LTE eNodeB

AZS-XXX-0200 Small Cell AZCMCK2KBXB1 Baseband Unit HMI User Guide

Document Revision: 4.5

Date: 15.06.2018

Trademarks and Permissions

Notice

The purchased products, services and features are stipulated by the contract made between Azcom and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute the warranty of any kind, express or implied.

Azcom Technology srl

Headquarters:

Centro Direzionale Milanofiori

Strada 6, Palazzo N/2

20089 Rozzano (MI) – Italy

Tel.: +39 02 82450311

Website: www.azcom.it

E-mail: info@azcom.it

REVISION HISTORY

Rev.	Date	Description
1.0	16-May-2016	First Issue
2.0	16-Mar-2017	AZS-XXX-0200 Small Cell added
3.0	28-Mar-2017	Chapter 5.13.2 updated
4.0	26-Jun-2017	Document restructured to include new CLI commands
4.1	23-Aug-2017	OAM traces configuration and L2 measurements selection SIB timers for klish command
4.2	31-Aug-2017	Added description of L2 measurements and Handover commands
4.3	25-Sep-17	SOM NIB and S1 refactoring Added description for additional L2L3 statistics
4.4	27-Feb-18	Updates on Handover commands VLAN id support for s1ap/s1u through snmp and klish Updates to log parameters for NIB on SOM Added description for user inactivity timer support Added description for alarm status parameters UeTable moved under azStatistics as PHY stats tree
4.5	15-Jun-18	Added description for Alarms commands (ALARM-MIB) Statistics aggregation introduced for MAC,RLC,PDCP,S1U categories Added description for DSP statistics Added reference to facilities available in standard MIBs Note about NIB on ARM configuration

ABBREVIATIONS AND ACRONYMS

3GPP	Third-Generation Partnership Project
ARM	Advanced RISC Machine
BBU	Base Band Unit
BLER	Block Error Rate
CPRI	Common Public Radio Interface standard
CQI	Channel Quality Indicator
DB	DataBase
DL	Downlink
DRB	Data Radio Bearer
DSP	Digital Signal Processor
EARFCN	E-UTRA Absolute Radio Frequency Channel Number
eNodeB	Evolved NodeB
EPC	Evolved Packet core
FDD	Frequency Division Duplex
GPS	Global Positioning System
HMI	Human Machine Interface
HW	Hardware
IP	Internet Protocol
I/F	Interface
KLISH	Kommand Line Interface
LCID	Logical Channel ID
LTE	Long Term Evolution
MAC	Media Access Control
M/C	Machine
MCC	Mobile Country Code
MIB	Management Information Base
MME	Mobility Management Entity
MNC	Mobile Network Code
NIB	Network In a Box
NTP	Network Time Protocol
NACK	Negative Acknowledgement

OAM	Operation Administration and Maintenance
OID	Object Identifier
PDCP	Packet Data Convergence Protocol
PGW	Packet Data Network Gateway
PHY	PHYsical Layer
PLMN	Public Land Mobile Network
PUCCH	Physical Uplink Control Channel
PUSCH	Physical Uplink Shared Channel
RACH	Random Access Channel
RLC	Radio Link Control
RLF	Radio Link Failure
RNTI	Radio Network Temporary Identifier
RRC	Radio Resource Control
RRH	Remote Radio Head
RSSI	Received Signal Strength Indicator
SFP	Small Form-factor Pluggable
S1AP	S1 Application Protocol
S1U	S1 User Plane Protocol
SGW	Serving Gateway
SNMP	Simple Network Management Protocol
SNR	Signal To Noise Ratio
SOM	System On Module
SW	Software
TAC	Tracking Area Code
TB	Transmission Block
TCP	Transmission Control Protocol
TDD	Time Division Duplex
UDP	User Datagram Protocol
UE	User Equipment
UL	Uplink
USB	Universal Serial Bus
VLAN	Virtual Local Area Network
X2AP	X2 Application Protocol
X2U	X2 User Plane Protocol

CONTENTS

1 Introduction 11
1.1 Scope 11
1.2 Intended Audience 11
1.3 References 11
2 eNodeB Overview 12
3 AZCOM eNodeB MIB model 13
3.1 The SNMP MIB 13
3.1.1 Additional standard MIBs 27
3.2 Administrative Status and System Status Coherence 29
4 SNMP Interface 30
4.1 SNMP commands 30
4.2 SNMP users configuration 31
4.3 SNMP errors 31
4.4 Commands reference 32
4.4.1 Cell operational status 33
4.4.2 Node administrative status 34
4.4.3 Node software version 34
4.4.4 Cell administrative status 34
4.4.5 Cell functional status 34
4.4.6 RF Parameters & Antenna Configuration 35
4.4.7 Air interface link 37
4.4.8 TDD configurations 37
4.4.9 SIB2 timers 38
4.4.10 Transmission Mode 39
4.4.11 Status reporting 39
4.4.12 Node administration 40
4.4.13 Cell RF identifier parameter 41
4.4.14 RF operational status 41
4.4.15 Node and cell addressing 41
4.4.16 S1 interfaces 42
4.4.17 X2 interface 43
4.4.18 Clock Sync Source 44

4.4.19 Log server ... 44
4.4.20 Software Management ... 46
4.4.21 CPRI parameters ... 48
4.4.22 GPS Info parameters ... 50
4.4.23 NTP configuration ... 52
4.4.24 Configuration save ... 52
4.4.25 L2 measurements ... 53
4.4.26 Handover configurations ... 59
4.4.27 Statistics ... 74
4.4.28 PDCP User Inactivity Timer ... 86
4.4.29 Cell Max UE Capacity ... 86
4.4.30 Standard MIB information ... 86
4.4.31 Alarms ... 88
5 Command Line Interface ... 97
5.1 CLI views ... 97
5.2 Commands reference ... 97
5.2.1 eNodeB administration ... 98
5.2.2 Sync_src ... 100
5.2.3 Cell(s) ... 101
5.2.4 RF module ... 104
5.2.5 Log Server configuration ... 105
5.2.6 S1 interfaces ... 107
5.2.7 X2 interfaces ... 110
5.2.8 CPRI link management ... 112
5.2.9 GPS information module ... 113
5.2.10 L2 Measurements ... 116
5.2.11 Handover configurations ... 124
5.2.12 Statistics ... 154
5.2.13 Alarms ... 171

List of tables

TABLE 1. GLOBAL L2 MEAS SELECTION FLAGS ... 54
TABLE 2. QCI BASED L2 MEAS IN UL ... 55

TABLE 3. QCI BASED L2 MEAS IN DL 56
TABLE 4 X2 NEIGHBOR CELLS TABLE 61
TABLE 5 S1 NEIGHBOR CELLS TABLE 62
TABLE 6 UTRA NEIGHBOR CELLS TABLE 63
TABLE 7 HO REPORT CONFIGURATIONS TABLE 67
TABLE 8 HO MEASUREMENT OBJECTS TABLE 69
TABLE 9 HO MEASUREMENTS ID TABLE 70
TABLE 10 HO MEASUREMENTS CONFIG TABLE 73
TABLE 11 ALARM MESSAGES 89
TABLE 12 ALARM MODEL TABLE 90
TABLE 13 ACTIVE ALARM TABLE 94
TABLE 14 ACTIVE STATS TABLE 95
TABLE 15 LEGEND OF PARAMETERS PERMISSIONS........ 98
TABLE 16. NODE 99
TABLE 17. SYNC_SRC 100
TABLE 18. CELL 102
TABLE 19. TIMER 103
TABLE 20. RF 104
TABLE 21. TX PATH 104
TABLE 22. RX PATH 105
TABLE 23. ANTENNA 105
TABLE 24. LOG SERVER........ 106
TABLE 25. PHY/L2/L3/OAM 106
TABLE 26 EVENTS 107
TABLE 27. ADDR 107
TABLE 28. S1 INTERFACE 108
TABLE 29. S1AP 108
TABLE 30. S1U 109
TABLE 31. X2 INTERFACE 111
TABLE 32. X2AP 111
TABLE 33. X2U 111
TABLE 34. CPRI 113

TABLE 35. GPS 114

TABLE 36. SATELLITE 115

TABLE 37. CATEGORIES OF L2 MEASUREMENT 116

TABLE 38. CATEGORIES OF SELECTION FLAGS/VALUES 116

TABLE 39. GLOBAL SELECTION FLAGS 117

TABLE 40. UL BASED SELECTION FLAGS 117

TABLE 41. DL BASED SELECTION FLAGS 118

TABLE 42. GLOBAL MEASUREMENT VALUES 118

TABLE 43. UL BASED MEASUREMENT VALUES 119

TABLE 44. DL BASED MEASUREMENT VALUES 119

TABLE 45 HANDOVER CONFIGURATIONS 124

TABLE 46 X2 NEIGHBORS 125

TABLE 47 S1 NEIGHBORS 126

TABLE 48 UTRA NEIGHBORS 127

TABLE 49 HO REPORT CONFIGURATION 130

TABLE 50 HO MEASUREMENT OBJECT CONFIGURATION 131

TABLE 51 HO MEASUREMENT ID CONFIGURATION 132

TABLE 52 HO MEASUREMENT CONFIGURATIONS 135

TABLE 53 STATISTICS CATEGORIES 155

TABLE 54 MAC STATISTICS 156

TABLE 55 RLC STATISTICS 156

TABLE 56 PDCP STATISTICS 156

TABLE 57 S1U STATISTICS 156

TABLE 58 RRC STATISTICS 157

TABLE 59 GLOBAL STATISTICS 158

TABLE 60 RRC RNTI STATISTICS 158

TABLE 61 PHY STATISTICS 158

TABLE 62 DSP STATISTICS 159

TABLE 63 DSP MEMORY STATISTICS 159

TABLE 64 DSP PACKETS STATISTICS 160

TABLE 65 ALARM CATEGORIES 171

TABLE 66 ALARM MODEL TABLE 171

TABLE 67 ACTIVE ALARM TABLE 171

TABLE 68 ACTIVE ALARM STATS 172

1 Introduction

1.1 Scope

This document describes Azcom LTE eNodeB OAM user interface.

1.2 Intended Audience

The present document is intended for the Customers of Azcom LTE eNodeB.

1.3 References

[1] AZB-NC0M-4304 BBU User Manual

[2] AZB-NC0M-4304 BBU Platform SW Installation

[3] AZB-NC0M-4304 BBU HW Installation manual

[4] AZS-XXX-0200 Small Cell User Manual

[5] AZS-XXX-0200 Small Cell HW Installation manual

[6] AZS-XXX-0200 Platform Command Line Interface

2 eNodeB Overview

The AZCOM eNodeB OAM module permits the operations, administration and maintenance of the node through the SNMP protocol and the CLI.

This manual applies to two different families of products: AZB-NC0M-4304 is a Baseband Unit that can operate with several remote radio heads (RRH) elements through optical connections; AZS-XXX-0200 is an all-in-one Small Cell system with RF elements physically plugged into the main board.

Although the two systems share most of their management concepts, some parameters may be applicable to either one or the other system only, or to specific radio elements that are connected. For example, the AZS-XXX-0200 systems do not need a CPRI connection to communicate with the radio equipment; and the possibility to use FDD or TDD duplex modes depends on the characteristics of the radio equipment of choice.

Finally, some functionalities may require a specific license in order to have them available.

3 AZCOM eNodeB MIB model

The MIB object model of the AZCOM eNodeB is composed by parameters grouped in several logic elements. These logic elements are:

- The eNodeB administration
- The cell(s)
- The RF module
- The S1 interface
- The X2 interface
- The log server configuration
- The Software Management
- The CPRI link management
- The GPS information module
- The NTP configuration
- The L2 measurements
- The configurations for handover
- The system statistics

