, 1288
curProfile, 1288
pExtErrCode, 1288
unblockLeft
personalizationStatus, 746
remainingRetries, 820
uim_personalizationStatus, 990
uim_remainingRetries, 994
unblockPIN
UIMUnblockPinReq, 1021
unblockRetriesLeft
unpack_dms_UIMGetControlKeyStatus_t, 1078
unpack_dms_UIMSetPINProtection_t, 1081
unpack_dms_UIMUnblockControlKey_t, 1082
unblockUIMPIN, 1031
newPINLen, 1032
newPINVal, 1032
pinID, 1032
pukLen, 1032
pukVal, 1032
uniqueID
CurrImageInfo, 203
image_info_t, 311
univPin
appStats, 124
appStatus, 127
uim_appStatus, 979
UniversalTime, 1032
day, 1033
dayOfWeek, 1033
hour, 1033
minute, 1033
month, 1033
second, 1033
year, 1033
universalTime
nasNetworkTime, 556
unpack_nas_SLQSNasNetworkTimeCallBack_ind-_t, 1146
unpack_QosFlowStat_t, 1193
bearerId, 1193
tx_bytes, 1193
tx_bytes_drp, 1193
tx_pkt, 1193
tx_pkt_drp, 1193
unpack_RMTransferStatistics_ind_t, 1194

RxDropConutTlv, 1195
RxOkByteCountTlv, 1195
RxOkConutTlv, 1195
TxDropConutTlv, 1195
TxOkByteCountTlv, 1195
TxOkConutTlv, 1195
unpack_dms_ActivateAutomatic
dms.h, 1466
unpack_dms_GetActivationState
dms.h, 1466
unpack_dms_GetActivationState_t, 1033
state, 1034
unpack_dms_GetBandCapability
dms.h, 1467
unpack_dms_GetBandCapability_t, 1034
BandCapability, 1035
Tlvresult, 1036
unpack_dms_GetCrashAction
dms.h, 1467
unpack_dms_GetCrashAction_t, 1036
DevCrashState, 1036
Tlvresult, 1036
unpack_dms_GetCustFeature
dms.h, 1467
unpack_dms_GetCustFeature_t, 1036
DHCPRelayEnabled, 1038
DisableIMSI, 1038
GPSLPM, 1038
GPSSel, 1038
GpsEnable, 1038
IPFamSupport, 1038
IsVoiceEnabled, 1038
RMAutoConnect, 1039
SMSSupport, 1039
Tlvresult, 1039
unpack_dms_GetCustFeaturesV2
dms.h, 1468
unpack_dms_GetCustFeaturesV2_t, 1039
GetCustomFeatureV2, 1039
Tlvresult, 1039
unpack_dms_GetDeviceCap
dms.h, 1468
unpack_dms_GetDeviceCap_t, 1039
DataServiceCapability, 1040
MaxRXChannelRate, 1040
MaxTXChannelRate, 1040
RadioIfaces, 1040
RadioIfacesSize, 1040
SimCapability, 1040
Tlvresult, 1040
unpack_dms_GetDeviceCapabilities
dms.h, 1468
unpack_dms_GetDeviceCapabilities_t, 1040
dataServiceCaCapability, 1041
maxRxChannelRate, 1041
maxTxChannelRate, 1041
RadioIfaces, 1042
radioIfacesSize, 1042
simCapability, 1042
unpack_dms_GetDeviceHardwareRev
dms.h, 1469
unpack_dms_GetDeviceHardwareRev_t, 1042
String, 1042
stringSize, 1042
Tlvresult, 1042
unpack_dms_GetDeviceMfr
dms.h, 1469
unpack_dms_GetDeviceMfr_t, 1042
String, 1043
stringSize, 1043
Tlvresult, 1043
unpack_dms_GetDeviceSerialNumbers
dms.h, 1469
unpack_dms_GetDeviceSerialNumbers_t, 1043
ESNString, 1044
esnSize, 1044
IMEIString, 1044
imeiSize, 1044
imeiSvnSize, 1044
ImeiSvnString, 1044
MEIDString, 1044
meidSize, 1044
Tlvresult, 1044
unpack_dms_GetFSN
dms.h, 1471
unpack_dms_GetFSN_t, 1048
String, 1049
Tlvresult, 1049
unpack_dms_GetFirmwareInfo
dms.h, 1470
unpack_dms_GetFirmwareInfo_t, 1044
appversion_str, 1046
bootversion_str, 1046
carrier_str, 1046
cur_carr_name, 1046
cur_carr_rev, 1046
modelid_str, 1046
packageid_str, 1046
priversion_str, 1046
sku_str, 1046
Tlvresult, 1046
unpack_dms_GetFirmwareRevision
dms.h, 1470
unpack_dms_GetFirmwareRevision_t, 1046
AMSSString, 1047
amssSize, 1047
PRIString, 1047
Tlvresult, 1047
unpack_dms_GetFirmwareRevisions
dms.h, 1470
unpack_dms_GetFirmwareRevisions_t, 1047
AMSSString, 1048
amssSize, 1048
bootSize, 1048
BootString, 1048
PRIString, 1048

priSize, 1048
Tlvresult, 1048
unpack_dms_GetHardwareRevision
dms.h, 1471
unpack_dms_GetHardwareRevision_t, 1049
hwVer, 1049
unpack_dms_GetIMSI
dms.h, 1471
unpack_dms_GetIMSI_t, 1049
imsi, 1049
Tlvresult, 1049
unpack_dms_GetManufacturer
dms.h, 1472
unpack_dms_GetManufacturer_t, 1050
manufacturer, 1050
Tlvresult, 1050
unpack_dms_GetModelID
dms.h, 1472
unpack_dms_GetModelID_t, 1050
modelid, 1050
Tlvresult, 1050
unpack_dms_GetNetworkTime
dms.h, 1472
unpack_dms_GetNetworkTime_t, 1051
source, 1051
timestamp, 1051
Tlvresult, 1051
unpack_dms_GetOfflineReason
dms.h, 1473
unpack_dms_GetOfflineReason_t, 1051
pReasonMask, 1052
pbPlatform, 1052
Tlvresult, 1052
unpack_dms_GetPRLVersion
dms.h, 1473
unpack_dms_GetPRLVersion_t, 1054
Tlvresult, 1054
u16PRLVersion, 1054
u8PRLPreference, 1054
unpack_dms_GetPower
dms.h, 1473
unpack_dms_GetPower_t, 1052
HardwareControlledMode, 1053
OfflineReason, 1053
OperationMode, 1053
Tlvresult, 1053
unpack_dms_GetSerialNumbers
dms.h, 1474
unpack_dms_GetSerialNumbers_t, 1054
esn, 1055
imei_no, 1055
imeisv_svn, 1055
meid, 1055
unpack_dms_GetUSBComp
dms.h, 1474
unpack_dms_GetUSBComp_t, 1055
NumSupUSBComps, 1057
SupUSBComps, 1057
Tlvresult, 1057
USBComp, 1057
unpack_dms_GetVoiceNumber
dms.h, 1474
unpack_dms_GetVoiceNumber_t, 1057
MIN, 1058
minSize, 1058
Tlvresult, 1058
VoiceNumber, 1058
voiceNumberSize, 1058
unpack_dms_ResetToFactoryDefaults
dms.h, 1475
unpack_dms_ResetToFactoryDefaults_t, 1058
Tlvresult, 1059
unpack_dms_SLQSDmsSwiGetResetInfo
dms.h, 1478
unpack_dms_SLQSDmsSwiGetResetInfo_Ind
dms.h, 1478
unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, 1062
source, 1063
Tlvresult, 1063
type, 1063
unpack_dms_SLQSDmsSwiGetResetInfo_t, 1063
source, 1064
Tlvresult, 1064
type, 1064
unpack_dms_SLQSDmsSwiIndicationRegister
dms.h, 1479
unpack_dms_SLQSDmsSwiIndicationRegister_t, 1064
Tlvresult, 1065
unpack_dms_SLQSGetBandCapability
dms.h, 1479
unpack_dms_SLQSGetBandCapability_t, 1065
bandCapability, 1068
LteBandCapability, 1068
TdsBandCapability, 1068
unpack_dms_SLQSGetERIFile
dms.h, 1480
unpack_dms_SLQSGetERIFile_t, 1068
eriFile, 1068
Tlvresult, 1068
unpack_dms_SLQSSwiClearDyingGaspStatistics
dms.h, 1480
unpack_dms_SLQSSwiClearDyingGaspStatistics_t, 1068
Tlvresult, 1069
unpack_dms_SLQSSwiGetCrashInfo
dms.h, 1480
unpack_dms_SLQSSwiGetCrashInfo_t, 1069
crashInfoParam, 1069
Tlvresult, 1069
unpack_dms_SLQSSwiGetDyingGaspCfg
dms.h, 1481
unpack_dms_SLQSSwiGetDyingGaspCfg_t, 1069
pGetDyingGaspCfg, 1070
Tlvresult, 1070
unpack_dms_SLQSSwiGetDyingGaspStatistics
dms.h, 1481

unpack_dms_SLQSSwiGetDyingGaspStatistics_t, 1070
pGetDyingGaspStatistics, 1070
Tlvresult, 1070
unpack_dms_SLQSSwiGetFirmwareCurr
dms.h, 1481
unpack_dms_SLQSSwiGetFirmwareCurr_t, 1070
carrier, 1071
fwvers, 1071
numEntries, 1071
pCurrImgInfo, 1071
pkgver, 1071
priver, 1071
unpack_dms_SLQSSwiGetFwUpdateStatus
dms.h, 1482
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 1072
imgType, 1073
logString, 1073
refData, 1073
refString, 1073
ResCode, 1073
Tlvresult, 1073
unpack_dms_SLQSSwiGetHostDevInfo
dms.h, 1482
unpack_dms_SLQSSwiGetHostDevInfo_t, 1073
manString, 1074
modelString, 1074
plasmaIDString, 1074
swVerString, 1074
Tlvresult, 1074
unpack_dms_SLQSSwiGetOSInfo
dms.h, 1482
unpack_dms_SLQSSwiGetOSInfo_t, 1074
nameString, 1074
Tlvresult, 1075
versionString, 1075
unpack_dms_SLQSSwiGetSerialNoExt
dms.h, 1483
unpack_dms_SLQSSwiGetSerialNoExt_t, 1075
meidString, 1075
Tlvresult, 1075
unpack_dms_SLQSSwiSetDyingGaspCfg
dms.h, 1483
unpack_dms_SLQSSwiSetDyingGaspCfg_t, 1075
Tlvresult, 1075
unpack_dms_SLQSSwiSetHostDevInfo
dms.h, 1483
unpack_dms_SLQSSwiSetHostDevInfo_t, 1076
Tlvresult, 1076
unpack_dms_SLQSSwiSetOSInfo
dms.h, 1484
unpack_dms_SLQSSwiSetOSInfo_t, 1076
Tlvresult, 1076
unpack_dms_SLQSUIMGetState
dms.h, 1484
unpack_dms_SLQSUIMGetState_t, 1076
state, 1077
Tlvresult, 1077
unpack_dms_SetActivationStatusCallback
dms.h, 1475
unpack_dms_SetActivationStatusCallback_t, 1059
Tlvresult, 1059
unpack_dms_SetCrashAction
dms.h, 1475
unpack_dms_SetCrashAction_t, 1059
notused, 1059
unpack_dms_SetCustFeature
dms.h, 1476
unpack_dms_SetCustFeature_t, 1059
Tlvresult, 1060
unpack_dms_SetCustFeaturesV2
dms.h, 1476
unpack_dms_SetCustFeaturesV2_t, 1060
Tlvresult, 1060
unpack_dms_SetEventReport
dms.h, 1476
unpack_dms_SetEventReport_ind
dms.h, 1477
unpack_dms_SetEventReport_ind_t, 1060
ActivationStatusTlv, 1061
OperatingModeTlv, 1061
Tlvresult, 1061
unpack_dms_SetEventReport_t, 1061
Tlvresult, 1061
unpack_dms_SetFirmwarePreference
dms.h, 1477
unpack_dms_SetFirmwarePreference_t, 1061
Tlvresult, 1061
unpack_dms_SetPower
dms.h, 1477
unpack_dms_SetPower_t, 1062
Tlvresult, 1062
unpack_dms_SetUSBComp
dms.h, 1478
unpack_dms_SetUSBComp_t, 1062
Tlvresult, 1062
unpack_dms_UIMChangePIN
dms.h, 1484
unpack_dms_UIMGetControlKeyStatus
dms.h, 1485
unpack_dms_UIMGetControlKeyStatus_t, 1077
facilityState, 1078
Tlvresult, 1078
unblockRetriesLeft, 1078
verifyRetriesLeft, 1078
unpack_dms_UIMGetICCID
dms.h, 1485
unpack_dms_UIMGetICCID_t, 1078
String, 1078
stringSize, 1078
Tlvresult, 1078
unpack_dms_UIMGetPINStatus
dms.h, 1485
unpack_dms_UIMGetPINStatus_t, 1078
p1Status, 1080
p1UnblockRetriesLeft, 1080
p1VerifyRetriesLeft, 1080