3.1 The SNMP MIB

Every logic group and element representing a parameter of the node is identified by an OID in the model, in a hierarchical manner. The complete list of eNodeB's parameters managed by the SNMP protocol can be represented by a tree, where every parameter is considered a leaf. This tree for the AZCOM eNodeB MIB model is reported here, where every parameter name is preceded by the access level (R, RW, CR), the type of the parameter (integer, enumerative, string) and is followed by the last number of the OID in brackets:

```
+--eNodeB(1)
   |
   +-- -RW- EnumVal   eNodeBadmStatus(1)
   |        Values: lock(0), unlock(1)
   +-- -RW- EnumVal   eNodeBopStatus(2)
   |        Values: idleState(0), startInprogress(1), active(2),
   |                stopInprogress(3), rebootInprogress(4),
   |                stopFail(5), startFail(6), invalid(7), start(8),
   |                halt(9), restart(10)
   +-- -R-- EnumVal   eNodeBalarm(3)
   |        Values: clear(0), stopFail(1), startFail(2), willReset(3),
   |                willReboot(4)
   +-- -RW- Integer32 azNodeInactivityTimer(4)
   |        Range: 1..3600
   +-- -RW- String    name(6)
   |        Size: 1..512
```

```
+-- -R-- EnumVal   type(7)
|        Values: macro(0), small(1), home(2)
+-- -RW- String mcc(8)
|        Size: 3
+-- -RW- String mnc(9)
|        Size: 2..3
+-- -RW- EnumVal   syncSource(10)
|        Values: extclock(1), gps(2), ieee1588(3)
+-- -R-- EnumVal   clockLockStatus(11)
|        Values: unlocked(0), locked(1)
+-- -RW- Integer32 tac(12)
|        Range: 0..65535
+-- -R-- String    azNodeVersion(13)
|        Size: 1..50
+-- -R-- String    azNodeReleaseID(14)
|        Size: 1..50
+-- -RW- EnumVal   azConfigSave (15)
|        Values: success(0), active(1), error(2), mismatch(3)
+-- -RW- EnumVal   azNIBMode(16)
|        Values: none(0), bbu(1), som(2)
|
+--cellsTable(17)
|  |
|  +--cellsEntry(1)
|     |  Index: cellIndex
|     |
|     +-- ---- Integer32 cellIndex(1)
|     |        Range: 0..1
|     +-- -RW- EnumVal   cellAdmStatus(2)
|     |        Values: lock(0), unlock(1)
|     +-- -RW- EnumVal   cellOpStatus(3)
|     |        Values: disabled(0), enabled(1), invalid(2)
|     +-- -R-- EnumVal   duplexMode(4)
|     |        Values: fdd(0), tdd(1)
|     +-- -RW- Integer32 cellId(5)
|     |        Range: 0..268435455
|     +-- -RW- EnumVal   tddSubFrameAss(6)
|     |        Values: sa0(0), sa1(1), sa2(2), sa3(3), sa4(4), sa5(5), sa6(6)
|     +-- -RW- EnumVal   tddSpecSubFrameAss(7)
|     |        Values: ssp0(0), ssp3(3), ssp4(4), ssp5(5), ssp7(7), ssp8(8)
|     +-- -RW- Integer32 phyCellId(8)
|     |        Range: 0..503
|     +-- -R-- EnumVal   cellFuncStatus(9)
|     |        Values: notoperational(0), operational(1)
|     +-- -RW- EnumVal   cellRf(10)
|     |        Values: manual(0), ngscbp(1), bh(2), lamarr0(3), lamarr24(4),
|     |                lamarr33(5), lamarr(6)
|     +-- -RW- EnumVal   txBandwidth(11)
|     |        Values: band1dot4(14), band3(30), band5(50), band10(100),
|     |                band15(150), band20(200)
|     +-- -RW- Integer32 txEarfcn(12)
|     |        Range: 0..65535
|     +-- -R-- Integer32 rxEarfcn(13)
|     +-- -R-- Integer32 dlCenterFreq(14)
|     +-- -R-- Integer32 ulCenterFreq(15)
|     +-- -R-- Integer32 bandId(16)
|     |        Range: 1..44
|     +-- -RW- EnumVal   t300(17)
|     |        Values: 600(600),1000(1000),1500(1500),2000(2000)
|     +-- -RW- EnumVal t301(18)
|     |        Values: 100(100),200(200),300(300),400(400),600(600),
|     |                1000(1000),1500(1500),2000(2000)
|     +-- -RW- EnumVal t310(19)
|     |        Values: 0(0),50(50),100(100),200(200),500(500),1000(1000),
|     |                2000(2000)
|     +-- -RW- EnumVal t311(20)
|     |        Values: 1000(1000),3000(3000),5000(5000),10000(10000),
|     |                15000(15000),20000(20000),30000(30000)
|     +-- -RW- EnumVal   transMode(21)
|     |        Values: siso(0), diversity(1), tm3(2), tm4(3)
|     +-- -R-- Integer32 numUE(22)
|     |        Range: 0..255
|     +-- -RW- Integer32 maxTxPower(23)
|     |        Range: -10..50
|     +-- -RW- Integer32 azAifLinkRef(24)
|     |        Range: 0..1
|     +-- -RW- EnumVal   azRefreshStats(25)
|     |        Values: auto(0), manual(1)
|     +-- -R-- EnumVal   azCellMaxUEReached(26)
|              Values: clear(0), alarmed(1)
|
+--rf(23)
|  |
|  +--rfOnBoardMibModule(1)
|     |
|     +--rfOnBoard(1)
|        |
|        +-- -R-- EnumVal    rfOpStatus(2)
|        |        Values: disabled(0), notoperational(1), enabled(2)
|        |
|        +--txSigPathEutraTable(5)
|        |  |
|        |  +--txSigPathEutraEntry(1)
|        |     |  Index: txSigPathEutraIndex
|        |     |
|        |     +-- ---- Integer32 txSigPathEutraIndex(1)
|        |     |        Range: 0
|        |     +-- -R-- Integer32 txPower(8)
|        |
|        +--rxSigPathEutraTable(6)
|        |  |
|        |  +--rxSigPathEutraEntry(1)
|        |     |  Index: rxSigPathEutraIndex
|        |     |
|        |     +-- ---- Integer32 rxSigPathEutraIndex(1)
|        |     |        Range: 0
|        |     +-- -R-- Integer32 rxPower(2)
|        |
|        +--rfBoardAntPortTable(7)
|           |
|           +--rfBoardAntPortEntry(1)
|              |  Index: rfBoardAntPortIndex
|              |
|              +-- ---- Integer32 rfBoardAntPortIndex(1)
|              |        Range: 1..4
|              +-- -R-- Integer32 rfAntTemp(2)
|                       Range: 0..65535
|
+--eNodeBLog(28)
|  |
|  +-- -RW- IpAddr    logServerIpAddr(1)
|  +-- -RW- Integer32 logServerPort(2)
|  |        Range: 0..65535
|  +-- -RW- EnumVal   logPhySeverity(3)
|  |        Textual Convention: LogSeverity
|  |        Values: critical(0), error(1), warning(2), info(3), debug(4)
|  +-- -RW- EnumVal   logL2Severity(4)
|  |        Textual Convention: LogSeverity
|  |        Values: critical(0), error(1), warning(2), info(3), debug(4)
|  +-- -RW- EnumVal   logL3Severity(5)
|  |        Textual Convention: LogSeverity
|  |        Values: critical(0), error(1), warning(2), info(3), debug(4)
|  +-- -RW- String    logPhyMask(6)
|  |        Size: 8
|  +-- -RW- String    logL2Mask(7)
|  |        Size: 16
|  +-- -RW- String    logL3Mask(8)
|  |        Size: 16
|  +-- -RW- String    logOamSeverity(11)
|  |        Textual Convention: LogSeverity
|  |        Values: critical(0), error(1), warning(2), info(3), debug(4)
|  +-- -RW- IpAddr    logEventsSeverity(12)
|  |        Textual Convention: LogSeverity
|  |        Values: critical(0), error(1), warning(2), info(3), debug(4)
|  +-- -RW- String    logOamMask(13)
|           Size: 16
+--softwareMgmt(29)
|  |
|  +-- -RW- EnumVal   swmOpStatus(1)
|  |        Values: idle(0), transfer(1), abort(2), ready(3), install(4),
|  |                failed(5)
|  +-- -RW- IpAddr    swmServerIpAddr(2)
|  +-- -RW- String    swmServerUserName(3)
|  |        Size: 1..15
|  +-- -RW- EnumVal   swmTransferMode(4)
|  |        Values: tftp(0), scp(1)
|  +-- -RW- String    swmTransferFile(5)
|           Size: 1..200
|
+--s1Interface(30)
|  |
|  +-- -R-- IpAddr    s1apeNBIpAddress(1)
|  +-- -RW- Integer32 s1apeNBPort(2)
|  |        Range: 0..65535
|  +-- -RW- IpAddr    s1apMmeIpAddress(3)
|  +-- -RW- Integer32 s1apMmePort(4)
|  |        Range: 0..65535
|  +-- -R-- IpAddr    s1ueNBIpAddress(5)
|  +-- -RW- Integer32 s1ueNBPort(6)
|  |        Range: 0..65535
|  +-- -RW- IpAddr    s1uGwIpAddress(7)
|  +-- -RW- Integer32 s1uGwPort(8)
|  |        Range: 0..65535
|  +-- -RW- EnumVal   s1apIface(9)
|  |        Values: eth0(0), eth1(1), eth2(2), eth3(3)
|  +-- -RW- EnumVal   s1uIface(10)
|  |        Values: eth0(0), eth1(1), eth2(2), eth3(3)
|  +-- -RW- Integer32 s1apVlanId(11)
|  |         Range: 1..4094
|  +-- -RW- Integer32 s1uVlanId(12)
|  |         Range: 1..4094
|  +-- -R-- EnumVal   s1SctpStatus(13)
|            Values: operational(0), notoperational(1)
|
+--x2Interface(31)
|  |
|  +-- -RW- IpAddr    x2eNBIpAddress(1)
|  +-- -RW- Integer32 x2apeNBPort(2)
|  |         Range: 0..65535
|  +-- -RW- Integer32 x2ueNBPort(3)
|  |         Range: 0..65535
|  +-- -R-- EnumVal   x2SetupStatus(4)
|  |         Values: clear(0), failed(1)
|  +-- -R-- EnumVal   x2NeighUnknownStatus(5)
|            Values: clear(0), alarmed(1)
|
+--cpriTable(33)
|  |
|  +--cpriEntry(1)
|     |  Index: cpriIndex
|     |
|     +-------- Integer32 cpriIndex(1)
|     |         Range: 0..1
|     +-- -R-- EnumVal   cpriOpStatus(3)
|     |         Values: disabled(0), enabled(1), invalid(2)
|     +-- -RW- Integer32 cpriAtPimin(5)
|     |         Range: 0..4194303
|     +-- -RW- Integer32 cpriDBmOffset(6)
|     |         Range: -2147483648..2147483647
|     +-- -RW- Unsigned  cpriHFsyncDelay(7)
|     |         Range: 0..4294967295
|     +-- -RW- Integer32 cpriAxcOffset(8)
|     |         Range: 0..33554431
|     +-- -RW- EnumVal   tcpipOverCpri(9)
|               Values: disable(0), enable(1)
|
+--azGpsInfo(34)
|  |
|  +-- -R-- EnumVal   azGpsFixtype(1)
|  |         Values: undef(0), none(1), twoDimension(2), threeDimension(3)
|  +-- -R-- String    azGpsTimeStamp(2)
|  |         Textual Convention: SnmpAdminString
|  |         Size: 0..255
|  +-- -R-- String    azGpsLatitude(3)
|  |         Textual Convention: SnmpAdminString
|  |         Size: 0..255
|  +-- -R-- String    azGpsLongitude(4)
|  |         Textual Convention: SnmpAdminString
|  |         Size: 0..255
|  +-- -R-- Integer32 azGpsHeight(5)
|  +-- -R-- Integer32 azGpsHeading(6)
|  +-- -R-- String    azGpsVelocityHorizontal(7)
|  |         Textual Convention: SnmpAdminString
|  |         Size: 0..255
|  +-- -R-- String    azGpsVelocityVertical(8)
|  |         Textual Convention: SnmpAdminString
|  |        Size: 0..255
|  +-- -R-- Integer32 azGpsSatelliteNumber(9)
|  |
|  +--azGpsSatelliteInfoTable(10)
|     |
|     +--azGpsSatelliteInfoEntry(1)
|        |  Index: azGpsSatelliteInfoIndex
|        |
|        +-- ---- Integer32 azGpsSatelliteInfoIndex(1)
|        |         Range: 0..1000
|        +-- -R-- Integer32 azGpsSatelliteElevation(2)
|        +-- -R-- Integer32 azGpsSatelliteAzimuth(3)
|        +-- -R-- Integer32 azGpsSatelliteSignalStrength(4)
|        +-- -R-- EnumVal   azGpsSatelliteUsed(5)
|                 Values: false(0), true(1)
+--azNtp(35)
|  |
|  +-- -RW- IpAddr    azNtpServerPrimary(1)
|  +-- -RW- IpAddr    azNtpServerSecondary(2)
|
+--azHandover(40)
|  |
|  +--azHoX2NeighCellsTable(1)
|  |  |
|  |  +--azHoX2NeighCellsEntry(1)
|  |     |  Index: cellIndex, azHoX2NeighCellsIndex
|  |     |
|  |     +-- ---- Integer32 azHoX2NeighCellsIndex(1)
|  |     |         Range: 0..31
|  |     +-- CR-- EnumVal   azHoX2NeighStatus(2)
|  |     |         Textual Convention: RowStatus
|  |     |         Values: active(1), notInService(2), notReady(3),
|  |     |                 createAndGo(4), createAndWait(5), destroy(6)
|  |     +-- -RW- Integer32 azHoX2NeighPhyCellId(3)
|  |     |         Range: 0..503
|  |     +-- -RW- IpAddr    azHoX2NeighIpAddr(4)
|  |     +-- -RW- Integer32 azHoX2NeighX2apPort(5)
|  |     |         Range: 0..65535
|  |     +-- -RW- String    azHoX2NeighMacAddr(6)
|  |     |         Textual Convention: MacAddress
|  |     |         Size: 6
|  |     +-- -RW- Integer32 azHoX2NeighX2uPort(7)
|  |               Range: 0..65535
|  |
|  +--azHoS1NeighCellsTable(2)
|  |  |
|  |  +--azHoS1NeighCellsTableEntry(1)
|  |     |  Index: cellIndex, azHoS1NeighCellsIndex
|  |     |
|  |     +-- ---- Integer32 azHoS1NeighCellsIndex(1)
|  |     |         Range: 0..31
|  |     +-- CR-- EnumVal   azHoS1NeighStatus(2)
|  |     |         Textual Convention: RowStatus
|  |     |         Values: active(1), notInService(2), notReady(3),
|  |     |                 createAndGo(4), createAndWait(5), destroy(6)
|  |     +-- -RW- Integer32 azHoS1NeighPhyCellId(3)
|  |     |         Range: 0..503
|  |     +-- -RW- EnumVal   azHoS1NeighType(4)
|  |     |         Values: macro(0), home(1)
|  |     +-- -RW- String    azHoS1NeighPlmnMcc(5)
|  |     |        Size: 3
|  |     +-- -RW- String    azHoS1NeighPlmnMnc(6)
|  |     |        Size: 2..3
|  |     +-- -RW- Integer32 azHoS1NeighCellId(7)
|  |              Range: 0..268435455
|  |
|  +--azHoUtraCellsTable(3)
|  |  |
|  |  +--azHoUtraCellsTableEntry(1)
|  |     |  Index: cellIndex, azHoUtraCellsIndex
|  |     |
|  |     +-- ---- Integer32 azHoUtraCellsIndex(1)
|  |     |        Range: 0..31
|  |     +-- CR-- EnumVal   azHoUtraStatus(2)
|  |     |        Textual Convention: RowStatus
|  |     |        Values: active(1), notInService(2), notReady(3),
|  |     |                createAndGo(4), createAndWait(5), destroy(6)
|  |     +-- -RW- String    azHoUtraPlmnMcc(3)
|  |     |        Size: 3
|  |     +-- -RW- String    azHoUtraPlmnMnc(4)
|  |     |        Size: 2..3
|  |     +-- -RW- Integer32 azHoUtraCellId(5)
|  |     |        Range: 0..65535
|  |     +-- -RW- Integer32 azHoUtraRncId(6)
|  |     |        Range: 0..65535
|  |     +-- -RW- Integer32 azHoUtraLAC(7)
|  |     |        Range: 0..65535
|  |     +-- -RW- Integer32 azHoUtraRAC(8)
|  |     |        Range: 0..255
|  |     +-- -RW- Integer32 azHoUtraPsc(9)
|  |              Range: 0..511
|  |
|  +--azHoMeasurements(4)
|     |
|     +--azHoReportConfigTable(100)
|     |  |
|     |  +--azHoReportConfigTableEntry(1)
|     |     |  Index: cellIndex, azHoReportConfigsIndex
|     |     |
|     |     +-- ---- Integer32 azHoReportConfigsIndex(1)
|     |     |        Range: 0..31
|     |     +-- CR-- EnumVal   azHoReportConfigStatus(2)
|     |     |        Textual Convention: RowStatus
|     |     |        Values: active(1), notInService(2), notReady(3),
|     |     |                createAndGo(4), createAndWait(5), destroy(6)
|     |     +-- -RW- EnumVal   azHoReportConfigChoice(4)
|     |     |        Values: eutra(1), utra(2)
|     |     +-- -RW- EnumVal   azHoReportConfigTriggerType(5)
|     |     |        Values: event(1), periodic(2)
|     |     +-- -RW- EnumVal   azHoReportConfigEvent(6)
|     |     |        Values: ab1(1), ab2(2), a3(3), a4(4), a5(5)
|     |     +-- -RW- EnumVal   azHoReportConfigThresholdChoice(7)
|     |     |        Values: rsrp(1), rsrq(2)
|     |     +-- -RW- Integer32 azHoReportConfigThresholdRsrp(8)
|     |     |        Range: 0..97
|     |     +-- -RW- Integer32 azHoReportConfigThresholdRsrq(9)
|     |     |        Range: 0..34
|     |     +-- -RW- Integer32 azHoReportConfigA3Offset(10)
|     |     |        Range: -30..30
|     |     +-- -RW- EnumVal   azHoReportConfigOnLeave(11)
|     |     |        Values: false(0), true(1)
|     |     +-- -RW- EnumVal   azHoReportConfigThreshold2Choice(12)
|     |     |        Values: rsrp(1), rsrq(2)
|     |     +-- -RW- Integer32 azHoReportConfigThreshold2Rsrp(13)
|     |     |        Range: 0..97
|     |     +-- -RW- Integer32 azHoReportConfigThreshold2Rsrq(14)
|     |     |        Range: 0..34
|     |     +-- -RW- EnumVal   azHoReportThrshldUtraChoice(15)
|     |     |        Values: rscp(1), ecno(2)
|     |     +-- -RW- Integer32 azHoReportConfigThresholdRSCP(16)
|     |     |        Range: -5..91
|     |     +-- -RW- Integer32 azHoReportConfigThresholdEcNO(17)
|     |     |        Range: 0..49
|     |     +-- -RW- Integer32 azHoReportConfigHysteresis(18)
|     |     |        Range: 0..30
|     |     +-- -RW- EnumVal   azHoReportConfigTimeToTrigger(19)
|     |     |        Values: ms0(0), ms40(1), ms64(2), ms80(3), ms100(4),
|     |     |                ms128(5), ms160(6), ms256(7), ms320(8),ms480(9),
|     |     |                ms512(10), ms640(11), ms1024(12), ms1280(13),
|     |     |                ms2560(14), ms5120(15)
|     |     +-- -RW- EnumVal   azHoReportConfigTriggerQuantity(20)
|     |     |        Values: rsrp(0), rsrq(1)
|     |     +-- -RW- EnumVal   azHoReportConfigQuantity(21)
|     |     |        Values: trigger(0), both(1)
|     |     +-- -RW- Integer32 azHoReportConfigMaxCells(22)
|     |     |        Range: 1..8
|     |     +-- -RW- EnumVal   azHoReportConfigInterval(23)
|     |     |        Values: ms120(0), ms240(1), ms480(2), ms640(3),
|     |     |                ms1024(4), ms2048(5), ms5120(6), ms10240(7),
|     |     |                min1(8), min6(9), min12(10), min30(11),
|     |     |                min60(12)
|     |     +-- -RW- EnumVal   azHoReportConfigAmount(24)
|     |              Values: ra1(0), ra2(1), ra4(2), ra8(3), ra16(4),
|     |                      ra32(5), ra64(6), inf(7)
|     |
|     +--azHoMeasObjectTable(200)
|     |  |
|     |  +--azHoMeasObjectTableEntry(1)
|     |     |  Index: cellIndex, azHoMeasObjectIndex
|     |     |
|     |     +-- ---- Integer32 azHoMeasObjectIndex(1)
|     |     |        Range: 0..31
|     |     +-- CR-- EnumVal   azHoMeasObjectStatus(2)
|     |     |        Textual Convention: RowStatus
|     |     |        Values: active(1), notInService(2), notReady(3),
|     |     |                createAndGo(4), createAndWait(5), destroy(6)
|     |     +-- -RW- EnumVal   azHoMeasObjectChoice(4)
|     |     |        Values: eutra(1), utra(2)
|     |     +-- -RW- Integer32 azHoMeasCarrierFreq(5)
|     |     |        Range: 0..65535
|     |     +-- -RW- EnumVal   azHoMeasResourceBlock(6)
|     |     |        Values: mbw6(0), mbw15(1), mbw25(2), mbw50(3), mbw75(4),
|     |     |                mbw100(5)
|     |     +-- -RW- EnumVal   azHoEutraqOffset(7)
|     |     |        Values: dBMinus24(0), dBMinus22(1), dBMinus20(2),
|     |     |                dBMinus18(3), dBMinus16(4), dBMinus14(5),
|     |     |                dBMinus12(6), dBMinus10(7), dBMinus8(8),
|     |     |                dBMinus6(9), dBMinus5(10), dBMinus4(11),
|     |     |                dBMinus3(12), dBMinus2(13), dBMinus1(14),
|     |     |                dB0(15), dB1(16), dB2(17), dB3(18), dB4(19),
|     |     |                dB5(20), dB6(21), dB8(22), dB10(23), dB12(24),
|     |     |                dB14(25), dB16(26), dB18(27), dB20(28),
|     |     |                dB22(29), dB24(30)
|     |     +-- -RW- Integer32 azHoCellsForWhichToReportCGI(8)
|     |              Range: 0..503
|     |
|     +--azHoMeasIdTable(300)
|     |  |
|     |  +--azHoMeasIdTableEntry(1)
|     |     |  Index: cellIndex, azHoMeasIdIndex
|     |     |
|     |     +-- ---- Integer32 azHoMeasIdIndex(1)
|     |     |        Range: 0..31
|     |     +-- CR-- EnumVal   azHoMeasIdStatus(2)
|     |     |        Textual Convention: RowStatus
|     |     |        Values: active(1), notInService(2), notReady(3),
|     |     |                createAndGo(4), createAndWait(5), destroy(6)
|     |     +-- -RW- Integer32 azHoMeasObjectId(4)
|     |     |        Range: 0..31
|     |     +-- -RW- Integer32 azHoReportId(5)
|     |              Range: 0..31
|     |
|     +--azHoMeasConfigTable(400)
|        |
|        +--azHoMeasConfigTableEntry(1)
|           |  Index: cellIndex
|           |
|           +-- -RW- EnumVal   azHoEutraFilterCoefficientRsrp(2)
|           |        Values: fc0(0), fc1(1), fc2(2), fc3(3), fc4(4), fc5(5),
|           |                fc6(6), fc7(7), fc8(8), fc9(9), fc11(10),
|           |                fc13(11), fc15(12), fc17(13), fc19(14)
|           +-- -RW- EnumVal   azHoEutraFilterCoefficientRsrq(3)
|           |        Values: fc0(0), fc1(1), fc2(2), fc3(3), fc4(4), fc5(5),
|           |                fc6(6), fc7(7), fc8(8), fc9(9), fc11(10),
|           |                fc13(11), fc15(12), fc17(13), fc19(14)
|           +-- -RW- EnumVal   azHoUtraMeasQuantityFdd(4)
|           |        Values: cpichRSCP(0), cpichEcNO(1)
|           +-- -RW- EnumVal   azHoUtraMeasQuantityTdd(5)
|           |        Values: pccpchRSCP(0)
|           +-- -RW- EnumVal   azHoUtraFilterCoefficient(6)
|           |        Values: fc0(0), fc1(1), fc2(2), fc3(3), fc4(4), fc5(5),
|           |                fc6(6), fc7(7), fc8(8), fc9(9), fc11(10),
|           |                fc13(11), fc15(12), fc17(13), fc19(14)
|           +-- -RW- EnumVal   azHoGapConfigChoice(7)
|           |        Values: release(1), setup(2)
|           +-- -RW- EnumVal   azHoGapOffsetChoice(8)
|           |        Values: gp0(0), gp1(1)
|           +-- -RW- Integer32 azHoGapOffset(9)
|           |        Range: 0..79
|           +-- -RW- Integer32 azHoHysteresisP(10)
|           |        Range: 0..30
|           +-- -RW- Integer32 azHoHysteresisQ(11)
|           |        Range: 0..30
|           +-- -RW- Integer32 azHoThresholdRSCP(12)
|           |        Range: -5..91
|           +-- -RW- Integer32 azHoThresholdECNO(13)
|           |        Range: 0..49
|           +-- -RW- Integer32 azHoThresholdRSRP(14)
|           |        Range: 0..97
|           +-- -RW- Integer32 azHoThresholdRSRQ(15)
|                    Range: 0..34
|           +-- -RW- Integer32 azHoThresholdRSRPMeasGapEnable(16)
|           |        Range: 0..97
|           +-- -RW- Integer32 azHoThresholdRSRPMeasGapDisable(17)
|           |        Range: 0..97
|           +-- -RW- Integer32 azHoThresholdRSRQMeasGapEnable(18)
|           |        Range: 0..34
|           +-- -RW- Integer32 azHoThresholdRSRQMeasGapDisable(19)
|           |        Range: 0..34
|
+--azL2Measurements(50)
|  +-- -RW- EnumVal   azMeasActive(5)
|  |        Values: disable(0), enable(1)
|  |
|  +--azL2MeasSelections(10)
|  |  |
|  |  +--azL2MeasSelectionsTable(1)
|  |  |  |
|  |  |  +--azL2MeasSelectionsEntry(1)
|  |  |     |  Index: cellIndex
|  |  |     |
|  |  |     +-- -RW- EnumVal   azUlTotalPrbUsageActive(1)
|  |  |     |        Values: disable(0), enable(1)
|  |  |     +-- -RW- EnumVal   azDlTotalPrbUsageActive(2)
|  |  |     |        Values: disable(0), enable(1)
|  |  |     +-- -RW- EnumVal   azUlRecvRandAccessPreamActive(3)
|  |  |              Values: disable(0), enable(1)
|  |  |
|  |  +--azL2MeasUlQciSelectionsTable(2)
|  |  |  |
|  |  |  +--azL2MeasUlQciSelectionsEntry(1)
|  |  |     |  Index: cellIndex, azUlQciMeasIndex
|  |  |     |
|  |  |     +-- ---- Integer32 azUlQciMeasIndex(1)
|  |  |     |        Range: 1..9
|  |  |     +-- -RW- EnumVal   azUlPrbUsagePerQciActive(2)
|  |  |     |        Values: disable(0), enable(1)
|  |  |     +-- -RW- EnumVal   azUlPktLossRateActive(3)
|  |  |     |        Values: disable(0), enable(1)
|  |  |     +-- -RW- EnumVal   azUlNumActiveUeActive(4)
|  |  |     |        Values: disable(0), enable(1)
|  |  |     +-- -RW- EnumVal   azUlSchedIpTptActive(5)
|  |  |              Values: disable(0), enable(1)
|  |  |
|  |  +--azL2MeasDlQciSelectionsTable(3)
|  |     |
|  |     +--azL2MeasDlQciSelectionsEntry(1)
|  |        |  Index: cellIndex, azDlQciMeasIndex
|  |        |
|  |        +-- ---- Integer32 azDlQciMeasIndex(1)
|  |        |        Range: 1..9
|  |        +-- -RW- EnumVal   azDlPrbUsagePerQciActive(2)
|  |        |        Values: disable(0), enable(1)
|  |        +-- -RW- EnumVal   azDlAvgPktDelayActive(3)
|  |        |        Values: disable(0), enable(1)
|  |         +-- -RW- EnumVal   azDlPktUuLossRateActive(4)
|  |         |         Values: disable(0), enable(1)
|  |         +-- -RW- EnumVal   azDlPktDiscardRateActive(5)
|  |         |         Values: disable(0), enable(1)
|  |         +-- -RW- EnumVal   azDlNumActiveUeActive(6)
|  |         |         Values: disable(0), enable(1)
|  |         +-- -RW- EnumVal   azDlSchedIpTptActive(7)
|  |                   Values: disable(0), enable(1)
|  |
|  +--azL2MeasValues (20)
|     |
|     +--azL2MeasValuesTable(1)
|     |  |
|     |  +--azL2MeasValuesEntry(1)
|     |     |  Index: cellIndex
|     |     |
|     |     +-- -R-- Integer32 azUlTotalPrbUsage(1)
|     |     |         Range: 0..100
|     |     +-- -R-- Integer32 azDlTotalPrbUsage(2)
|     |     |         Range: 0..100
|     |     +-- -R-- Counter   azUlRecvRandAccessPreamTotal(3)
|     |     +-- -R-- Counter   azUlRecvRandAccessPreamDedicated(4)
|     |     +-- -R-- Counter   azUlRecvRandAccessPreamLow(5)
|     |     +-- -R-- Counter   azUlRecvRandAccessPreamHigh(6)
|     |
|     +--azL2MeasUlQciValuesTable(2)
|     |  |
|     |  +--azL2MeasUlQciValuesEntry(1)
|     |     |  Index: cellIndex, azUlQciMeasIndex
|     |     |
|     |     +-- -R-- Integer32 azUlPrbUsagePerQci(1)
|     |     |         Range: 0..100
|     |     +-- -R-- Integer32 azUlPktUuLossRate(2)
|     |     |         Range: 0..100
|     |     +-- -R-- Integer32 azUlNumActiveUe(3)
|     |     |         Range: 0..64
|     |     +-- -R-- Integer32 azUlSchedIpTpt(4)
|     |               Range: 0..204800
|     |
|     +--azL2MeasDlQciValuesTable(3)
|        |
|        +--azL2MeasDlQciValuesEntry(1)
|           |  Index: cellIndex, azDlQciMeasIndex
|           |
|           +-- -R-- Integer32 azDlPrbUsagePerQci(1)
|           |         Range: 0..100
|           +-- -R-- Integer32 azDlAvgPktDelay(2)
|           |         Range: 0..65535
|           +-- -R-- Integer32 azDlPktUuLossRate(3)
|           |         Range: 0..100
|           +-- -R-- Integer32 azDlPktDiscardRate(4)
|           |         Range: 0..100
|           +-- -R-- Integer32 azDlNumActiveUe(5)
|           |         Range: 0..64
|           +-- -R-- Integer32 azDlSchedIpTpt(6)
|                     Range: 0..204800
|
+--azStatistics(60)
   |
   +--azMACStatisticsTable(1)
   |   |
   |   +--azMACStatisticsEntry(1)
   |      |   Index: cellIndex, azMacUeIndex
   |      |
   |      +-- ---- Integer32 azMacUeIndex(1)
   |      |        Range: 10..65523
   |      +-- -R-- Integer32 azUlSnrPusch(2)
   |      |        Range: 0..255
   |      +-- -R-- Integer32 azUlSnrPucch(3)
   |      |        Range: 0..255
   |      +-- -R-- Integer32 azDlSnr(4)
   |      |        Range: 0..15
   |      +-- -R-- Integer32 azMacDlTput(5)
   |      +-- -R-- Integer32 azMacUlTput(6)
   |      +-- -R-- Integer32 azDlBlerOfTB1(7)
   |      |        Range: 0..100
   |      +-- -R-- Integer32 azDlBlerOfTB2(8)
   |      |        Range: 0..10
   |      +-- -R-- Integer32 azUlBler(9)
   |                Range: 0..100
   |
   +--azRLCStatisticsTable(2)
   |   |
   |   +--azRLCStatisticsEntry(1)
   |      |   Index: cellIndex, azRlcUeIndex
   |      |
   |      +-- ---- Integer32 azMacUeIndex(1)
   |      |        Range: 10..65523
   |      +-- -R-- Integer32 azRlcDlTput(2)
   |      +-- -R-- Integer32 azRlcUlTput(3)
   |      +-- -R-- Integer32 azRlcPktWholeNackRecv(4)
   |      +-- -R-- Integer32 azRlcPktSegNackRecv(5)
   |
   +--azPDCPStatisticsTable(3)
   |   |
   |   +--azPDCPStatisticsEntry(1)
   |      |   Index: cellIndex, azPdcpUeIndex
   |      |
   |      +-- ---- Integer32 azPdcpUeIndex(1)
   |      |        Range: 10..65523
   |      +-- -R-- Integer32 azDlTput(2)
   |      +-- -R-- Integer32 azUlTput(3)
   |      +-- -R-- Integer32 azUlPktLost(4)
   |      +-- -R-- Integer32 azDlPktLost(5)
   |
   +--azS1UStatisticsTable(4)
   |   |
   |   +--azS1UStatisticsEntry(1)
   |      |   Index: cellIndex, azS1uUeIndex
   |      |
   |      +-- ---- Integer32 azS1uUeIndex(1)
   |      |        Range: 10..65523
   |      +-- -R-- Integer32 azSgwUlTput(2)
   |      +-- -R-- Integer32 azSgwDlTput(3)
   |
   +--azRRCStatisticsTable(5)
   |   |
   |   +--azRRCStatisticsEntry(1)
   |     |  Index: cellIndex
   |     |
   |     +-- -R-- Integer32 azRrcConnRequestReceived(1)
   |     +-- -R-- Integer32 azRrcConnSetupCompleteNtReceived(2)
   |     +-- -R-- Integer32 azRrcConnReconfigCompleteNtRcved(3)
   |     +-- -R-- Integer32 azRrcConnReestablishmentReceived(4)
   |     +-- -R-- Integer32 azRlfReportedLayer2(5)
   |     +-- -R-- Integer32 azUEsConfigured(6)
   |     +-- -R-- Integer32 azUEsReleased(7)
   |
   +--azGlobalStatisticsTable(6)
   |  |
   |  +--azGlobalStatisticsEntry(1)
   |     |  Index: cellIndex
   |     |
   |     +-- -R-- Integer32 azNumRach(1)
   |     +-- -R-- Integer32 azNumMsg2(2)
   |     +-- -R-- Integer32 azNumMsg3(3)
   |     +-- -R-- Integer32 azNumPdcpInactivityTimer(4)
   |     +-- -R-- Integer32 azNumRLFatMac(5)
   |     +-- -R-- Integer32 azNumOfReleaseTriggeredatMac(6)
   |     +-- -R-- Integer32 azUlSnrPucchUeRelease(7)
   |     +-- -R-- Integer32 azUlSnrPuschUeRelease(8)
   |     +-- -R-- Integer32 azDlCrcUeRelease(9)
   |     +-- -R-- Integer32 azRlcNumRlf(10)
   |
   +--azRRCRntiStatisticsTable(7)
   |  |
   |  +--azRRCRntiStatisticsEntry(1)
   |     |  Index: cellIndex, azRntiUeIndex
   |     |
   |     +-- ---- Integer32 azRntiUeIndex(1)
   |     |        Range: 0..64
   |     +-- -R-- Integer32 azRnti(2)
   |
   |
   +--azPHYStatisticsTable(8)
   |  |
   |  +--azPHYStatisticsEntry(1)
   |     |  Index: cellIndex, azPhyUeIndex
   |     |
   |     +-- ---- Integer32 azPhyUeIndex(1)
   |     |        Range: 0..65535
   |     +-- -R-- Integer32 azUlCqiAnt0(2)
   |     |        Range: -64..63
   |     +-- -R-- Integer32 azUlCqiAnt1(3)
   |     |        Range: -64..63
   |     +-- -R-- Integer32 azNoiseVarAnt0(4)
   |     |        Range: -128..127
   |     +-- -R-- Integer32 azNoiseVarAnt1(5)
   |     |        Range: -128..127
   |     +-- -R-- Integer32 azRssiAnt0(6)
   |     |        Range: -128..127
   |     +-- -R-- Integer32 azRssiAnt1(7)
   |              Range: -128..127
   |
   +--azDspMemoryStatisticsTable(9)
   |  |
   |  +--azDspMemoryStatisticsTableEntry(1)
   |     |  Index: azDspMemIndex, azDspMemHeapIndex
   |     |
   |     +-- ---- Unsigned  azDspMemIndex(1)
   |     |        Range: 0..1
   |     +-- ---- Unsigned  azDspMemHeapIndex(2)
   |     |        Range: 0..14
   |     +-- -R-- Unsigned  azDspMemHeapIdentity(3)
   |     +-- -R-- Unsigned  azDspMemTotalSize(4)
   |     +-- -R-- Unsigned  azDspMemTotalFreeSize(5)
   |     +-- -R-- Unsigned  azDspMemLargestFreeSize(6)
   |     +-- -R-- Unsigned  azDspMemMaxAllocatedBlocks(7)
   |     +-- -R-- Unsigned  azDspMemNumAllocatedBlocks(8)
   |
   +--azDspPacketStatisticsTable(10)
   |  |
   |  +--azDspPacketStatisticsTableEntry(1)
   |     |  Index: azDspPktIndex, azDspPktHeapIndex
   |     |
   |     +-- ---- Unsigned  azDspPktIndex(1)
   |     |        Range: 0..1
   |     +-- ---- Unsigned  azDspPktHeapIndex(2)
   |     |        Range: 0..14
   |     +-- -R-- String    azDspPktHeapName(3)
   |     |        Size: 35
   |     +-- -R-- Unsigned  azDspPktNumFreeDataPackets(4)
   |     +-- -R-- Unsigned  azDspPktNumZeroBufferPackets(5)
   |     +-- -R-- Unsigned  azDspPktNumPacketsInGarbage(6)
   |     +-- -R-- EnumVal   azDspPktDataBufThrStatus(7)
   |     |        Values: clear(0), starving(1)
   |     +-- -R-- Unsigned  azDspPktDataBufStarvationCnt(8)
   |     |        Range: 0..255
   |     +-- -R-- EnumVal   azDspPktZeroDataBufThrStatus(9)
   |     |        Values: clear(0), starving(1)
   |     +-- -R-- Unsigned  azDspPktZeroDataBufStarvationCnt(10)
   |              Range: 0..255
   |
   +--azMACStatisticsAggregated(100)
   |  |
   |  +-- -R-- Unsigned  azMacDlTputTotal(4)
   |  +-- -R-- Unsigned  azMacUlTputTotal(5)
   |  +-- -R-- Unsigned  azDlBlerOfTB1Total(6)
   |  |        Range: 0..100
   |  +-- -R-- Unsigned  azDlBlerOfTB2Total(7)
   |  |        Range: 0..100
   |  +-- -R-- Unsigned  azUlBlerTotal(8)
   |           Range: 0..100
   |
   +--azRLCStatisticsAggregated(101)
   |  |
   |  +-- -R-- Unsigned  azRlcDlTputTotal(1)
   |  +-- -R-- Unsigned  azRlcUlTputTotal(2)
   |  +-- -R-- Unsigned  azRlcPktWholeNackRecvTotal(3)
   |  +-- -R-- Unsigned  azRlcPktSegNackRecvTotal(4)
   |
   +--azPDCPStatisticsAggregated(102)
   |  |
   |  +-- -R-- Unsigned  azDlTputTotal(1)
   |  +-- -R-- Unsigned  azUlTputTotal(2)
   |  +-- -R-- Unsigned  azUlPktLostTotal(3)
```

```
      |  +-- -R-- Unsigned  azDlPktLostTotal(4)
      |
      +--azS1UStatisticsAggregated(103)
         |
         +-- -R-- Unsigned  azSgwUlTputTotal(1)
         +-- -R-- Unsigned  azSgwDlTputTotal(2)
```

The same tree can be regenerated from actualized MIBs with the snmp command:

```
snmptranslate -IR -Tp eNodeB
```

Some parameters related to the same entity are grouped in a table. Every instance of the table entry is managed through the table index which is a read-only parameter. Some tables are managed through more than one index to identify an entry so to get/set a parameter for that entry a values for every index must be specified.

The “rf” branch of the tree is the one where the RF/RRH parameters boards sub-tree is linked. This can be done due to the definition of a MIB for every module. The SNMP tool load every desired MIB and link together to create a single MIB tree.

3.1.1 Additional standard MIBs

Standard MIBs that are not supported off-the-shelf by snmpd daemon on platform follow.

3.1.1.1 ALARM-MIB

ALARM-MIB as depicted by RFC 3877 is shown below. The ALARM-MIB consists of alarm models and lists of active alarms and other optional facilities for configuring notifications, providing logs, and more.

In this release the following restrictions apply:

- Alarm models are readonly: every action on the RowStatus parameter will be rejected.
- Only alarm models, statistics and the list of active alarms are supported. The table for cleared alarms is omitted from the MIB.
- No optional information is supported.

```
+--alarmMIB(118)
   |
   +--alarmObjects(1)
      |
      +--alarmModel(1)
      |  |
      |  +-- -R-- TimeTicks alarmModelLastChanged(1)
      |  |
      |  +--alarmModelTable(2)
      |     |
      |     +--alarmModelEntry(1)
      |        |  Index: alarmListName, alarmModelIndex, alarmModelState
      |        |
      |        +-- ---- Unsigned  alarmModelIndex(1)
      |        |        Range: 1..4294967295
      |        +-- ---- Unsigned  alarmModelState(2)
      |        |        Range: 1..4294967295
      |        +-- R-- ObjID     alarmModelNotificationId(3)
      |        +-- R-- Unsigned  alarmModelVarbindIndex(4)
      |        +-- R-- Integer32 alarmModelVarbindValue(5)
      |        +-- R-- String    alarmModelDescription(6)
      |        |        Textual Convention: SnmpAdminString
      |        |        Size: 0..255
      |        +-- R-- ObjID     alarmModelSpecificPointer(7)
      |        |        Textual Convention: RowPointer
      |        +-- R-- ObjID     alarmModelVarbindSubtree(8)
      |        +-- R-- ObjID     alarmModelResourcePrefix(9)
      |        +-- CR-- EnumVal   alarmModelRowStatus(10)
      |                 Textual Convention: RowStatus
      |                 Values: active(1), notInService(2), notReady(3),
      |                         createAndGo(4), createAndWait(5), destroy(6)
      |
      +--alarmActive(2)
         |
         +-- -R-- TimeTicks alarmActiveLastChanged(1)
         |
         +--alarmActiveTable(2)
         |  |
         |  +--alarmActiveEntry(1)
         |     |  Index: alarmListName, alarmActiveDateAndTime, alarmActiveIndex
         |     |
         |     +-- ---- String    alarmListName(1)
         |     |        Textual Convention: SnmpAdminString
         |     |        Size: 0..32
         |     +-- ---- String    alarmActiveDateAndTime(2)
         |     |        Textual Convention: DateAndTime
         |     |        Size: 8 | 11
         |     +-- ---- Unsigned  alarmActiveIndex(3)
         |     |        Range: 1..4294967295
         |     +-- -R-- String    alarmActiveEngineID(4)
         |     |        Textual Convention: LocalSnmpEngineOrZeroLenStr
         |     |        Size: 0 | 5..32
         |     +-- -R-- EnumVal   alarmActiveEngineAddressType(5)
         |     |        Textual Convention: InetAddressType
         |     |        Values: unknown(0), ipv4(1), ipv6(2), ipv4z(3),
         |     |                ipv6z(4),dns(16)
         |     +-- -R-- String    alarmActiveEngineAddress(6)
         |     |        Textual Convention: InetAddress
         |     |        Size: 0..255
         |     +-- -R-- String    alarmActiveContextName(7)
         |     |        Textual Convention: SnmpAdminString
         |     |        Size: 0..32
         |     +-- -R-- Unsigned  alarmActiveVariables(8)
         |     +-- -R-- ObjID     alarmActiveNotificationID(9)
         |     +-- -R-- ObjID     alarmActiveResourceId(10)
         |     |        Textual Convention: ResourceId
         |     +-- -R-- String    alarmActiveDescription(11)
         |     |        Textual Convention: SnmpAdminString
         |     |        Size: 0..255
         |     +-- -R-- ObjID     alarmActiveLogPointer(12)
         |     |        Textual Convention: RowPointer
         |     +-- -R-- ObjID     alarmActiveModelPointer(13)
         |     |        Textual Convention: RowPointer
         |     +-- -R-- ObjID     alarmActiveSpecificPointer(14)
         |              Textual Convention: RowPointer
         +--alarmActiveStatsTable(4)
            |
            +--alarmActiveStatsEntry(1)
               |  Index: alarmListName
               |
               +-- -R-- Gauge     alarmActiveStatsActiveCurrent(1)
               +-- -R-- Gauge     alarmActiveStatsActives(2)
               |        Textual Convention: ZeroBasedCounter32
               +-- -R-- TimeTicks alarmActiveStatsLastRaise(3)
               +-- -R-- TimeTicks alarmActiveStatsLastClear(4)
```

The same tree can be regenerated from actualized MIBs with the snmp command:

```
snmptranslate -IR -Tp alarmMIB
```

3.2 Administrative Status and System Status Coherence

The OAM HMI implements an advanced administrative (lock/unlock) status.

In order to write certain parameters, eNodeB, a cell, or other administered sub-objects must be set as locked.

Note that, unlike the common standard-defined locking behavior which requires an object to be disabled, the Azcom eNodeB is able to receive several configurations while active that, when not enforceable at runtime, will be enforced at unlock (some parameters have effect on the fly instead).

Unlocking the node or a cell will automatically enforce the needed board restart/soft restart for the system to apply the modified parameters (if any is necessary). The action that will be enforced can be monitored through the read-only eNodeBalarm.0 parameter. The most relevant action will be enforced; which means that a node Restart will take precedence over the need for a Soft Restart.