p2Status, 1080
p2UnblockRetriesLeft, 1080
p2VerifyRetriesLeft, 1080
Tlvresult, 1080
unpack_dms_UIMSetControlKeyProtection
dms.h, 1486
unpack_dms_UIMSetControlKeyProtection_t, 1080
Tlvresult, 1080
verifyRetriesLeft, 1080
unpack_dms_UIMSetPINProtection
dms.h, 1486
unpack_dms_UIMSetPINProtection_t, 1080
Tlvresult, 1081
unblockRetriesLeft, 1081
verifyRetriesLeft, 1081
unpack_dms_UIMUnblockControlKey
dms.h, 1486
unpack_dms_UIMUnblockControlKey_t, 1081
Tlvresult, 1082
unblockRetriesLeft, 1082
unpack_dms_UIMUnblockPIN
dms.h, 1487
unpack_dms_UIMVerifyPIN
dms.h, 1487
unpack_dms_ValidateSPC
dms.h, 1488
unpack_fms_GetImagesPreference
fms.h, 1490
unpack_fms_GetImagesPreference_t, 1082
ImageListSize, 1082
pImageList, 1082
Tlvresult, 1082
unpack_fms_GetStoredImages
fms.h, 1491
unpack_fms_GetStoredImages_t, 1082
imageList, 1083
imagelistSize, 1083
Tlvresult, 1083
unpack_fms_SetImagesPreference
fms.h, 1491
unpack_fms_SetImagesPreference_t, 1083
ImageTypes, 1083
ImageTypesSize, 1083
Tlvresult, 1083
unpack_loc_BestAvailPos_Ind
loc.h, 1501
unpack_loc_BestAvailPos_Ind_t, 1084
pAltitudeWrtEllipsoid, 1088
pAltitudeWrtMeanSeaLevel, 1088
pGpsTime, 1088
pHeading, 1088
pHeadingUnc, 1088
pHorCirConf, 1088
pHorEllpConf, 1088
pHorReliability, 1088
pHorUncCircular, 1088
pHorUncEllipseOrientAzimuth, 1088
pHorUncEllipseSemiMajor, 1088
pHorUncEllipseSemiMinor, 1088
pLatitude, 1089
pLongitude, 1089
pMagneticDeviation, 1089
pPrecisionDilution, 1089
pSensorDataUsage, 1089
pSpeedHorizontal, 1089
pSpeedUnc, 1089
pSpeedVertical, 1089
pSpeedVerticalUnc, 1089
pSvUsedforFix, 1089
pTechnologyMask, 1089
pTimeSrc, 1089
pTimeUnc, 1089
pTimestampUtc, 1089
pVertConfidence, 1089
pVertReliability, 1089
pVertUnc, 1089
pXid, 1089
status, 1089
Tlvresult, 1089
unpack_loc_Delete_Assist_Data_t, 1089
Tlvresult, 1090
unpack_loc_DeleteAssistData
loc.h, 1501
unpack_loc_DeleteAssistData_Ind
loc.h, 1502
unpack_loc_DeleteAssistData_Ind_t, 1090
status, 1090
Tlvresult, 1090
unpack_loc_EngineState_Ind
loc.h, 1502
unpack_loc_EngineState_Ind_t, 1091
engineState, 1091
Tlvresult, 1091
unpack_loc_EventNMEA_Ind
loc.h, 1503
unpack_loc_EventNMEA_Ind_t, 1091
NMEAData, 1092
Tlvresult, 1092
unpack_loc_EventRegister
loc.h, 1503
unpack_loc_EventRegister_t, 1092
Tlvresult, 1092
unpack_loc_GetServer_Ind
loc.h, 1503
unpack_loc_GetServer_Ind_t, 1092
pIPv4AddrInfo, 1093
pIPv6AddrInfo, 1093
pURL, 1093
serverStatus, 1093
serverType, 1093
Tlvresult, 1093
unpack_loc_GnssSvInfo_Ind
loc.h, 1504
unpack_loc_GnssSvInfo_Ind_t, 1094
altitudeAssumed, 1094
pSatelliteInfo, 1094

Tlvresult, 1094
unpack_loc_PositionRpt_Ind
loc.h, 1504
unpack_loc_PositionRpt_Ind_t, 1094
pAltitudeAssumed, 1100
pAltitudeWrtEllipsoid, 1100
pAltitudeWrtMeanSeaLevel, 1100
pFixId, 1100
pGpsTime, 1100
pHeading, 1100
pHeadingUnc, 1100
pHorConfidence, 1100
pHorReliability, 1100
pHorUncCircular, 1100
pHorUncEllipseOrientAzimuth, 1100
pHorUncEllipseSemiMajor, 1100
pHorUncEllipseSemiMinor, 1100
pLatitude, 1100
pLeapSeconds, 1100
pLongitude, 1100
pMagneticDeviation, 1100
pPrecisionDilution, 1100
pSensorDataUsage, 1100
pSpeedHorizontal, 1100
pSpeedUnc, 1100
pSpeedVertical, 1100
pSvUsedforFix, 1100
pTechnologyMask, 1100
pTimeSrc, 1100
pTimeUnc, 1101
pTimestampUtc, 1100
pVertConfidence, 1101
pVertReliability, 1101
pVertUnc, 1101
sessionId, 1101
sessionStatus, 1101
Tlvresult, 1101
unpack_loc_SLQSLOCGetBestAvailPos
loc.h, 1506
unpack_loc_SLQSLOCGetBestAvailPos_t, 1103
Tlvresult, 1103
unpack_loc_SLQSLOCGetServer
loc.h, 1506
unpack_loc_SLQSLOCInjectPosition
loc.h, 1506
unpack_loc_SLQSLOCInjectSensorData
loc.h, 1507
unpack_loc_SLQSLOCInjectUTCTime
loc.h, 1507
unpack_loc_SLQSLOCSetCradleMountConfig
loc.h, 1507
unpack_loc_SetExtPowerConfig_Ind
loc.h, 1504
unpack_loc_SetExtPowerConfig_Ind_t, 1101
status, 1101
Tlvresult, 1101
unpack_loc_SetExtPowerState
loc.h, 1505
unpack_loc_SetExtPowerState_t, 1102
Tlvresult, 1102
unpack_loc_SetOperationMode
loc.h, 1505
unpack_loc_SetOperationMode_Ind
loc.h, 1505
unpack_loc_SetOperationMode_Ind_t, 1102
status, 1103
Tlvresult, 1103
unpack_loc_SetOperationMode_t, 1103
Tlvresult, 1103
unpack_loc_Start
loc.h, 1508
unpack_loc_Start_t, 1104
Tlvresult, 1104
unpack_loc_Stop
loc.h, 1508
unpack_loc_Stop_t, 1104
Tlvresult, 1104
unpack_nas_GetACCOLC
nas.h, 1534
unpack_nas_GetANAAAAuthenticationStatus
nas.h, 1534
unpack_nas_GetCDMANetworkParameters
nas.h, 1535
unpack_nas_GetCDMANetworkParameters_t, 1104
Application, 1106
Broadcast, 1106
CustomSCP, 1106
ForceRev0, 1106
Protocol, 1106
RegForeignNID, 1106
RegForeignSID, 1106
RegHomeSID, 1106
Roaming, 1106
SCI, 1106
SCM, 1107
unpack_nas_GetHomeNetwork
nas.h, 1535
unpack_nas_GetHomeNetwork3GPP2
nas.h, 1536
unpack_nas_GetHomeNetwork3GPP2_t, 1107
nameSize, 1108
pMCC, 1108
pMNC, 1108
pNID, 1108
pName, 1108
pNw2DescDisp, 1109
pNw2DescEnc, 1109
pNw2DescLen, 1109
pNw2MCC, 1109
pNw2MNC, 1109
pNw2Name, 1109
pSID, 1109
unpack_nas_GetHomeNetwork_t, 1109
mcc, 1110
mnc, 1110
name, 1110

nid, 1110
sid, 1110
unpack_nas_GetNetworkPreference
nas.h, 1536
unpack_nas_GetNetworkPreference_t, 1110
ActiveTechPref, 1111
Duration, 1111
PersistentTechPref, 1111
Tlvresult, 1111
unpack_nas_GetRFInfo
nas.h, 1536
unpack_nas_GetRFInfo_t, 1111
instancesSize, 1112
RFBandInfoElements, 1112
unpack_nas_GetServingNetwork
nas.h, 1537
unpack_nas_GetServingNetwork_t, 1112
CSDomain, 1114
DataCaps, 1114
DataCapsLen, 1114
MCC, 1114
MNC, 1114
Name, 1114
nameSize, 1114
PSDomain, 1114
RAN, 1114
RadioIfaces, 1114
RadioIfacesSize, 1114
RegistrationState, 1114
Roaming, 1114
unpack_nas_GetServingNetworkCapabilities
nas.h, 1537
unpack_nas_GetServingNetworkCapabilities_t, 1114
DataCaps, 1115
DataCapsLen, 1115
unpack_nas_GetSignalStrengths
nas.h, 1537
unpack_nas_GetSignalStrengths_t, 1115
len, 1115
radio, 1116
rssi, 1116
unpack_nas_InitiateDomainAttach
nas.h, 1538
unpack_nas_PerformNetworkScan
nas.h, 1538
unpack_nas_PerformNetworkScan_t, 1116
p3GppNetworkInfoInstances, 1117
p3GppNetworkInstanceSize, 1117
pPCSInstance, 1117
pPCSInstanceSize, 1117
pRATINstance, 1117
pRATInstanceSize, 1117
pScanResult, 1117
unpack_nas_SLQSConfigSigInfo
nas.h, 1542
unpack_nas_SLQSGetErrorRate
nas.h, 1542
unpack_nas_SLQSGetErrorRate_t, 1120
pCDMAFrameErrRate, 1121
pGSMBER, 1121
pHDRPackErrRate, 1121
pWCDMABER, 1121
unpack_nas_SLQSGetNetworkTime
nas.h, 1543
unpack_nas_SLQSGetNetworkTime_t, 1122
p3GPP2TimeInfo, 1122
p3GPPTimeInfo, 1122
unpack_nas_SLQSGetOperatorNameData
nas.h, 1543
unpack_nas_SLQSGetOperatorNameData_t, 1122
pNITZInformation, 1123
pOperatorNameString, 1123
pOperatorPLMNList, 1123
pPLMNNetworkName, 1123
pSrvcProviderName, 1123
unpack_nas_SLQSGetPLMNName
nas.h, 1543
unpack_nas_SLQSGetPLMNName_t, 1123
longName, 1125
longNameCI, 1125
longNameEn, 1125
longNameLen, 1125
longNameSB, 1125
shortName, 1125
shortNameCI, 1125
shortNameEn, 1126
shortNameLen, 1126
shortNameSB, 1126
spn, 1126
spnEncoding, 1126
spnLength, 1126
unpack_nas_SLQSGetServingSystem
nas.h, 1544
unpack_nas_SLQSGetServingSystem_t, 1126
BasestationID, 1129
BasestationLatitude, 1129
BasestationLongitude, 1129
CDMASystemInfoExt, 1129
CallBarStatus, 1129
CellID, 1129
ConcSvcInfo, 1129
CurrentPLMN, 1129
DTMInd, 1129
DataSrvCapabilities, 1129
DefaultRoamInd, 1129
DetailedSvcInfo, 1129
Gpp2TimeZone, 1129
GppNetworkDSTAdjustment, 1129
GppTimeZone, 1129
HdrPersonality, 1129
Lac, 1129
NetworkID, 1129
PRLInd, 1130
RoamIndicatorVal, 1130
RoamingIndicatorList, 1130
ServingSystem, 1130

SystemID, 1130
TrackAreaCode, 1130
unpack_nas_SLQSGetSignalStrength
nas.h, 1544
unpack_nas_SLQSGetSignalStrength_t, 1130
ecioList, 1132
ecioListLen, 1132
errorRateList, 1132
errorRateListLen, 1132
Io, 1132
ltersrp, 1132
ltesnr, 1132
rsrqInfo, 1132
rxSignalStrengthList, 1132
rxSignalStrengthListLen, 1132
signalStrengthReqMask, 1132
sinr, 1132
unpack_nas_SLQSGetSysInfo
nas.h, 1545
unpack_nas_SLQSGetSysInfo_t, 1132
pAddCDMASysInfo, 1135
pAddGSMSysInfo, 1135
pAddHDRSysInfo, 1135
pAddLTESysInfo, 1135
pCDMASrvStatusInfo, 1135
pCDMASysInfo, 1135
pGSMCallBarringSysInfo, 1135
pGSMCipherDomainSysInfo, 1135
pGSMSrvStatusInfo, 1135
pGSMSysInfo, 1135
pHDRSrvStatusInfo, 1135
pHDRSysInfo, 1135
pLTESrvStatusInfo, 1135
pLTESysInfo, 1135
pLTEVoiceSupportSysInfo, 1135
pWCDMASysInfo, 1136
unpack_nas_SLQSGetSysSelectionPref
nas.h, 1545
unpack_nas_SLQSGetSysSelectionPref_t, 1136
pBandPref, 1140
pEmerMode, 1140
pGWAcqOrderPref, 1140
pLTEBandPref, 1140
pModePref, 1140
pNetSelPref, 1140
pPRLPref, 1140
pRoamPref, 1140
pSrvDomainPref, 1140
unpack_nas_SLQSInitiateNetworkRegistration
nas.h, 1545
unpack_nas_SLQSNASSwiGetChannelLock
nas.h, 1549
unpack_nas_SLQSNASSwiGetChannelLock_t, 1147
unpack_nas_SLQSNASSwiSetChannelLock
nas.h, 1550
unpack_nas_SLQSNasConfigSigInfo2
nas.h, 1546
unpack_nas_SLQSNasGet3GPP2Subscription
nas.h, 1546
unpack_nas_SLQSNasGet3GPP2Subscription_t, 1140
pDirNum, 1141
unpack_nas_SLQSNasGetCellLocationInfo
nas.h, 1546
unpack_nas_SLQSNasGetCellLocationInfo_t, 1141
pCDMAInfo, 1142
pLTEInfoInterfreq, 1143
pLTEInfoIntrafreq, 1143
pUMTSInfo, 1143
unpack_nas_SLQSNasGetHDRColorCode
nas.h, 1547
unpack_nas_SLQSNasGetHDRColorCode_t, 1143
pColorCode, 1143
unpack_nas_SLQSNasGetSigInfo
nas.h, 1547
unpack_nas_SLQSNasGetSigInfo_t, 1143
CDMASSInfo, 1144
GSMSSInfo, 1144
HDRSSInfo, 1144
LTESSInfo, 1144
unpack_nas_SLQSNasGetTxRxInfo
nas.h, 1548
unpack_nas_SLQSNasGetTxRxInfo_t, 1144
pRXChain0Info, 1145
pRXChain1Info, 1145
pTXInfo, 1145
unpack_nas_SLQSNasIndicationRegisterExt
nas.h, 1548
unpack_nas_SLQSNasNetworkTimeCallBack_ind
nas.h, 1548
unpack_nas_SLQSNasNetworkTimeCallBack_ind_t, 1145
pDayltSavAdj, 1146
pRadioInterface, 1146
pTimeZone, 1146
universalTime, 1146
unpack_nas_SLQSNasSigInfoCallback_ind
nas.h, 1549
unpack_nas_SLQSNasSigInfoCallback_ind_t, 1146
pRscp, 1147
unpack_nas_SLQSNasSwiIndicationRegister
nas.h, 1549
unpack_nas_SLQSNasSwiModemStatus
nas.h, 1550
unpack_nas_SLQSNasSwiModemStatus_t, 1148
commonInfo, 1148
pLTEInfo, 1148
unpack_nas_SLQSNasSwiOTAMessageCallback_ind
nas.h, 1550
unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t, 1148
Info, 1149
Tlvresult, 1149
unpack_nas_SLQSNasSysInfoCallback_ind
nas.h, 1551
unpack_nas_SLQSNasTimerCallback_ind
nas.h, 1551

unpack_nas_SLQSNasTimerCallback_ind_t, 1149
t3396_apn, 1149
t3396_plmn_id, 1149
t3396_val, 1149
unpack_nas_SLQSSetBandPreference
nas.h, 1551
unpack_nas_SLQSSetSignalStrengthsCallback
nas.h, 1552
unpack_nas_SLQSSetSysSelectionPref
nas.h, 1552
unpack_nas_SLQSSetSysSelectionPrefCallBack_ind
nas.h, 1552
unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t, 1150
Info, 1150
Tlvresult, 1150
unpack_nas_SLQSSwiGetHDRPersonality
nas.h, 1552
unpack_nas_SLQSSwiGetHDRPersonality_t, 1150
pCurrentPersonality, 1151
pPersonalityListLength, 1151
pProtocolSubtypeElement, 1151
unpack_nas_SLQSSwiGetHDRProtSubtype
nas.h, 1553
unpack_nas_SLQSSwiGetHDRProtSubtype_t, 1151
pAppSubType, 1151
pCurrentPrsnlty, 1151
pPersonalityListLength, 1151
unpack_nas_SLQSSwiGetHRPDStats
nas.h, 1553
unpack_nas_SLQSSwiGetHRPDStats_t, 1152
pPilotSetData, 1152
pUATI, 1152
unpack_nas_SLQSSwiGetLteCQI
nas.h, 1554
unpack_nas_SLQSSwiGetLteCQI_t, 1152
ValidityCW0, 1153
ValidityCW1, 1153
unpack_nas_SLQSSwiGetLteSccRxInfo
nas.h, 1554
unpack_nas_SLQSSwiGetLteSccRxInfo_t, 1153
pSccRxInfo, 1153
unpack_nas_SLQSSwiNetworkDebug
nas.h, 1554
unpack_nas_SLQSSwiNetworkDebug_t, 1154
pDataStatusDetail, 1154
pDeviceConfigDetail, 1154
pNetworkStat1x, 1154
pNetworkStatEVDO, 1154
pObjectVer, 1154
unpack_nas_SLQSSwiPSDetach
nas.h, 1555
unpack_nas_SLQSSysInfoCallback_ind_t, 1154
pGSMSysInfo, 1157
pHDRSysInfo, 1157
pLTESysInfo, 1157
pSysInfoNoChange, 1157
unpack_nas_SetACCOLC
nas.h, 1538
unpack_nas_SetCDMANetworkParameters
nas.h, 1539
unpack_nas_SetDataCapabilitiesCallback_ind
nas.h, 1539
unpack_nas_SetDataCapabilitiesCallback_ind_t, 1117
dataCaps, 1117
dataCapsSize, 1117
unpack_nas_SetEventReportInd
nas.h, 1539
unpack_nas_SetEventReportInd_t, 1117
RFTlv, 1118
RRTlv, 1118
SLQSSSTlv, 1118
SSTlv, 1118
unpack_nas_SetLURejectCallback
nas.h, 1540
unpack_nas_SetNasLTECphyCaIndCallback_ind
nas.h, 1540
unpack_nas_SetNasLTECphyCaIndCallback_ind_t, 1118
unpack_nas_SetNetworkPreference
nas.h, 1540
unpack_nas_SetNetworkPreference_t, 1119
Tlvresult, 1119
unpack_nas_SetRFInfoCallback
nas.h, 1541
unpack_nas_SetRoamingIndicatorCallback_ind
nas.h, 1541
unpack_nas_SetRoamingIndicatorCallback_ind_t, 1119
roaming, 1119
unpack_nas_SetServingSystemCallback_ind
nas.h, 1541
unpack_nas_SetServingSystemCallback_ind_t, 1120
SSInfo, 1120
Tlvresult, 1120
unpack_nas_SlqsGetLTECphyCAInfo
nas.h, 1542
unpack_nas_SlqsGetLTECphyCAInfo_t, 1121
LTECphyCaInfo, 1122
Tlvresult, 1122
unpack_omaDmConfigTlv_t, 1157
alertmsg, 1158
alertmsglength, 1158
state, 1158
userInputReq, 1158
userInputTimeout, 1158
unpack_omaDmFotaTlv_t, 1158
description, 1160
descriptionlength, 1160
fwdloadsize, 1160
fwloadComplete, 1160
namelength, 1160
package_name, 1160
sessionType, 1160
severity, 1160
state, 1160
updateCompleteStatus, 1160

userInputReq, 1160
userInputTimeout, 1160
version, 1160
versionlength, 1160
unpack_omaDmNotificationsTlv_t, 1160
notification, 1161
sessionStatus, 1161
unpack_pds_ForceXTRADownload
pds.h, 1568
unpack_pds_GetPDSDefaults
pds.h, 1569
unpack_pds_GetPDSDefaults_t, 1161
pAccuracy, 1162
pInterval, 1162
pOperation, 1162
pTimeout, 1162
unpack_pds_GetPDSState
pds.h, 1569
unpack_pds_GetPDSState_t, 1162
pEnabledStatus, 1163
pTrackingStatus, 1163
unpack_pds_GetPortAutomaticTracking
pds.h, 1569
unpack_pds_GetPortAutomaticTracking_t, 1163
pbAuto, 1163
unpack_pds_GetServiceAutomaticTracking
pds.h, 1570
unpack_pds_GetServiceAutomaticTracking_t, 1163
pbAuto, 1164
unpack_pds_GetXTRAAutomaticDownload
pds.h, 1570
unpack_pds_GetXTRAAutomaticDownload_t, 1164
pInterval, 1164
pbEnabled, 1164
unpack_pds_GetXTRANetwork
pds.h, 1570
unpack_pds_GetXTRANetwork_t, 1164
pPreference, 1165
unpack_pds_GetXTRAValidity
pds.h, 1571
unpack_pds_GetXTRAValidity_t, 1165
pDuration, 1165
pGPSWeek, 1165
pGPSWeekOffset, 1165
unpack_pds_PDSInjectTimeReference
pds.h, 1571
unpack_pds_ResetPDSData
pds.h, 1571
unpack_pds_SLQSGetAGPSConfig
pds.h, 1575
unpack_pds_SLQSGetAGPSConfig_t, 1168
pServerAddress, 1169
pServerPort, 1169
pServerURL, 1169
unpack_pds_SLQSGetGPSStateInfo
pds.h, 1575
unpack_pds_SLQSGetGPSStateInfo_t, 1169
Altitude, 1172
EngineState, 1172
HorizontalUncertainty, 1172
Iono_valid, 1172
Latitude, 1172
Longitude, 1172
Time_uncert_ms, 1172
ValidMask, 1172
VerticalUncertainty, 1173
unpack_pds_SLQSPDSDeterminePosition
pds.h, 1575
unpack_pds_SLQSPDSInjectAbsoluteTimeReference
pds.h, 1576
unpack_pds_SLQSPDSInjectPositionData
pds.h, 1576
unpack_pds_SLQSSetAGPSConfig
pds.h, 1576
unpack_pds_SLQSSetPositionMethodState
pds.h, 1577
unpack_pds_SetEventReport_Ind
pds.h, 1572
unpack_pds_SetEventReport_Ind_t, 1165
dLatitude, 1167
dLongitude, 1167
has_PositionDataNMEA, 1167
has_SessionStatus, 1167
has_dLatitude, 1167
has_dLongitude, 1167
has_posSrc, 1167
posSrc, 1167
PositionDataNMEA, 1167
SessionStatus, 1167
unpack_pds_SetEventReportCallback
pds.h, 1572
unpack_pds_SetPDSDefaults
pds.h, 1572
unpack_pds_SetPDSState
pds.h, 1573
unpack_pds_SetPdsState_Ind
pds.h, 1573
unpack_pds_SetPdsState_Ind_t, 1167
EnabledStatus, 1168
TlvPresent, 1168
TrackingStatus, 1168
unpack_pds_SetPortAutomaticTracking
pds.h, 1573
unpack_pds_SetServiceAutomaticTracking
pds.h, 1574
unpack_pds_SetXTRAAutomaticDownload
pds.h, 1574
unpack_pds_SetXTRANetwork
pds.h, 1574
unpack_pds_StartPDSTrackingSessionExt
pds.h, 1577
unpack_pds_StopPDSTrackingSession
pds.h, 1577
unpack_qmi_t, 1173
msgid, 1173
type, 1173

xid, 1173
unpack_qos_IPv4Addr_t, 1174
addr, 1174
subnetMask, 1174
unpack_qos_IPv6Addr_t, 1174
addr, 1174
prefixLen, 1175
unpack_qos_IPv6TrafCls_t, 1175
mask, 1175
val, 1175
unpack_qos_Port_t, 1176
port, 1176
range, 1176
unpack_qos_QosFlowInfo_t, 1176
BearerID, 1177
is_RxQFlowGranted_Available, 1177
is_TxQFlowGranted_Available, 1177
NumRxFilters, 1177
NumTxFilters, 1177
QFlowState, 1177
RxQFilter, 1177
RxQFlowGranted, 1177
TxQFilter, 1177
TxQFlowGranted, 1177
unpack_qos_QosFlowInfoState_t, 1178
id, 1178
isNewFlow, 1178
state, 1178
unpack_qos_SLQSQosGetNetworkStatus
qos.h, 2006
unpack_qos_SLQSQosGetNetworkStatus_t, 1178
NWQoSStatus, 1179
unpack_qos_SLQSQosSwiReadApnExtraParams
qos.h, 2007
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 1179
ambr_dl, 1179
ambr_dl_ext, 1180
ambr_dl_ext2, 1180
ambr_ul, 1180
ambr_ul_ext, 1180
ambr_ul_ext2, 1180
apnId, 1180
unpack_qos_SLQSQosSwiReadDataStats
qos.h, 2007
unpack_qos_SLQSQosSwiReadDataStats_t, 1180
apnId, 1181
numQosFlow, 1181
qosFlow, 1181
total_rx_bytes, 1181
total_rx_pkt, 1181
total_tx_bytes, 1181
total_tx_pkt, 1181
unpack_qos_SLQSSetQosEventCallback
qos.h, 2008
unpack_qos_SLQSSetQosEventCallback_ind
qos.h, 2008
unpack_qos_SLQSSetQosEventCallback_ind_t, 1181
NumFlows, 1182
QosFlowInfo, 1182
unpack_qos_SLQSSetQosNWStatusCallback_ind
qos.h, 2008
unpack_qos_SLQSSetQosNWStatusCallback_ind_t, 1182
status, 1182
unpack_qos_SLQSSetQosPriEventCallback_ind
qos.h, 2009
unpack_qos_SLQSSetQosPriEventCallback_ind_t, 1182
event, 1182
unpack_qos_SLQSSetQosStatusCallback_ind
qos.h, 2010
unpack_qos_SLQSSetQosStatusCallback_ind_t, 1183
event, 1184
id, 1184
reason, 1184
status, 1184
unpack_qos_Tos_t, 1192
mask, 1193
val, 1193
unpack_qos_dataRate_t, 1173
dataRateMax, 1174
guaranteedRate, 1174
unpack_qos_pktErrRate_t, 1175
exponent, 1175
multiplier, 1175
unpack_qos_swiQosFilter_t, 1184
EspSpi, 1186
IPv4DstAddr, 1186
IPv4SrcAddr, 1186
IPv4Tos, 1186
IPv6DstAddr, 1186
IPv6Label, 1187
IPv6SrcAddr, 1187
IPv6TrafCls, 1187
Id, 1186
index, 1186
is_EspSpi_Available, 1187
is_IPv4DstAddr_Available, 1187
is_IPv4SrcAddr_Available, 1187
is_IPv4Tos_Available, 1187
is_IPv6DstAddr_Available, 1187
is_IPv6Label_Available, 1187
is_IPv6SrcAddr_Available, 1187
is_IPv6TrafCls_Available, 1187
is_Id_Available, 1187
is_NxtHdrProto_Available, 1187
is_Precedence_Available, 1187
is_TCPDstPort_Available, 1187
is_TCPSrcPort_Available, 1187
is_TranDstPort_Available, 1187
is_TranSrcPort_Available, 1187
is_UDPDstPort_Available, 1187
is_UDPSrcPort_Available, 1187
NxtHdrProto, 1187
Precedence, 1187