Several parameters require that the containing object is locked in order to be changed. Some exceptions to this rule is the setting of the node operational status (start, restart, halt), which will not need a previous lock but will indeed be rejected if the node is locked by another user.

In case of stop or soft restart failure, eNodeBalarm.0 (SNMP) or 'show node alarm' (CLI) will show an error value. Whenever a parameter requiring a reboot is changed (it must happen while the node or cell is locked), the alarm status will automatically become willReboot, and a configuration save and system reboot will be enforced on unlock as described above, unless of course the reboot is triggered explicitly.

The alarm condition (e.g. willReboot) will be enforced also when the configuration is saved through azConfigSave.0 (SNMP) or 'node configuration apply' (CLI). The system thus automatically applies the due operations to make the saved configuration become actual.

Locking or unlocking the node will also lock or unlock the children objects, such as cells, in cascade. Locking a parent object will fail if another user is locking a child object. Unlocking of a child object will fail if an ancestor object is locked. Different users can simultaneously lock objects that are not hierarchically related.

4 SNMP Interface

The SNMP protocol includes commands to get and set values of the eNodeB configuration parameters, defined in an object model called MIB. SNMP is based on the manager/agent model consisting of a manager, an agent, and a database of management information, managed objects and the network protocol. The manager provides the interface between the human network manager and the management system. The agent provides the interface between the manager and the eNodeB being managed.

4.1 SNMP commands

Users can adopt any SNMP v3 compliant client or use CLI commands on the node itself. An open source, yet complete and portable graphic SNMP client tool can be found at https://www.manageengine.com/products/mibbrowser-free-tool/.

Assuming a remote net-snmp CLI client, the commands used to get/set parameters values read/written from the eNodeB DB will be:

1. SNMPGET: used to get the value of a parameter
2. SNMPSET: used to set the value of a parameter
3. SNMPWALK: used to get the values of many parameters together
4. SNMPTABLE: used to get well formatted values from a table

Every SNMP command will be followed by some command option like the version protocol, the user name, the authentication protocol, and the credentials:

```
snmpget <options> hostToContact parameterName

snmpset <options> hostToContact parameterName parameterType
valueToSet

snmpwalk <options> hostToContact parameterGroupName

snmptable <options> hostToContact tableName
```

An example of the options follow:

```
snmpget -v 3 –t 15 -u azcomsha -a SHA -A "mypassword" -x DES -
X “mypassword” -l authNoPriv Server_ENodeB eNodeBopStatus.0
```

where:

-v 3: is the version 3 of the SNMP protocol

-t 15: defines the timeout for the command to be acknowledged, in seconds

-u azcomsha: is the name of the user

-a SHA: is the authentication algorithm

-A "mypassword": is the password for the user authentication

-x DES: is the encryption algorithm

-X “mypassword”: is the passphrase for encryption

-l authNoPriv: is the privacy level

Server_EnodeB: is the host name or IP address of the server to contact (the eNodeB board)

eNodeBopStatus.0: is the name of the parameter

Note: the client-side command timeout must be configured in order to leave enough time for the command to be completed at server side, also in cases when a single command reconfigures several aspects of the system or when the system is overloaded. A timeout of 5-6 seconds should be enough in most cases, yet using a timeout of 15 seconds is suggested.

In order to obtain an usage help about a parameter, the *snmptranslate* command is useful again:

```
root@AZB_NCOM_4300:~# snmptranslate -IR -Td eNodeBadmStatus.0
AZCOM-LTE-OAM-BBU-MIB::eNodeBadmStatus.0
eNodeBadmStatus OBJECT-TYPE
  -- FROM  AZCOM-LTE-OAM-BBU-MIB
  SYNTAX   INTEGER {lock(0), unlock(1)}
  MAX-ACCESS     read-write
  STATUS   current
  DESCRIPTION    "(LOCK, UNLOCK)
     on UNLOCK execute pending full reboot or
     sw reset according to alarm."
  DEFVAL   { lock }
::= { iso(1) org(3) dod(6) internet(1) private(4) enterprises(1)
azcom(48112) azcomLteOamBbuMibModule(1) eNodeB(1) eNodeBadmStatus(1)
0 }
```

4.2 SNMP users configuration

SNMP users credentials (both username and password) are always kept in synch with those of unix login users. Refer to [6] AZS-XXX-0200 Platform Command Line Interface for configuring the users.

4.3 SNMP errors

During the SNMP usage some generic or SNMP specific errors can be encountered.

SNMP error type	Meaning
Timeout	A get/set command can’t be completed due to missed response from the remote host
Host unreachable	The specified SNMP host is unreachable or SNMP is not running
Authentication failure	The specified password/community or key are incorrect or the specified authentication algorithm is wrong
Unknown user name	The specified SNMP user does not exist on the node
Decryption error	The specified encryption algorithm is not supported
Unknown Object Identifier	The specified parameter does not exists

No Such Instance currently exists at this OID	The instance of the specified parameter does not exists
Needs type and value	A set command for a parameter is sent without type and value specifications
Bad variable type	The specified variable type in the set command does not match the parameter definition type
wrongValue (The set value is illegal or unsupported in some way)	A value specified in the set command is not included in the admitted parameter values range
Commit Failed	A set command is applied to a parameter that require a locked node or cell status

The list of SNMP specific errors types returned by the SNMP commands follows:

SNMP error type	Meaning
SNMP_ERR_GENERR	Generic error. Communication problems with internal subsystems are also reported through this error type.
SNMP_ERR_WRONGVALUE	A not admitted value for a parameter is given in the SNMP set command
SNMP_ERR_NOTWRITABLE	A SNMP SET command is applied to a read-only parameter
SNMP_ERR_COMMITFAILED	A SNMP SET command is applied on a parameter that requires the node/cell lock when the is not set
SNMP_ERR_BADVALUE	A SNMP GET is applied on a parameter with no corresponding instance
SNMP_ERR_NOACCESS	A SNMP GET applied to a non- accessible parameter instance
SNMP_ERR_RESOURCEUNAVAILABLE	A SNMP action is applied on an unavailable resource
SNMP_NOSUCHOBJECT	A SNMP action is applied on a parameter that does not exists in the MIB model

4.4 Commands reference

This section describes how to configure eNodeB system parameters via SNMP. These steps allow the user to “prepare” new configuration values.

Depending on the option type, additional action may be required to activate the updated configuration. There are three possible cases:

a) Setting applied immediately.

b) eNodeB soft restart required at unlock. This operation takes about 1500 milliseconds.

c) eNodeB restart required at unlock. This operation takes about 25 seconds.

We would need to configure following environment variables which shall be used throughout the document.

- $USER
- $HOST

- $PWD1
- $PWD2

If a parameter is an enumerative, it can be set through either the symbolic enumeration string or the corresponding numeric value. For example,

```
snmpset  -v  3 -t 15 -u  $USER -a  SHA  -A $PWD1 -x DES -X $PWD2 -l
authPriv $HOST eNodeBadmStatus.0  =  0
```

is equivalent to

```
snmpset -v 3 -t 15 -u $USER -a SHA -A $PWD1 -x DES -X $PWD2 -l
authPriv $HOST eNodeBadmStatus.0  =  lock
```

The output for a successful command is like the following:

```
AZCOM-LTE-OAM-BBU-MIB:: eNodeBadmStatus.0 = INTEGER: lock(0)
```

Parameters can be read individually:

```
snmpget   -v  3 -t 15  -u   $USER   -a   SHA -A $PWD1 -x DES -X
$PWD2 -l authPriv  $HOST duplexMode.0
```

Tabled parameters can be read collectively:

```
snmptable -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST cellsTable
```

A whole OID hierarchy can be walked through:

```
snmpwalk -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST eNodeB
```

4.4.1 Cell operational status

This command enables or disables a cell. A disabled cell will turn its RF power to OFF.

This command is executed immediately.

The ON/OFF transmission status is stored persistently. When the eNodeB restarts, the RF is turned ON/OFF depending on the stored RF status.

Note that the system will not try to align the CPRI connection to the RRH after startup until the *cellOpStatus.0* is enabled. This applies to configurations with remote radio elements only. Refer to *Delayed synch with RRH* procedure in [1] .

The SNMP HMI requires that the cell is locked by the user and associated RF board not disabled (i.e. it must be the one selected via *cellRf.0*).

RF behavior is defined for Azcom's RF and RRH boards; support for other RRH boards may have a slightly different operational behavior, yet the HMI will always try to disable any traffic on the indexed cell when disabled.

Note: some 3rd party RRH management interfaces foresee that power is switched off when the cell is locked. In this case the cell operational status becomes read-only and completely in synch with the cell administrative status, as soon as the RF is selected through *cellRf.0*. Consequently, also the delayed synch with RRH procedure will be driven by the administrative status which will become persistent after a reboot.
The same behavior will be progressively emulated for all the radio elements in future, including Azcom equipment’s.

1) enable cell 0

```
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST cellOpStatus.0 =     enabled
```

2) disable cell 0

```
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST cellOpStatus.0 =     disabled
```

4.4.2 Node administrative status

This command locks and unlocks the node.

The lock command will fail if a subordinate object (e.g. a cell) is already locked by another user.

1) lock the node

```
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST eNodeBadmStatus.0 = lock
```

2) unlock the node

```
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST eNodeBadmStatus.0 = unlock
```

4.4.3 Node software version

Two read-only parameters report the currently running application software version as readable string (*azNodeVersion.0*) and as an unique md5sum (*azNodeReleaseID.0*).

The md5sum version is the one relevant to Azcom to identify the exact build that the customer is asking support for.

1) software version as a readable string

```
snmpget   -v  3  -t 15 -u   $USER   -a   SHA -A $PWD1 -x DES -X
$PWD2 -l authPriv  $HOST  azNodeVersion.0
```

2) software version as a md5sum string

```
snmpget   -v  3  -t 15 -u   $USER   -a   SHA -A $PWD1 -x DES -X
$PWD2 -l authPriv  $HOST   azNodeReleaseID.0
```

4.4.4 Cell administrative status

This command locks and unlocks a cell.

The unlock command will fail if an ancestor object (e.g. the node) is still locked.

1) lock the cell 0

```
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST cellAdmStatus.0 = lock
```

2) unlock the cell 0

```
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST cellAdmStatus.0 = unlock
```

4.4.5 Cell functional status

This read-only parameter reports the runtime cell status.

While the operational status reports the *commanded* status of the cell, the functional status reports the actual runtime condition which may be different because of contingent external causes. For example, RF power is turned OFF if the SCTP link is detected down, irrespective of the operator's command, and will be recovered as soon the degrading condition is relaxed.

Possible values are *notoperational* and *operational*.

```
snmpget   -v  3  -t 15 -u   $USER   -a   SHA -A $PWD1 -x DES -X
$PWD2 -l authPriv  $HOST  cellFuncStatus.0
```

4.4.6 RF Parameters & Antenna Configuration

These commands are common for every RF/RRH. However, they may behave slightly differently according to the specific RF/RRH in use. For example the Azcom ngRF RRH will accept values from 0 to 25 dBm, while Azcom Lamarr solutions may accept values depending to the RF module in use, spanning from -10 to 33 dBm.

The SNMP HMI requires that the cell is locked by the user for these commands to work, except for the TX power.

To apply the updated configuration, “eNodeB Restart” is enforced by the HMI on cell unlock. The node alarm status *eNodeBalarm.0* reports the action to be performed while the cell is still locked, in this case *willReboot*.

4.4.6.1 Max TX power

Set the max TX power of cell 0 to "10": will apply to selected RF/RRH

```
snmpset -v 3 –t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST maxTxPower.0 =  10
```

The max TX power parameter is applied to the configuration of the specific radio equipment selected so different min and max range values are admitted. Please refer to the vendor’s manuals.

The SNMP HMI *may* require that the cell is locked depending on the radio element.

Note that this is a parameter specific to the radio equipment referenced by the addressed cell (it isn’t a node, nor a cell parameter), so setting it when *cellRf.0* is in *manual* mode will not have an effect on the output power of your equipment. However, setting the max tx power is mandatory in *manual* mode too, so that the protocol stack can properly adjust dependent parameters, such as the *Reference Signal Power*.

4.4.6.2 Earfcn, center frequency, band Id

Setting the Earfcn of cell 0 via *txEarfcn.0* will:

- change the band (*bandId.0*) accordingly, which is read-only
- change the duplexMode.0 (read-only)
- change the *rxEarfcn.0* (read-only) to the proper shift, according to the duplex mode involved
- change the same representations in frequencies (*dlCenterFreq.0* and *ulCenterFreq.0*), read-only

The cell must be locked, otherwise *commitFailed* will be raised. An “eNodeB Restart” is needed to apply the change and will be automatically performed at unlock. The node alarm status *eNodeBalarm.0* reports the action to be performed while the cell is still locked, in this case *willReboot*.

All the above is applied to the current duplex mode configuration. The center frequency plus the bandwidth (*txBandwidth.0*) shall be contained within the band itself, otherwise *inconsistent value* will be returned.

In other terms, the Downlink Center Frequency shall be configured within the following interval:

```
[min_DL_freq + ((bandwidth/2)*1000, max_DL_freq - ((bandwidth/2)*1000))] Khz
```

where min_DL_freq and max_DL_freq are respectively the minimum and the maximum frequency boundaries of the band.

If an UL Earfcn is given, *wrongValue* is returned.

Examples:

```
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST txEarfcn.0 =  900
AZCOM-LTE-OAM-BBU-MIB::txEarfcn.0 = INTEGER: 900

snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST txEarfcn.0 =  600
WARNING: frequency for current band id 2 and bandwidth 20 MHz should
be within 1940000 and 1980000 KHz (1930000 KHz given)
Error in packet.
Reason: inconsistentValue (The set value is illegal or unsupported
in some way)
Failed object: AZCOM-LTE-OAM-BBU-MIB::txEarfcn.0

snmpget   -v  3  -t 15 -u   $USER   -a   SHA -A $PWD1 -x DES -X
$PWD2 -l authPriv  $HOST  rxEarfcn.0
AZCOM-LTE-OAM-BBU-MIB::rxEarfcn.0 = INTEGER: 18900

snmpget   -v  3  -t 15 -u   $USER   -a   SHA -A $PWD1 -x DES -X
$PWD2 -l authPriv  $HOST  dlCenterFreq.0
AZCOM-LTE-OAM-BBU-MIB::dlCenterFreq.0 = INTEGER: 1960000 KHz

snmpget   -v  3  -t 15 -u   $USER   -a   SHA -A $PWD1 -x DES -X
$PWD2 -l authPriv  $HOST  ulCenterFreq.0
AZCOM-LTE-OAM-BBU-MIB::ulCenterFreq.0 = INTEGER: 1880000 KHz

snmpget   -v  3  -t 15 -u   $USER   -a   SHA -A $PWD1 -x DES -X
$PWD2 -l authPriv  $HOST  bandId.0
AZCOM-LTE-OAM-BBU-MIB::bandId.0 = INTEGER: 2
```

4.4.6.3 Bandwidth

The standard defined bandwidths values, for both FDD and TDD modes, are:

- 1,4 MHz (band1dot4 = 14)
- 3 MHz (band3 = 30)
- 5 MHz (band5 = 50)

- 10 MHz (band10 = 100)
- 15 MHz (band15 = 150)
- 20 MHz (band20 = 200)

Units are in MHz/10. Nevertheless, each RF/RRH will support a subset of these, so the bandwidth setting can be rejected with *inconsistentValue* if the connected RF/RRH board does not support that bandwidth value.

The cell must be locked, otherwise *commitFailed* will be raised. An “eNodeB Restart” is needed to apply this change and to adapt the AxC Offset consequently, and will be automatically performed at unlock. The node alarm status *eNodeBalarm.0* reports the action to be performed while the cell is still locked, in this case *willReboot*.

The *rxBandwidth.0* is read-only and is set accordingly.

Set the txBandwidth to "5 MHz":

```
snmpset -v 3 –t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST txBandwidth.0 = band5
```

4.4.7 Air interface link

The *azAifLinkRef.0* parameter defines the air interface link (0 or 1) to which the RF/RRH referenced by the cell is connected.

On a RRH setup, this corresponds to the physical optical SFP link. On a setup with on-board RF, this corresponds to the RF’s physical slot.

```
snmpset -v 3 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azAifLinkRef.0 = 0
```

4.4.8 TDD configurations

These commands are used to update the system TDD configuration, in particular the TDD UL/DL Configuration and TDD Special Sub-frame Configuration.

The supported TDD configurations are:

- TDD UL/DL Configuration: 0,1,2,3,4,5,6
- TDD Special Sub-frame Configuration: 3,4,7,8

The SNMP HMI requires that the cell is locked by the user, and that *txEarfcn.0* to be set which lies in TDD mode, as in FDD mode TDD parameters cannot be changed.

To apply the updated configuration a “eNodeB Restart” will be automatically performed at unlock. The node alarm status *eNodeBalarm.0* reports the action to be performed while the cell is still locked, in this case *willReboot*.

Set the *tddSubFrameAss.0* to "sa2":

```
snmpset -v 3 –t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST tddSubFrameAss.0 = sa2
```

Set the tddSpecSubFrameAss.0 to "ssp4":

```
snmpset -v 3 –t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST tddSpecSubFrameAss.0 = ssp4
```

4.4.9 SIB2 timers

This command is used to set SIB2 timers: t300, t301, t310 and t311 (used by the UE).

The SNMP HMI requires that the cell is locked by the user, and the configuration will apply for the current duplex mode.

To apply the updated configuration, “eNodeB Restart” is usually enforced by the HMI on cell unlock. The node alarm status *eNodeBalarm.0* reports the action to be performed while the cell is still locked, in this case *willReboot*.

Timer descriptions and valid values are provided in the following Table:

Timer	Description	Valid values[ms]
t300	UE waiting time for RRC connection request message response	600 1000 1500 2000
t301	UE waiting time before going to idle mode once RRC connection re-establishment request message is sent	100 200 300 400 600 1000 1500 2000
t310	Started after receiving a defined number of out-of-sync indicators	0 50 100 200 500 1000 2000
t311	UE waiting time before going to idle mode if connection re-establishment procedure has started and it is unable to locate a suitable cell indicators	1000 3000 5000 10000 15000 20000 30000

Table 2.1: Timers description and valid values

1) set the t300 to "600":

```
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST t300.0 = 600
```

2) set the t301 to "200":

```
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST t301.0 = 200
```

3) set the t310 to "50":

```
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST t310.0 = 50
```

4) set the t311 to "20000":

```
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST t311.0 = 20000
```

4.4.10 Transmission Mode

This command is used to configure single antenna (SISO) or dual antenna transmission mode; in the latter case it’s possible to enable the TM3 capability or to force TX diversity:

0. for SISO mode
1. for TX Diversity mode
2. for TM3 mode
3. for TM4 mode

Please note that once enabled, TM3 requires proper conditions to be active and it’s not necessary always active.

The SNMP HMI requires that the cell is locked by the user, and the configuration will apply for the current duplex mode.

To apply the updated configuration, “eNodeB Restart” is usually enforced by the HMI on cell unlock. The node alarm status *eNodeBalarm.0* reports the action to be performed while the cell is still locked, in this case *willReboot*.

Set the transMode to "siso":

```
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST transMode.0 = siso
```

Set the transMode to "diversity":

```
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST transMode.0 = diversity
```

Set the transMode to "tm3":

```
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST transMode.0 = tm3
```

Set the transMode to "tm4":

```
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST transMode.0 = tm4
```

4.4.11 Status reporting

The *snmpwalk* command can help to retrieve an overall view of some parameters in a single shot.

4.4.11.1 RF Status

A walk over the *rfOnBoard* object (for Azcom RF/RRH) will provide a snapshot of the main parameters.

```
      snmpwalk -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
      authPriv $HOST rfOnBoard
```

The output for a successful command may look like:

```
      snmpwalk -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
      authPriv $HOST rfOnBoard
       AZCOM-LTE-OAM-RF-MIB::rfOpStatus.0 = INTEGER: enabled(2)
       AZCOM-LTE-OAM-RF-MIB::txPower.0 = INTEGER: 10
       AZCOM-LTE-OAM-RF-MIB::rxPower.0 = INTEGER: 10
```

Note: these parameters are not currently available on the AZS-XXX-0200 Small Cell family.

4.4.12 Node administration

The options described in this section allow the user to change the eNodeB operational state:

- eNodeB Start
- eNodeB Halt
- eNodeB Restart

Changes to the node operational status do not require the node to be locked, but any object lock by another user will block the operation.

4.4.12.1 eNodeB Start

The commands starts the idle eNodeB after the board has been powered up or restarted. This is needed only on systems that are not preconfigured for autostart.

Setting the *eNodeBopStatus.0* to “start” will start the node:

snmpset -v 3 –t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l authPriv $HOST eNodeBopStatus.0 = start

4.4.12.2 eNodeB Halt

Setting the *eNodeBopStatus.0* to “halt” will halt the node:

```
      snmpset -v 3 –t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
      authPriv $HOST eNodeBopStatus.0 = halt
```

4.4.12.3 eNodeB Restart

Setting the *eNodeBopStatus.0* to “restart” will reboot the node.

The ARM core of the base band board is halted and then the eNodeB board is restarted (power cycle).

The procedure is necessary in some cases, e.g. to apply Duplex Mode change, but a node unlock will automatically do that already.

Set the *eNodeBopStatus.0* to “Restart”:

```
      snmpset -v 3 –t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
      authPriv $HOST eNodeBopStatus.0 = restart
```

4.4.13 Cell RF identifier parameter

This parameter permits to select the RF/RRH board serving a cell. The values defined for the cellRf parameters provide the capability to trigger a manual or dedicated management for the selected RF/RRH board, and consequently define the context of application of cell parameters.

The SNMP HMI requires that the node is locked by the user.

1. enable the manual RRH management mode:
```
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST cellRf.0 = manual
```
2. enable the O&M managed Azcom RRH mode:
```
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST cellRf.0 = ngscbp
```
3. enable the O&M managed Azcom RF mode:
```
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST cellRf.0 = lamarr
```

The cellRf value setting triggers the alignment of the cell parameters with the saved configuration on the RF/RRH board. In cases when the upcoming radio element does not support the current band or bandwidth that were used with the former setup, a suitable band for the current duplex mode and/or a suitable bandwidth will be automatically set.

If no suitable band for the current duplex mode is available, an error would be emitted. In some cases, it will therefore be necessary to switch through a *manual* mode first.

Note: if a configuration package is loaded holding incompatible settings (e.g. current band not foreseen by the radio element) a similar automatic switch will happen as well. In case of an irreconcilable condition, the cellRf setting will automatically switch to *manual* mode.

On read, an AZS-XXX-0200 family system, using on-board Azcom RF radio elements, will report back the specifically detected/configured radio element type that is plugged in.

4.4.14 RF operational status

This read-only parameter shows the enabled, notoperational or disabled state of an RF/RRH board. A disabled or not operational board has its RF power OFF. If you want to turn the power off for a cell only, then refer to the cell operational status described in paragraph 4.4.1. The RF operational status parameter is specific to the RF/RRH selected through the cellRf parameter. For Azcom's RRH the parameter is named rfOpStatus.0.

4.4.15 Node and cell addressing

Addresses are defined through node's *mcc.0*, *mnc.0*, *tac.0*, and cell's *cellId.X* and *phyCellId.X* parameters, where *X* is the index of the cell.

The node must be locked (locking the cell is sufficient for *cellId* and *phyCellId*).

An "eNodeB Restart" is enforced by the HMI on node or cell unlock.

Example:
```
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST mnc.0 = 299
AZCOM-LTE-OAM-BBU-MIB::mnc.0 = STRING: "299"
```

```
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST mcc.0 = 01
AZCOM-LTE-OAM-BBU-MIB::mcc.0 = STRING: "01"
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST cellId.0 = 333
AZCOM-LTE-OAM-BBU-MIB::cellId.0 = INTEGER: 333
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST phyCellId.0 = 440
AZCOM-LTE-OAM-BBU-MIB::phyCellId.0 = INTEGER: 440
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST tac.0 = 150
AZCOM-LTE-OAM-BBU-MIB::tac.0 = INTEGER: 150
```

4.4.16 S1 interfaces

IPs and port addresses for S1AP and S1U interfaces can be defined for both the local node and the peer elements. They are grouped into the s1Interface table which comprehends the parameter listed below:

- s1apeNBIpAddress: node IP address for S1AP
- s1apMmeIpAddress: IP address of MME
- s1uGwIpAddress: IP address of SGW
- s1ueNBPort: Node IP port for S1U
- s1uGwPort: IP port of SGW
- s1apeNBPort: Node IP port for S1AP
- s1apMmePort: IP port of MME
- s1ueNBIpAddress: Node IP address used for S1U
- s1apIface: Interface used to connect S1AP from eNB to EPC.
- s1uIface: Interface used to connect S1U from eNB to EPC.
- s1apVlanId: VLAN id for s1ap for NIB on SOM.
- s1uVlanId: VLAN id for s1u for NIB on SOM.
- s1SctpStatus: Specifies the SCTP connection status between eNB and MME. It reports notOperational value due to connection loss between eNB and MME. It reports operational when the connection is re-established.

The s1Interface can be walked as follows:

Example:

```
snmpwalk -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST s1Interface
AZCOM-LTE-OAM-BBU-MIB::s1apeNBIpAddress.0 = IpAddress: 192.168.3.1
AZCOM-LTE-OAM-BBU-MIB::s1apeNBPort.0 = INTEGER: 5000
AZCOM-LTE-OAM-BBU-MIB::s1apMmeIpAddress.0 = IpAddress: 192.168.1.1
AZCOM-LTE-OAM-BBU-MIB::s1apMmePort.0 = INTEGER: 5000
AZCOM-LTE-OAM-BBU-MIB::s1ueNBIpAddress.0 = IpAddress: 192.168.3.2
AZCOM-LTE-OAM-BBU-MIB::s1ueNBPort.0 = INTEGER: 2152
```

```
AZCOM-LTE-OAM-BBU-MIB::s1uGwIpAddress.0 = IpAddress: 192.168.3.10
AZCOM-LTE-OAM-BBU-MIB::s1uGwPort.0 = INTEGER: 2152
AZCOM-LTE-OAM-BBU-MIB::s1apIface.0 = INTEGER: eth2(2)
AZCOM-LTE-OAM-BBU-MIB::s1uIface.0 = INTEGER: eth3(3)
AZCOM-LTE-OAM-BBU-MIB::s1apVlanId.0 = INTEGER: 1
AZCOM-LTE-OAM-BBU-MIB::s1uVlanId.0 = INTEGER: 1
AZCOM-LTE-OAM-BBU-MIB::s1SctpStatus.0 = INTEGER: operational(0)
```

The S1 interface Ip addresses and port numbers modifications applied through SNMP are set to the database and eNodeB configuration also at operative system level. The S1 interface parameters require a reboot of the systems to be applied.

If the EPC resides on the BBU, the *azNIBMode.0* parameter must be set to bbu(1).

If the EPC resides on a SOM, the *azNIBMode.0* parameter must be set to som(2). VLAN IDs necessary for this configuration can be changed via the MIB.

Local IPs are read-only. They are read from the defined Ethernet interfaces.

The only exception is with the S1U local IP when the EPC resides on the BBU (NIB on ARM configuration, see below).

s1SctpStatus is read-only.

4.4.16.1 NIB on ARM particular case

The ARM NIB configuration requires an Ethernet loopback among two Ethernet interfaces (refer to [3]), say ethA the one connected via fastpath to the DSPs and ethB the one connected with EPC on ARM.

EthA must not have an IP address set, but it must be configured as the *s1uIface.0*. Still, the stack needs to have one IP address which falls into the same netmask as the IP set on ethB at network level. This IP address for ethA is only set via *s1ueNBIpAddress.0*, which is made writable only in this case.

4.4.17 X2 interface

X2 is the interface between two eNBs.

IP and port addresses for X2AP and X2U interfaces can be defined for the local node. They are grouped into the x2Interface table, which can be walked.

X2 setup status reports the outcome of X2 connection attempts. eNB1 initiates the X2 setup procedure by sending the X2 SETUP REQUEST message to eNB2. If eNB2 cannot accept the setup it shall respond with an X2 SETUP FAILURE message and this is reported as *failed* state. This is cleared when X2 setup is established between the two.

X2 unknown neighbor reports an *alarmed* status whenever a peer eNB receives a PDU from eNB which is not valid to it or vice versa.

Example:

```
snmpwalk -v 3 –t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST x2Interface
AZCOM-LTE-OAM-BBU-MIB::x2eNBIpAddress.0 = IpAddress: 192.168.3.1
AZCOM-LTE-OAM-BBU-MIB::x2apeNBPort.0 = INTEGER: 36000
AZCOM-LTE-OAM-BBU-MIB::x2ueNBPort.0 = INTEGER: 36100
```

```
AZCOM-LTE-OAM-BBU-MIB::x2SetupStatus.0 = INTEGER: clear(0)
AZCOM-LTE-OAM-BBU-MIB::x2NeighUnknownStatus.0 = INTEGER: clear(0)
```

As for the S1 interface also for the X2 interface Ip addresses and port numbers modifications applied through SNMP are set to the database and eNodeB configuration files but not at operative system level, so the same values must be changed in the operative system network configuration. The X2 interface parameters require a reboot of the system to be applied.

x2SetupStatus and x2NeighUnknownStatus are read-only.