TCPDstPort, 1187
TCPSrcPort, 1187
TranDstPort, 1187
TranSrcPort, 1187
UDPDstPort, 1187
UDPSrcPort, 1187
version, 1188
unpack_qos_swiQosFlow_t, 1188
DataRate, 1190
index, 1190
is_DataRate_Available, 1190
is_Jitter_Available, 1191
is_Latency_Available, 1191
is_LteQci_Available, 1191
is_MaxAllowedPktSz_Available, 1191
is_MinPolicedPktSz_Available, 1191
is_PktErrRate_Available, 1191
is_ProfileId3GPP2_Available, 1191
is_TokenBucket_Available, 1191
is_TrafficClass_Available, 1191
is_val_3GPP2Pri_Available, 1191
is_val_3GPPImCn_Available, 1191
is_val_3GPPSigInd_Available, 1191
Jitter, 1191
Latency, 1191
LteQci, 1191
MaxAllowedPktSz, 1191
MinPolicedPktSz, 1191
PktErrRate, 1191
ProfileId3GPP2, 1191
TokenBucket, 1191
TrafficClass, 1191
val_3GPP2Pri, 1191
val_3GPPImCn, 1191
val_3GPPResResidualBER, 1191
val_3GPPSigInd, 1191
val_3GPPTraHdlPri, 1191
unpack_qos_tokenBucket_t, 1192
bucketSz, 1192
peakRate, 1192
tokenRate, 1192
unpack_result_code_only
common.h, 1436
unpack_rms_GetSMSWake
rms.h, 2011
unpack_rms_GetSMSWake_t, 1193
enabled, 1194
wake_mask, 1194
unpack_rms_SetSMSWake
rms.h, 2012
unpack_sar_SLQSGetRfSarState
sar.h, 2013
unpack_sar_SLQSGetRfSarState_t, 1195
pSarRFState, 1196
unpack_sar_SLQSSetRfSarState
sar.h, 2013
unpack_sms_GetSMSCAddress
sms.h, 2026
unpack_sms_GetSMSCAddress_t, 1196
addressSize, 1196
pSMSCAddress, 1196
pSMSCType, 1196
typeSize, 1196
unpack_sms_SLQSDeleteSMS
sms.h, 2028
unpack_sms_SLQSDeleteSMS_t, 1199
unpack_sms_SLQSGetIndicationRegister
sms.h, 2028
unpack_sms_SLQSGetIndicationRegister_t, 1199
pGetIndicationRegInfo, 1199
unpack_sms_SLQSGetMessageWaiting
sms.h, 2028
unpack_sms_SLQSGetMessageWaiting_t, 1199
pGetMsgWaitingInfoResp, 1200
unpack_sms_SLQSGetSMS
sms.h, 2029
unpack_sms_SLQSGetSMS_t, 1200
message, 1200
messageFormat, 1200
messageSize, 1200
messageTag, 1200
unpack_sms_SLQSGetSMSList
sms.h, 2029
unpack_sms_SLQSGetSMSList_t, 1201
messageList, 1202
messageListSize, 1202
unpack_sms_SLQSGetSmsBroadcastConfig
sms.h, 2029
unpack_sms_SLQSGetSmsBroadcastConfig_t, 1201
pBroadcastConfig, 1201
pCDMABroadcastConfig, 1201
unpack_sms_SLQSGetTransLayerInfo
sms.h, 2030
unpack_sms_SLQSGetTransLayerInfo_t, 1202
pGetTransLayerInfo, 1202
unpack_sms_SLQSGetTransNWRegInfo
sms.h, 2030
unpack_sms_SLQSGetTransNWRegInfo_t, 1202
unpack_sms_SLQSModifySMSStatus
sms.h, 2031
unpack_sms_SLQSModifySMSStatus_t, 1202
unpack_sms_SLQSSendAsyncSMS
sms.h, 2031
unpack_sms_SLQSSetIndicationRegister
sms.h, 2031
unpack_sms_SLQSSetSmsBroadcastActivation
sms.h, 2032
unpack_sms_SLQSSetSmsBroadcastConfig
sms.h, 2032
unpack_sms_SLQSSetSmsStorage
sms.h, 2032
unpack_sms_SLQSSmsGetMaxStorageSize
sms.h, 2032
unpack_sms_SLQSSmsGetMaxStorageSize_t, 1203
pMaxStorageSizeResp, 1203
unpack_sms_SLQSSmsGetMessageProtocol

sms.h, 2033
unpack_sms_SLQSSmsGetMessageProtocol_t, 1203
pMessageProtocol, 1203
unpack_sms_SLQSSmsSetRoutes
sms.h, 2033
unpack_sms_SLQSSwiGetSMSStorage
sms.h, 2034
unpack_sms_SLQSSwiGetSMSStorage_t, 1204
pSmsStorage, 1204
unpack_sms_SLQSWmsMemoryFullCallBack_ind
sms.h, 2034
unpack_sms_SLQSWmsMemoryFullCallBack_ind_t, 1204
messageMode, 1204
storageType, 1204
unpack_sms_SaveSMS
sms.h, 2026
unpack_sms_SaveSMS_t, 1196
pMessageIndex, 1197
unpack_sms_SendSMS
sms.h, 2026
unpack_sms_SendSMS_t, 1197
messageFailureCode, 1197
messageID, 1197
unpack_sms_SetNewSMSCallback
sms.h, 2027
unpack_sms_SetNewSMSCallback_ind
sms.h, 2027
unpack_sms_SetNewSMSCallback_ind_t, 1197
ETWSTlv, 1198
IMSTlv, 1198
MMTlv, 1198
NewMMTlv, 1198
SMSCTlv, 1198
TRMessageTlv, 1198
unpack_sms_SetNewSMSCallback_t, 1198
unpack_sms_SetSMSCAddress
sms.h, 2027
unpack_swiloc_SwiLocGetAutoStart
swiloc.h, 2037
unpack_swiloc_SwiLocGetAutoStart_t, 1205
fix_rate, 1206
fix_rate_reported, 1206
fix_type, 1206
fix_type_reported, 1206
function, 1206
function_reported, 1206
max_dist, 1206
max_dist_reported, 1206
max_time, 1206
max_time_reported, 1206
unpack_swiloc_SwiLocSetAutoStart
swiloc.h, 2037
unpack_swioma_SLQSOMADMAlertCallback
swioma.h, 2043
unpack_swioma_SLQSOMADMAlertCallback_ind
swioma.h, 2043
unpack_swioma_SLQSOMADMAlertCallback_ind_t, 1206
eventType, 1207
unpack_swioma_SLQSOMADMCancelSession
swioma.h, 2044
unpack_swioma_SLQSOMADMGetSessionInfo
swioma.h, 2044
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1207
Date, 1209
DateLength, 1209
PkgDescLength, 1209
PkgDescription, 1209
PkgName, 1209
PkgNameLength, 1209
RetryCount, 1209
SessionState, 1209
SessionType, 1209
Severity, 1209
Source, 1209
SourceLength, 1209
Status, 1209
Time, 1210
TimeLength, 1210
UpdateCompleteStatus, 1210
unpack_swioma_SLQSOMADMGetSettings
swioma.h, 2044
unpack_swioma_SLQSOMADMGetSettings_t, 1210
Autosdm, 1211
FOTAdownload, 1211
FwAutoCheck, 1211
unpack_swioma_SLQSOMADMSendSelection
swioma.h, 2045
unpack_swioma_SLQSOMADMSetSettings
swioma.h, 2045
unpack_swioma_SLQSOMADMStartSession
swioma.h, 2046
unpack_swioma_SLQSOMADMStartSession_t, 1211
FwAvailability, 1212
unpack_uim_ChangePin
uim.h, 2057
unpack_uim_ChangePin_t, 1212
pEncryptedPIN1, 1212
pIndicationToken, 1212
pRemainingRetries, 1213
Tlvresult, 1213
unpack_uim_GetCardStatus
uim.h, 2057
unpack_uim_GetCardStatus_t, 1213
pCardStatus, 1213
pHotSwapStatus, 1213
Tlvresult, 1213
unpack_uim_ReadTransparent
uim.h, 2058
unpack_uim_ReadTransparent_t, 1213
pCardResult, 1214
pEncryptedData, 1214
pIndicationToken, 1214
pReadResult, 1214

Tlvresult, 1214
unpack_uim_SLQSUIMAuthenticate
uim.h, 2059
unpack_uim_SLQSUIMAuthenticate_t, 1216
pAuthenticateResult, 1216
pCardResult, 1216
pIndicationToken, 1216
unpack_uim_SLQSUIMDepersonalization
uim.h, 2059
unpack_uim_SLQSUIMDepersonalization_t, 1216
pRemainingRetries, 1217
unpack_uim_SLQSUIMEventRegister
uim.h, 2059
unpack_uim_SLQSUIMEventRegister_t, 1217
eventMask, 1217
unpack_uim_SLQSUIMGetConfiguration
uim.h, 2060
unpack_uim_SLQSUIMGetConfiguration_t, 1217
pAutoSelection, 1218
pHaltSubscription, 1218
pPersonalizationStatus, 1218
unpack_uim_SLQSUIMGetFileAttributes
uim.h, 2060
unpack_uim_SLQSUIMGetFileAttributes_t, 1218
pCardResult, 1219
pFileAttributes, 1219
pIndicationToken, 1219
unpack_uim_SLQSUIMGetSlotsStatus
uim.h, 2060
unpack_uim_SLQSUIMGetSlotsStatus_t, 1219
pNumberOfPhySlot, 1219
pUimSlotsStatus, 1219
unpack_uim_SLQSUIMPowerDown
uim.h, 2061
unpack_uim_SLQSUIMPowerUp
uim.h, 2061
unpack_uim_SLQSUIMRefreshComplete
uim.h, 2061
unpack_uim_SLQSUIMRefreshGetLastEvent
uim.h, 2062
unpack_uim_SLQSUIMRefreshGetLastEvent_t, 1219
pRefreshEvent, 1220
unpack_uim_SLQSUIMRefreshOK
uim.h, 2062
unpack_uim_SLQSUIMRefreshRegister
uim.h, 2062
unpack_uim_SLQSUIMReset
uim.h, 2063
unpack_uim_SLQSUIMSetStatusChangeCallBack_ind
uim.h, 2063
unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t, 1220
unpack_uim_SLQSUIMSwitchSlot
uim.h, 2064
unpack_uim_SetPinProtection
uim.h, 2058
unpack_uim_SetPinProtection_t, 1214
pEncryptedPIN1, 1215
pIndicationToken, 1215
pRemainingRetries, 1215
Tlvresult, 1215
unpack_uim_SetUimSlotStatusChangeCallback_ind
uim.h, 2058
unpack_uim_SetUimSlotStatusChangeCallback_ind_t, 1215
bNumberOfPhySlots, 1215
slotsstatusChange, 1216
unpack_uim_UnblockPin
uim.h, 2064
unpack_uim_UnblockPin_t, 1220
pEncryptedPIN1, 1221
pIndicationToken, 1221
pRemainingRetries, 1221
Tlvresult, 1221
unpack_uim_VerifyPin
uim.h, 2064
unpack_uim_VerifyPin_t, 1221
pEncryptedPIN1, 1222
pIndicationToken, 1222
pRemainingRetries, 1222
Tlvresult, 1222
unpack_valid_nas_GetCDMANetworkParameters
nas.h, 1555
unpack_valid_nas_SLQSGetServingSystem
nas.h, 1555
unpack_valid_nas_SLQSGetSignalStrength
nas.h, 1556
unpack_valid_nas_SLQSNasGetSigInfo
nas.h, 1557
unpack_voice_AnswerUSSD
voice.h, 2081
unpack_voice_CancelUSSD
voice.h, 2082
unpack_voice_OriginateUSSD
voice.h, 2082
unpack_voice_SLQSOriginateUSSD
voice.h, 2082
unpack_voice_SLQSOriginateUSSD_t, 1222
pAlphaIDInfo, 1225
pCCSuppsType, 1225
pCallId, 1225
pCcResultType, 1225
pUSSDInfo, 1225
pfailureCause, 1225
unpack_voice_SLQSVoiceALSSelectLine
voice.h, 2083
unpack_voice_SLQSVoiceALSSetLineSwitching
voice.h, 2083
unpack_voice_SLQSVoiceAnswerCall
voice.h, 2083
unpack_voice_SLQSVoiceAnswerCall_t, 1225
pCallId, 1226
unpack_voice_SLQSVoiceBindSubscription
voice.h, 2084
unpack_voice_SLQSVoiceBurstDTMF
voice.h, 2084

unpack_voice_SLQSVoiceBurstDTMF_t, 1226
pCallID, 1226
unpack_voice_SLQSVoiceDialCall
voice.h, 2084
unpack_voice_SLQSVoiceDialCall_t, 1226
pAlphaIDInfo, 1227
pCCResultType, 1227
pCCSUPSType, 1227
pCallID, 1227
unpack_voice_SLQSVoiceEndCall
voice.h, 2085
unpack_voice_SLQSVoiceEndCall_t, 1227
pCallId, 1228
unpack_voice_SLQSVoiceGetAllCallInfo
voice.h, 2085
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1228
pArrAlertingPattern, 1230
pArrAlertingType, 1230
pArrAlphaID, 1230
pArrCallEndReason, 1230
pArrCallInfo, 1230
pArrCalledPartyNum, 1230
pArrConnectPartyNum, 1230
pArrDiagInfo, 1230
pArrRedirPartyNum, 1230
pArrRemotePartyName, 1230
pArrRemotePartyNum, 1230
pArrSvcOption, 1230
pArrUUSInfo, 1230
pVoicePrivacy, 1230
unpack_voice_SLQSVoiceGetCLIP
voice.h, 2087
unpack_voice_SLQSVoiceGetCLIP_t, 1238
pAlphaIDInfo, 1239
pCCResType, 1239
pCallID, 1239
pFailCause, 1239
unpack_voice_SLQSVoiceGetCLIR
voice.h, 2087
unpack_voice_SLQSVoiceGetCLIR_t, 1239
pAlphaIDInfo, 1240
pCCResType, 1240
pCallID, 1240
pFailCause, 1241
unpack_voice_SLQSVoiceGetCNAP
voice.h, 2087
unpack_voice_SLQSVoiceGetCNAP_t, 1241
pAlphaIDInfo, 1242
pCCResType, 1242
pCallID, 1242
pFailCause, 1242
unpack_voice_SLQSVoiceGetCOLP
voice.h, 2088
unpack_voice_SLQSVoiceGetCOLP_t, 1242
pAlphaIDInfo, 1243
pCCResType, 1243
pCallID, 1243
pFailCause, 1243
unpack_voice_SLQSVoiceGetCOLR
voice.h, 2088
unpack_voice_SLQSVoiceGetCOLR_t, 1243
pAlphaIDInfo, 1245
pCCResType, 1245
pCallID, 1245
pFailCause, 1245
unpack_voice_SLQSVoiceGetCallBarring
voice.h, 2085
unpack_voice_SLQSVoiceGetCallBarring_t, 1230
pAlphaIDInfo, 1232
pCCResType, 1232
pCallID, 1232
pFailCause, 1232
pSvcClass, 1232
unpack_voice_SLQSVoiceGetCallForwardingStatus
voice.h, 2086
unpack_voice_SLQSVoiceGetCallForwardingStatus_t, 1232
pAlphaIDInfo, 1233
pCCResType, 1233
pCallID, 1233
pFailCause, 1233
unpack_voice_SLQSVoiceGetCallInfo
voice.h, 2086
unpack_voice_SLQSVoiceGetCallInfo_t, 1233
pAlertType, 1236
pAlertingPattern, 1236
pAlphaIDInfo, 1236
pCallInfo, 1236
pConnectNumInfo, 1236
pDiagInfo, 1236
pOTASPStatus, 1236
pRemotePartyName, 1236
pRemotePartyNum, 1236
pSrvOpt, 1236
pUUSInfo, 1236
pVoicePrivacy, 1236
unpack_voice_SLQSVoiceGetCallWaiting
voice.h, 2086
unpack_voice_SLQSVoiceGetCallWaiting_t, 1237
pAlphaIDInfo, 1238
pCCResType, 1238
pCallID, 1238
pFailCause, 1238
pSvcClass, 1238
unpack_voice_SLQSVoiceGetConfig
voice.h, 2088
unpack_voice_SLQSVoiceGetConfig_t, 1245
pAirTimerCnt, 1247
pAutoAnswerStat, 1247
pCurAMRConfig, 1247
pCurPrefVoiceSO, 1247
pCurVoiceDomainPref, 1247
pCurVoicePrivacyPref, 1247
pCurrTTYMode, 1247
pRoamTimerCnt, 1247
unpack_voice_SLQSVoiceIndicationRegister

voice.h, 2089
unpack_voice_SLQSVoiceManageCalls
voice.h, 2089
unpack_voice_SLQSVoiceManageCalls_t, 1247
pFailCause, 1247
unpack_voice_SLQSVoiceOrigUSSDNoWait
voice.h, 2089
unpack_voice_SLQSVoiceSendFlash
voice.h, 2090
unpack_voice_SLQSVoiceSendFlash_t, 1247
pCallID, 1248
unpack_voice_SLQSVoiceSetCallBarringPassword
voice.h, 2090
unpack_voice_SLQSVoiceSetCallBarringPassword_t, 1248
pAlphaIDInfo, 1249
pCCResType, 1249
pCallID, 1249
pFailCause, 1249
unpack_voice_SLQSVoiceSetConfig
voice.h, 2090
unpack_voice_SLQSVoiceSetConfig_t, 1249
pAirTimerStatus, 1250
pAutoAnsStatus, 1250
pPrefVoiceSOStatus, 1250
pRoamTimerStatus, 1251
pTTYConfigStatus, 1251
pVoiceDomainPrefStatus, 1251
unpack_voice_SLQSVoiceSetPreferredPrivacy
voice.h, 2091
unpack_voice_SLQSVoiceSetSUPSService
voice.h, 2091
unpack_voice_SLQSVoiceSetSUPSService_t, 1251
pCallID, 1252
pFailCause, 1252
unpack_voice_SLQSVoiceStartContDTMF
voice.h, 2091
unpack_voice_SLQSVoiceStartContDTMF_t, 1252
pCallID, 1252
unpack_voice_SLQSVoiceStopContDTMF
voice.h, 2092
unpack_voice_SLQSVoiceStopContDTMF_t, 1252
callID, 1253
unpack_wds_DHCPv4ClientLease_ind
wds.h, 2118
unpack_wds_DHCPv4ClientLease_ind_t, 1253
DHCPv4LeaseOptTlv, 1253
DHCPv4LeaseStateTlv, 1253
IPv4AddrTlv, 1253
ProfileIdTlv, 1253
unpack_wds_DHCPv4ClientLeaseChange
wds.h, 2118
unpack_wds_GetAutoconnect
wds.h, 2119
unpack_wds_GetAutoconnect_t, 1253
psetting, 1254
unpack_wds_GetByteTotals
wds.h, 2119
unpack_wds_GetByteTotals_t, 1254
pRXTotalBytes, 1254
pTXTotalBytes, 1254
unpack_wds_GetConnectionRate
wds.h, 2119
unpack_wds_GetConnectionRate_t, 1254
currentChannelRXRate, 1255
currentChannelTXRate, 1255
maxChannelRXRate, 1255
maxChannelTXRate, 1255
unpack_wds_GetDataBearerTechnology
wds.h, 2120
unpack_wds_GetDataBearerTechnology_t, 1255
pDataBearer, 1256
unpack_wds_GetDefaultProfile
wds.h, 2120
unpack_wds_GetDefaultProfile_t, 1256
apnname, 1258
apnsize, 1258
auth, 1258
ipaddr, 1258
ipaddrv6, 1258
name, 1258
namesize, 1258
pdptype, 1258
pridns, 1258
pridnsv6, 1258
secdns, 1258
secdnsv6, 1258
username, 1258
usersize, 1258
unpack_wds_GetDefaultProfileNum
wds.h, 2120
unpack_wds_GetDefaultProfileNum_t, 1258
index, 1259
unpack_wds_GetDormancyState
wds.h, 2121
unpack_wds_GetDormancyState_t, 1259
dormancyState, 1259
unpack_wds_GetLastMobileIPError
wds.h, 2121
unpack_wds_GetLastMobileIPError_t, 1259
error, 1260
unpack_wds_GetMobileIP
wds.h, 2121
unpack_wds_GetMobileIP_t, 1260
mipMode, 1260
unpack_wds_GetMobileIPProfile
wds.h, 2122
unpack_wds_GetMobileIPProfile_t, 1260
AAASPI, 1262
AAAState, 1262
address, 1262
enabled, 1262
HASPI, 1262
HAState, 1262
NAI, 1262
naiSize, 1262

primaryHA, 1262
revTunneling, 1262
secondaryHA, 1262
unpack_wds_GetPacketStatistics
wds.h, 2122
unpack_wds_GetPacketStatistics_t, 1262
pRXDroppedCount, 1264
pRXOKBytesLastCall, 1264
pRXOkBytesCount, 1264
pRXPacketErrors, 1264
pRXPacketOverflows, 1264
pRXPacketSuccesses, 1264
pTXDroppedCount, 1264
pTXOKBytesLastCall, 1264
pTXOkBytesCount, 1264
pTXPacketErrors, 1264
pTXPacketOverflows, 1264
pTXPacketSuccesses, 1264
unpack_wds_GetPacketStatus
wds.h, 2122
unpack_wds_GetPacketStatus_t, 1264
rXDroppedCount, 1265
rXOKBytesLastCall, 1266
rXOkBytesCount, 1265
rXPacketErrors, 1266
rXPacketOverflows, 1266
rXPacketSuccesses, 1266
tXDroppedCount, 1266
tXOKBytesLastCall, 1266
tXOkBytesCount, 1266
tXPacketErrors, 1266
tXPacketOverflows, 1266
tXPacketSuccesses, 1266
unpack_wds_GetSessionDuration
wds.h, 2123
unpack_wds_GetSessionDuration_t, 1266
callDuration, 1266
unpack_wds_GetSessionState
wds.h, 2123
unpack_wds_GetSessionState_t, 1266
connectionStatus, 1267
unpack_wds_RMSetTransferStatistics
wds.h, 2123
unpack_wds_RMSetTransferStatistics_t, 1267
unpack_wds_RMTransferStatistics_ind
wds.h, 2124
unpack_wds_SLQSCreateProfile
wds.h, 2126
unpack_wds_SLQSCreateProfile_t, 1267
pCreateProfileOut, 1268
pProfileID, 1268
Tlvresult, 1268
unpack_wds_SLQSDeleteProfile
wds.h, 2127
unpack_wds_SLQSDeleteProfile_t, 1268
extendedErrorCode, 1268
unpack_wds_SLQSGet3GPPConfigItem
wds.h, 2127
unpack_wds_SLQSGet3GPPConfigItem_t, 1268
_3gppRelease, 1270
defaultPDNEnabled, 1270
LTEAttachProfile, 1270
profileList, 1270
unpack_wds_SLQSGetCurrDataSystemStat
wds.h, 2127
unpack_wds_SLQSGetCurrDataSystemStat_t, 1270
currNetworkInfo, 1271
networkInfoLen, 1271
prefNetwork, 1271
unpack_wds_SLQSGetCurrentChannelRate
wds.h, 2128
unpack_wds_SLQSGetCurrentChannelRate_t, 1271
max_channel_rx_rate, 1272
max_channel_tx_rate, 1272
unpack_wds_SLQSGetDUNCallInfo
wds.h, 2128
unpack_wds_SLQSGetDUNCallInfo_t, 1272
callEndReason, 1275
channelRate, 1275
connectionStatus, 1275
dataBearerTech, 1275
dormancyStatus, 1275
lastCallDataBearerTech, 1275
mdmCallDurationActive, 1275
rxOKBytesCount, 1275
txOKBytesCount, 1275
unpack_wds_SLQSGetDataBearerTechnology
wds.h, 2128
unpack_wds_SLQSGetDataBearerTechnology_t, 1272
curDataBearerTechnology, 1272
dataBearerMask, 1272
lastCallDataBearerTechnology, 1272
unpack_wds_SLQSGetProfileSettings
wds.h, 2129
unpack_wds_SLQSGetProfileSettings_t, 1275
pProfileSettings, 1276
ProfileType, 1276
Tlvresult, 1276
unpack_wds_SLQSGetRuntimeSettings
wds.h, 2129
unpack_wds_SLQSGetRuntimeSettings_t, 1276
APNName, 1278
Authentication, 1278
DomainList, 1278
GPRSGrantedQoS, 1278
GWAddressV4, 1278
IMCNflag, 1278
IPFamilyPreference, 1278
IPV6AddrInfo, 1278
IPV6GWAddrInfo, 1278
IPv4, 1278
Mtu, 1278
PDPType, 1278
PrimaryDNSV4, 1279
PrimaryDNSV6, 1279
ProfileID, 1279

ProfileName, 1279
SecondaryDNSV4, 1279
SecondaryDNSV6, 1279
ServerAddrList, 1279
SubnetMaskV4, 1279
Technology, 1279
UMTSGrantedQoS, 1279
Username, 1279
unpack_wds_SLQSModifyProfile
wds.h, 2129
unpack_wds_SLQSModifyProfile_t, 1279
pExtErrorCode, 1279
unpack_wds_SLQSResetPacketStatics
wds.h, 2130
unpack_wds_SLQSSGetDHCPv4ClientConfig
wds.h, 2131
unpack_wds_SLQSSGetDHCPv4ClientConfig_t, 1284
pHwConfig, 1284
unpack_wds_SLQSSGetLoopback
wds.h, 2132
unpack_wds_SLQSSGetLoopback_t, 1284
ByteLoopbackMode, 1285
ByteLoopbackMultiplier, 1285
unpack_wds_SLQSSSetDHCPv4ClientConfig
wds.h, 2132
unpack_wds_SLQSSSetLoopback
wds.h, 2132
unpack_wds_SLQSSet3GPPConfigItem
wds.h, 2130
unpack_wds_SLQSSetIPFamilyPreference
wds.h, 2130
unpack_wds_SLQSSetIPFamilyPreference_t, 1279
Tlvresult, 1280
unpack_wds_SLQSSetPacketSrvStatusCallback
wds.h, 2130
unpack_wds_SLQSSetPacketSrvStatusCallback_t, 1280
bearerID, 1281
conn_status, 1281
ipFamily, 1281
reconfigReqd, 1281
sessionEndReason, 1281
techName, 1281
verboseSessnEndReason, 1282
verboseSessnEndReasonType, 1282
unpack_wds_SLQSSetWdsEventCallback
wds.h, 2131
unpack_wds_SLQSSetWdsEventCallback_ind
wds.h, 2131
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1282
currDBTechAvail, 1283
currNWInfo, 1283
dBTechAvail, 1283
dBTechnology, 1283
dBtechExtAvail, 1283
dBtechnologyExt, 1283
dataSysStatAvail, 1283
dormancyStatAvail, 1283
dormancyStatus, 1283
mipStatus, 1283
mipstatAvail, 1283
netInfoLen, 1283
prefNetwork, 1283
ratMask, 1283
rx_bytes, 1283
rx_pkts, 1283
soMask, 1283
tx_bytes, 1283
tx_pkts, 1283
xferStatAvail, 1283
unpack_wds_SLQSStartDataSession
wds.h, 2133
unpack_wds_SLQSStartDataSession_t, 1285
pFailureReason, 1285
pVerboseFailReasonType, 1285
pVerboseFailureReason, 1285
psid, 1285
unpack_wds_SLQSStopDataSession
wds.h, 2133
unpack_wds_SLQSWdsGoActive
wds.h, 2133
unpack_wds_SLQSWdsGoDormant
wds.h, 2134
unpack_wds_SLQSWdsSetEventReport
wds.h, 2134
unpack_wds_SLQSWdsSwiPDPRuntimeSettings
wds.h, 2134
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 1285
apnName, 1287
bearerId, 1287
contextId, 1287
ipv4Address, 1287
ipv4GWAddress, 1287
ipv6Address, 1287
ipv6GWAddress, 1287
unpack_wds_SetAutoconnect
wds.h, 2124
unpack_wds_SetDefaultProfile
wds.h, 2124
unpack_wds_SetDefaultProfileNum
wds.h, 2125
unpack_wds_SetMobileIP
wds.h, 2125
unpack_wds_SetMobileIPParameters
wds.h, 2125
unpack_wds_SetMobileIPProfile
wds.h, 2126
unpack_wds_SetMobileIPProfile_t, 1267
unpack_wds_SetMuxID
wds.h, 2126
UnpackQmiProfileInfo
wds.h, 2098
unpackWdsProfileParam, 1288
SlqsProfile3GPP, 1289
SlqsProfile3GPP2, 1289