4.4.18 Clock Sync Source

Azcom eNodeB can use three kinds of sources for internal syncing: a gps source, ieee1588 protocol, or an external 10Mhz sync source connected to the board. Values for the *syncSource.0* parameter are encoded as follows:

- 0 = intclock
- 1 = extclock (10 Mhz reference)
- 2 = gps
- 3 = ieee1588

The SNMP HMI requires that the node is locked by the user. The system will reboot at unlock.

Each subsystem will support a subset of the values, so the clock source setting can be rejected if the system does not support that particular source value.

Note that the *intclock* (internal clock sync source) can be a viable option to use only if your board is equipped with an High Performance clock, otherwise the quality will not be sufficient to make the system work properly in a long run.

Set the clock sync source to “gps”:

```
snmpset -v 3 –t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST syncSource.0 = gps
```

Status of the synch in case of 10MHz External reference is available with the command:

```
snmpget   -v  3  –t 15 -u   $USER   -a   SHA -A $PWD1 -x DES -X
$PWD2 -l authPriv  $HOST  clockLockStatus.0
```

4.4.19 Log server

Azcom eNodeB can efficiently provide traces to a server which opens an UDP port. This set of parameters is able to define the server details and to trim the desired log severity levels.

Disclaimer: depending on the severity level and the number of enabled mask flags, the logging system may introduce a computational weight which may compromise the normal functionality of the system, in particular when involving PHY or L2 subsystems.

4.4.19.1 Log server addresses

Local network information needed by the log setup at the DSPs (i.e. the local IP) is retrieved from the interface assigned to s1u. The remote IP of the log Server can be assigned via the *logServerIpAddr.0* parameter.

Example:

```
      snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
      authPriv $HOST logServerIpAddr.0 = 192.168.3.8
```

4.4.19.2 Logs severity levels

Different severity levels can be defined for OAM, L3, L2 and PHY subsystems, and for trapped events. Traces with a less critical severity will be filtered out.

The setting is persistent and therefore it will be in use at the next reboot as well.

Valid severities are:

critical(0), error(1), warning(2), info(3), debug(4)

Examples:

```
      snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
      authPriv $HOST logPhySeverity.0 = critical

      snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
      authPriv $HOST logL2Severity.0 = 2

      snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
      authPriv $HOST logL3Severity.0 = warning

      snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
      authPriv $HOST logOamSeverity.0 = warning

      snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
      authPriv $HOST logEventsSeverity.0 = warning
```

4.4.19.3 Logs mode masks

Different masks can be set to filter the desired logs for OAM, L3, L2 and PHY subsystems. Trapped events use the same bitmask as OAM. Traces related to a bit flag not enabled by the masks will be filtered out. A 32-bit mask is defined for the PHY subsystem and 64-bit masks are defined for OAM, L2 and L3 subsystems. To set the masks an 8 or 16 nibbles long hexadecimal format string must be set for the corresponding parameters for PHY or OAM/L2/L3 subsystems respectively. A mask string with a length not exactly equal to 8 or 16 nibbles string will cause an error and a ‘wrong value’ message will be returned by the sset command.

The mask will take effect only after the snmpset on log severity parameter is applied. So the snmpset command on a specific subsystem mask must precede the snmpset command on log severity for the same subsystem. The setting is persistent and therefore it will be in use at the next reboot as well.

Examples:

```
      snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
      authPriv $HOST logPhyMask.0 = 0001ffff

      snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
      authPriv $HOST logL2Mask.0 = aabbccddeeff0011

      snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
      authPriv $HOST logL3Mask.0 = 0000000000000004
```

```
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST logOAMMask.0 = 0000000000000999
```

4.4.20 Software Management

Software Management allows to transfer and activate application software images.

Transfers can be done either through *tftp* or *scp* protocols. The latter is the default because of improved security, but it additionally needs a user name for the remote server.

In both cases, a server must be set up properly, and have a folder containing all-in-one software packages, aggregating both an image and its installation script. Also, the server IP address and the package file name must be set for a transfer to succeed.

In order to have an overall view of the software management status and its related parameters, a *swalk* command over the container label is very handy:

```
snmpwalk -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST softwareMgmt
AZCOM-LTE-OAM-BBU-MIB::swmOpStatus.0 = INTEGER: idle(0)
AZCOM-LTE-OAM-BBU-MIB::swmServerIpAddr.0 = IpAddress: 192.168.3.2
AZCOM-LTE-OAM-BBU-MIB::swmServerUserName.0 = STRING: "swmgmt"
AZCOM-LTE-OAM-BBU-MIB::swmTransferMode.0 = INTEGER: scp(1)
AZCOM-LTE-OAM-BBU-MIB::swmTransferFile.0 = STRING: "path/Enb-
tws8300bbu_08.00.04.azpkg"
```

In order to configure a tftp server please refer to the manual of the product you are using.

In order to configure the *scp* transfer type using a Linux server box, you need to append the node’s public ssh key in your server user’s *~/.ssh/authorized_keys* file. The configured ssh public key might be specially crafted for the systems delivered to you.

4.4.20.1 Software Management operational status

The operational status reports the current status through some literals (idle, transfer, abort, ready, install, failed) that can also be used as command when set. The parameter can be set without node lock except for the “install” value, provided that no other user is locking it.

The meaning of every value provided for the swmOpStatus parameter if used during a get or a set command is described in the following table:

	GET	SET
idle	current image is running	clear abort/ready/failed conditions
transfer	image is being transferred locally	transfer swmTransferFile.0 file locally
abort	error occurred	abort transfer
ready	transferred image ready to install	-
install	image is being installed	install image if ready (NEEDS LOCK)
failed	image install failed	-

To initiate a file transfer (after all needed parameters are properly set as mentioned above) use the *transfer* value, which will also be reported until the transfer ends:

```
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST swmOpStatus.0 = transfer

snmpget   -v  3  -t 15 -u   $USER   -a   SHA -A $PWD1 -x DES -X
$PWD2 -l authPriv  $HOST  swmOpStatus.0
AZCOM-LTE-OAM-BBU-MIB::swmOpStatus.0 = INTEGER: transfer(1)
```

Note that other functionalities are forbidden in this state and only *abort* value can be set to stop an ongoing transfer.

The parameter will report *abort* if the transfer is aborted due to an abort command or a transfer error occurred.

```
snmpget   -v  3  -t 15 -u   $USER   -a   SHA -A $PWD1 -x DES -X
$PWD2 -l authPriv  $HOST  swmOpStatus.0
AZCOM-LTE-OAM-BBU-MIB::swmOpStatus.0 = INTEGER: transfer(1)

snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST swmOpStatus.0 = abort

snmpget   -v  3  -t 15 -u   $USER   -a   SHA -A $PWD1 -x DES -X
$PWD2 -l authPriv  $HOST  swmOpStatus.0
AZCOM-LTE-OAM-BBU-MIB::swmOpStatus.0 = INTEGER: transfer(1)
  …
snmpget   -v  3  -t 15 -u   $USER   -a   SHA -A $PWD1 -x DES -X
$PWD2 -l authPriv  $HOST  swmOpStatus.0
AZCOM-LTE-OAM-BBU-MIB::swmOpStatus.0 = INTEGER: abort(2)
```

When no current operation is ongoing, the status reported is *idle*. The *idle* value can be used as a command to clear an *abort* status indication or to clear the current ready image.

```
snmpget   -v  3  -t 15 -u   $USER   -a   SHA -A $PWD1 -x DES -X
$PWD2 -l authPriv  $HOST  swmOpStatus.0
AZCOM-LTE-OAM-BBU-MIB::swmOpStatus.0 = INTEGER: abort(2)

snmpset -v 3 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST swmOpStatus.0 = idle
snmpget   -v  3  -t 15 -u   $USER   -a   SHA -A $PWD1 -x DES -X
$PWD2 -l authPriv  $HOST  swmOpStatus.0
AZCOM-LTE-OAM-BBU-MIB::swmOpStatus.0 = INTEGER: idle(0)
```

Once the file is successfully transferred the get command on the swmOpStatus shows the *ready* value and the package can be installed by setting the *install* value. If the install procedure fails the get command on the swmOpStatus return the *failed* value otherwise the BBU will reboot automatically to activate the new package image.

```
snmpget   -v  3  -t 15 -u   $USER   -a   SHA -A $PWD1 -x DES -X
$PWD2 -l authPriv  $HOST  swmOpStatus.0
       AZCOM-LTE-OAM-BBU-MIB::swmOpStatus.0 = INTEGER: ready(3)

snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST swmOpStatus.0 = install
AZCOM-LTE-OAM-BBU-MIB::swmOpStatus.0 = INTEGER: install(4)
…
snmpget   -v  3  -t 15 -u   $USER   -a   SHA -A $PWD1 -x DES -X
$PWD2 -l authPriv  $HOST  swmOpStatus.0
AZCOM-LTE-OAM-BBU-MIB::swmOpStatus.0 = INTEGER: failed(5)
```

4.4.20.2 Server address

The server IP address is needed for an image file transfer. It can be set without node lock, provided that no other user is locking it.

Example:

```
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST swmServerIpAddr.0 = 192.168.3.2
```

4.4.20.3 Server user name

This parameter needs to be set for scp transfers only. It can be set without node lock, provided that no other user is locking it.

Example:

```
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST swmServerUserName.0 = swmanagement
```

4.4.20.4 Transfer Mode

Defines the protocol used for the file transfer: *tftp* or *scp* (default). It can be set without node lock, provided that no other user is locking it.

Example:

```
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST swmTransferMode.0 = scp
```

4.4.20.5 Transfer File Name

Defines the filename to be looked for in the remote system for transfer. For the *scp* protocol, it should include an absolute path or a relative path from the username's home directory. *Tftp* does not support paths.

It can be set without node lock, provided that no other user is locking it and no other long lasting operation (file transfer or image activation) is executing.

Note that different images that, on server, have different paths but the same base name, will conflict as the node only keeps the base name, and therefore a transfer will fail as if it was duplicated.

Example:

```
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST swmTransferFile.0 = remotePath/Enb-
tws8300bbu_08.00.04.azpkg
```

Note that platform tool used by SNMP to install the package applies some checks on the file name and content. Only official release packages provided by AZCOM Technology with the name starting with “Enb” and with “.azpkg” extension can be installed.

4.4.21 CPRI parameters

The *cpriTable* object contains parameters relevant for the support and trimming of different RRH boards. Not every parameter is actually used on every radio element. In particular, systems in the AZS-XXX-0200 family do not use a CPRI connection towards the radio elements. Your system should come preconfigured with proper trim values for your radio element of choice, so use these only if you really know what you are doing.

4.4.21.1 CPRI op status

Read-only parameter which reports the operational status of the CPRI link (disabled, enabled, invalid). An invalid status means that the indexed CPRI entry is not meaningful in the current context.

Example:

```
snmpget   -v  3  -t 15 -u   $USER   -a   SHA -A $PWD1 -x DES -X
$PWD2 -l authPriv  $HOST  cpriOpStatus.0
AZCOM-LTE-OAM-BBU-MIB::cpriOpStatus.0 = INTEGER: enabled(1)
```

4.4.21.2 AT PIMIN

This parameter defines the CPRI receive window: the antenna interface checks that the K28.5 synch character falls within the expected Pi window specified by Pi_min and Pi_max (=Pi_min+200). The smallest recommended Pi_min is 380 byte clock (245.76MHz for CPRI).

Value ranges between 0 and 4194303.

Example:

```
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST cpriAtPimin.0 = 580
```

4.4.21.3 dBm offset

RF factor used for the conversion from digital dB bit to antenna RX dBm power:

dBm = dB bit + DBM_OFFSET

Value ranges between -2147483648 and 2147483647.

Example:

```
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST cpriDBmOffset.0 = -119
```

4.4.21.4 HF synch delay

Waiting time before the enabling of RX link. The RX link is enabled only if CPRI Slave already in HFNSYNC and start CPRI transmit properly.

This is to avoid eNB CPRI synchronized to improper UL signal from CPRI Slave. Unit is LTE radio frame. This parameter is automatically set to a default suitable to the selected RRH when setting cellRf.0 to a value other than *manual*.

Value ranges between 0 and 4294967295.

Example:

```
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST cpriHFsyncDelay.0 = 1500
```

4.4.21.5 AxC offset

Antenna Carrier Offset programmed in the number of four samples, relative to the link (PHY) frame boundary.

The AxC offset is used to compensate the time propagation between the base station and the remote radio head when there is a long delay on fiber.

Value ranges between 0 and 33554431.

The cpriAxcOffset default values are pre-configured for different RF boards depending also on the txBandwidth and duplexMode set. These tabled values can be overwritten through the set command when the corresponding configuration is in operation.

Example:

```
snmpget   -v  3  -t 15 -u   $USER   -a   SHA -A $PWD1 -x DES -X
$PWD2 -l authPriv  $HOST  cpriAxcOffset.0
      AZCOM-LTE-OAM-BBU-MIB:: cpriAxcOffset.0 = INTEGER: 0

snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST cpriAxcOffset.0 = 100
AZCOM-LTE-OAM-BBU-MIB:: cpriAxcOffset.0 = INTEGER: 100
```

4.4.21.6 TCP/IP over CPRI

The TCP/IP over CPRI is used to enable the TCP/IP protocol to communicate over CPRI through the channel connected to the RRH.

Values are 0 or 1 that correspond respectively to “disabled” or “enabled” status.

Example:

```
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST tcpipOverCpri.0 = enable
```

4.4.22 GPS Info parameters

The GPS Info parameters group provides information about the available GPS satellites. This group of parameters is constituted by the selected GPS satellite information:

- azGpsFixtype: represents the NMEA mode (undefined, none, two dimensions, three dimensions)
- azGpsTimeStamp: timestamp from GPS device
- azGpsLatitude: is the latitude in degrees where +/- signifies North/South. This value is set if azGpsFixtype has a value of 2 (two dimensions) or 3 (three dimensions)
- azGpsLongitude: is the longitude in degrees where +/- signifies East/West. This value is set if azGpsFixtype has a value of 2 (two dimensions) or 3 (three dimensions)
- azGpsHeight: is the altitude in meters. This value is set if azGpsFixtype has a value of 3 (three dimensions)
- azGpsHeading: is the course over ground in degrees from true north
- azGpsVelocityHorizontal: is the speed over ground in meters per second
- azGpsVelocityVertical: is the limb (positive) or sink (negative) rate in meters per second
- azGpsSatelliteNumber: the number of visible satellites as displayed in azGpsSatelliteInfoTable

The GPS satellites that are visible during the location fix are listed in the azGpsSatelliteInfoTable table where the values of the following parameters are shown:

- azGpsSatelliteElevation: elevation in degrees

- azGpsSatelliteAzimuth: Azimuth of the current satellite in context referenced by the Satellite InfoIndex. Azimuth is the degree of rotation of the satellites dish on its vertical plane
- azGpsSatelliteSignalStrength: Signal to Noise Ratio(SNR) of received GPS signal in dB. SNR is referred to as the signal strength in GPS standards
- azGpsSatelliteUsed: is this satellite in line of sight to the GPS device and used in calculating the GPS location

Example:

```
snmpget   -v  3  -t 15 -u   $USER   -a   SHA -A $PWD1 -x DES -X
$PWD2 -l authPriv  $HOST  azGpsFixtype.0
AZCOM-LTE-OAM-BBU-MIB::azGpsFixtype.0 = INTEGER: threeDimension(3)

snmpget   -v  3  -t 15 -u   $USER   -a   SHA -A $PWD1 -x DES -X
$PWD2 -l authPriv  $HOST  azGpsTimeStamp.0
AZCOM-LTE-OAM-BBU-MIB::azGpsTimeStamp.0 = STRING: Thu May 4 11:30:56
2006

snmpget   -v  3  -t 15 -u   $USER   -a   SHA -A $PWD1 -x DES -X
$PWD2 -l authPriv  $HOST  azGpsLatitude.0
AZCOM-LTE-OAM-BBU-MIB::azGpsLatitude.0 = STRING: 47.064471 degrees

snmpget   -v  3  -t 15 -u   $USER   -a   SHA -A $PWD1 -x DES -X
$PWD2 -l authPriv  $HOST  azGpsLongitude.0
AZCOM-LTE-OAM-BBU-MIB::azGpsLongitude.0 = STRING: 15.453376 degrees

snmpget   -v  3  -t 15 -u   $USER   -a   SHA -A $PWD1 -x DES -X
$PWD2 -l authPriv  $HOST  azGpsHeight.0
AZCOM-LTE-OAM-BBU-MIB::azGpsHeight.0 = INTEGER: 386 m

snmpget   -v  3  -t 15 -u   $USER   -a   SHA -A $PWD1 -x DES -X
$PWD2 -l authPriv  $HOST  azGpsHeading.0
AZCOM-LTE-OAM-BBU-MIB::azGpsHeading.0 = INTEGER: 75 degrees

snmpget   -v  3  -t 15 -u   $USER   -a   SHA -A $PWD1 -x DES -X
$PWD2 -l authPriv  $HOST  azGpsVelocityHorizontal.0
AZCOM-LTE-OAM-BBU-MIB::azGpsVelocityHorizontal.0 = STRING: 0.014000
m/s

snmpget   -v  3  -t 15 -u   $USER   -a   SHA -A $PWD1 -x DES -X
$PWD2 -l authPriv  $HOST  azGpsVelocityVertical.0
AZCOM-LTE-OAM-BBU-MIB::azGpsVelocityVertical.0 = INTEGER: 44 degrees
```

All the GPS Info parameters can be read at once with the walk command.

```
snmpwalk -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azGpsInfo
AZCOM-LTE-OAM-BBU-MIB::azGpsFixtype.0 = INTEGER: threeDimension(3)
AZCOM-LTE-OAM-BBU-MIB::azGpsTimeStamp.0 = STRING: Thu May 4 11:30:56
2006
AZCOM-LTE-OAM-BBU-MIB::azGpsLatitude.0 = STRING: 47.064471 degrees
AZCOM-LTE-OAM-BBU-MIB::azGpsLongitude.0 = STRING: 15.453376 degrees
```

```
AZCOM-LTE-OAM-BBU-MIB::azGpsHeight.0 = INTEGER: 386 m
AZCOM-LTE-OAM-BBU-MIB::azGpsHeading.0 = INTEGER: 75 degrees
AZCOM-LTE-OAM-BBU-MIB::azGpsVelocityHorizontal.0 = STRING: 0.014000
m/s
AZCOM-LTE-OAM-BBU-MIB:: azGpsSatelliteNumber.0 = = INTEGER: 3
AZCOM-LTE-OAM-BBU-MIB::azGpsSatelliteElevation.0 = INTEGER: 44
degrees
AZCOM-LTE-OAM-BBU-MIB::azGpsSatelliteElevation.1 = INTEGER: 68
degrees
AZCOM-LTE-OAM-BBU-MIB::azGpsSatelliteElevation.2 = INTEGER: 42
degrees
AZCOM-LTE-OAM-BBU-MIB::azGpsSatelliteAzimuth.0 = INTEGER: 181
degrees
AZCOM-LTE-OAM-BBU-MIB::azGpsSatelliteAzimuth.1 = INTEGER: 293
degrees
AZCOM-LTE-OAM-BBU-MIB::azGpsSatelliteAzimuth.2 = INTEGER: 62 degrees
AZCOM-LTE-OAM-BBU-MIB::azGpsSignalStrength.0 = INTEGER: 27 dB
AZCOM-LTE-OAM-BBU-MIB::azGpsSignalStrength.1 = INTEGER: 50 dB
AZCOM-LTE-OAM-BBU-MIB::azGpsSignalStrength.2 = INTEGER: 48 dB
AZCOM-LTE-OAM-BBU-MIB::azGpsSatelliteUsed.0 = INTEGER: true(1)
AZCOM-LTE-OAM-BBU-MIB::azGpsSatelliteUsed.1 = INTEGER:true(1)
AZCOM-LTE-OAM-BBU-MIB::azGpsSatelliteUsed.2 = INTEGER:true(1)
```

4.4.23 NTP configuration

The NTP configuration group allows the configuration of a primary (*azNtpServerPrimary.*0) and (optional) secondary (*azNtpServerSecondary.*0) NTP time server references.

Examples:

```
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azNtpServerPrimary.0 = 192.168.50.50

snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azNtpServerSecondary.0 = 192.168.50.51
```

4.4.24 Configuration save

The platform configuration command is piloted by the *azConfigSave.0* parameter. The set operation on *azConfigSave.0* with the active (1) value will save node configuration to flash. Missing to do so will make latest settings volatile, and they will not be recovered after a system restart.

On read, the status may be success (0), error (1) or mismatch(3). The reported status of *azConfigSave.0* refers to the outcome of last save operation and if the *mismatch* value is returned by the Get command means that a save command must be done because some configuration parameters have been changed. This situation is notified also through the "Configuration save needed" log printed on console during the Get command.

In cases when a reboot is needed to enforce some parameters, *eNodeBalarm.0* will notify this with the *willReboot* (4) status. A reboot will be automatically triggered just after a save command in this case. This will bring the system in the same condition throughout a reboot, yet with *eNodeBalarm.0* status cleared. If the configuration is not saved and an unlock is attempted when *eNodeBalarm.0* is not cleared, the configuration will be automatically saved and the needed reboot is automatically triggered at unlock.

Note that a save command also redefines the conditions for the following startup, including the conditions defined for the *Delayed sync with RRH procedure*.

The command sequence that follows shows a configuration *mismatch* event:

```
snmpget   -v  3  -t 15 -u   $USER   -a   SHA -A $PWD1 -x DES -X
$PWD2 -l authPriv  $HOST  azConfigSave.0
Configuration save needed
AZCOM-LTE-OAM-BBU-MIB:: azConfigSave.0 = INTEGER: mismatch(3)

snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azConfigSave.0 = active
AZCOM-LTE-OAM-BBU-MIB:: azConfigSave.0 = INTEGER: active(1)

snmpget   -v  3  -t 15 -u   $USER   -a   SHA -A $PWD1 -x DES -X
$PWD2 -l authPriv  $HOST  azConfigSave.0
AZCOM-LTE-OAM-BBU-MIB:: azConfigSave.0 = INTEGER: success(0)
```

4.4.25 L2 measurements

eNodeB performs layer 2 measurements which are signaled across OAM. Multiple measurements can be requested in one single request.

The azMeasActive.0 parameter triggers the activation/stop of the selected measurements.

4.4.25.1 Selection flags and values

The user can ask for multiple measurements in single request for different QCI/measurement type. This is supported by setting (enabling) the measurement selection flags for relevant L2 measurement parameters before actually activating the measurement request.

Selection flags can be changed without lock, but trying to change any of them while measurements are active will result in *inconsistentValue* error. Flags are retained after deactivation of measurements, but they are nevertheless not persistent after a reboot.

Measurement values are reported only if a measurement was selected and only after it is provided by L2 after the last activation, otherwise the parameter will return *No Such Instance* (will result as hidden when walking the values).

The measurements will also be logged and trapped by the OAM subsystem at warning severity over the bit 9 in the OAM log bitmask.

There are three types of measurements:

- Global
- UL QCI based
- DL QCI based

Global Selection Flags can be set by parameters under the azL2MeasSelectionsTable as described in the table below:

MIB Parameter(selection/value)	**Measurement**
Selection: azUlTotalPrbUsageActive.0 *Value:* azUlTotalPrbUsage.0	Total physical resource block usage in uplink. Example Measurement Log: *Total UL prb usage: 86*
Selection: azDlTotalPrbUsageActive.0 *Value:* azDlTotalPrbUsage.0	Total physical resource block usage in downlink. Example Measurement Log: *Total DL prb usage: 96*
Selection: azUlRecvRandAccessPreamActive.0 *Values:* azUlRecvRandAccessPreamTotal.0 azUlRecvRandAccessPreamDedicated.0 azUlRecvRandAccessPreamLow.0 azUlRecvRandAccessPreamHigh.0	Number of received Random Access preambles during a time period over all PRACHs configured in a cell. The measurement is composed by 4 different values: • total number of preambles • number of Dedicated preambles • number of preambles in the low range • number of preambles in the high range Example Measurement Log: *UL Received RA Preambles Total:87* *UL Received RA Preambles Dedicated:76* *UL Received RA Preambles Low:11* *UL Received RA Preambles High:0*

Table 1. Global L2 meas selection flags

Selection flags for QCI based measurements in UL can be set by parameters under the azL2MeasUlQciSelectionsTable using a second index to indicate the QCI in a range (1,9) as described in the table below:

MIB Parameter(selection/value)	**Selected Measurement**
Selection: azUlPrbUsagePerQciActive.0.x *Value:* azUlPrbUsagePerQci.0.x	Physical resource blocks usage per traffic class for uplink DTCH or per QCI x Example Measurement Log: *UL prb usage for QCI 1: 68*
Selection: azUlPktLossRateActive.0.x *Value:* azUlPktLossRate.0.x	Packet loss rate in the uplink per QCI x for data radio bearers Example Measurement Log: *Packet loss rate in uplink for QCI 1 (packets per million): 85000*
Selection: azUlNumActiveUeActive.0.x *Value:* azUlNumActiveUe.0.x	Number of active UE's in uplink per QCI x. It refers to UEs for which there is buffered data for uplink in data radio bearers Example Measurement Log: *Num active UE in UL for QCI 1: 2*
Selection: azUlSchedIpTptActive.0.x *Value:* azUlSchedIpTpt.0.x	IP throughput in uplink per QCI x Example Measurement Log: *IP throughput in uplink for UE 1 (RNTI 11) QCI 1: 3946 kbps*

Table 2. QCI based L2 meas in UL

Selection flags for QCI based measurements in DL can be set by parameters under the azL2MeasDlQciSelectionsTable using a second index to indicate the QCI in a range (1,9) as described in the table below:

MIB Parameter(selection/value)	Selected Measurement
Selection: azDlPrbUsagePerQciActive.0.x *Value:* azDlPrbUsagePerQci.0.x	Physical resource block usage per traffic class for downlink DTCH or per QCI x Example Measurement Log: *DL prb usage for QCI 1: 76*
Selection: azDlAvgPktDelayActive.0.x *Value:* azDlAvgPktDelay.0.x	Measure packet delay in downlink per QCI x for data radio bearers Example Measurement Log: *Packet delay in downlink for QCI 1: 10*
Selection: azDlPktUuLossRateActive.0.x *Value:* azDlPktUuLossRate.0.x	Packet loss rate in the downlink per QCI x for data radio bearers Example Measurement Log: *Packet Uu loss rate in downlink for QCI 1 (packets per million): 1360*
Selection: azDlPktDiscardRateActive.0.x *Value:* azDlPktDiscardRate.0.x	Packet discard rate in the downlink per QCI x for data radio bearers Example Measurement Log: *Discard rate in downlink for QCI 1 (packets per million):10500*
Selection: azDlNumActiveUeActive.0.x *Value:* azDlNumActiveUe.0.x	Number of active UE's in downlink per QCI x Example Measurement Log: *Num active UE in DL for QCI 1: 3*
Selection: azDlSchedIpTptActive.0.x *Value:* azDlSchedIpTpt.0.x	IP throughput in downlink per QCI x Example Measurement Log: *IP throughput in downlink for UE 1 (RNTI 11) QCI 1: 3946 kbps*

Table 3. QCI based L2 meas in DL

Examples of setting flags:

```
snmpset -v 3 -t  15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azUlTotalPrbUsageActive.0 1
AZCOM-LTE-OAM-L2MEAS-MIB:: azUlTotalPrbUsageActive.0 = INTEGER:
enable(1)

snmpset -v 3 -t  15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azUlPrbUsagePerQciActive.0.9 1
```

```
AZCOM-LTE-OAM-L2MEAS-MIB:: azUlPrbUsagePerQciActive.0.9 = INTEGER:
enable(1)

snmpset -v 3 -t  15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azUlPrbUsagePerQciActive.0.6 1
AZCOM-LTE-OAM-L2MEAS-MIB:: azUlPrbUsagePerQciActive.0.6 = INTEGER:
enable(1)
```

Example of walking current flags and activation settings (case not active):

```
snmpwalk -v 3 -t 15   -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -
l authPriv $HOST azL2Measurements
AZCOM-LTE-OAM-L2MEAS-MIB:: azMeasActive.0 = INTEGER: disable(0)
AZCOM-LTE-OAM-L2MEAS-MIB:: azUlTotalPrbUsageActive.0 = INTEGER:
enable(1)
AZCOM-LTE-OAM-L2MEAS-MIB:: azDlTotalPrbUsageActive.0 = INTEGER:
disable(0)
AZCOM-LTE-OAM-L2MEAS-MIB:: azUlRecvRandAccessPreamActive.0 =
INTEGER: disable(0)
AZCOM-LTE-OAM-L2MEAS-MIB:: azUlPrbUsagePerQciActive.0.1 = INTEGER:
disable(0)
AZCOM-LTE-OAM-L2MEAS-MIB:: azUlPrbUsagePerQciActive.0.2 = INTEGER:
disable(0)
AZCOM-LTE-OAM-L2MEAS-MIB:: azUlPrbUsagePerQciActive.0.3 = INTEGER:
disable(0)
AZCOM-LTE-OAM-L2MEAS-MIB:: azUlPrbUsagePerQciActive.0.4 = INTEGER:
disable(0)
AZCOM-LTE-OAM-L2MEAS-MIB:: azUlPrbUsagePerQciActive.0.5 = INTEGER:
disable(0)
AZCOM-LTE-OAM-L2MEAS-MIB:: azUlPrbUsagePerQciActive.0.6 = INTEGER:
enable(1)
AZCOM-LTE-OAM-L2MEAS-MIB:: azUlPrbUsagePerQciActive.0.7 = INTEGER:
disable(0)
AZCOM-LTE-OAM-L2MEAS-MIB:: azUlPrbUsagePerQciActive.0.8 = INTEGER:
disable(0)
AZCOM-LTE-OAM-L2MEAS-MIB:: azUlPrbUsagePerQciActive.0.9 = INTEGER:
enable(1)
… (continues)
```

The user can disable selection flags:

```
snmpset -v 3 -t  15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azUlPrbUsagePerQciActive.0.6 0
AZCOM-LTE-OAM-L2MEAS-MIB:: azUlPrbUsagePerQciActive.0.6 = INTEGER:
disable(0)
```

Activation of measurements according the flags selected:.

```
snmpset -v 3 -t  15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azMeasActive.0 1
AZCOM-LTE-OAM-L2MEAS-MIB:: azMeasActive.0 = INTEGER: enable(1)
```

Once the measurements are active, selection flags cannot be modified any more. On any attempt, “inconsistent value” error will be returned until the measurements are deactivated.

```
snmpset -v 3 -t  15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azUlPrbUsagePerQciActive.0.6 1
Error in packet.
```

```
Reason: inconsistentValue (The set value is illegal or unsupported
in some way)
Failed object: AZCOM-LTE-OAM-L2MEAS-MIB::
azUlPrbUsagePerQciActive.0.6
```

Measurements are stopped by setting the azMeasActive.0 parameter *disable*.

```
snmpset -v 3 -t  15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azMeasActive.0 0
AZCOM-LTE-OAM-L2MEAS-MIB:: azMeasActive.0 = INTEGER: disable(0)
```

Once measurement request is sent to L3 based on measurements selected. User can read the measurement values on snmp interface.

```
snmpset -v 3 -t  15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azUlTotalPrbUsageActive.0 1
AZCOM-LTE-OAM-L2MEAS-MIB:: azUlTotalPrbUsageActive.0 = INTEGER:
enable(1)

snmpset -v 3 -t  15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azUlPrbUsagePerQciActive.0.9 1
AZCOM-LTE-OAM-L2MEAS-MIB:: azUlPrbUsagePerQciActive.0.9 = INTEGER:
enable(1)

snmpset -v 3 -t  15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azUlPrbUsagePerQciActive.0.6 1
AZCOM-LTE-OAM-L2MEAS-MIB:: azUlPrbUsagePerQciActive.0.6 = INTEGER:
enable(1)

snmpset -v 3 -t  15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azMeasActive.0 1
AZCOM-LTE-OAM-L2MEAS-MIB:: azMeasActive.0 = INTEGER: enable(1)
```

Measurement read at snmp interface:

```
snmpget -v 3 -t  15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azUlPrbUsagePerQci.0.9
AZCOM-LTE-OAM-L2MEAS-MIB:: azUlPrbUsagePerQci.0.9 = INTEGER: 68

snmpget -v 3 -t  15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azUlTotalPrbUsage.0
AZCOM-LTE-OAM-L2MEAS-MIB:: azUlTotalPrbUsage.0 = INTEGER: 96

snmpget -v 3 -t  15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azUlPrbUsagePerQci.0.6
AZCOM-LTE-OAM-L2MEAS-MIB:: azUlPrbUsagePerQci.0.6 = INTEGER: 69
```

User can read all the measurements which have been received since the last activation by:

```
snmpwalk -v 3 -t  15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -
l authPriv $HOST azL2MeasValues
AZCOM-LTE-OAM-L2MEAS-MIB::azUlTotalPrbUsage.0 = INTEGER: 96
AZCOM-LTE-OAM-L2MEAS-MIB::azUlPrbUsagePerQci.0.6 = INTEGER: 69
AZCOM-LTE-OAM-L2MEAS-MIB::azUlPrbUsagePerQci.0.9 = INTEGER: 68
```

All the three measurement values which are present can be read by user at once.

User can read all the global measurements which are present in system by:

```
snmpwalk -v 3 -t  15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -
l authPriv $HOST azL2MeasValuesEntry
AZCOM-LTE-OAM-L2MEAS-MIB::azUlTotalPrbUsage.0 = INTEGER: 96
```

User can read all the UL QCI based measurements which are present in system by :

```
snmpwalk -v 3 -t  15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -
l authPriv $HOST azL2MeasUlQciValuesTable
AZCOM-LTE-OAM-L2MEAS-MIB::azUlPrbUsagePerQci.0.6 = INTEGER: 69
AZCOM-LTE-OAM-L2MEAS-MIB::azUlPrbUsagePerQci.0.9 = INTEGER: 68
```

User can read all the DL QCI based measurements which are present in system by:

```
snmpwalk -v 3 -t  15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -
l authPriv $HOST azL2MeasDlQciValuesTable
```

Since in the examples above no DL QCI selection flag was selected, the table is reported empty:

```
AZCOM-LTE-OAM-L2MEAS-MIB::azL2MeasDlQciValuesTable = No Such Object
available on this agent at this OID
```

4.4.26 Handover configurations

Handover configurations are necessary to control the policy adopted by the system to decide that a UE could shift to another cell, and to define the candidate neighbor cells.

As the UE is moving towards another cell, signal strength of that cell becomes greater than the one under system control and therefore UE shall undergo a handover to that cell. The network triggers the handover procedure based on radio conditions. To facilitate this, the network configures the UE to perform measurement reporting. The network may also initiate handover blindly, i.e. without having received measurement reports from the UE.

Based on measurement reports from UE, the network makes a decision on whether it will let UE handover or not. To avoid blind handover, a list of neighbor cells is also required in configuration.

Handovers directed to adjacent EUTRA cells can be managed either through the X2 or S1 (EPC-managed) protocols. As a last resort, an handover can be directed towards an UTRA adjacent cell.

The following configurations are supported for handover:

1. azHoX2NeighCellsTable specifies a list of configurations available for EUTRA X2 neighbor cells.
2. azHoS1NeighCellsTable specifies a list of configurations available for EUTRA S1 neighbor cells.
3. azHoUtraCellsTable specifies a list of configurations available for UTRA neighbor cells.
4. azHoReportConfigTable specifies a list of configurations available for measurement reports.
5. azHoMeasObjectTable specifies a list of configurations available for measurement objects.
6. azHoMeasIdTable specifies a list of configurations available for measurements composed by one meas report and one meas object from the above tables.

7. azHoMeasConfigTable specifies a list of general parameters related to the measurements for handover.

Except for azHoMeasConfigTable, which holds a single configuration row for the cell, each table supports row additions and deletions through a row management field of standard SNMP type RowStatus.

Since most tables require data for which a defined default value cannot be suitable for every network, the RowStatus entry will usually only allow to create temporary rows to be further filled with values (*createAndWait*), while an attempt to create and activate a row with default values (*createAndGo*) will return an error. Setting a parameter over a row that does not exist will imply that the *createAndWait* is automatically done.

All the tables are explained in detail in the following sub sections.

4.4.26.1 RowStatus management

The RowStatus column is used to manage the creation and deletion of rows in a table. This column has six defined values which are as follows:

1. *active*: indicates that the row with all columns is in use.
2. *notInService*: indicates that the row is ready to be activated, but is not actually used by the system.
3. *notReady*: indicates that one or more columns in the row required for activation are not instantiated.
4. *createAndGo*: used to create a new row and make it available for use, filling missing fields with defaults. Will fail if at least one default value cannot be determined.
5. *createAndWait*: used to create a new row but not making it available for use.
6. destroy: to delete all of the columns associated with an existing row.

All values except *notReady* may be specified in a set operation, however, only three values will be returned in response to a get command: *notReady*, *notInService* or *active*. That is, when queried, an existing row can only have one out of three states:

1. It is either in use (RowStatus column has value *active*)
2. It is not used, though sufficient information is provided in order to attempt to activate it (RowStatus column has value *notInService*)
3. It is not used and an attempt to activate it would fail because insufficient information are available (RowStatus column has value *notReady*).

Available actions are:

1. To make the system use a row, a row in *notInService* RowStatus shall be set to *active*.
2. To delete a row, RowStatus shall be set to destroy which will remove the row from the table.
3. To make the system stop using a row, yet keep that available for further processing, the operator can set an active RowStatus to *notInService*.

Note that rows with temporary values – i.e. not in *active* status – will not be persistent

after a system restart. After a system restart, active rows will have indices compacted to the low numbers, i.e. filling the gaps left by deleted or temporary rows.

The *Presence* column in tables below indicates whether a parameter is expected to be given a value explicitly, or it can be omitted and given a default value, before a row is switched to *active*.

4.4.26.2 azHoX2NeighCellsTable

This table covers neighbor cells configuration for Intra LTE handover performed by X2 interface.

X2 is the interface between two eNBs, a serving eNB and a target eNB. When X2 interface is present, then handover is performed using X2 interface without involving EPC. Handover decision is performed based on neighboring cell measurement reports.

The SNMP HMI requires that the cell is locked by the user, and the configuration will apply for the current duplex mode. The parameters require a system restart to be applied.

Semantics of the parameters in azHoX2NeighCellsTable are defined as follows:

MIB Parameter	Presence	Explanation	Type
azHoX2NeighStatus	Mandatory	It is used to manage the creation and deletion of rows.	RowStatus
azHoX2NeighPhyCellId	Mandatory	Physical Cell ID is an identification of a cell at physical layer.	Integer (0..503)
azHoX2NeighIpAddr	Mandatory	Indicates IP address of the node required for socket creation for X2 interface.	IpAddress
azHoNeighX2apPort	Mandatory	Indicates X2ap port of the node required for socket creation for X2 interface.	Integer (0..65535)
azHoX2NeighMacAddr	Mandatory	Indicates MAC address of the node.	MacAddress
azHoX2NeighX2uPort	Mandatory	Indicates X2u port of the node.	Integer (0..65535)

Table 4 X2 neighbor cells table

Examples:

```
snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azHoX2NeighPhyCellId.0.0 123
AZCOM-LTE-OAM-BBU-MIB::azHoX2NeighPhyCellId.0.0 = INTEGER: 123

snmpset -v 3 -t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv   $HOST azHoX2NeighIpAddr.0.0 1.2.3.4
AZCOM-LTE-OAM-BBU-MIB::azHoX2NeighIpAddr.0.0 = IpAddress: 1.2.3.4

snmpset -v 3 -t  15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azHoX2NeighMacAddr.0.0 1:2:3:4:5:6
```

```
AZCOM-LTE-OAM-BBU-MIB::azHoX2NeighMacAddr.0.0 = STRING: 1:2:3:4:5:6
```

4.4.26.3 azHoS1NeighCellsTable

This table covers neighbor cells configuration for Intra LTE handover performed by S1 interface. S1 interface is the interface between eNB and EPC. In case when X2 interface is not available, then handover is carried out through S1 interface. Handover decision is performed based on neighboring cell measurement reports.

The SNMP HMI requires that the cell is locked by the user, and the configuration will apply for the current duplex mode. The parameters require a system restart to be applied.

Semantics of the parameters in azHoS1NeighCellsTable are defined as follows:

MIB Parameter	Presence	Explanation	Type
azHoS1NeighStatus	Mandatory	It is used to manage the creation and deletion of rows.	RowStatus
azHoS1NeighPhyCellId	Mandatory	Physical Cell ID is an identification of a cell at physical layer.	Integer (0..503)
azHoS1NeighType	Mandatory	Indicates the node type (small, macro or home).	Enum (small, macro,home)
azHoS1NeighPlmnMcc	Mandatory	Public Land Mobile Network Identity (PLMN-ID) have a three digit mobile country code (MCC).	String size : 3
azHoS1NeighPlmnMnc	Mandatory	Public Land Mobile Network Identity (PLMN-ID) have a three a two or three digit mobile network code (MNC).	String size: 2..3
azHoS1NeighCellId	Mandatory	It is used to uniquely identify the cell within a network.	Integer (1..268435455)

Table 5 S1 neighbor cells table

Examples:

```
snmpset -v 3 -t 15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azHoS1NeighPhyCellId.0.0 123
AZCOM-LTE-OAM-BBU-MIB::azHoS1NeighPhyCellId.0.0= INTEGER: 123

snmpset -v 3 -t 15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv   $HOST azHoS1NeighType.0.0 1
AZCOM-LTE-OAM-BBU-MIB::azHoS1NeighType.0.0 = INTEGER: home(1)

snmpset -v 3 -t 15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azHoS1NeighPlmnMcc.0.0 123
AZCOM-LTE-OAM-BBU-MIB::azHoS1NeighPlmnMcc.0.0 = STRING: “123”
```

4.4.26.4 azHoUtraCellsTable

This table covers neighbor cells configuration for inter RAT handover. It contains all the parameters to uniquely identify an UTRA cell and other necessary information.

The SNMP HMI requires that the cell is locked by the user, and the configuration will apply for the current duplex mode. The parameters require a system restart to be applied.

Semantics of the parameters in azHoUtraCellsTable are defined as follows:

MIB Parameter	Presence	Explanation	Type
azHoUtraStatus	Mandatory	It is used to manage the creation and deletion of rows.	RowStatus
azHoUtraPlmnMcc	Mandatory	PLMN ID consists of 3 digit Mobile Country Code (MCC).	String size: 3
azHoUtraPlmnMnc	Mandatory	PLMN ID consists of 2 or 3 digit Mobile Network Code (MNC).	String size : 2..3
azHoUtraCellId	Mandatory	It identifies a cell within an RNS.	Integer (0..65535)
azHoUtraRncId	Mandatory	It uniquely identifies an RNC within a PLMN.	Integer (0..65535)
azHoUtraLAC	Mandatory	The served area of a cellular radio network is usually divided into location areas. Each location area is given a unique number within the network, the Location Area Code (LAC).	Integer (0..65535)
azHoUtraRAC	Mandatory	Routing area is the sub-division of location area and is used by mobiles which are GPRS attached. Routing Area Code is used to uniquely identify a routing area.	Integer (0..255)
azHoUtraPsc	Mandatory	It is primary scrambling code. It improves synchronization.	Integer (0..511)

Table 6 UTRA neighbor cells table

Examples:

```
snmpset -v 3 -t 15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv  $HOST azHoUtraCellId.0.0 1234
AZCOM-LTE-OAM-BBU-MIB::azHoUtraCellId.0.0 = INTEGER: 1234

snmpset -v 3 -t 15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv  $HOST azHoUtraRncId.0.0 123
AZCOM-LTE-OAM-BBU-MIB::azHoUtraRncId.0.0 = INTEGER: 123
```

```
snmpset -v 3 -t 15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv  $HOST azHoUtraLAC.0.0 78
AZCOM-LTE-OAM-BBU-MIB::azHoUtraLAC.0.0 = INTEGER: 78
```

4.4.26.5 azHoReportConfigTable

This table contains parameters related to connected mode mobility. It specifies criteria for triggering of an E-UTRA and UTRA measurement reporting event.

All the mandatory parameters need to be set and the optional parameters can take a default value in order to commit a row.

The SNMP HMI requires that the cell is locked by the user, and the configuration will apply for the current duplex mode. The parameters require a system restart to be applied.

Semantics of the parameters in azHoReportConfigTable are defined as follows:

MIB Parameter	Presence	Explanation	Type
azHoReportConfigStatus	Mandatory	It is used to manage the creation and deletion of rows.	RowStatus
azHoReportConfigChoice	Mandatory	It specifies that the report is for which type of cell i.e EUTRA(1) or UTRA(2) cell.	Enum (eutra,utra)
azHoReportConfigTriggerType	Mandatory	It specifies how the report will be triggered either event based or periodical.	Enum (event, periodic)
azHoReportConfigEvent	Optional Default : ab1	It specifies the choice of E-UTRA or UTRA event triggered reporting criteria. EUTRA events are from a1 to a5 and UTRA events are b1 and b2. Value ab1 means A1 event for EUTRA cell or B1 event for UTRA cell.	Enum (ab1, ab2, a3, a4, a5)
azHoReportConfigThresholdChoice	Optional Default : rsrp	It specifies threshold quantity to be used for measurement report triggering condition for events A1, A2, A4 and A5	Enum (rsrp, rsrq)
azHoReportConfigThresholdRsrp	Mandatory	Threshold to be used in	Integer

		EUTRA measurement report triggering condition for events A1, A2, A3 and A5 provided threshold_choice is RSRP	(0..97)
azHoReportConfigThresholdRsrq	Mandatory	Threshold to be used in EUTRA measurement report triggering condition for event A1, A2, A3 and A5 provided threshold_choice is RSRQ	Integer (0..34)
azHoReportConfigA3Offset	Mandatory	Offset value to be used in EUTRA measurement report triggering condition for event A3	Integer (-30..30)
azHoReportConfigOnLeave	Optional Default : false	Indicates whether or not the UE shall initiate the measurement reporting procedure when the leaving condition is met for a cell.	Enum (false, true)
azHoReportConfigThreshold2Choice	Optional Default : rsrp	It specifies threshold quantity to be used for measurement report triggering condition for event A5	Enum (rsrp, rsrq)
azHoReportConfigThreshold2Rsrp	Mandatory	Threshold2 to be used in EUTRA measurement report triggering condition for event A5.	Integer (0..97)
azHoReportConfigThreshold2Rsrq	Mandatory	Threshold2 to be used in EUTRA measurement report triggering condition for event A5.	Integer (0..34)
azHoReportConfigThresholdUtraChoice	Optional Default : rscp	It specifies threshold quantity to be used for measurement report triggering condition for event B1 and B2	Enum (ecno, rscp)
azHoReportConfigThresholdRSCP	Mandatory	RSCP threshold to be	Integer

		used in UTRA measurement report triggering condition for event B1 and B2.	(-5..91)
azHoReportConfigThresholdEcNO	Mandatory	EcNO threshold to be used in UTRA measurement report triggering condition for event B1 and B2.	Integer (0..49)
azHoReportConfigHysteresis	Optional Default : 0	Hysteresis value applied to entry and leave condition of a report triggering event. It is expressed in dB.	Integer (0..30)
azHoReportConfigTimeToTrigger	Optional Default : ms40	Time period during which measurement report triggering condition needs to be met in order to trigger a measurement report.	Enum (ms0, ms40, ms64, ms80, ms100, ms128, ms160, ms256, ms320, ms480, ms512, ms640, ms1024, ms1280, ms2560, ms5120)
azHoReportConfigTriggerQuantity	Optional Default : rsrp	Quantities used to evaluate a measurement report triggering condition.	Enum (rsrp, rsrq)
azHoReportConfigQuantity	Optional default : both	The quantities to be included in the measurement report. The value both means that both the rsrp and rsrq quantities are to be included in the measurement report.	Enum (trigger, both)
azHoReportConfigMaxCells	Optional	Maximum number of	Integer

	default : 8	cells that can be included in a measurement report.	(1..8)
azHoReportConfigInterval	Optional Default : ms480	Interval between successive measurement reports.	Enum (ms120, ms240, ms480, ms640, ms1024, ms2048, ms5120, ms10240, min1, min6, min12, min30, min60)
azHoReportConfigAmount	Optional default: inf	Number of times a measurement report is sent.	Enum (ra1, ra2, ra4, ra8, ra16, ra32, ra64, inf)

Table 7 HO Report Configurations table

Examples:

```
        snmpset -v 3 -t 15   -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
        authPriv $HOST azHoReportConfigChoice.0.0 1
        AZCOM-LTE-OAM-BBU-MIB::azHoReportConfigChoice.0.0= INTEGER: eutra(1)

        snmpset -v 3 -t 15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
        authPriv $HOST azHoReportConfigTriggerType.0.0 1
        AZCOM-LTE-OAM-BBU-MIB::azHoReportConfigTriggerType.0.0= INTEGER:
        event(1)

        snmpset -v 3 -t 15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
        authPriv $HOST azHoReportConfigThresholdRsrp.0.0.0.0 78
        AZCOM-LTE-OAM-BBU-MIB::azHoReportConfigThresholdRsrp.0.0= INTEGER:
        78
```

4.4.26.6 azHoMeasObjectTable

This table specifies information applicable for intra-frequency or inter-frequency E-UTRA cells and inter-RAT UTRA neighboring cells.

All the mandatory parameters need to be set and the optional parameters can take a default value in order to commit a row.

The SNMP HMI requires that the cell is locked by the user, and the configuration will apply for the current duplex mode. The parameters require a system restart to be applied.

Semantics of the parameters in azHoMeasObjectTable are defined as follows:

MIB Parameter	Presence	Explanation	Type
azHoMeasObjectStatus	Mandatory	It is used to manage the creation and deletion of rows.	RowStatus
azHoUtraMeasObjectChoice	Mandatory	It is used to distinguish between measurement object of EUTRA (1) and UTRA (2) cell.	Enum (eutra, utra)
azHoCarrierFreq	Mandatory	E-UTRA (or ARFCN), uniquely identifies the LTE band and carrier frequency and does not take into account the channel bandwidth. If EARFCN of the serving cell is same as that of the neighbor cell, then it means that the latter is an intra frequency neighbor, otherwise it is inter frequency neighbor.	Integer (0..65535)
azHoMeasResourceBlock	Optional Default : mbw15	It is used to indicate the maximum allowed measurement bandwidth on a carrier frequency as defined by the parameter Transmission Bandwidth Configuration "NRB”. The values 6, 15, 25, 50, 75, 100 indicate 6, 15, 25, 50, 75 and 100 resource blocks respectively.	Enum (mbw6, mbw15, mbw25, mbw50, mbw75, mbw100)
azHoEutraqOffset	Optional Default : dBMinus1	Indicates a frequency specific offset of the frequency of the neighbor cell, specified in dB. It is used for evaluating triggering conditions for measurement reporting in connected mode.	Enum: (dBMinus24, dBMinus22, dBMinus20, dBMinus18, dBMinus16, dBMinus14, dBMinus12, dBMinus10, dBMinus8, dBMinus6, dBMinus5,

			dBMinus4, dBMinus3, dBMinus2, dBMinus1, dB0, dB1, dB2, dB3, dB4, dB5, dB6, dB8, dB10, dB12, dB14, dB16, dB18, dB20, dB22, dB24)
azHoCellsForWhichToReportCGI	Optional default : 0	It indicates the PCI of that cell for which to report CGI.	Integer (0..503)

Table 8 HO Measurement Objects table

Examples:

```
snmpset -v 3 -t 15   -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azHoMeasObjectChoice.0.0 1
AZCOM-LTE-OAM-BBU-MIB::azHoMeasObjectChoice.0.0 = INTEGER: eutra(1)

snmpset -v 3 -t 15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azHoMeasCarrierFreq.0.0 1234
AZCOM-LTE-OAM-BBU-MIB::azHoMeasCarrierFreq.0.0= INTEGER: 1234

snmpset -v 3 -t 15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azHoMeasObjectStatus.0.0 1
AZCOM-LTE-OAM-BBU-MIB::azHoMeasObjectStatus.0.0 = INTEGER: active(1)

snmpwalk -v 3 -t 15   -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -
l authPriv $HOST azHoMeasObjectTable
AZCOM-LTE-OAM-BBU-MIB::azHoMeasObjectStatus.0.0 = INTEGER: active(1)
AZCOM-LTE-OAM-BBU-MIB::azHoMeasObjectChoice.0.0 = INTEGER: eutra(1)
AZCOM-LTE-OAM-BBU-MIB::azHoMeasCarrierFreq.0.0 = INTEGER: 3100
AZCOM-LTE-OAM-BBU-MIB::azHoMeasResourceBlock.0.0 = INTEGER: mbw15(1)
AZCOM-LTE-OAM-BBU-MIB::azHoEutraqOffset.0.0 = INTEGER: dBMinus1 (14)
AZCOM-LTE-OAM-BBU-MIB::azHoCellsForWhichToReportCGI.0.0 = INTEGER: 0

snmpset -v 3 -t 15   -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azHoMeasObjectStatus.0.0 6
AZCOM-LTE-OAM-BBU-MIB::azHoMeasObjectStatus.0.0 = INTEGER:
destroy(6)

snmpwalk -v 3 -t 15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azHoMeasObjectTable
AZCOM-LTE-OAM-BBU-MIB::azHoMeasObjectTable = No Such Object
available on this agent at this OID
```

4.4.26.7 azHoMeasIdTable

This table is used to identify a measurement configuration, i.e., linking of a measurement object and a reporting configuration. This is done by creating rows containing the indexes of existing Meas Objects and Report Configs.

The SNMP HMI requires that the cell is locked by the user, and the configuration will apply for the current duplex mode. The parameters require a system restart to be applied.

The meas id table is created by adding a reference of meas objects and report configs. So reference to a non-existent meas object ID or report ID will report an error.

On deleting a meas object or report entry, its corresponding entries in meas ID table referencing it will also be deleted. As a side effect, all meas id rows will be resynched with the persistency database, therefore the rows that are not yet active will be wiped even if referencing other existing active meas objects and reports. Conversely, active rows with valid references will still remain persistent.

Semantics of the parameters in azHoMeasIdTable are defined as follows:

MIB Parameter	**Presence**	**Explanation**	**Type**
azHoMeasIdStatus	Mandatory	It is used to manage the creation and deletion of rows.	RowStatus
azHoMeasObjectId	Mandatory	It is used to identify a measurement object configuration. It is a foreign key from measObject.	Integer (0..32)
azHoReportId	Mandatory	It is used to identify the measurement report. It is a foreign key from reportConfig.	Integer (0..32)

Table 9 HO Measurements Id table

Examples:

```
snmpset -v 3 -t 15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv  $HOST azHoMeasObjectId.0.0 1
AZCOM-LTE-OAM-BBU-MIB::azHoMeasObjectId.0.0 = INTEGER: 1

snmpset -v 3 -t 15   -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv  $HOST azHoReportId.0.0 2
AZCOM-LTE-OAM-BBU-MIB::azHoReportId.0.0 = INTEGER: 2

snmpset -v 3 -t 15   -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv  $HOST azHoMeasIdStatus.0.0 1
AZCOM-LTE-OAM-BBU-MIB::azHoMeasIdStatus.0.0 = INTEGER: active(1)
```

4.4.26.8 azHoMeasConfigTable

This table specifies general configuration for measurements to be performed by the UE in a cell, and covers intra-frequency, inter-frequency and inter-RAT mobility as well as configuration of measurement gaps.

The SNMP HMI requires that the cell is locked by the user, and the configuration will apply for the current duplex mode. The parameters require a system restart to be applied.

In order to enable measurement gaps, azHoGapOffset must be set to any valid value in range (0..79). To disable the measurement gaps, azHoGapOffset must be set to -1.