upLink
NSSAudioCtrl, 586
UpdateCompleteStatus
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1210
updateCompleteStatus
omaDmFotaTlv, 592
unpack_omaDmFotaTlv_t, 1160
upgrade_mc77xx_fw
qaGobiApiFms.h, 1730
UpgradeFirmware2k
qaGobiApiFms.h, 1730
upinRetries
slotInf, 879
slotInfo, 881
uim_slotInfo, 998
upinState
slotInf, 880
slotInfo, 881
uim_slotInfo, 998
UpkQmiCbkCatEventReportInd
qaCbkCatEventReportInd.h, 1579
UpkQmiCbkSwiOmaDmEventReportInd
qaCbkSwiOmaDmEventReportInd.h, 1580
UpkQmiCbkSwiOmaDmEventReportIndExt
qaCbkSwiOmaDmEventReportInd.h, 1580
UpkQmiNasGetRFBandInfo
qaNasGetRFBandInfo.h, 1994
UpkQmiNasPerformNetworkScan
qaNasPerformNetworkScan.h, 1995
upukRetries
slotInf, 880
slotInfo, 881
uim_slotInfo, 998
urlAddr, 1289
address, 1289
TlvPresent, 1289
usageMask
loc_sensorDataUsage, 377
sensorDataUsage_s, 843
t_sensor, 955
User Identity Module Service (UIM), 52
userData
SMSAsyncRawSend_s, 912
userInputReq
omaDmConfigTlv, 587
omaDmConfigTlvExt, 590
omaDmFotaTlv, 592
unpack_omaDmConfigTlv_t, 1158
unpack_omaDmFotaTlv_t, 1160
userInputTimeout
omaDmConfigTlv, 587
omaDmConfigTlvExt, 590
omaDmFotaTlv, 592
omaDmFotaTlvExt, 594
unpack_omaDmConfigTlv_t, 1158
unpack_omaDmFotaTlv_t, 1160
Username
unpack_wds_SLQSGetRuntimeSettings_t, 1279
username
unpack_wds_GetDefaultProfile_t, 1258
usersize
unpack_wds_GetDefaultProfile_t, 1258
ussDCS
pack_voice_SLQSOriginateUSSD_t, 695
USSInfo, 1294
voice_USSInfo, 1332
ussData
pack_voice_SLQSOriginateUSSD_t, 695
USSInfo, 1294
voice_USSInfo, 1331
ussLen
pack_voice_SLQSOriginateUSSD_t, 695
USSInfo, 1294
voice_USSInfo, 1332
uusInfo
allCallsUUSInfo, 118
voice_allCallsUUSInfo, 1298

v4sessionId
qaQmiInterfaceInfo, 775
ssdatasession_params, 932
WdsRunTimeSettings, 1426
v6sessionId
qaQmiInterfaceInfo, 775
ssdatasession_params, 932
WdsRunTimeSettings, 1426
VOICE_SUPS_SRV_CLASS_DATA
qaGobiApiVoice.h, 1933
VOICE_SUPS_SRV_CLASS_DATACIRCUITASYNC
qaGobiApiVoice.h, 1933
VOICE_SUPS_SRV_CLASS_DATACIRCUITSYNC
qaGobiApiVoice.h, 1933
VOICE_SUPS_SRV_CLASS_FAX
qaGobiApiVoice.h, 1933
VOICE_SUPS_SRV_CLASS_NONE
qaGobiApiVoice.h, 1933
VOICE_SUPS_SRV_CLASS_PACKETACCESS
qaGobiApiVoice.h, 1933
VOICE_SUPS_SRV_CLASS_PADACCESS
qaGobiApiVoice.h, 1933
VOICE_SUPS_SRV_CLASS_SMS
qaGobiApiVoice.h, 1933
VOICE_SUPS_SRV_CLASS_VOICE
qaGobiApiVoice.h, 1933
VDOP
loc_precisionDilution, 374
precisionDilution_s, 758
VOICE_SRV
qaGobiApiCbk.h, 1595
val
IPv6TrafCls, 335
Tos, 970
unpack_qos_IPv6TrafCls_t, 1175
unpack_qos_Tos_t, 1193
val_3GPP2Pri
unpack_qos_swiQosFlow_t, 1191

val_3GPPImCn
unpack_qos_swiQosFlow_t, 1191
val_3GPPResResidualBER
unpack_qos_swiQosFlow_t, 1191
val_3GPPSigInd
unpack_qos_swiQosFlow_t, 1191
val_3GPPTraHdlPri
unpack_qos_swiQosFlow_t, 1191
ValidMask
GPSStateInfo, 293
unpack_pds_SLQSGetGPSStateInfo_t, 1172
validMask
loc_satelliteInfo, 376
satelliteInfo, 838
ValidateSPC
qaGobiApiDms.h, 1711
ValidityCW0
LteCQIParm, 394
unpack_nas_SLQSSwiGetLteCQI_t, 1153
ValidityCW1
LteCQIParm, 394
unpack_nas_SLQSSwiGetLteCQI_t, 1153
Value
GetM2MSpkrGainResp, 281
SetM2MSpkrGainReq, 865
value
pack_dms_UIMSetPINProtection_t, 610
pack_dms_UIMVerifyPIN_t, 611
value_length
custSettingInfo, 211
DMScustSettingInfo, 230
pack_dms_SetCustFeaturesV2_t, 602
setCustomSettingV2, 855
verbFailReason
ssdatasession_params, 932
verbFailReasonType
ssdatasession_params, 932
verboseSessnEndReason
_packetSrvStatus, 70
unpack_wds_SLQSSetPacketSrvStatusCallback_t, 1282
verboseSessnEndReasonType
_packetSrvStatus, 70
unpack_wds_SLQSSetPacketSrvStatusCallback_t, 1282
verifyLeft
personalizationStatus, 746
remainingRetries, 820
uim_personalizationStatus, 990
uim_remainingRetries, 994
verifyPIN
pack_uim_VerifyPin_t, 693
UIMVerifyPinReq, 1022
verifyRetriesLeft
unpack_dms_UIMGetControlKeyStatus_t, 1078
unpack_dms_UIMSetControlKeyProtection_t, 1080
unpack_dms_UIMSetPINProtection_t, 1081
verifyUIMPIN, 1296
pinID, 1296
pinLen, 1296
pinVal, 1296
version
omaDmFotaTlv, 592
omaDmFotaTlvExt, 594
swiQosFilter, 947
unpack_omaDmFotaTlv_t, 1160
unpack_qos_swiQosFilter_t, 1188
versionString
pack_dms_SLQSSwiSetOSInfo_t, 607
unpack_dms_SLQSSwiGetOSInfo_t, 1075
versionlength
omaDmFotaTlv, 592
omaDmFotaTlvExt, 594
unpack_omaDmFotaTlv_t, 1160
vertConfidence
pack_loc_SLQSLOCInjectPosition_t, 623
vertReliability
pack_loc_SLQSLOCInjectPosition_t, 623
vertUnc
pack_loc_SLQSLOCInjectPosition_t, 623
VerticalUncertainty
GPSStateInfo, 293
unpack_pds_SLQSGetGPSStateInfo_t, 1173
VirtStream
nas_protocolSubtypeElement, 496
protocolSubtypeElement, 773
Voice Service (VOICE), 50
voice.h
LITE_VOICE_SUPS_SRV_CLASS_DATA, 2069
LITE_VOICE_SUPS_SRV_CLASS_DATACIRCUITASYNC, 2070
LITE_VOICE_SUPS_SRV_CLASS_DATACIRCUITSYNC, 2070
LITE_VOICE_SUPS_SRV_CLASS_FAX, 2069
LITE_VOICE_SUPS_SRV_CLASS_NONE, 2069
LITE_VOICE_SUPS_SRV_CLASS_PACKETACCESS, 2070
LITE_VOICE_SUPS_SRV_CLASS_PADACCESS, 2070
LITE_VOICE_SUPS_SRV_CLASS_SMS, 2070
LITE_VOICE_SUPS_SRV_CLASS_VOICE, 2069
voice.h, 2065
liteServiceClassInformation, 2069
MAXVOICEUSSDLENGTH, 2069
pack_voice_AnswerUSSD, 2070
pack_voice_CancelUSSD, 2070
pack_voice_OriginateUSSD, 2070
pack_voice_SLQSOriginateUSSD, 2071
pack_voice_SLQSVoiceALSSelectLine, 2071
pack_voice_SLQSVoiceALSSetLineSwitching, 2071
pack_voice_SLQSVoiceAnswerCall, 2072
pack_voice_SLQSVoiceBindSubscription, 2072
pack_voice_SLQSVoiceBurstDTMF, 2073
pack_voice_SLQSVoiceDialCall, 2073
pack_voice_SLQSVoiceEndCall, 2073

pack_voice_SLQSVoiceGetAllCallInfo, 2074
pack_voice_SLQSVoiceGetCLIP, 2076
pack_voice_SLQSVoiceGetCLIR, 2076
pack_voice_SLQSVoiceGetCNAP, 2076
pack_voice_SLQSVoiceGetCOLP, 2077
pack_voice_SLQSVoiceGetCOLR, 2077
pack_voice_SLQSVoiceGetCallBarring, 2074
pack_voice_SLQSVoiceGetCallForwardingStatus, 2074
pack_voice_SLQSVoiceGetCallInfo, 2075
pack_voice_SLQSVoiceGetCallWaiting, 2075
pack_voice_SLQSVoiceGetConfig, 2077
pack_voice_SLQSVoiceIndicationRegister, 2078
pack_voice_SLQSVoiceManageCalls, 2078
pack_voice_SLQSVoiceOrigUSSDNoWait, 2078
pack_voice_SLQSVoiceSendFlash, 2079
pack_voice_SLQSVoiceSetCallBarringPassword, 2079
pack_voice_SLQSVoiceSetConfig, 2079
pack_voice_SLQSVoiceSetPreferredPrivacy, 2080
pack_voice_SLQSVoiceSetSUPSService, 2080
pack_voice_SLQSVoiceStartContDTMF, 2081
pack_voice_SLQSVoiceStopContDTMF, 2081
unpack_voice_AnswerUSSD, 2081
unpack_voice_CancelUSSD, 2082
unpack_voice_OriginateUSSD, 2082
unpack_voice_SLQSOriginateUSSD, 2082
unpack_voice_SLQSVoiceALSSelectLine, 2083
unpack_voice_SLQSVoiceALSSetLineSwitching, 2083
unpack_voice_SLQSVoiceAnswerCall, 2083
unpack_voice_SLQSVoiceBindSubscription, 2084
unpack_voice_SLQSVoiceBurstDTMF, 2084
unpack_voice_SLQSVoiceDialCall, 2084
unpack_voice_SLQSVoiceEndCall, 2085
unpack_voice_SLQSVoiceGetAllCallInfo, 2085
unpack_voice_SLQSVoiceGetCLIP, 2087
unpack_voice_SLQSVoiceGetCLIR, 2087
unpack_voice_SLQSVoiceGetCNAP, 2087
unpack_voice_SLQSVoiceGetCOLP, 2088
unpack_voice_SLQSVoiceGetCOLR, 2088
unpack_voice_SLQSVoiceGetCallBarring, 2085
unpack_voice_SLQSVoiceGetCallForwarding-Status, 2086
unpack_voice_SLQSVoiceGetCallInfo, 2086
unpack_voice_SLQSVoiceGetCallWaiting, 2086
unpack_voice_SLQSVoiceGetConfig, 2088
unpack_voice_SLQSVoiceIndicationRegister, 2089
unpack_voice_SLQSVoiceManageCalls, 2089
unpack_voice_SLQSVoiceOrigUSSDNoWait, 2089
unpack_voice_SLQSVoiceSendFlash, 2090
unpack_voice_SLQSVoiceSetCallBarringPassword, 2090
unpack_voice_SLQSVoiceSetConfig, 2090
unpack_voice_SLQSVoiceSetPreferredPrivacy, 2091
unpack_voice_SLQSVoiceSetSUPSService, 2091
unpack_voice_SLQSVoiceStartContDTMF, 2091
unpack_voice_SLQSVoiceStopContDTMF, 2092
voice_CLIPResp, 1314
ActiveStatus, 1315
ProvisionStatus, 1315
voice_CLIRResp, 1315
ActiveStatus, 1316
ProvisionStatus, 1316
voice_CNAPResp, 1316
ActiveStatus, 1316
ProvisionStatus, 1316
voice_COLPResp, 1317
ActiveStatus, 1317
ProvisionStatus, 1317
voice_COLRResp, 1317
ActiveStatus, 1318
ProvisionStatus, 1318
voice_CUGInfo, 1320
CUGIndex, 1320
SuppOA, 1320
SuppPrefCUG, 1320
voice_DTMFLengths, 1322
DTMFInterdigitInterval, 1322
DTMFPulseWidth, 1322
voice_USSInfo, 1331
ussDCS, 1332
ussData, 1331
ussLen, 1332
voice_UUSInfo, 1332
UUSData, 1333
UUSDatalen, 1333
UUSDcs, 1333
UUSType, 1333
voice_airTimer, 1296
airTimerValue, 1297
namID, 1297
voice_allCallsAlphaIDInfo, 1297
AlphaIDInfo, 1297
callID, 1297
voice_allCallsDiagInfo, 1297
callID, 1298
DiagInfo, 1298
voice_allCallsUUSInfo, 1298
callID, 1298
uusInfo, 1298
voice_alphaIDInfo, 1298
alphaDcs, 1299
alphaLen, 1299
alphaText, 1299
voice_arrAlertingPattern, 1299
alertingPattern, 1300
callID, 1300
numInstances, 1300
voice_arrAlertingType, 1300
AlertingType, 1301
callID, 1301
numInstances, 1301
voice_arrAlphaID, 1301
allCallsAlphaIDInfoArr, 1301