Semantics of the parameters in azHoMeasConfigTable are defined as follows:

MIB Parameter	Presence	Explanation	Type
azHoEutraFilterCoefficientRsrp	Optional default : fc0	Filtering coefficient used for RSRP measurements. The UE filters the measured result, before using for evaluation of reporting criteria or for measurement reporting.	Enum: (fc0, fc1, fc2, fc3, fc4, fc5, fc6, fc7, fc8, fc9, fc11, fc13, fc15, fc17, fc19)
azHoEutraFilterCoefficientRsrq	Optional default : fc0	Filtering coefficient used for RSRQ measurements. The UE filters the measured result, before using for evaluation of reporting criteria or for measurement reporting.	Enum: (fc0, fc1, fc2, fc3, fc4, fc5, fc6, fc7, fc8, fc9, fc11, fc13, fc15, fc17, fc19)
azHoUtraMeasQuantityFdd	Optional default : 1	Measurement quantity used for UTRA measurements in FDD.	Enum (cpich_RSCP, cpich_EcNO)
azHoUtraMeasQuantityTdd	Optional default : 0	Measurement quantity used for UTRA measurements in TDD.	Enum (pccpch-RSCP)
azHoUtraFilterCoefficient	Optional default : fc4	Filtering coefficient used for UTRA measurements.	Enum: (fc0, fc1, fc2, fc3, fc4, fc5, fc6, fc7, fc8, fc9, fc11, fc13, fc15, fc17, fc19)
azHoGapConfigChoice	Optional default : setup	It controls setup/ release of measurement gaps.	Enum (release, setup)
azHoGapOffsetChoice	Optional default : gp1	It indicates the gap pattern to be used for configuring measurement gaps.	Enum (gp0, gp1)
azHoGapOffset	Optional	It indicates the value of	Integer

	default : -1	gap offset to be used, specified in ms(milli second). If gap pattern ID 0 is used , then max value is 40. If gap pattern ID 1 is used, then max value is 80. If value is -1, then it specifies that measurement gaps are disabled currently.	(-1..79)
azHoHysteresisP	Optional default : 9	It indicates the value of threshold to be used in EUTRA measurement report triggering condition for periodic reports. Valid if triggeringQuantity is RSRP.	Integer (0..30)
azHoHysteresisQ	Optional default : 5	It indicates the value of threshold to be used in EUTRA measurement report triggering condition for periodic reports. Valid if triggeringQuantity is RSRQ.	Integer (0..30)
azHoThresholdRSCP	Optional default : 60	It indicates the value of threshold to be used in UTRA measurement report triggering condition for periodic reports. Valid if triggeringQuantity is RSCP.	Integer (-5..91)
azHoThresholdECNO	Optional default : 43	It indicates the value of threshold to be used in UTRA measurement report triggering condition for periodic reports. Valid if triggeringQuantity is EcNO.	Integer (0..49)
azHoThresholdRSRP	Optional	It indicates the value of	Integer

	default : 60	threshold to be used in UTRA measurement report triggering condition for periodic reports. Valid if triggeringQuantity is RSRP.	(0..97)
azHoThresholdRSRQ	Optional default : 28	It indicates the value of threshold to be used in UTRA measurement report triggering condition for periodic reports. Valid if triggeringQuantity is RSRQ	Integer (0..34)
azHoThresholdRSRPMeasGapEnable	Optional default : 60	RSRP threshold value below which the measurement gaps will be enabled.	Integer (0..97)
azHoThresholdRSRPMeasGapDisable	Optional default : 69	RSRP threshold value above which the measurement gaps will be disabled.	Integer (0..97)
azHoThresholdRSRQMeasGapEnable	Optional default : 24	RSRQ threshold value below which the measurement gaps will be enabled	Integer (0..34)
azHoThresholdRSRQMeasGapDisable	Optional default : 29	RSRQ threshold value above which the measurement gaps will be disabled	Integer (0..34)

Table 10 HO Measurements Config Table

Examples:

```
snmpset -v 3 -t 15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv  $HOST cellAdmStatus.0 0
AZCOM-LTE-OAM-BBU-MIB::cellAdmStatus.0 = INTEGER: lock(0)

snmpset -v 3 -t 15   -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azHoEutraFilterCoefficientRsrp.0 11
AZCOM-LTE-OAM-BBU-MIB::azHoEutraFilterCoefficientRsrp.0 = INTEGER:
11

snmpset -v 3 -t 15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azHoEutraFilterCoefficientRsrq.0.0 12
```

```
AZCOM-LTE-OAM-BBU-MIB::azHoEutraFilterCoefficientRsrq.0.0 = INTEGER:
12

snmpwalk -v 3 -t 15   -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -
l authPriv $HOST azHoMeasConfigTable
AZCOM-LTE-OAM-BBU-MIB::azHoEutraFilterCoefficientRsrp.0 = INTEGER:
fc9(9)
AZCOM-LTE-OAM-BBU-MIB::azHoEutraFilterCoefficientRsrq.0 = INTEGER:
fc9(9)
AZCOM-LTE-OAM-BBU-MIB::azHoUtraMeasQuantityFdd.0 = INTEGER:
cpichEcNO(1)
AZCOM-LTE-OAM-BBU-MIB::azHoUtraMeasQuantityTdd.0 = INTEGER:
pccpchRSRP(0)
AZCOM-LTE-OAM-BBU-MIB::azHoUtraFilterCoefficient.0 = INTEGER: fc9(9)
AZCOM-LTE-OAM-BBU-MIB::azHoGapConfigChoice.0 = INTEGER: release(2)
AZCOM-LTE-OAM-BBU-MIB::azHoGapOffsetChoice.0 = INTEGER: gp1(1)
AZCOM-LTE-OAM-BBU-MIB::azHoGapOffset.0 = INTEGER: 0
AZCOM-LTE-OAM-BBU-MIB::azHoHysteresisP.0 = INTEGER: 9
AZCOM-LTE-OAM-BBU-MIB::azHoHysteresisQ.0 = INTEGER: 5
AZCOM-LTE-OAM-BBU-MIB::azHoThresholdRSCP.0 = INTEGER: 60
AZCOM-LTE-OAM-BBU-MIB::azHoThresholdECNO.0 = INTEGER: 43
AZCOM-LTE-OAM-BBU-MIB::azHoThresholdRSRP.0 = INTEGER: 60
AZCOM-LTE-OAM-BBU-MIB::azHoThresholdRSRQ.0 = INTEGER: 28
AZCOM-LTE-OAM-BBU-MIB:: azHoThresholdRSRPMeasGapEnable.0 = INTEGER:
60
AZCOM-LTE-OAM-BBU-MIB:: azHoThresholdRSRPMeasGapDisable.0 = INTEGER:
69
AZCOM-LTE-OAM-BBU-MIB:: azHoThresholdRSRQMeasGapEnable.0 = INTEGER:
24
AZCOM-LTE-OAM-BBU-MIB:: azHoThresholdRSRQMeasGapDisable.0 = INTEGER:
29
```

4.4.27 Statistics

Layer 1, Layer 2 and Layer 3 statistics provide an effective overview of the running system status.

Layer 1 provides stats from the physical layer. Layer 2 provides stats about MAC, PDCP, RLC and S1U, and about memory and packets on DSPs used by the layer. Layer 3 provides stats about RRC. In addition global statistics are also represented. The statistics data is Read-Only.

azRefreshStats.0 is available under the cellsTable to allow the operator to instruct the system to capture statistics data either manually or automatically. This parameter can be changed without lock.

Any query over stats data in manual mode would be answered from the captured stats at the time when the refresh command is issued (parameter is set to *manual*) and will remain unchanged afterwards, until the next command. In auto mode the query would be answered from the captured data at the time when they are queried, yet they will be returned from cache if data is less than 6 seconds old. A walk over all the statistics will therefore always return data captured at the same time.

Querying over the statistics before the source subsystems have completely started will return NO SUCH INSTANCE as output.

The aggregated statistics data will represent a logical summation/averaging (where meaningful, as per the statistic type meaning) over a set of data available for all currently available rnti's on the running system for a particular stat. For example, azMacDlTputTotal will represent an aggregated dl tput value summed up for all available rnti's.

Semantics of the parameters in azMACStatisticsTable are defined per rnti as follows:

MIB Parameter	Description
azUlSnrPusch	SNR calculated in UL over PUSCH
azUlSnrPucch	SNR calculated in UL over PUCCH
azDlSnr	SNR calculated in DL
azMacDlTput	DL throughput calculated at MAC
azMacUlTput	UL throughput calculated at MAC
azDlBlerOfTB1	DL Bler calculated for TB1 (Percentage of NACKs received in DL for TB1)
azDlBlerOfTB2	DL Bler calculated for TB2 (Percentage of NACKs received in DL for TB2)
azUlBler	Bler calculated in UL (Percentage of NACKs sent for UL data)

Semantics of the parameters in azMACStatisticsAggregated are defined as follows:

MIB Parameter	Description
azMacDlTputTotal	DL throughput calculated at MAC for all available rnti's
azMacUlTputTotal	UL throughput calculated at MAC for all available rnti's
azDlBlerOfTB1Total	DL Bler average calculated for TB1 (Percentage of NACKs received in DL for TB1) for all available rnti's
azDlBlerOfTB 2Total	DL Bler average calculated for TB2 (Percentage of NACKs received in DL for TB2) for all available rnti's
azUlBlerTotal	Bler average calculated in UL (Percentage of NACKs sent for UL data) for all available rnti's

Semantics of the parameters in azRLCStatisticsTable are defined per rnti as follows:

MIB Parameter	Description
azRlcDlTput	DL throughput at RLC
azRlcUlTput	UL throughput at RLC
azRlcPktWholeNackRecv	Number of NACKs received for Complete RLC PDU
azRlcPktSegNackRecv	Number of NACKs received for Segments of RLC PDUs

Semantics of the parameters in azRLCStatisticsAggregated are defined as follows:

MIB Parameter	Description
azRlcDlTputTotal	DL throughput at RLC for all available rnti’s
azRlcUlTputTotal	UL throughput at RLC for all available rnti’s
azRlcPktWholeNackRecvTotal	Number of NACKs received for Complete RLC PDU for all available rnti’s
azRlcPktSegNackRecvTotal	Number of NACKs received for Segments of RLC PDUs for all available rnti’s

Semantics of the parameters in azPDCPStatisticsTable are defined per rnti as follows:

MIB Parameter	Description
azDlTput	DL Throughput at PDCP
azUlTput	UL Throughput at PDCP
azUlPktLost	Packet loss in UL at PDCP
azDlPktLost	Packet loss in DL at PDCP

Semantics of the parameters in azPDCPStatisticsAggregated are defined as follows:

MIB Parameter	Description
azDlTputTotal	DL Throughput at PDCP for all available rnti's
azUlTputTotal	UL Throughput at PDCP for all available rnti's
azUlPktLostTotal	Packet loss in UL at PDCP for all available rnti's
azDlPktLostTotal	Packet loss in DL at PDCP for all available rnti's

Semantics of the parameters in azS1UStatisticsTable are defined per rnti as follows:

MIB Parameter	Description
azDlTput	S-GW UL throughput
azUlTput	S-GW DL throughput

Semantics of the parameters in azS1UStatisticsAggregated are defined as follows:

MIB Parameter	Description
azDlTputTotal	S-GW UL throughput for all available rnti's
azUlTputTotal	S-GW DL throughput for all available rnti's

Semantics of the parameters in azRRCStatisticsTable are defined as follows:

MIB Parameter	Description
azRrcConnRequestReceived	Number of RRC Connection Requests received
azRrcConnSetupCompleteNtReceived	Number of RRC Connnection Setup Complete not received
azRrcConnReconfigCompleteNtRcved	Number of RRC Connection Reconfiguration Complete Not Received
azRrcConnReestablishmentReceived	Number of RRCConnectionReestablishments received
azRlfReportedLayer2	Number of RLFs reported by Layer-2
azUEsConfigured	Total number of UEs configured at Layer-3
azUEsReleased	Total number of UEs released at Layer-3

Semantics of the parameters in azGlobalStatisticsTable are defined as follows:

MIB Parameter	Description
azNumRach	Number of RACH received
azNumMsg2	Number of msg2 sent
azNumMsg3	Number of msg3 received
azNumPdcpInactivityTimer	Number of times PDCP Inactivity Timer Expired Called
azNumRLFatMac	Number of RLFs triggered at MAC
azNumOfReleaseTriggeredatMac	Number of releases triggered at MAC
azUlSnrPucchUeRelease	Number of UEs released based on low UL SNR over PUCCH
azUlSnrPuschUeRelease	Number of UEs released based on low UL SNR over PUSCH
azDlCrcUeRelease	Number of UEs released based on High DL Bler
azRlcNumRlf	Number of RLFs triggered at RLC

Semantics of the parameters in azRRCRntiStatisticsTable are defined as follows:

MIB Parameter	Description
azRntisPresent	RNTI present at Layer-3

Semantics of the parameters in azPHYStatisticsTable are defined per UE entry as follows:

MIB Parameter	Description
azUlCqiAnt0	UL CQI on the first configured antenna in dB
azUlCqiAnt1	UL CQI on the second configured antenna in dB
azNoiseVarAnt0	Noise variance on the first configured antenna in dBm
azNoiseVarAnt1	Noise variance on the second configured antenna in dBm
azRssiAnt0	RSSI on the first configured antenna
azRssiAnt1	RSSI on the second configured antenna

Semantics of the parameters in azDspMemoryStatisticsTable are defined per DSP and heap as follows:

MIB Parameter	Description
azDspMemHeapIdentity	A numeric identifier of the memory heap
azDspMemTotalSize	Total memory available in the heap
azDspMemTotalFreeSize	The current size of free memory in the heap
azDspMemLargestFreeSize	The current largest contiguous free memory block in the heap
azDspMemMaxAllocatedBlocks	The maximum number of blocks allocated from this heap at any single point in time, during its lifetime
azDspMemNumAllocatedBlocks	The total number of blocks currently allocated in the heap

Semantics of the parameters in azDspPacketStatisticsTable are defined per DSP and heap as follows:

MIB Parameter	**Description**
azDspPktHeapName	A string identifier of the memory heap
azDspPktNumFreeDataPackets	The current number of free data packets which are available in the heap
azDspPktNumZeroBufferPackets	The current number of free packets (with no buffers) which are available in the heap
azDspPktNumPacketsInGarbage	The current number of packets (with and without) data buffers which are in the garbage queue of the heap
azDspPktDataBufThrStatus	This field is set to indicate that the heap has hit the specified data buffer threshold. It can take values *clear* and *starving*. It is applicable only if the 'starvation data threshold' was specified during heap creation in stack code
azDspPktDataBufStarvationCnt	Indicates the number of times the data buffer queue was starved of packets. This is set only if the heap was configured to use starvation queues in stack code
azDspPktZeroDataBufThrStatus	This field is set to indicate that the heap has hit the specified zero data buffer threshold. It can take values *clear* and *starving*. It is applicable only if the 'starvation zero buffer threshold' was specified during heap creation in stack code
azDspPktZeroDataBufStarvationCnt	Indicates the number of times the zero buffer queue was starved of packets. This is set only if the heap was configured to use starvation queues in stack code

Examples in auto mode:

Assuming 2 UEs are attached in the system with rnti values 10, 11 respectively.

```
snmpget -v 3 -t  15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azRefreshStats.0
AZCOM-LTE-OAM-L2MEAS-MIB:: azRefreshStats.0 = INTEGER: auto(0)
```

User can read all the statistics which are present in system by:

```
snmpwalk -v 3 -t 15   -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azStatistics
AZCOM-LTE-OAM-STATS-MIB::azUlSnrPusch.0.10 = INTEGER: 20
AZCOM-LTE-OAM-STATS-MIB::azUlSnrPusch.0.11 = INTEGER: 22
AZCOM-LTE-OAM-STATS-MIB::azUlSnrPucch.0.10 = INTEGER: 21
AZCOM-LTE-OAM-STATS-MIB::azUlSnrPucch.0.11 = INTEGER: 23
AZCOM-LTE-OAM-STATS-MIB::azDlSnr.0.10 = INTEGER: 22
AZCOM-LTE-OAM-STATS-MIB::azDlSnr.0.11 = INTEGER: 24
AZCOM-LTE-OAM-STATS-MIB::azMacDlTput.0.10 = INTEGER: 23 kbps
AZCOM-LTE-OAM-STATS-MIB::azMacDlTput.0.11 = INTEGER: 25 kbps
AZCOM-LTE-OAM-STATS-MIB::azMacUlTput.0.10 = INTEGER: 24 kbps
AZCOM-LTE-OAM-STATS-MIB::azMacUlTput.0.11 = INTEGER: 26 kbps
AZCOM-LTE-OAM-STATS-MIB::azDlBlerOfTB1.0.10 = INTEGER: 25 ratio/100
AZCOM-LTE-OAM-STATS-MIB::azDlBlerOfTB1.0.11 = INTEGER: 27 ratio/100
AZCOM-LTE-OAM-STATS-MIB::azDlBlerOfTB2.0.10 = INTEGER: 26 ratio/100
AZCOM-LTE-OAM-STATS-MIB::azDlBlerOfTB2.0.11 = INTEGER: 28 ratio/100
AZCOM-LTE-OAM-STATS-MIB::azUlBler.0.10 = INTEGER: 27 ratio/100
AZCOM-LTE-OAM-STATS-MIB::azUlBler.0.11 = INTEGER: 29 ratio/100
AZCOM-LTE-OAM-STATS-MIB::azRlcDlTput.0.10 = INTEGER: 36 kbps
AZCOM-LTE-OAM-STATS-MIB::azRlcDlTput.0.11 = INTEGER: 38 kbps
AZCOM-LTE-OAM-STATS-MIB::azRlcUlTput.0.10 = INTEGER: 37 kbps
AZCOM-LTE-OAM-STATS-MIB::azRlcUlTput.0.11 = INTEGER: 39 kbps
AZCOM-LTE-OAM-STATS-MIB::azRlcPktWholeNackRecv.0.10 = INTEGER: 38
AZCOM-LTE-OAM-STATS-MIB::azRlcPktWholeNackRecv.0.11 = INTEGER: 40
AZCOM-LTE-OAM-STATS-MIB::azRlcPktSegNackRecv.0.10 = INTEGER: 39
AZCOM-LTE-OAM-STATS-MIB::azRlcPktSegNackRecv.0.11 = INTEGER: 41
AZCOM-LTE-OAM-STATS-MIB::azDlTput.0.10 = INTEGER: 32 kbps
AZCOM-LTE-OAM-STATS-MIB::azDlTput.0.11 = INTEGER: 34 kbps
AZCOM-LTE-OAM-STATS-MIB::azUlTput.0.10 = INTEGER: 33 kbps
AZCOM-LTE-OAM-STATS-MIB::azUlTput.0.11 = INTEGER: 35 kbps
AZCOM-LTE-OAM-STATS-MIB::azUlPktLost.0.10 = INTEGER: 34
AZCOM-LTE-OAM-STATS-MIB::azUlPktLost.0.11 = INTEGER: 36
AZCOM-LTE-OAM-STATS-MIB::azDlPktLost.0.10 = INTEGER: 35
AZCOM-LTE-OAM-STATS-MIB::azDlPktLost.0.11 = INTEGER: 37
AZCOM-LTE-OAM-STATS-MIB::azSgwUlTput.0.10 = INTEGER: 30 kbps
AZCOM-LTE-OAM-STATS-MIB::azSgwUlTput.0.11 = INTEGER: 32 kbps
AZCOM-LTE-OAM-STATS-MIB::azSgwDlTput.0.10 = INTEGER: 31 kbps
AZCOM-LTE-OAM-STATS-MIB::azSgwDlTput.0.11 = INTEGER: 33 kbps
AZCOM-LTE-OAM-STATS-MIB::azRrcConnRequestReceived.0 = INTEGER: 14
AZCOM-LTE-OAM-STATS-MIB::azRrcConnSetupCompleteNtReceived.0 = INTEGER: 12
AZCOM-LTE-OAM-STATS-MIB::azRrcConnReconfigCompleteNtRcved.0 = INTEGER: 13
AZCOM-LTE-OAM-STATS-MIB::azRrcConnReestablishmentReceived.0 = INTEGER: 11
AZCOM-LTE-OAM-STATS-MIB::azRlfReportedLayer2.0 = INTEGER: 11
AZCOM-LTE-OAM-STATS-MIB::azUEsConfigured.0 = INTEGER: 21
AZCOM-LTE-OAM-STATS-MIB::azUEsReleased.0 = INTEGER: 20
AZCOM-LTE-OAM-STATS-MIB::azNumRach.0 = INTEGER: 31
AZCOM-LTE-OAM-STATS-MIB::azNumMsg2.0 = INTEGER: 32
AZCOM-LTE-OAM-STATS-MIB::azNumMsg3.0 = INTEGER: 33
AZCOM-LTE-OAM-STATS-MIB::azNumPdcpInactivityTimer.0 = INTEGER: 34
AZCOM-LTE-OAM-STATS-MIB::azNumRLFatMac.0 = INTEGER: 35
AZCOM-LTE-OAM-STATS-MIB::azNumOfReleaseTriggeredatMac.0 = INTEGER: 36
AZCOM-LTE-OAM-STATS-MIB::azUlSnrPucchUeRelease.0 = INTEGER: 37
```

```
AZCOM-LTE-OAM-STATS-MIB::azUlSnrPuschUeRelease.0 = INTEGER: 38
AZCOM-LTE-OAM-STATS-MIB::azDlCrcUeRelease.0 = INTEGER: 39
AZCOM-LTE-OAM-STATS-MIB::azRlcNumRlf.0 = INTEGER: 40
AZCOM-LTE-OAM-STATS-MIB::azRntisPresent.0.0 = INTEGER: 10
AZCOM-LTE-OAM-STATS-MIB::azRntisPresent.0.1 = INTEGER: 11
AZCOM-LTE-OAM-STATS-MIB::azUlCqiAnt0.0.10 = INTEGER: 0 dB
AZCOM-LTE-OAM-STATS-MIB::azUlCqiAnt0.0.11 = INTEGER: 0 dB
AZCOM-LTE-OAM-STATS-MIB::azUlCqiAnt1.0.10 = INTEGER: 16 dB
AZCOM-LTE-OAM-STATS-MIB::azUlCqiAnt1.0.11 = INTEGER: 17 dB
AZCOM-LTE-OAM-STATS-MIB::azNoiseVarAnt0.0.10 = INTEGER: -93 dBm
AZCOM-LTE-OAM-STATS-MIB::azNoiseVarAnt0.0.11 = INTEGER: -94 dBm
AZCOM-LTE-OAM-STATS-MIB::azNoiseVarAnt1.0.10 = INTEGER: -94 dBm
AZCOM-LTE-OAM-STATS-MIB::azNoiseVarAnt1.0.11 = INTEGER: -94 dBm
AZCOM-LTE-OAM-STATS-MIB::azRssiAnt0.0.10 = INTEGER: -90 dBm
AZCOM-LTE-OAM-STATS-MIB::azRssiAnt0.0.11 = INTEGER: -90 dBm
AZCOM-LTE-OAM-STATS-MIB::azRssiAnt1.0.10 = INTEGER: -77 dBm
AZCOM-LTE-OAM-STATS-MIB::azRssiAnt1.0.11 = INTEGER: -77 dBm
AZCOM-LTE-OAM-STATS-MIB::azDspMemHeapIdentity.0.0 = Gauge32: 1
AZCOM-LTE-OAM-STATS-MIB::azDspMemHeapIdentity.1.0 = Gauge32: 1
AZCOM-LTE-OAM-STATS-MIB::azDspMemHeapIdentity.1.1 = Gauge32: 2
AZCOM-LTE-OAM-STATS-MIB::azDspMemTotalSize.0.0 = Gauge32: 32768
AZCOM-LTE-OAM-STATS-MIB::azDspMemTotalSize.1.0 = Gauge32: 32768
AZCOM-LTE-OAM-STATS-MIB::azDspMemTotalSize.1.1 = Gauge32: 32768
AZCOM-LTE-OAM-STATS-MIB::azDspMemTotalFreeSize.0.0 = Gauge32: 32440
AZCOM-LTE-OAM-STATS-MIB::azDspMemTotalFreeSize.1.0 = Gauge32: 32440
AZCOM-LTE-OAM-STATS-MIB::azDspMemTotalFreeSize.1.1 = Gauge32: 32440
AZCOM-LTE-OAM-STATS-MIB::azDspMemLargestFreeSize.0.0 = Gauge32: 3244
AZCOM-LTE-OAM-STATS-MIB::azDspMemLargestFreeSize.1.0 = Gauge32: 6488
AZCOM-LTE-OAM-STATS-MIB::azDspMemLargestFreeSize.1.1 = Gauge32: 6488
AZCOM-LTE-OAM-STATS-MIB::azDspMemMaxAllocatedBlocks.0.0 = Gauge32: 10
AZCOM-LTE-OAM-STATS-MIB::azDspMemMaxAllocatedBlocks.1.0 = Gauge32: 10
AZCOM-LTE-OAM-STATS-MIB::azDspMemMaxAllocatedBlocks.1.1 = Gauge32: 10
AZCOM-LTE-OAM-STATS-MIB::azDspMemNumAllocatedBlocks.0.0 = Gauge32: 2
AZCOM-LTE-OAM-STATS-MIB::azDspMemNumAllocatedBlocks.1.0 = Gauge32: 2
AZCOM-LTE-OAM-STATS-MIB::azDspMemNumAllocatedBlocks.1.1 = Gauge32: 2
AZCOM-LTE-OAM-STATS-MIB::azDspPktHeapName.0.0 = STRING: "pktHeap_001"
AZCOM-LTE-OAM-STATS-MIB::azDspPktHeapName.0.1 = STRING: "pktHeap_002"
AZCOM-LTE-OAM-STATS-MIB::azDspPktHeapName.1.0 = STRING: "pktHeap_001"
AZCOM-LTE-OAM-STATS-MIB::azDspPktNumFreeDataPackets.0.0 = Gauge32: 99
AZCOM-LTE-OAM-STATS-MIB::azDspPktNumFreeDataPackets.0.1 = Gauge32: 99
AZCOM-LTE-OAM-STATS-MIB::azDspPktNumFreeDataPackets.1.0 = Gauge32: 99
AZCOM-LTE-OAM-STATS-MIB::azDspPktNumZeroBufferPackets.0.0 = Gauge32: 0
AZCOM-LTE-OAM-STATS-MIB::azDspPktNumZeroBufferPackets.0.1 = Gauge32: 0
AZCOM-LTE-OAM-STATS-MIB::azDspPktNumZeroBufferPackets.1.0 = Gauge32: 0
AZCOM-LTE-OAM-STATS-MIB::azDspPktNumPacketsInGarbage.0.0 = Gauge32: 1
AZCOM-LTE-OAM-STATS-MIB::azDspPktNumPacketsInGarbage.0.1 = Gauge32: 1
AZCOM-LTE-OAM-STATS-MIB::azDspPktNumPacketsInGarbage.1.0 = Gauge32: 1
AZCOM-LTE-OAM-STATS-MIB::azDspPktDataBufThrStatus.0.0 = INTEGER:
starving(1)
AZCOM-LTE-OAM-STATS-MIB::azDspPktDataBufThrStatus.0.1 = INTEGER:
starving(1)
AZCOM-LTE-OAM-STATS-MIB::azDspPktDataBufThrStatus.1.0 = INTEGER:
starving(1)
```

```
AZCOM-LTE-OAM-STATS-MIB::azDspPktDataBufStarvationCnt.0.0 = Gauge32: 2
AZCOM-LTE-OAM-STATS-MIB::azDspPktDataBufStarvationCnt.0.1 = Gauge32: 2
AZCOM-LTE-OAM-STATS-MIB::azDspPktDataBufStarvationCnt.1.0 = Gauge32: 2
AZCOM-LTE-OAM-STATS-MIB::azDspPktZeroDataBufThrStatus.0.0 = INTEGER:
starving(1)
AZCOM-LTE-OAM-STATS-MIB::azDspPktZeroDataBufThrStatus.0.1 = INTEGER:
starving(1)
AZCOM-LTE-OAM-STATS-MIB::azDspPktZeroDataBufThrStatus.1.0 = INTEGER:
starving(1)
AZCOM-LTE-OAM-STATS-MIB::azDspPktZeroDataBufStarvationCnt.0.0 = Gauge32:
4
AZCOM-LTE-OAM-STATS-MIB::azDspPktZeroDataBufStarvationCnt.0.1 = Gauge32:
4
AZCOM-LTE-OAM-STATS-MIB::azDspPktZeroDataBufStarvationCnt.1.0 = Gauge32:
4
AZCOM-LTE-OAM-STATS-MIB::azMacDlTputTotal.0 = Gauge32: 28 kbps
AZCOM-LTE-OAM-STATS-MIB::azMacUlTputTotal.0 = Gauge32: 30 kbps
AZCOM-LTE-OAM-STATS-MIB::azDlBlerOfTB1Total.0 = Gauge32: 16 ratio/100
AZCOM-LTE-OAM-STATS-MIB::azDlBlerOfTB2Total.0 = Gauge32: 17 ratio/100
AZCOM-LTE-OAM-STATS-MIB::azUlBlerTotal.0 = Gauge32: 18 ratio/100
AZCOM-LTE-OAM-STATS-MIB::azRlcDlTputTotal.0 = Gauge32: 54 kbps
AZCOM-LTE-OAM-STATS-MIB::azRlcUlTputTotal.0 = Gauge32: 56 kbps
AZCOM-LTE-OAM-STATS-MIB::azRlcPktWholeNackRecvTotal.0 = Gauge32: 58
AZCOM-LTE-OAM-STATS-MIB::azRlcPktSegNackRecvTotal.0 = Gauge32: 60
AZCOM-LTE-OAM-STATS-MIB::azDlTputTotal.0 = Gauge32: 46 kbps
AZCOM-LTE-OAM-STATS-MIB::azUlTputTotal.0 = Gauge32: 48 kbps
AZCOM-LTE-OAM-STATS-MIB::azUlPktLostTotal.0 = Gauge32: 50
AZCOM-LTE-OAM-STATS-MIB::azDlPktLostTotal.0 = Gauge32: 52
AZCOM-LTE-OAM-STATS-MIB::azSgwUlTputTotal.0 = Gauge32: 42 kbps
AZCOM-LTE-OAM-STATS-MIB::azSgwDlTputTotal.0 = Gauge32: 44 kbps
```

Snmpwalk in manual mode, or after a time lapse of 6 seconds in auto mode, will show fresh values.

A scenario when there are no UEs present in the system:

```
snmpwalk -v 3 -t 15   -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azStatistics
AZCOM-LTE-OAM-STATS-MIB::azRrcConnRequestReceived.0 = INTEGER: 1
AZCOM-LTE-OAM-STATS-MIB::azRrcConnSetupCompleteNtReceived.0 = INTEGER: 0
AZCOM-LTE-OAM-STATS-MIB::azRrcConnReconfigCompleteNtRcved.0 = INTEGER: 0
AZCOM-LTE-OAM-STATS-MIB::azRrcConnReestablishmentReceived.0 = INTEGER: 0
AZCOM-LTE-OAM-STATS-MIB::azRlfReportedLayer2.0 = INTEGER: 1
AZCOM-LTE-OAM-STATS-MIB::azUEsConfigured.0 = INTEGER: 0
AZCOM-LTE-OAM-STATS-MIB::azUEsReleased.0 = INTEGER: 1
AZCOM-LTE-OAM-STATS-MIB::azNumRach.0 = INTEGER: 3
AZCOM-LTE-OAM-STATS-MIB::azNumMsg2.0 = INTEGER: 3
AZCOM-LTE-OAM-STATS-MIB::azNumMsg3.0 = INTEGER: 1
AZCOM-LTE-OAM-STATS-MIB::azNumPdcpInactivityTimer.0 = INTEGER: 1
AZCOM-LTE-OAM-STATS-MIB::azNumRLFatMac.0 = INTEGER: 0
AZCOM-LTE-OAM-STATS-MIB::azNumOfReleaseTriggeredatMac.0 = INTEGER: 0
AZCOM-LTE-OAM-STATS-MIB::azUlSnrPucchUeRelease.0 = INTEGER: 0
AZCOM-LTE-OAM-STATS-MIB::azUlSnrPuschUeRelease.0 = INTEGER: 0
```

```
AZCOM-LTE-OAM-STATS-MIB::azDlCrcUeRelease.0 = INTEGER: 0
AZCOM-LTE-OAM-STATS-MIB::azRlcNumRlf.0 = INTEGER: 0
AZCOM-LTE-OAM-STATS-MIB::azDspMemHeapIdentity.0.0 = Gauge32: 0
AZCOM-LTE-OAM-STATS-MIB::azDspMemHeapIdentity.1.0 = Gauge32: 0
AZCOM-LTE-OAM-STATS-MIB::azDspMemHeapIdentity.1.1 = Gauge32: 1
AZCOM-LTE-OAM-STATS-MIB::azDspMemTotalSize.0.0 = Gauge32: 16384
AZCOM-LTE-OAM-STATS-MIB::azDspMemTotalSize.1.0 = Gauge32: 16384
AZCOM-LTE-OAM-STATS-MIB::azDspMemTotalSize.1.1 = Gauge32: 16384
AZCOM-LTE-OAM-STATS-MIB::azDspMemTotalFreeSize.0.0 = Gauge32: 16384
AZCOM-LTE-OAM-STATS-MIB::azDspMemTotalFreeSize.1.0 = Gauge32: 16384
AZCOM-LTE-OAM-STATS-MIB::azDspMemTotalFreeSize.1.1 = Gauge32: 16384
AZCOM-LTE-OAM-STATS-MIB::azDspMemLargestFreeSize.0.0 = Gauge32: 1638
AZCOM-LTE-OAM-STATS-MIB::azDspMemLargestFreeSize.1.0 = Gauge32: 3276
AZCOM-LTE-OAM-STATS-MIB::azDspMemLargestFreeSize.1.1 = Gauge32: 3276
AZCOM-LTE-OAM-STATS-MIB::azDspMemMaxAllocatedBlocks.0.0 = Gauge32: 0
AZCOM-LTE-OAM-STATS-MIB::azDspMemMaxAllocatedBlocks.1.0 = Gauge32: 0
AZCOM-LTE-OAM-STATS-MIB::azDspMemMaxAllocatedBlocks.1.1 = Gauge32: 0
AZCOM-LTE-OAM-STATS-MIB::azDspMemNumAllocatedBlocks.0.0 = Gauge32: 0
AZCOM-LTE-OAM-STATS-MIB::azDspMemNumAllocatedBlocks.1.0 = Gauge32: 0
AZCOM-LTE-OAM-STATS-MIB::azDspMemNumAllocatedBlocks.1.1 = Gauge32: 0
AZCOM-LTE-OAM-STATS-MIB::azDspPktHeapName.0.0 = STRING: "pktHeap_000"
AZCOM-LTE-OAM-STATS-MIB::azDspPktHeapName.0.1 = STRING: "pktHeap_001"
AZCOM-LTE-OAM-STATS-MIB::azDspPktHeapName.1.0 = STRING: "pktHeap_000"
AZCOM-LTE-OAM-STATS-MIB::azDspPktNumFreeDataPackets.0.0 = Gauge32: 0
AZCOM-LTE-OAM-STATS-MIB::azDspPktNumFreeDataPackets.0.1 = Gauge32: 0
AZCOM-LTE-OAM-STATS-MIB::azDspPktNumFreeDataPackets.1.0 = Gauge32: 0
AZCOM-LTE-OAM-STATS-MIB::azDspPktNumZeroBufferPackets.0.0 = Gauge32: 1
AZCOM-LTE-OAM-STATS-MIB::azDspPktNumZeroBufferPackets.0.1 = Gauge32: 1
AZCOM-LTE-OAM-STATS-MIB::azDspPktNumZeroBufferPackets.1.0 = Gauge32: 1
AZCOM-LTE-OAM-STATS-MIB::azDspPktNumPacketsInGarbage.0.0 = Gauge32: 0
AZCOM-LTE-OAM-STATS-MIB::azDspPktNumPacketsInGarbage.0.1 = Gauge32: 0
AZCOM-LTE-OAM-STATS-MIB::azDspPktNumPacketsInGarbage.1.0 = Gauge32: 0
AZCOM-LTE-OAM-STATS-MIB::azDspPktDataBufThrStatus.0.0 = INTEGER: clear(0)
AZCOM-LTE-OAM-STATS-MIB::azDspPktDataBufThrStatus.0.1 = INTEGER: clear(0)
AZCOM-LTE-OAM-STATS-MIB::azDspPktDataBufThrStatus.1.0 = INTEGER: clear(0)
AZCOM-LTE-OAM-STATS-MIB::azDspPktDataBufStarvationCnt.0.0 = Gauge32: 1
AZCOM-LTE-OAM-STATS-MIB::azDspPktDataBufStarvationCnt.0.1 = Gauge32: 1
AZCOM-LTE-OAM-STATS-MIB::azDspPktDataBufStarvationCnt.1.0 = Gauge32: 1
AZCOM-LTE-OAM-STATS-MIB::azDspPktZeroDataBufThrStatus.0.0 = INTEGER:
clear(0)
AZCOM-LTE-OAM-STATS-MIB::azDspPktZeroDataBufThrStatus.0.1 = INTEGER:
clear(0)
AZCOM-LTE-OAM-STATS-MIB::azDspPktZeroDataBufThrStatus.1.0 = INTEGER:
clear(0)
AZCOM-LTE-OAM-STATS-MIB::azDspPktZeroDataBufStarvationCnt.0.0 = Gauge32:
3
AZCOM-LTE-OAM-STATS-MIB::azDspPktZeroDataBufStarvationCnt.0.1 = Gauge32:
3
AZCOM-LTE-OAM-STATS-MIB::azDspPktZeroDataBufStarvationCnt.1.0 = Gauge32:
3
AZCOM-LTE-OAM-STATS-MIB::azMacDlTputTotal.0 = Gauge32: 0 kbps
AZCOM-LTE-OAM-STATS-MIB::azMacUlTputTotal.0 = Gauge32: 0 kbps
AZCOM-LTE-OAM-STATS-MIB::azDlBlerOfTB1Total.0 = Gauge32: 0 ratio/100
```

```
AZCOM-LTE-OAM-STATS-MIB::azDlBlerOfTB2Total.0 = Gauge32: 0 ratio/100
AZCOM-LTE-OAM-STATS-MIB::azUlBlerTotal.0 = Gauge32: 0 ratio/100
AZCOM-LTE-OAM-STATS-MIB::azRlcDlTputTotal.0 = Gauge32: 0 kbps
AZCOM-LTE-OAM-STATS-MIB::azRlcUlTputTotal.0 = Gauge32: 0 kbps
AZCOM-LTE-OAM-STATS-MIB::azRlcPktWholeNackRecvTotal.0 = Gauge32: 0
AZCOM-LTE-OAM-STATS-MIB::azRlcPktSegNackRecvTotal.0 = Gauge32: 0
AZCOM-LTE-OAM-STATS-MIB::azDlTputTotal.0 = Gauge32: 0 kbps
AZCOM-LTE-OAM-STATS-MIB::azUlTputTotal.0 = Gauge32: 0 kbps
AZCOM-LTE-OAM-STATS-MIB::azUlPktLostTotal.0 = Gauge32: 0
AZCOM-LTE-OAM-STATS-MIB::azDlPktLostTotal.0 = Gauge32: 0
AZCOM-LTE-OAM-STATS-MIB::azSgwUlTputTotal.0 = Gauge32: 0 kbps
AZCOM-LTE-OAM-STATS-MIB::azSgwDlTputTotal.0 = Gauge32: 0 kbps
```

User can read protocol or DSP statistics separately, by querying:.

```
snmpwalk -v 3 -t 15   -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azMACStatisticsTable
snmpwalk -v 3 -t 15   -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azMACStatisticsAggregated
snmpwalk -v 3 -t 15   -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azRLCStatisticsTable
snmpwalk -v 3 -t 15   -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azRLCStatisticsAggregated
snmpwalk -v 3 -t 15   -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azPDCPStatisticsTable
snmpwalk -v 3 -t 15   -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azPDCPStatisticsAggregated
snmpwalk -v 3 -t 15   -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azS1UStatisticsTable
snmpwalk -v 3 -t 15   -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azS1UStatisticsAggregated
snmpwalk -v 3 -t 15   -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azRRCStatistics
snmpwalk -v 3 -t 15   -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azGlobalStatistics
snmpwalk -v 3 -t 15   -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azRRCRntiStatistics
snmpwalk -v 3 -t 15   -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azPHYStatistics
snmpwalk -v 3 -t 15   -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azDspMemoryStatisticsTable
snmpwalk -v 3 -t 15   -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azDspPacketStatisticsTable
```

Setting the refresh mode to manual will always refresh the stats:

```
snmpset -v 3 -t  15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azRefreshStats.0 manual
AZCOM-LTE-OAM-L2MEAS-MIB:: azRefreshStats.0 = INTEGER: manual(1)
```

Setting the refresh mode back to automatic (cached for 6 seconds):

```
snmpset -v 3 -t  15  -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azRefreshStats.0 auto
AZCOM-LTE-OAM-L2MEAS-MIB:: azRefreshStats.0 = INTEGER: auto(0)
```

4.4.28 PDCP User Inactivity Timer

This command is used to configure the user inactivity timer which is used by PDCP layer in L2 independently for each of the attached UEs. A single interval configuration is applied at node level. If inactivity happens for any of the attached UE the corresponding timer expires followed by the release procedure.
The parameter available to configure the timer is azNodeInactivityTimer.0. The valid value range is 1-3600 seconds.

The SNMP HMI doesn’t require the node to be explicitly locked by the user to perform any value change operation on the timer. The changes shall be applied on the fly.

1) set the azNodeInactivityTimer.0 to "600":

```
snmpset -v 3 –t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azNodeInactivityTimer.0 = 600
AZCOM-LTE-OAM-BBU-MIB::azNodeInactivityTimer.0 = INTEGER: 600
```

2) get the azNodeInactivityTimer .0 :

```
snmpget -v 3 –t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azNodeInactivityTimer.0
AZCOM-LTE-OAM-BBU-MIB::azNodeInactivityTimer.0 = INTEGER: 600
```

4.4.29 Cell Max UE Capacity

This is used to check whether the max UE capacity of a cell has been reached or not. It reports alarmed status when the number of active UEs matches the max number of supported UEs. It is cleared whenever a UE is detached from the system resulting in decrement of number of active UEs.

azCellMaxUEReached is read-only.

Example:

```
snmpget -v 3 –t 15 -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST azCellMaxUEReached.0
AZCOM-LTE-OAM-BBU-MIB::azCellMaxUEReached.0 = INTEGER: clear(0)
```

4.4.30 Standard MIB information

The Platform layer offers an implementation of standard SNMPv2 MIBS. Other standard MIBs (e.g. ALARM-MIB) are not provided by Platform and implemented explicitly, as described in other sections.

In particular, Platform support for *HOST-RESOURCES-MIB::host* provides interesting information about the status of memory heaps and the CPU load over the ARM cores.

Information about this MIB can be found on official RFC at https://tools.ietf.org/html/rfc2790. Similar documentation for other standard MIBs can be likewise easily found in the internet.

Note that similar information related to DSPs is non standard and implemented explicitly and documented in other sections about statistics.

Example:

identify the CPU devices

```
snmpwalk -v 3 -t 15   -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST hrDeviceType
HOST-RESOURCES-MIB::hrDeviceType.196608 = OID: HOST-RESOURCES-
TYPES::hrDeviceProcessor
HOST-RESOURCES-MIB::hrDeviceType.196609 = OID: HOST-RESOURCES-
TYPES::hrDeviceProcessor
HOST-RESOURCES-MIB::hrDeviceType.196610 = OID: HOST-RESOURCES-
TYPES::hrDeviceProcessor
HOST-RESOURCES-MIB::hrDeviceType.196611 = OID: HOST-RESOURCES-
TYPES::hrDeviceProcessor
HOST-RESOURCES-MIB::hrDeviceType.262145 = OID: HOST-RESOURCES-
TYPES::hrDeviceNetwork
HOST-RESOURCES-MIB::hrDeviceType.262146 = OID: HOST-RESOURCES-
TYPES::hrDeviceNetwork
HOST-RESOURCES-MIB::hrDeviceType.262147 = OID: HOST-RESOURCES-
TYPES::hrDeviceNetwork
HOST-RESOURCES-MIB::hrDeviceType.262148 = OID: HOST-RESOURCES-
TYPES::hrDeviceNetwork
HOST-RESOURCES-MIB::hrDeviceType.262149 = OID: HOST-RESOURCES-
TYPES::hrDeviceNetwork
HOST-RESOURCES-MIB::hrDeviceType.262150 = OID: HOST-RESOURCES-
TYPES::hrDeviceNetwork
HOST-RESOURCES-MIB::hrDeviceType.262151 = OID: HOST-RESOURCES-
TYPES::hrDeviceNetwork
HOST-RESOURCES-MIB::hrDeviceType.262152 = OID: HOST-RESOURCES-
TYPES::hrDeviceNetwork
HOST-RESOURCES-MIB::hrDeviceType.262153 = OID: HOST-RESOURCES-
TYPES::hrDeviceNetwork
HOST-RESOURCES-MIB::hrDeviceType.262154 = OID: HOST-RESOURCES-
TYPES::hrDeviceNetwork
HOST-RESOURCES-MIB::hrDeviceType.786432 = OID: HOST-RESOURCES-
TYPES::hrDeviceCoprocessor
```

Having found that device ID 196608 is of type hrDeviceProcessor, we can grep all information related to it, discovering for example that the core is running and CPU load is 19% (it is actually running the walk command and the grep):

```
snmpwalk -v 3 -t 15   -u $USER -a SHA -A "$PWD1" -x DES -X "$PWD2" -l
authPriv $HOST host | grep 196608
HOST-RESOURCES-MIB::hrDeviceIndex.196608 = INTEGER: 196608
HOST-RESOURCES-MIB::hrDeviceType.196608 = OID: HOST-RESOURCES-
TYPES::hrDeviceProcessor
HOST-RESOURCES-MIB::hrDeviceDescr.196608 = STRING:
HOST-RESOURCES-MIB::hrDeviceID.196608 = OID: SNMPv2-SMI::zeroDotZero
HOST-RESOURCES-MIB::hrDeviceStatus.196608 = INTEGER: running(2)
HOST-RESOURCES-MIB::hrProcessorFrwID.196608 = OID: SNMPv2-
SMI::zeroDotZero
HOST-RESOURCES-MIB::hrProcessorLoad.196608 = INTEGER: 19
HOST-RESOURCES-MIB::hrSWRunParameters.3514 = STRING: "196608"
```

4.4.31 Alarms

An alarm signifies an undesired condition of a network element for which an operator attention is required. It can represent:

- an error:
  unrecoverable fault generally requiring a full restart
- a failure:
  recoverable fault
- a system warning
  for example the exhaustion or degradation of some kind of resource not compromising minimal functionalities, but which will prevent the successful completion of some further scenario.

The operator is informed about the condition through the notification generated by the node.

The following alarms are currently supported:

1. SCTP connection down
2. X2 setup failure
3. X2 unknown neighbor
4. Max cell capacity reached

Detailed description of the alarms is as follows:

1. SCTP connection down
   triggered due to connection loss between eNB and MME. This alarm is cleared when the connection is re-established.
2. X2 setup failure
   X2 is the interface between two nodes, say eNB1 and eNB2. eNB1 initiates the procedure by sending the X2 SETUP REQUEST message to eNB2. If eNB2 cannot accept the setup it shall respond with an X2 SETUP FAILURE message with appropriate cause value and raises the alarm. This alarm is cleared when X2 setup is established between the two.
3. X2 unknown neighbor
   whenever a node receives a PDU from a cell which is not known to it, then it raises an alarm for x2 setup unknown neighbor.
4. Max cell capacity
   alarm is raised when the number of active UEs matches the max number of supported UEs. The alarm is cleared whenever a UE is detached from the system resulting in decrement of number of active UEs.

User will be informed about the alarm by:

1. SNMP trap
2. SNMP and Klish interfaces when queried
3. Log traces

Signaled alarm	Message (trap, traces, active alarm reports)
SCTP connection is down	sctp alarmed
SCTP connection is up	sctp clear
X2 setup failure occurred	x2ap setup failure alarmed
X2 setup failure is removed	x2ap setup failure clear
X2 unknown neighbor occurred	x2ap unknown neighbour alarmed
X2 unknown neighbor is removed	x2ap unknown neighbour clear
Max cell capacity is reached	max cell capacity alarmed
Max cell capacity condition relaxed	max cell capacity clear

Table 11 Alarm Messages

The user can query the read-only alarm model table, the alarms statistics table and the active alarm table.

4.4.31.1 alarmModel

It contains the models of the alarm states that could occur in the system. It is supported as read-only, therefore the last changed time of the models currently report a fixed 0 value.

Semantics of the parameters in alarmModel are defined as follows:

MIB Parameter	Explanation	Type
alarmModelLastChanged	Specifies the value of sysUpTime at the time of the last creation, deletion or modification of an entry in the alarmModelTable. In current implementation, value will be 0.	TimeTicks
alarmModelTable	See 4.4.31.2	

4.4.31.2 alarmModelTable

The alarmModelTable contains information about all the possible states that instances of any alarm can take. It is populated with all alarms that could be signaled in the system.

Semantics of the parameters in alarmModelTable are defined as follows:

MIB Parameter	Explanation	Type

alarmModelNotificationId	Identifier for the MIB object of the specific notification sent for this alarm's state transition. In current implementation, value will be 0.0.	ObjID
AlarmModelVarbindIndex	The index into the varbind listing of the notification indicated by alarmModelNotificationId, which helps signal that the given alarm has changed state. In current implementation, value will be 0	Unsigned integer
alarmModelVarbindValue	Defines the value that the varbind indicated by alarmModelVarbindIndex takes to indicate that the alarm has entered this state. In current implementation, value will be 0.	Signed integer
alarmModelDescription	Describes the alarm state in readable format.	String size 0..255
alarmModelSpecificPointer	This object must refer to the first accessible object in a corresponding row of the model definition in one model-specific MIB, or '0.0' if no model-specific MIB is supported. In current implementation, no model extensions are supported.	ObjID
alarmModelVarbindSubtree	This object is used in calculation of alarmActiveResourceId. In current implementation, value will always be 0.0.	ObjID
alarmModelResourcePrefix	The value of alarmActiveResourceId is computed by appending any indexes extracted in accordance with the description of alarmModelVarbindSubtree onto the value of this object. In current implementation, value will always be 0.0.	ObjID
alarmModelRowStatus	Control for creating and deleting entries, as defined in RFC2579. Entries may be modified while alarmModelRowStatus is active. Current implementation will always report pre-filled and immutable active rows, rejecting every setting to this object.	RowStatus

Table 12 alarm model table

Example:

```
root@AZB_NCOM_4300:~# swalk alarmModelTable
ALARM-MIB::alarmModelNotificationId."enb".1.1 = OID: SNMPv2-
SMI::zeroDotZero
ALARM-MIB::alarmModelNotificationId."enb".1.2 = OID: SNMPv2-
SMI::zeroDotZero
ALARM-MIB::alarmModelNotificationId."enb".2.1 = OID: SNMPv2-
SMI::zeroDotZero
ALARM-MIB::alarmModelNotificationId."enb".2.2 = OID: SNMPv2-
SMI::zeroDotZero
ALARM-MIB::alarmModelNotificationId."enb".3.1 = OID: SNMPv2-
SMI::zeroDotZero
ALARM-MIB::alarmModelNotificationId."enb".3.2 = OID: SNMPv2-
SMI::zeroDotZero
ALARM-MIB::alarmModelNotificationId."enb".4.1 = OID: SNMPv2-
SMI::zeroDotZero
ALARM-MIB::alarmModelNotificationId."enb".4.2 = OID: SNMPv2-
SMI::zeroDotZero
ALARM-MIB::alarmModelVarbindIndex."enb".1.1 = Gauge32: 0
ALARM-MIB::alarmModelVarbindIndex."enb".1.2 = Gauge32: 0
ALARM-MIB::alarmModelVarbindIndex."enb".2.1 = Gauge32: 0
ALARM-MIB::alarmModelVarbindIndex."enb".2.2 = Gauge32: 0
ALARM-MIB::alarmModelVarbindIndex."enb".3.1 = Gauge32: 0
ALARM-MIB::alarmModelVarbindIndex."enb".3.2 = Gauge32: 0
ALARM-MIB::alarmModelVarbindIndex."enb".4.1 = Gauge32: 0
ALARM-MIB::alarmModelVarbindIndex."enb".4.2 = Gauge32: 0
ALARM-MIB::alarmModelVarbindValue."enb".1.1 = INTEGER: 0
ALARM-MIB::alarmModelVarbindValue."enb".1.2 = INTEGER: 0
ALARM-MIB::alarmModelVarbindValue."enb".2.1 = INTEGER: 0
ALARM-MIB::alarmModelVarbindValue."enb".2.2 = INTEGER: 0
ALARM-MIB::alarmModelVarbindValue."enb".3.1 = INTEGER: 0
ALARM-MIB::alarmModelVarbindValue."enb".3.2 = INTEGER: 0
ALARM-MIB::alarmModelVarbindValue."enb".4.1 = INTEGER: 0
ALARM-MIB::alarmModelVarbindValue."enb".4.2 = INTEGER: 0
ALARM-MIB::alarmModelDescription."enb".1.1 = STRING: sctp clear
ALARM-MIB::alarmModelDescription."enb".1.2 = STRING: sctp alarmed
ALARM-MIB::alarmModelDescription."enb".2.1 = STRING: x2ap setup failure
clear
ALARM-MIB::alarmModelDescription."enb".2.2 = STRING: x2ap setup failure
alarmed
ALARM-MIB::alarmModelDescription."enb".3.1 = STRING: x2ap unknown
neighbour clear
ALARM-MIB::alarmModelDescription."enb".3.2 = STRING: x2ap unknown
neighbour alarmed
ALARM-MIB::alarmModelDescription."enb".4.1 = STRING: max cell capacity
clear
ALARM-MIB::alarmModelDescription."enb".4.2 = STRING: max cell capacity
alarmed
ALARM-MIB::alarmModelSpecificPointer."enb".1.1 = OID: SNMPv2-
SMI::zeroDotZero
ALARM-MIB::alarmModelSpecificPointer."enb".1.2 = OID: SNMPv2-
SMI::zeroDotZero
```

```
ALARM-MIB::alarmModelSpecificPointer."enb".2.1 = OID: SNMPv2-
SMI::zeroDotZero
ALARM-MIB::alarmModelSpecificPointer."enb".2.2 = OID: SNMPv2-
SMI::zeroDotZero
ALARM-MIB::alarmModelSpecificPointer."enb".3.1 = OID: SNMPv2-
SMI::zeroDotZero
ALARM-MIB::alarmModelSpecificPointer."enb".3.2 = OID: SNMPv2-
SMI::zeroDotZero
ALARM-MIB::alarmModelSpecificPointer."enb".4.1 = OID: SNMPv2-
SMI::zeroDotZero
ALARM-MIB::alarmModelSpecificPointer."enb".4.2 = OID: SNMPv2-
SMI::zeroDotZero
ALARM-MIB::alarmModelVarbindSubtree."enb".1.1 = OID: SNMPv2-
SMI::zeroDotZero
ALARM-MIB::alarmModelVarbindSubtree."enb".1.2 = OID: SNMPv2-
SMI::zeroDotZero
ALARM-MIB::alarmModelVarbindSubtree."enb".2.1 = OID: SNMPv2-
SMI::zeroDotZero
ALARM-MIB::alarmModelVarbindSubtree."enb".2.2 = OID: SNMPv2-
SMI::zeroDotZero
ALARM-MIB::alarmModelVarbindSubtree."enb".3.1 = OID: SNMPv2-
SMI::zeroDotZero
ALARM-MIB::alarmModelVarbindSubtree."enb".3.2 = OID: SNMPv2-
SMI::zeroDotZero
ALARM-MIB::alarmModelVarbindSubtree."enb".4.1 = OID: SNMPv2-
SMI::zeroDotZero
ALARM-MIB::alarmModelVarbindSubtree."enb".4.2 = OID: SNMPv2-
SMI::zeroDotZero
ALARM-MIB::alarmModelResourcePrefix."enb".1.1 = OID: SNMPv2-
SMI::zeroDotZero
ALARM-MIB::alarmModelResourcePrefix."enb".1.2 = OID: SNMPv2-
SMI::zeroDotZero
ALARM-MIB::alarmModelResourcePrefix."enb".2.1 = OID: SNMPv2-
SMI::zeroDotZero
ALARM-MIB::alarmModelResourcePrefix."enb".2.2 = OID: SNMPv2-
SMI::zeroDotZero
ALARM-MIB::alarmModelResourcePrefix."enb".3.1 = OID: SNMPv2-
SMI::zeroDotZero
ALARM-MIB::alarmModelResourcePrefix."enb".3.2 = OID: SNMPv2-
SMI::zeroDotZero
ALARM-MIB::alarmModelResourcePrefix."enb".4.1 = OID: SNMPv2-
SMI::zeroDotZero
ALARM-MIB::alarmModelResourcePrefix."enb".4.2 = OID: SNMPv2-
SMI::zeroDotZero
ALARM-MIB::alarmModelRowStatus."enb".1.1 = INTEGER: active(1)
ALARM-MIB::alarmModelRowStatus."enb".1.2 = INTEGER: active(1)
ALARM-MIB::alarmModelRowStatus."enb".2.1 = INTEGER: active(1)
ALARM-MIB::alarmModelRowStatus."enb".2.2 = INTEGER: active(1)
ALARM-MIB::alarmModelRowStatus."enb".3.1 = INTEGER: active(1)
ALARM-MIB::alarmModelRowStatus."enb".3.2 = INTEGER: active(1)
ALARM-MIB::alarmModelRowStatus."enb".4.1 = INTEGER: active(1)
ALARM-MIB::alarmModelRowStatus."enb".4.2 = INTEGER: active(1)
```

4.4.31.3 alarmActive

The alarmActive context contains a list of alarms that are currently occurring on a system along with their last changed time.

Semantics of the parameters in alarmActive are defined as follows:

MIB Parameter	Explanation	Type
alarmActiveLastChanged	Specifies the value of sysUpTime at the time of the last creation or deletion of an entry in the alarmActiveTable, or 0 if no change occurred since a restart.	Timeticks
alarmActiveTable	See 4.4.31.4	

4.4.31.4 alarmActiveTable

The alarmActiveTable contains a list of alarms that are currently occurring on a system. It is intended that this table be queried to determine which alarms are currently active on the system.