numInstances, 1301
voice_arrCallEndReason, 1302
callEndReason, 1302
callID, 1302
numInstances, 1302
voice_arrCallInfo, 1302
getAllCallInfo, 1303
numInstances, 1303
voice_arrCalledPartyNum, 1301
CalledPartyNum, 1302
numInstances, 1302
voice_arrConnectPartyNum, 1303
ConnectedPartyNum, 1303
numInstances, 1303
voice_arrDiagInfo, 1304
DiagInfo, 1304
numInstances, 1304
voice_arrRedirPartyNum, 1304
numInstances, 1304
RedirPartyNum, 1305
voice_arrRemotePartyName, 1305
GetAllCallRmtPtyName, 1305
numInstances, 1305
voice_arrRemotePartyNum, 1305
numInstances, 1306
RmtPtyNum, 1306
voice_arrSvcOption, 1306
callID, 1306
numInstances, 1306
srvOption, 1306
voice_arrUUSInfo, 1306
AllCallsUUSInfo, 1307
numInstances, 1307
voice_burstDTMFInfo, 1307
digitCnt, 1307
pCallID, 1307
pDigitBuff, 1307
voice_callFWExtInfo, 1309
noReplyTimer, 1311
numLen, 1311
numPlan, 1311
numType, 1311
number, 1311
PI, 1311
SI, 1311
SvcClass, 1311
SvcStatus, 1311
voice_callFWInfo, 1311
noReplyTimer, 1312
numLen, 1312
number, 1312
SvcClass, 1312
SvcStatus, 1312
voice_callFwdTypeAndPlan, 1309
numberPlan, 1309
numberType, 1309
voice_callInfo, 1312
callID, 1313
callState, 1313
callType, 1314
direction, 1314
mode, 1314
voice_calledPartySubAdd, 1308
extBit, 1308
oddEvenInd, 1308
subAddr, 1308
subAddrLen, 1308
subAddrType, 1308
voice_ccSUPSType, 1314
reason, 1314
svcType, 1314
voice_connectNumInfo, 1318
callerID, 1319
callerIDLen, 1319
numPlan, 1319
numPresInd, 1319
numType, 1320
screeningInd, 1320
voice_curAMRConfig, 1320
gsmAmrStat, 1321
wcdmaAmrStat, 1321
voice_diagInfo, 1321
diagInfoLen, 1321
diagnosticInfo, 1321
voice_getAllCallInformation, 1322
ALS, 1323
Callinfo, 1323
isEmpty, 1323
voice_getAllCallRmtPtyName, 1323
callID, 1323
RemotePartyName, 1323
voice_getAllCallRmtPtyNum, 1323
callID, 1324
RemotePartyNum, 1324
voice_getCallFWExtInfo, 1324
CallFWExtInfo, 1324
numInstances, 1324
voice_getCallFWInfo, 1325
CallFWInfo, 1325
numInstances, 1325
voice_peerNumberInfo, 1325
callID, 1327
numLen, 1327
numPI, 1327
numPlan, 1327
numSI, 1327
numType, 1327
number, 1327
voice_prefVoiceSO, 1327
evrcCapability, 1329
homeOrigVoiceSO, 1329
homePageVoiceSO, 1329
namID, 1329
roamOrigVoiceSO, 1329
voice_remotePartyName, 1329
callerName, 1329

codingScheme, 1330
nameLen, 1330
namePI, 1330
voice_remotePartyNum, 1330
numLen, 1330
presentationInd, 1330
remPartyNumber, 1330
voice_roamTimer, 1330
namID, 1331
roamTimerValue, 1331
voiceALSSelectLineInfo, 1333
lineValue, 1333
voiceALSSetLineSwitchInfo, 1333
switchOption, 1333
voiceAnswerCall, 1334
pCallId, 1334
voiceBindSubscriptionInfo, 1334
subsType, 1334
voiceBurstDTMFInfo, 1334
BurstDTMFInfo, 1335
pBurstDTMFLengths, 1335
voiceCallInfoReq, 1335
callID, 1335
voiceCallInfoResp, 1335
pAlertType, 1338
pAlertingPattern, 1338
pAlphaIDInfo, 1338
pCallInfo, 1338
pConnectNumInfo, 1338
pDiagInfo, 1338
pOTASPStatus, 1338
pRemotePartyName, 1338
pRemotePartyNum, 1338
pSrvOpt, 1338
pUUSInfo, 1338
pVoicePrivacy, 1338
voiceCallRequestParams, 1338
callNumber, 1340
pCLIRType, 1340
pCUGInfo, 1340
pCallPartySubAdd, 1340
pCallType, 1340
pEmergencyCategory, 1340
pSvcType, 1340
pUUSInFo, 1340
voiceCallResponseParams, 1340
pAlphaIDInfo, 1341
pCCResultType, 1341
pCCSUPSType, 1341
pCallID, 1341
voiceContDTMFinfo, 1341
DTMFdigit, 1341
pCallID, 1341
voiceDTMFEventInfo, 1342
DTMFInformation, 1342
pOffLength, 1342
pOnLength, 1342
voiceFlashInfo, 1342
pCallID, 1343
pFlashPayLd, 1343
pFlashType, 1343
voiceGetAllCallInfo, 1343
pArrAlertingPattern, 1345
pArrAlertingType, 1345
pArrAlphaID, 1345
pArrCallEndReason, 1345
pArrCallInfo, 1345
pArrCalledPartyNum, 1345
pArrConnectPartyNum, 1345
pArrDiagInfo, 1345
pArrRedirPartyNum, 1345
pArrRemotePartyName, 1345
pArrRemotePartyNum, 1345
pArrSvcOption, 1345
pArrUUSInfo, 1346
pOTASPStatus, 1346
pVoicePrivacy, 1346
voiceGetCLIPResp, 1351
pAlphaIDInfo, 1352
pCCResType, 1352
pCCSUPSType, 1352
pCLIPResp, 1352
pCallID, 1352
pFailCause, 1352
voiceGetCLIRResp, 1352
pAlphaIDInfo, 1353
pCCResType, 1354
pCCSUPSType, 1354
pCLIRResp, 1354
pCallID, 1353
pFailCause, 1354
voiceGetCNAPResp, 1354
pAlphaIDInfo, 1355
pCCResType, 1355
pCCSUPSType, 1355
pCNAPResp, 1355
pCallID, 1355
pFailCause, 1355
voiceGetCOLPResp, 1355
pAlphaIDInfo, 1356
pCCResType, 1356
pCCSUPSType, 1356
pCOLPResp, 1356
pCallID, 1356
pFailCause, 1356
voiceGetCOLRResp, 1356
pAlphaIDInfo, 1357
pCCResType, 1357
pCCSUPSType, 1357
pCOLRResp, 1357
pCallID, 1357
pFailCause, 1357
voiceGetCallBarringReq, 1346
pSvcClass, 1346
reason, 1346
voiceGetCallBarringResp, 1346

pAlphaIDInfo, 1347
pCCResType, 1348
pCCSUPSType, 1348
pCallID, 1348
pFailCause, 1348
pSvcClass, 1348
voiceGetCallFWReq, 1348
pSvcClass, 1348
Reason, 1348
voiceGetCallFWResp, 1348
pAlphaIDInfo, 1350
pCCResType, 1350
pCCSUPSType, 1350
pCallID, 1350
pFailCause, 1350
pGetCallFWExtInfo, 1350
pGetCallFWInfo, 1350
voiceGetCallWaitInfo, 1350
pAlphaIDInfo, 1351
pCCResType, 1351
pCCSUPSType, 1351
pCallID, 1351
pFailCause, 1351
pSvcClass, 1351
voiceGetConfigReq, 1357
pAMRStatus, 1359
pAirTimer, 1359
pAutoAnswer, 1359
pNamID, 1359
pPrefVoicePrivacy, 1359
pPrefVoiceSO, 1359
pRoamTimer, 1359
pTTYMode, 1359
pVoiceDomainPref, 1359
voiceGetConfigResp, 1359
pAirTimerCnt, 1361
pAutoAnswerStat, 1361
pCurAMRConfig, 1361
pCurPrefVoiceSO, 1361
pCurVoiceDomainPref, 1361
pCurVoicePrivacyPref, 1361
pCurrTTYMode, 1361
pRoamTimerCnt, 1361
voiceIndicationRegisterInfo, 1361
pRegDTMFEvents, 1362
pRegVoicePrivacyEvents, 1362
pSuppsNotifEvents, 1362
voiceInfoRec, 1362
callID, 1364
pCLIRCause, 1364
pCallWaitInd, 1364
pCalledPartyInfo, 1364
pCallerIDInfo, 1364
pCallerNameInfo, 1364
pCallingPartyInfo, 1364
pConnectNumInfo, 1364
pDispInfo, 1364
pExtDispInfo, 1364
pExtDispRecInfo, 1364
pLineCtrlInfo, 1364
pNSSAudioCtrl, 1364
pNSSRelease, 1364
pRedirNumInfo, 1364
pSignalInfo, 1364
voiceManageCallsReq, 1364
pCallID, 1365
SUPSType, 1365
voiceManageCallsResp, 1365
pFailCause, 1366
VoiceNumber
unpack_dms_GetVoiceNumber_t, 1058
voiceNumberSize
unpack_dms_GetVoiceNumber_t, 1058
voiceOTASPStatusInfo, 1366
callID, 1367
OTASPStatus, 1367
voiceOrigUSSDNoWaitInfo, 1366
USSInformation, 1366
voicePrivacy
voicePrivacyInfo, 1368
voicePrivacyInfo, 1367
callID, 1368
voicePrivacy, 1368
voiceSUPSInfo, 1380
pAlphaIDInfo, 1382
pCLIPstatus, 1382
pCLIRstatus, 1382
pCNAPstatus, 1382
pCOLPstatus, 1382
pCOLRstatus, 1382
pCallBarPasswd, 1382
pCallFWNum, 1382
pCallFWTimerVal, 1382
pCallFwdInfo, 1382
pCallID, 1382
pDataSrc, 1382
pFailCause, 1382
pNewPwdData, 1382
pReason, 1382
pSvcClass, 1382
pUSSInfo, 1382
SUPSInformation, 1382
voiceSUPSNotification, 1382
callID, 1384
notifType, 1384
pCUGIndex, 1384
pECTNum, 1384
voiceSetAllCallStatusCbkInfo, 1368
arrCallInfomation, 1370
pArrAlertingPattern, 1370
pArrAlertingType, 1370
pArrAlphaID, 1370
pArrCallEndReason, 1370
pArrCalledPartyNum, 1370
pArrConnectPartyNum, 1370
pArrDiagInfo, 1370

pArrRedirPartyNum, 1370
pArrRemotePartyName, 1370
pArrRemotePartyNum, 1370
pArrSvcOption, 1370
voiceSetCallBarringPwdInfo, 1370
newPasswd, 1371
newPasswdAgain, 1371
oldPasswd, 1371
Reason, 1371
voiceSetCallBarringPwdResp, 1371
pAlphaIDInfo, 1372
pCCResType, 1372
pCCSUPSType, 1372
pCallID, 1372
pFailCause, 1372
voiceSetConfigReq, 1372
pAirTimerConfig, 1373
pAutoAnswer, 1374
pPrefVoiceDomain, 1374
pPrefVoiceSO, 1374
pRoamTimerConfig, 1374
pTTYMode, 1374
voiceSetConfigResp, 1374
pAirTimerStatus, 1375
pAutoAnsStatus, 1375
pPrefVoiceSOStatus, 1375
pRoamTimerStatus, 1375
pTTYConfigStatus, 1375
pVoiceDomainPrefStatus, 1375
voiceSetPrefPrivacy, 1375
privacyPref, 1376
voiceSetSUPSServiceReq, 1376
pCallBarringPasswd, 1378
pCallForwardingNumber, 1378
pCallFwdTypeAndPlan, 1378
pServiceClass, 1378
pTimerVal, 1378
reason, 1378
voiceSvc, 1378
voiceSetSUPSServiceResp, 1378
pAlphaIDInfo, 1379
pCCResultType, 1379
pCCSUPSType, 1379
pCallID, 1379
pFailCause, 1379
voiceStopContDTMFinfo, 1379
callID, 1380
voiceSvc
pack_voice_SLQSVoiceSetSUPSService_t, 711
voiceSetSUPSServiceReq, 1378
VolValue
SetAudioVolTLBConfigReq, 854
Volume
GetAudioProfileResp, 261
GetAudioVolTLBConfigReq, 262
GetM2MAudioProfileResp, 278
SetAudioProfileReq, 853
SetAudioVolTLBConfigReq, 854
voteForInit
registerRefresh, 820
uim_registerRefresh, 994

WAMSParaChanged
_SwiAvmsEventInd, 100
wAlertMsgLength
_QmiSwiAvmsEventReportConfigResp, 74
WCDMACellInfo
lteWcdmaCellInfo, 414
nas_lteWcdmaCellInfo, 476
WCDMAECIOThresh, 1385
pWCDMAECIOThreshList, 1385
WCDMAECIOThreshListLen, 1385
WCDMAECIOThreshListLen
nas_WCDMAECIOThresh, 529
WCDMAECIOThresh, 1385
WCDMAInfoLTENeighborCell, 1385
UMTSLTENbrCell, 1386
umtsLTENbrCellLen, 1386
wcdmaRRCState, 1386
WCDMARSSIThresh, 1390
pWCDMARSSIThreshList, 1391
WCDMARSSIThreshListLen, 1391
WCDMARSSIThreshListLen
nas_WCDMARSSIThresh, 531
WCDMARSSIThresh, 1391
WCDMASSInfo
unpack_nas_SLQSNasGetSigInfo_t, 1144
WCDMASysInfo, 1391
cellId, 1393
cellIdValid, 1394
hsCallStatus, 1394
hsCallStatusValid, 1394
hsInd, 1394
hsIndValid, 1394
lac, 1394
lacValid, 1394
MCC, 1394
MNC, 1394
networkIdValid, 1394
psc, 1394
pscValid, 1394
regRejectInfoValid, 1394
rejCause, 1394
rejectSrvDomain, 1394
sysInfoWCDMA, 1394
WDS_IsGobiDevice
qaGobiApiWds.h, 1993
WDS_SRV
qaGobiApiCbk.h, 1595
WDSGetLoopbackData, 1421
ByteLoopbackMode, 1421
ByteLoopbackMultiplier, 1421
WDSSWICurrentChannelRates, 1429
current_channel_rx_rate, 1429
current_channel_tx_rate, 1429
max_channel_rx_rate, 1429
max_channel_tx_rate, 1429

WDSSetLoopbackData, 1428
pLoopbackMode, 1429
pLoopbackMultiplier, 1429
wDescriptionLength
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 73
wErrorCode
_QmiSwiAvmsEventReportDataSessionStatusResp, 75
wHTTPStatus
_QmiSwiAvmsEventReportHTTPStatusResp, 76
wNameLength
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 73
WORD
SwiDataTypes.h, 2036
wSessionStatus
_QmiSwiAvmsEventReportNotificationResp, 78
wUpdateCompeteStatus
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 73
wUserInputTimeout
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 73
_QmiSwiAvmsEventReportConfigResp, 74
_QmiSwiAvmsEventReportConnectionRequestResp, 75
wVersionLength
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 73
wWamsChangeMask
_QmiSwiAvmsEventReportWAMSParamChangeResp, 79
wake_mask
pack_rms_SetSMSWake_t, 667
unpack_rms_GetSMSWake_t, 1194
wcdmaAmrStat
curAMRConfig, 199
voice_curAMRConfig, 1321
wcdmaCellInfo, 1384
cpich_ecno, 1384
cpich_rscp, 1384
psc, 1384
srxlev, 1385
wcdmaLongMsgDecodingParams, 1386
Date, 1387
pIsUDHPresent, 1387
pMessage, 1387
pPartNum, 1388
pReferenceNum, 1388
pScAddr, 1388
pScAddrLength, 1388
pSenderAddr, 1388
pSenderAddrLength, 1388
pTextMsg, 1388
pTextMsgLength, 1388
pTotalNum, 1388
Time, 1388
wcdmaMsgDecodingParams, 1388
Date, 1389
pMessage, 1389
pScAddr, 1389
pScAddrLength, 1389
pSenderAddr, 1389
pSenderAddrLength, 1389
pTextMsg, 1389
pTextMsgLength, 1389
Time, 1389
wcdmaMsgEncodingParams, 1389
alphabet, 1390
messageSize, 1390
pDestAddr, 1390
pPDUMessage, 1390
pTextMsg, 1390
wcdmaRRCState
nas_WCDMAInfoLTENeighborCell, 530
WCDMAInfoLTENeighborCell, 1386
wcdmaUARFCN, 1394
status, 1395
uarfcn, 1395
wds.h
QMI_LITE_WDS_CURRENT_CALL_DB_MASK, 2098
QMI_LITE_WDS_LAST_CALL_DB_MASK, 2098
wds.h, 2092
BYT_STAT_STAT_MASK, 2098
liteQmiDataBearerMasks, 2098
PACK_WDS_IPV4, 2098
PACK_WDS_IPV6, 2098
pack_wds_DHCPv4ClientLeaseChange, 2099
pack_wds_GetAutoconnect, 2099
pack_wds_GetByteTotals, 2099
pack_wds_GetConnectionRate, 2100
pack_wds_GetDataBearerTechnology, 2100
pack_wds_GetDefaultProfile, 2100
pack_wds_GetDefaultProfileNum, 2101
pack_wds_GetDormancyState, 2101
pack_wds_GetLastMobileIPError, 2102
pack_wds_GetMobileIP, 2102
pack_wds_GetMobileIPProfile, 2103
pack_wds_GetPacketStatistics, 2103
pack_wds_GetPacketStatus, 2103
pack_wds_GetSessionDuration, 2104
pack_wds_GetSessionState, 2104
pack_wds_RMSetTransferStatistics, 2105
pack_wds_SLQSCreateProfile, 2108
pack_wds_SLQSDeleteProfile, 2108
pack_wds_SLQSGet3GPPConfigItem, 2109
pack_wds_SLQSGetCurrDataSystemStat, 2109
pack_wds_SLQSGetCurrentChannelRate, 2110
pack_wds_SLQSGetDUNCallInfo, 2111
pack_wds_SLQSGetDataBearerTechnology, 2110
pack_wds_SLQSGetProfileSettings, 2111
pack_wds_SLQSGetRuntimeSettings, 2111
pack_wds_SLQSModifyProfile, 2112
pack_wds_SLQSResetPacketStatics, 2112

pack_wds_SLQSSGetDHCPv4ClientConfig, 2114
pack_wds_SLQSSGetLoopback, 2114
pack_wds_SLQSSSetDHCPv4ClientConfig, 2115
pack_wds_SLQSSSetLoopback, 2115
pack_wds_SLQSSet3GPPConfigItem, 2113
pack_wds_SLQSSetIPFamilyPreference, 2113
pack_wds_SLQSSetWdsEventCallback, 2113
pack_wds_SLQSStartDataSession, 2115
pack_wds_SLQSStopDataSession, 2116
pack_wds_SLQSWdsGoActive, 2116
pack_wds_SLQSWdsGoDormant, 2117
pack_wds_SLQSWdsSetEventReport, 2117
pack_wds_SLQSWdsSwiPDPRuntimeSettings, 2118
pack_wds_SetAutoconnect, 2105
pack_wds_SetDefaultProfile, 2105
pack_wds_SetDefaultProfileNum, 2106
pack_wds_SetMobileIP, 2106
pack_wds_SetMobileIPParameters, 2107
pack_wds_SetMobileIPProfile, 2107
pack_wds_SetMuxID, 2108
unpack_wds_DHCPv4ClientLease_ind, 2118
unpack_wds_DHCPv4ClientLeaseChange, 2118
unpack_wds_GetAutoconnect, 2119
unpack_wds_GetByteTotals, 2119
unpack_wds_GetConnectionRate, 2119
unpack_wds_GetDataBearerTechnology, 2120
unpack_wds_GetDefaultProfile, 2120
unpack_wds_GetDefaultProfileNum, 2120
unpack_wds_GetDormancyState, 2121
unpack_wds_GetLastMobileIPError, 2121
unpack_wds_GetMobileIP, 2121
unpack_wds_GetMobileIPProfile, 2122
unpack_wds_GetPacketStatistics, 2122
unpack_wds_GetPacketStatus, 2122
unpack_wds_GetSessionDuration, 2123
unpack_wds_GetSessionState, 2123
unpack_wds_RMSetTransferStatistics, 2123
unpack_wds_RMTransferStatistics_ind, 2124
unpack_wds_SLQSCreateProfile, 2126
unpack_wds_SLQSDeleteProfile, 2127
unpack_wds_SLQSGet3GPPConfigItem, 2127
unpack_wds_SLQSGetCurrDataSystemStat, 2127
unpack_wds_SLQSGetCurrentChannelRate, 2128
unpack_wds_SLQSGetDUNCallInfo, 2128
unpack_wds_SLQSGetDataBearerTechnology, 2128
unpack_wds_SLQSGetProfileSettings, 2129
unpack_wds_SLQSGetRuntimeSettings, 2129
unpack_wds_SLQSModifyProfile, 2129
unpack_wds_SLQSResetPacketStatics, 2130
unpack_wds_SLQSSGetDHCPv4ClientConfig, 2131
unpack_wds_SLQSSGetLoopback, 2132
unpack_wds_SLQSSSetDHCPv4ClientConfig, 2132
unpack_wds_SLQSSSetLoopback, 2132
unpack_wds_SLQSSet3GPPConfigItem, 2130
unpack_wds_SLQSSetIPFamilyPreference, 2130
unpack_wds_SLQSSetPacketSrvStatusCallback, 2130
unpack_wds_SLQSSetWdsEventCallback, 2131
unpack_wds_SLQSSetWdsEventCallback_ind, 2131
unpack_wds_SLQSStartDataSession, 2133
unpack_wds_SLQSStopDataSession, 2133
unpack_wds_SLQSWdsGoActive, 2133
unpack_wds_SLQSWdsGoDormant, 2134
unpack_wds_SLQSWdsSetEventReport, 2134
unpack_wds_SLQSWdsSwiPDPRuntimeSettings, 2134
unpack_wds_SetAutoconnect, 2124
unpack_wds_SetDefaultProfile, 2124
unpack_wds_SetDefaultProfileNum, 2125
unpack_wds_SetMobileIP, 2125
unpack_wds_SetMobileIPParameters, 2125
unpack_wds_SetMobileIPProfile, 2126
unpack_wds_SetMuxID, 2126
UnpackQmiProfileInfo, 2098
wds_DHCPLeaseOptTlv, 1397
numOpt, 1397
optList, 1397
optListData, 1397
TlvPresent, 1398
wds_DHCPLeaseStateTlv, 1398
leaseState, 1398
TlvPresent, 1398
wds_DHCPOpt, 1398
optCode, 1399
optValLen, 1399
pOptVal, 1399
wds_DHCPProfileIdTlv, 1399
profileId, 1399
profileType, 1399
TlvPresent, 1399
wds_DHCPv4HWConfig, 1399
chaddr, 1400
chaddrLen, 1400
hwType, 1400
wds_DHCPv4Option, 1400
optCode, 1400
optVal, 1400
optValLen, 1401
wds_DHCPv4OptionList, 1401
numOpt, 1401
pOptList, 1401
wds_DHCPv4ProfileId, 1401
profileId, 1401
profileType, 1402
wds_DataULongLongTlv, 1396
TlvPresent, 1396
ullData, 1397
wds_DataULongTlv, 1397
TlvPresent, 1397
ulData, 1397
wds_Domain, 1402

domainLen, 1402
domainName, 1402
wds_DomainNameList, 1402
domain, 1403
numInstances, 1403
wds_GPRSQoS, 1403
delayClass, 1403
meanThroughputClass, 1403
peakThroughputClass, 1403
precedenceClass, 1403
reliabilityClass, 1403
wds_IPV6AddressInfo, 1404
IPAddressV6, 1404
IPV6PrefixLen, 1404
wds_IPV6GWAddressInfo, 1404
gwAddressV6, 1405
gwV6PrefixLen, 1405
wds_IPv4AdTlv, 1404
IPv4Addr, 1404
TlvPresent, 1404
wds_PCSCFFQDNAddress, 1405
fqdnAddr, 1405
fqdnLen, 1405
wds_PCSCFFQDNAddressList, 1406
numInstances, 1406
pcsfFQDNAddress, 1406
wds_PCSCFIPv4ServerAddressList, 1406
numInstances, 1407
pscsfIPv4Addr, 1407
wds_ProfileIdentifier, 1407
profileIndex, 1407
profileType, 1407
wds_TrStatInd, 1408
statsMask, 1409
statsPeriod, 1409
wds_UMTSMinQoS, 1409
deliveryErrSDU, 1411
grntDownlinkBitrate, 1411
grntUplinkBitrate, 1411
maxDownlinkBitrate, 1411
maxSDUSize, 1411
maxUplinkBitrate, 1411
qosDeliveryOrder, 1411
resBerRatio, 1411
sduErrorRatio, 1411
trafficClass, 1411
trafficPriority, 1411
transferDelay, 1411
wds_currNetworkInfo, 1395
NetworkType, 1396
RATMask, 1396
SOMask, 1396
wds_profileInfo, 1407
SlqsProfile3GPP, 1408
SlqsProfile3GPP2, 1408
wds_transferStatInd, 1408
StatsMask, 1408
StatsPeriod, 1408
WdsByteTotals, 1411
ByteTotalsElmntsV4, 1412
ByteTotalsElmntsV6, 1412
pV4sessionId, 1412
pV6sessionId, 1412
WdsByteTotalsElmnts, 1412
pRXTotalBytes, 1413
pTXTotalBytes, 1413
WdsClientLeaseChange, 1413
pEnableNotification, 1413
WdsConnectionRate, 1413
ConnRateElmntsV4, 1414
ConnRateElmntsV6, 1414
pV4sessionId, 1414
pV6sessionId, 1414
WdsConnectionRateElmnts, 1414
pCurrentChannelRXRate, 1415
pCurrentChannelTXRate, 1415
pMaxChannelRXRate, 1415
pMaxChannelTXRate, 1415
WdsDHCPv4ClientLeaseInd, 1415
pIPv4Addr, 1415
pLeaseState, 1415
pOptList, 1415
pProfileId, 1416
WdsDHCPv4Config, 1416
pHwConfig, 1416
pProfileId, 1416
pRequestOptionList, 1416
WdsDHCPv4HWConfig, 1416
chaddr, 1417
chaddrLen, 1417
hwType, 1417
WdsDHCPv4Option, 1418
optCode, 1418
optVal, 1419
optValLen, 1419
WdsDHCPv4OptionList, 1419
numOpt, 1420
pOptList, 1420
WdsDHCPv4ProfileId, 1420
profileId, 1420
profileType, 1420
wdsDhcpv4HwConfig, 1417
chaddr, 1417
chaddrLen, 1417
hwType, 1417
wdsDhcpv4Option, 1417
optCode, 1418
optVal, 1418
optValLen, 1418
wdsDhcpv4OptionList, 1419
numOpt, 1419
pOptList, 1419
wdsDhcpv4ProfileId, 1420
profileId, 1421
profileType, 1421
WdsIpAddressInfoReq, 1421

ip, 1422
pv4sessionId, 1422
pv6sessionId, 1422
WdsPktStatisticsElmnts, 1422
pRXDroppedCount, 1423
pRXOKBytesLastCall, 1423
pRXOkBytesCount, 1423
pRXPacketErrors, 1423
pRXPacketOverflows, 1423
pRXPacketSuccesses, 1423
pTXDroppedCount, 1424
pTXOKBytesLastCall, 1424
pTXOkBytesCount, 1424
pTXPacketErrors, 1424
pTXPacketOverflows, 1424
pTXPacketSuccesses, 1424
WdsPktStatisticsReq, 1424
pStatMask, 1424
WdsPktStatisticsResp, 1424
pV4sessionId, 1425
pV6sessionId, 1425
PktStatElmntsV4, 1425
PktStatElmntsV6, 1425
WdsProfileParam, 1425
SlqsProfile3GPP, 1425
SlqsProfile3GPP2, 1426
WdsRunTimeSettings, 1426
rts, 1426
v4sessionId, 1426
v6sessionId, 1426
wdsSetEventReportReq, 1426
pCurrChannelRateInd, 1428
pCurrDataBearerTechInd, 1428
pCurrPrefDataSysInd, 1428
pDataBearerTechInd, 1428
pDataCallStatusChangeInd, 1428
pDataSystemStatusChangeInd, 1428
pDormancyStatusInd, 1428
pEVDOPageMonPerChangeInd, 1428
pMIPStatusInd, 1428
pTransferStatInd, 1428
Wireless Data Service (WDS), 38

xAxis
sensorData, 841
sensorData_t, 842
xferStatAvail
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1283
xid
pack_loc_SLQSLOCGetBestAvailPos_t, 617
pack_qmi_t, 665
unpack_qmi_t, 1173
xtra_start_gps_minutes
GPSStateInfo, 293
unpack_pds_SLQSGetGPSStateInfo_t, 1173
xtra_start_gps_week
GPSStateInfo, 293
unpack_pds_SLQSGetGPSStateInfo_t, 1173
xtra_valid_duration_hours
GPSStateInfo, 293
unpack_pds_SLQSGetGPSStateInfo_t, 1173

yAxis
sensorData, 841
sensorData_t, 842
year
nas_timeInfo, 522
nas_UniversalTime, 528
timeInfo, 966
UniversalTime, 1033

zAxis
sensorData, 841
sensorData_t, 842