Semantics of the parameters in alarmActiveTable are defined as follows:

MIB Parameter	Explanation	Type
alarmActiveEngineID	The identification of the SNMP engine at which the alarm originated.	String Size: 0 | 5..32
alarmActiveEngineAddressType	This object indicates the type of address which is stored in the alarmActiveEngineAddress object	Enum (unknown, ipv4, ipv6, ipv4z, ipv6z, dns)
alarmActiveEngineAddress	The address of the SNMP engine on which the alarm is occurring.	String size: 0..255
alarmActiveContextName	The name of the SNMP MIB context from which the alarm came. In current implementation, value will be an empty string.	String size: 0..32
alarmActiveVariables	It specifies the number of variables in alarmActiveVariableTable for this alarm. In current implementation, this will always be 0.	Unsigned integer
alarmActiveNotificationID	Identifier of the alarm state transition that is occurring. In current implementation, value will be 0.0.	ObjID

alarmActiveResourceId	Identifies the resource under alarm. In current implementation, value will be 0.0	ObjID
alarmActiveDescription	Provides a textual description of the active alarm.	String size: 0..255
alarmActiveLogPointer	A pointer to the corresponding row in a notification logging MIB where the state change notification for this active alarm is logged. In current implementation, value will be 0.0.	ObjID
alarmActiveModelPointer	A pointer to the corresponding row in the alarmModelTable for this active alarm. This points not only to the alarm model being instantiated, but also to the specific alarm state that is active.	ObjID
alarmActiveSpecificPointer	When a model-specific Alarm MIB is supported, then this object is the instance pointer to the specific model-specific active alarm list. In current implementation, value will be 0.0.	ObjID

Table 13 active alarm table

Example:

Assuming that x2ap unknown neighbor and x2ap setup failure alarms are raised currently.

```
root@AZB_NCOM_4300:~# swalk alarmActiveTable
ALARM-MIB::alarmActiveEngineID."enb"."1970-02-27,17:55:34.3,+00:00".1 =
Hex-STRING: 80 00 1F 88 80 59 9A E7 5C FD 45 BD 58 00 00 00 00
ALARM-MIB::alarmActiveEngineID."enb"."1970-02-27,17:58:19.6,+00:00".2 =
Hex-STRING: 80 00 1F 88 80 59 9A E7 5C FD 45 BD 58 00 00 00 00
ALARM-MIB::alarmActiveEngineAddressType."enb"."1970-02-
27,17:55:34.3,+00:00".1 = INTEGER: ipv4(1)
ALARM-MIB::alarmActiveEngineAddressType."enb"."1970-02-
27,17:58:19.6,+00:00".2 = INTEGER: ipv4(1)
ALARM-MIB::alarmActiveEngineAddress."enb"."1970-02-
27,17:55:34.3,+00:00".1 = STRING: "192.168.3.1"
ALARM-MIB::alarmActiveEngineAddress."enb"."1970-02-
27,17:58:19.6,+00:00".2 = STRING: "192.168.3.1"
ALARM-MIB::alarmActiveContextName."enb"."1970-02-27,17:55:34.3,+00:00".1
= STRING: “”
ALARM-MIB::alarmActiveContextName."enb"."1970-02-27,17:58:19.6,+00:00".2
= STRING: “”
ALARM-MIB::alarmActiveVariables."enb"."1970-02-27,17:55:34.3,+00:00".1 =
Gauge32: 0
ALARM-MIB::alarmActiveVariables."enb"."1970-02-27,17:58:19.6,+00:00".2 =
Gauge32: 0
```

```
ALARM-MIB::alarmActiveNotificationID."enb"."1970-02-
27,17:55:34.3,+00:00".1 = OID: SNMPv2-SMI::zeroDotZero
ALARM-MIB::alarmActiveNotificationID."enb"."1970-02-
27,17:58:19.6,+00:00".2 = OID: SNMPv2-SMI::zeroDotZero
ALARM-MIB::alarmActiveResourceId."enb"."1970-02-27,17:55:34.3,+00:00".1 =
OID: SNMPv2-SMI::zeroDotZero
ALARM-MIB::alarmActiveResourceId."enb"."1970-02-27,17:58:19.6,+00:00".2 =
OID: SNMPv2-SMI::zeroDotZero
ALARM-MIB::alarmActiveDescription."enb"."1970-02-27,17:55:34.3,+00:00".1
= STRING: x2ap unknown neighbour alarmed
ALARM-MIB::alarmActiveDescription."enb"."1970-02-27,17:58:19.6,+00:00".2
= STRING: x2ap setup failure alarmed
ALARM-MIB::alarmActiveLogPointer."enb"."1970-02-27,17:55:34.3,+00:00".1 =
OID: SNMPv2-SMI::zeroDotZero
ALARM-MIB::alarmActiveLogPointer."enb"."1970-02-27,17:58:19.6,+00:00".2 =
OID: SNMPv2-SMI::zeroDotZero
ALARM-MIB::alarmActiveModelPointer."enb"."1970-02-27,17:55:34.3,+00:00".1
= OID: ALARM-MIB::alarmModelIndex.”enb”.3.2
ALARM-MIB::alarmActiveModelPointer."enb"."1970-02-27,17:58:19.6,+00:00".2
= OID: ALARM-MIB::alarmModelIndex.”enb”.2.2
ALARM-MIB::alarmActiveSpecificPointer."enb"."1970-02-
27,17:55:34.3,+00:00".1 = OID: SNMPv2-SMI::zeroDotZero
ALARM-MIB::alarmActiveSpecificPointer."enb"."1970-02-
27,17:58:19.6,+00:00".2 = OID: SNMPv2-SMI::zeroDotZero
```

4.4.31.5 alarmActiveStatsTable

The alarmActiveStatsTable represents the alarm statistics information. It contains the current and total raised alarm counts in addition to the time of the last alarm raise and alarm clears per named alarm list.

Semantics of the parameters in alarmActiveStatsTable are defined as follows:

MIB Parameter	Explanation	Type
alarmActiveStatsActiveCurrent	Represents total number of currently active alarms on the system.	Gauge
alarmActiveStatsActives	Represents total number of active alarms since system restarted.	Gauge
alarmActiveStatsLastRaise	Represents value of sysUpTime at the time of the last alarm raise for this alarm list.	TimeTicks
alarmActiveStatsLastClear	Represents value of sysUpTime at the time of the last alarm clear for this alarm list.	TimeTicks

Table 14 active stats table

Example:

```
root@AZB_NCOM_4300:~# swalk alarmActiveStatsTable
ALARM-MIB::alarmActiveStatsActiveCurrent."enb" = Gauge32: 3
```

```
ALARM-MIB::alarmActiveStatsActives."enb" = Gauge32: 7
ALARM-MIB::alarmActiveStatsLastRaise."enb" = Timeticks: (708) 0:00:07.08
ALARM-MIB::alarmActiveStatsLastClear."enb" = Timeticks: (628) 0:00:06.28
root@AZB_NC0M_4300:~#

root@AZB_NC0M_4300:~# sget alarmActiveStatsActiveCurrent.\"enb\"
ALARM-MIB::alarmActiveStatsActiveCurrent."enb" = Gauge32: 3
root@AZB_NC0M_4300:~#
```

5 Command Line Interface

The AZCOM eNodeB OAM module permits the operations, administration and maintenance of the node through a user friendly CLI interface based on Klish. Klish simplifies the user interface by providing a set of command line operations to configure and manage the eNodeB while guiding the user into the available options through tab auto-completion.

The CLI utility supports a similar set as compared to SNMP. Missing parameters are provided by the platform's CLI commands within the same shell; please refer to [6] AZS-XXX-0200 Platform Command Line Interface for details.

Parameters can be read individually or in collective manner (logical, useful aggregated walks of parameters is provided under show commands).

Please refer to the section dedicated to SNMP for a deeper explanation of each parameter.

5.1 CLI views

Three views are defined:

- An info view where an unprivileged user can run show commands
- An enable view where a site installator can access to network configuration commands. The view can be accessed from the info view via the enable command
- A configuration terminal view where all the configuration options are available. The view can be accessed from the enable view cia the configuration terminal command. Access to previous views' commands can be achieved through the do command prefix followed by the intended command (e.g. do show node).

The system comes with a unprivileged and a full-privileged users, Refer to [6] AZS-XXX-0200 Platform Command Line Interface for further details.

5.2 Commands reference

This section describes how to configure eNodeB system parameters. These steps allow the user to "prepare" new configuration values.

Depending on the option type, additional action may be required to activate the updated configuration. There are three possible cases:

a) Setting applied immediately.

b) eNodeB soft restart required at unlock. This operation takes about 1500 milliseconds.

c) eNodeB restart required at unlock. This operation takes about 25 seconds.

The *permissions* column in tables reported in this section not only define the read/write capabilities over each parameter, but also the need for a container object lock and whether a restart will be automatically triggered at unlock. Please refer to the legend here below:

Permission	Meaning
RO	parameter is read-only
RW	parameter can be written, no lock required. The setting is applied at runtime
RW-Lock	parameter can be written, and it requires a lock of the container object. Unlock will not trigger a restart
RW-Restart	parameter can be written, and it requires a lock of the container object. Unlock will trigger a restart

Table 15 Legend of parameters permissions

5.2.1 eNodeB administration

To retrieve node administration level elements.

The parameters are listed below:

Parameter	Description	Permissions R-read, W-write	Valid Values
ast	admin status of node	RW	lock, unlock
opst	operational status of node	RW-Lock	idle, startInprogress, active, stopInprogress, rebootInprogress, stopFail, startFail, invalid,start, halt, restart
alarm	alarm status of node	RO	clear, stopFail, startFail, willReset, willReboot
inactivity_timer	pdcp user inactivity timer	RW	1-3600 (in seconds)
name	specifies the node name	RW-Lock	1..512 octets
type	specifies the type of node	RO	macro, small, home
mcc	specifies the mobile country code	RW-Restart	3 octets
mnc	specifies the mobile network code	RW-Restart	2 or 3 octects
tac	specifies the tracking area code	RW-Restart	0..65535
version	running application software version	RO	1..50 octets
rel_id	identify the exact build that the customer is asking support for	RO	1..50 octets

nib_mode	specifies nib mode	RW-Restart	none, bbu, som
configuration	save a configuration from memory to flash	W-Restart	apply

Table 16. Node

* The administrative status change also supports handy shortcuts like ‘node lock’ and ‘node unlock’ along with ‘node ast lock’ and ‘node ast unlock’

5.2.1.1 Show node contents

show node

Operation	Successful Outcome
show node	node ast : lock(0) opst : active(2) alarm : willReboot(4) name : "azcom" type : small(1) mcc : "219" mnc : "01" tac : 1 version : "20161129_LTE_02_ENB_INT_vXX.YY.ZZ" rel_id : "bb0b5aafa871fe379287f114801bbbd5" nib_mode : none(0)

5.2.1.2 Display node parameters

show node paramName

paramName : any parameter present in Table 16 with read permission.

Operation	Successful Outcome
show node ast	node ast : lock(0)

5.2.1.3 Set node parameter

node paramName value

paramName - any parameter present in the above mentioned list with write permission.

value - value of the parameter to be set.

Operation	
node ast unlock	

5.2.2 Sync_src

Azcom eNodeB can use three kind of sources for internal syncing: a gps source, ieee1588 protocol, or an external 10Mhz sync source connected to the board.

Parameter	Description	Permissions R-read, W-write	Valid Values
type	type of source for internal syncing	RW-Restart	extclock, gps, ieee1588
status	specifies the clock lock status	RO	unlocked, locked

Table 17. Sync_src

5.2.2.1 Display sync_src contents

show sync_src

Operation	Successful Outcome
show sync_src	sync_src type : extclock(1) status : locked(1)

5.2.2.2 Display sync_src parameter

show sync_src paramName

paramName : any parameter present in the above mentioned list with read permission.

Operation	Successful Outcome
show sync_src type	sync_src type : extclock(1)

5.2.2.3 Set sync_src parameter

sync_src paramName value

paramName - any parameter present in the above mentioned list with write permission.

value - value of the parameter to be set.

Operation	
sync_src type ieee1588	

5.2.3 Cell(s)

Cell specific parameters are grouped together. They are listed below:

Parameter	**Description**	**Permissions R-read, W-write**	**Valid Values**
ast	lock/unlock admin state of cell	RW	lock, unlock
opst	enables or disables a cell	RW-Lock	disabled, enabled, invalid
fst	reports the runtime cell status	RO	notoperational, operational
dup_mode	specifies the cell duplex mode	RO	fdd, tdd
id	specifies the cell identity	RW-Restart	0..268435455
tdd_sf	specifies the tdd sub frame	RW-Restart	sa0, sa1, sa2, sa3, sa4, sa5, sa6
tdd_ssf	specifies the tdd special sub frame	RW-Restart	ssp3, ssp4, ssp7, ssp8
pci	specifies the physical cell id	RW-Restart	0..503
rf	specifies the type of the radio element used by the cell	RW-Restart	manual, ngscbp, bh, lamarr
tx_bw	specifies the transmission bandwidth in MHz	RW-Restart	1.4, 3, 5, 10, 15, 20
tx_earfcn	specifies the transmission earfcn	RW-Restart	0..65535
rx_earfcn	specifies the reception earfcn	RO	0..65535
dl_freq	specifies the downlink center frequency in KHz.	RO	integer
ul_freq	specifies the uplink center frequency in KHz	RO	integer
band_id	specifies the band identifier	RO	1..44

sib	specifies the UE waiting timer values. See Table 19		
tm	used to configure single antenna (SISO) or dual antenna transmission mode	RW-Restart	siso, diversity, tm3, tm4
num_ue	number of UEs attached to a specific cell	RO	0..255
max_tx_pwr	maximum transmission power	RW, RW-Lock or RW-Restart (depending from RF type)	-10..50
aif_link	defines the AIF link on which the RRH is connected	RW-Restart	0..1
refresh_stats	specifies the trigger type for collection of statistics	RW	auto, manual
max_ue_reached	specifies whether max UE capacity has been reached or not.	RO	clear, alarmed

Table 18. Cell

* The administrative status change also supports handy shortcuts like ‘cell 0 lock’ and ‘cell 0 unlock’ along with ‘cell 0 ast lock’ and ‘cell 0 ast unlock’

Parameter	**Description**	**Permissions R-read, W-write**	**Valid Values (ms)**
t300	UE waiting time for RRC connection request message response	RW-Restart	600 1000 1500 2000
t301	UE waiting time before going to idle mode once RRC connection re-establishment request message is sent	RW-Restart	100 200 300 400 600 1000 1500 2000
t310	started after receiving	RW-Restart	0

	a defined number of out-of-sync indicators		50 100 200 500 1000 2000
t311	UE waiting time before going to idle mode if connection re-establishment procedure has started and it is unable to locate a suitable cell indicators	RW-Restart	1000 3000 5000 10000 15000 20000 30000

Table 19. Timer

5.2.3.1 Display specific cell parameters

show cell cell_no

cell_no - cell entry.

Operation	Successful Outcome
show cell 0	cell 0 ast : lock(0) opst : enabled(1) dup_mode : fdd(0) ...

cell_no is within range but instance is not available

Operation	Successful Outcome
show cell 1	>

5.2.3.2 Display cell parameter

show cell cell_no paramName

cell_no - cell entry.

paramName - any parameter from above mentioned list with read permission.

Operation	Successful Outcome
show cell 0 ast	cell 0 ast : lock(0)

5.2.3.3 Set cell parameter

cell cell_no paramName value

cell_no - cell entry.

paramName - any parameter from above mentioned list with write permission.

value - value of the parameter to be set.

Operation	
cell 0 pci 270	

5.2.4 RF module

These commands are specific for the Azcom RF/RRH. Similar parameters shall be provided when the node is using a different RRH, and may behave differently.

Note: these parameters are not currently available on the AZS-XXX-0200 Small Cell family.

Parameter	Description	Permissions W-write,R-read	Valid Values
opst	operational status of rf on board	RO	disabled, not operational, enabled
txpath	See Table 21	RO	
rxpath	See Table 22	RO	
antenna	See Table 23	RO	

Table 20. RF

Parameter	Description	Permissions W-write,R-read	Valid Values
power	transmission power	RO	integer

Table 21. Tx Path

Parameter	Description	Permissions W-write,R-read	Valid Values
Power	reception power	RO	integer

Table 22. Rx Path

Parameter	Description	Permissions W-write,R-read	Valid Values
temperature	Rf antenna temperature	RO	0..65535

Table 23. Antenna

5.2.4.1 Display RF module

show ngrf

Operation	Successful Outcome
show ngrf	ngrf opst :disabled(0) txpath 0 power : 5 rxpath 0 power : 6 antenna 0 temperature : 20 antenna 1 temperature : 20

5.2.4.2 Display RF module parameter

show ngrf paramName

paramName – any parameter present in the above mentioned list with read permission.

Operation	Successful Outcome
show ngrf opst	ngrf opst : disabled(0)

5.2.5 Log Server configuration

Azcom eNodeB can efficiently provide traces to a server which opens an UDP port. This set of parameters is able to define the server details and to trim the desired log severity levels.

Disclaimer: Depending on the severity level and the number of enabled mask flags, the logging system may introduce a computational weight which may compromise the normal functionality of the system.

Parameter	Description	Permissions W-write, R-read	Valid Values

ip	log server remote ip	RW-Lock	a.b.c.d
			a = 0..255
			b = 0..255
			c = 0..255
			d = 0..255

port	log server port number	RW-Lock	0..65535
severity	shows severity for Phy, L2,L3	RO	critical, warning, error, info, debug
mask	shows masks for Phy, L2,L3	RO	String[16]
phy	See Table 25		
l2	See Table 25		
l3	See Table 25		
oam	See Table 25		
events	See Table 26		
addr	See Table 27		

Table 24. Log Server

Parameter	Description	Permissions W-write, R-read	Valid Values
severity	different severity levels can be defined. The setting is persistent and therefore it will be in use at the next reboot as well.	RW-Lock	critical, warning, error, info, debug
mask	Hex mask for the protocol level.	RW-Lock	String[16]/ String[8]*

Table 25. Phy/L2/L3/Oam

* String[8] is only for PHY mask

Parameter	Description	Permissions W-write, R-read	Valid Values
severity	Different severity levels can be defined for snmp trap events.	RW-Lock	Critical, warning, error, info, debug

Table 26 Events

Parameter	Description	Permissions W-write, R-read	Valid Values

ip	network information needed by the log server setup are retrieved from the s1u interface .	RW-Lock	a.b.c.d
			a = 0..255
			b = 0..255
			c = 0..255
			d = 0..255

port	log server port number	RW-Lock	0..65535

Table 27. Addr

5.2.5.1 Display log server configuration parameter

show logger paramName

paramName - any parameter mentioned in the above list with read permission.

Operation	Successful Outcome
show logger ip	logger ip : 192.168.3.40

5.2.5.2 Set log server configuration parameter

logger paramName value

paramName - any parameter mentioned in the above list with write permission.

value - value of the specific parameter.

Operation	
logger ip 192.168.3.40	

5.2.6 S1 interfaces

IPs and port addresses for S1AP and S1U interfaces can be defined for both the local node and the peer elements. They are grouped into the s1Interface table.

Parameter	Description	Permissions W-write, R-read	Valid Values
sctp_status	Specifies the status of SCTP connection between eNB and MME.	RO	operational, notoperational
s1ap	See Table 29		
s1u	See Table 30		

Table 28. S1 Interface

Parameter	Description	Permissions W-write,R-read	Valid Values
interface	interface used to connect S1AP from eNB to EPC	RW-Restart	eth0, eth1, eth2, eth3

node ip	node IP address for S1AP	R	a.b.c.d
			a = 0..255
			b = 0..255
			c = 0..255
			d = 0..255

node port	node IP port for S1AP	RW-Restart	0..65535

mme ip	IP address of MME	RW-Restart	a.b.c.d
			a = 0..255
			b = 0..255
			c = 0..255
			d = 0..255

mme port	IP port of MME	RW-Restart	0..65535
vlan_id	Virtual lan id when NIB on SOM is used.	RW-Restart	1..4094

Table 29. S1AP

Parameter	Description	Permissions W-write,R-read	Valid Values
interface	interface used to connect S1U from eNB to EPC	RW-Restart	eth0, eth1, eth2, eth3
node ip	node IP address used for S1U	R (RW-Restart in ARM NIB mode)	a.b.c.d a = 0..255

			b = 0..255
			c = 0..255
			d = 0..255

node port	Node IP port for S1U	RW-Restart	0..65535

gw ip	IP address of GW	RW-Restart	a.b.c.d
			a = 0..255
			b = 0..255
			c = 0..255
			d = 0..255

gw port	IP port of GW (Not used if azNIBMode.0 is enabled)	RW-Restart	0..65535
vlan_id	Virtual lan id when NIB on SOM is used.	RW-Restart	1..4094

Table 30. S1U

5.2.6.1 Display s1 interface

show s1

Operation	**Successful Outcome**
show s1	s1 sctp_status : operational(1) s1ap node : 192.168.3.1:36412 mme : 192.168.3.10:36412 interface : eth0(0) vlan_id : 1 s1u node : 192.168.3.1:2152 gw : 192.168.3.10:4000 interface : eth0(0) vlan_id : 1

5.2.6.2 Display s1 parameter

show s1 paramName

paramName - any parameter with read permission from Table 28Table 28.

Operation	
show s1 sctp_status	s1 sctp_status : operational(0)

5.2.6.3 Display s1ap parameter

show s1ap paramName

paramName - any parameter with read permission from Table 29.

Operation	Successful Outcome
show s1ap interface	s1ap interface : eth0(0)

5.2.6.4 Display s1u parameter

show s1u paramName

paramName - any parameter with read permission from Table 30.

Operation	Successful Outcome
show s1u interface	s1u interface : eth0(0)

5.2.6.5 Set s1ap parameter

s1ap paramName value

paramName - any parameter with write permission from Table 29.

value - value to be set for the parameter.

Operation	
s1ap interface eth1	

5.2.6.6 Set s1u parameter

s1u paramName value

paramName - any parameter with write permission from Table 30.

value - value to be set for the parameter.

Operation	
s1u interface eth0	

5.2.7 X2 interfaces

IP and port addresses for X2AP and X2U interfaces can be defined for the local node. They are grouped into the x2 Interface table.

Parameter	Description	Permissions R-read, W-write	Valid Values

ip	X2 interface IP address for eNodeB	RW-Restart	a.b.c.d
			a = 0..255
			b = 0..255
			c = 0..255
			d = 0..255

setup_status	Specifies the status of X2 connection between two eNBs.	RO	clear, failed
unknown_neighbor_status	Specifies whether the candidate eNB is a known X2 neighbor or not.	RO	clear, alarmed

Table 31. X2 Interface

Parameter	Description	Permissions R-read, W-write	Valid Values
port	X2ap interface port for eNodeB	RW-Restart	0..65535

Table 32. X2AP

Parameter	Description	Permissions R-read, W-write	Valid Values
port	X2u interface port for eNodeB	RW-Restart	0..65535

Table 33. X2U

5.2.7.1 Display x2 interface

Operation	Successful Outcome
show x2	x2ap : 192.168.3.1:36000 x2u : 192.168.3.1:36100 x2_setup_status : clear(0) x2_unknown_neighbor_status : clear(0)

5.2.7.2 Display x2 interface parameter

show x2 paramName

paramName - any parameter with read permission from Table 31.

Operation	Successful Outcome
show x2 ip	x2 ip : 192.168.3.1

5.2.7.3 Set content of x2 interface

x2 paramName value

paramName-any parameter with write permission from Table 31.

value - value of the parameter

Operation	
x2 ip 192.168.3.2	

5.2.8 CPRI link management

Parameter	Description	Permissions R-read, W-write	Valid Values
opst	reports the operational status of the CPRI link	RO	disabled, enabled, invalid
pimin	defines the CPRI receive window	RW-Restart	0..4194303
dbm_offset	RF factor used for the conversion from digital dB bit to antenna RX dBm power: dBm = dB bit + DBM_OFFSET	RW-Restart	-2147483648.. 2147483647
hfsync_delay	Waiting time before the enabling of RX link	RW-Restart	0..4294967295
axc_offset	Antenna Carrier Offset programmed in the number of four samples, relative to the link (PHY) frame boundary	RW-Restart	0..33554431
tcp_ip	The TCP/IP over CPRI is used to enable the TCP/IP protocol to communicate over CPRI through the channel connected to the RRH.	RW-Restart	Disable, enable

Table 34. CPRI

5.2.8.1 Display CPRI link parameter

show cpri cpri_no

cpri_no - cpri table entry.

Operation	Successful Outcome
show cpri 0	cpri0 opst : disabled(0) pimin : 380 dbm_offset : -125 hfsync_delay : 5 axc_offset : 1 tcp_ip : disabled(0)

5.2.8.2 Set CPRI link parameter

cpri cpri_no paramName value

cpri_no - cpri table entry.

paramName - any parameter present in above mentioned list with write permission.

value - parameter value.

Operation	
cpri 0 pimin 380	

5.2.9 GPS information module

Parameter	Description	Permissions R-read, W-write	Valid Values
fixtype	fixtype, represents the NMEA mode	RO	undef, none, twoDimension, threeDimension
timestamp	timeStamp, shows timestamp from GPS device	RO	0..255 octets
lat	shows Gps latitude in	RO	0..255 octets

	degrees where +/- signifies North/South. This value is set if azGpsFixtype has a value of 2 (two dimensions) or 3 (three dimensions)		
lon	shows Gps longitude in degrees where +/- signifies East/West. This value is set if azGpsFixtype has a value of 2 (two dimensions) or 3 (three dimensions)	RO	0..255 octets
alt	shows altitude in meters. This value is set if azGpsFixtype has a value of 3 (three dimensions)	RO	integer
heading	shows the course over ground in degrees from true north	RO	integer
hvelocity	shows speed over ground in meters per second	RO	0..255 octets
vvelocity	shows the limb (positive) or sink (negative) rate in meters per second	RO	0..255 octets
satellite_count	no. of satellites	RO	positive integer
satellite	Table 36	RO	

Table 35. GPS

Parameter	**Description**	**Permissions R-read, W-write**	**Valid Values**
elevation	elevation in degrees	RO	integer
azimuth	azimuth of the current satellite in context referenced by the	RO	integer

	Satellite Info Index. Azimuth is the degree of rotation of the satellites dish on its vertical plane		
strength	signal to Noise Ratio(SNR) of received GPS signal in dB.	RO	integer
used	satellite in line of sight to the GPS device and used in calculating the GPS location	RO	False, true

Table 36. Satellite

5.2.9.1 Display GPS module

show gps

Operation	Successful Outcome
show gps	Gps fixtype : threeDimension timestamp : 2010-09-08T13:33:06.095Z lat : 40.035093060 degrees lon : -75.519748733 degrees heading : 99 degrees hvelocity : 0.123 m/s vvelocity : 0.001 m/s satellite_count : 2 satellite 0 elevation : 55 degrees satellite 0 azimuth : 42 degrees satellite 1 elevation : 295 degrees satellite 1 azimuth : 263 degrees

5.2.9.2 Display content of GPS module

show gps paramName

Operation	Successful Outcome
show gps fixtype	Gps fixtype : none(1)

5.2.10 L2 Measurements

The following L2 Measurements are available for a single cell object via the *show measurements cell_no* and *measurements cell_no* commands.

Parameter	Description	Permissions R-read,W-write	Valid Value
selection	contains all the selection flags of the measurement parameters. See Table 39, Table 40 and Table 41	RW	
values	contains all the values of the measurement parameters. See Table 42, Table 43 and Table 44	RO	
active	controls the measurement request. Selection flags can be modified only when this is disabled. On enabling, request is sent for measurements whose selection flags were enabled	RW	enable/disable

Table 37. Categories of L2 Measurement

The following table showcases the hierarchy inside both the selection and values groupings.

Parameter	Description	Permissions R-read,W-write	Valid Value
global	contains global level measurements. See Table 39 and Table 42		
UL	contains QCI based UL measurements. See Table 40 and Table 43		
DL	contains QCI based DL measurements. See Table 41 and Table 44		

Table 38. Categories of Selection Flags/Values

Parameter	Description	Permissions R-read,W-write	Valid Value
ul_prb_total	selection flag for total physical	RW	enable/disable

	resource block usage in uplink		
dl_prb_total	selection flag for total physical resource block usage in downlink	RW	enable/disable
ul_pream	selection flag for number of received Random Access preambles during a time period over all PRACHs configured in a cell. It enables (see Table 42): • ul_pream_total • ul_pream_dedicated • ul_pream_low • ul_pream_high	RW	enable/disable

Table 39. Global Selection Flags

Parameter*	Description	Permissions R-read,W-write	Valid Value
prb_usage x	selection flag for physical resource blocks usage per traffic class for uplink DTCH or per QCI x	RW	enable/disable
pkt_loss x	selection flag for packet loss rate in the uplink per QCI x for data radio bearers	RW	enable/disable
num_ue x	selection flag for number of active UE's in uplink per QCI x. It refers to UEs for which there is buffered data for uplink in data radio bearers	RW	enable/disable
sched_tpt x	selection flag for IP throughput in uplink per QCI x	RW	enable/disable

Table 40. UL based Selection Flags

Parameter*	Description	Permissions R-read,W-write	Valid Value

prb_usage x	selection flag for physical resource block usage per traffic class for downlink DTCH or per QCI x	RW	enable/disable
pkt_delay x	selection flag to measure packet delay in downlink per QCI x for data radio bearers	RW	enable/disable
pkt_loss x	selection flag for packet loss rate in the downlink per QCI x for data radio bearers	RW	enable/disable
pkt_discard x	selection flag for packet discard rate in the downlink per QCI x for data radio bearers	RW	enable/disable
num_ue x	selection flag for number of active UE's in downlink per QCI x	RW	enable/disable
sched_tpt x	selection flag for IP throughput in downlink per QCI x	RW	enable/disable

Table 41. DL based Selection Flags

Parameter	**Description**	**Permissions R-read,W-write**	**Valid Value**
ul_prb_total	total physical resource block usage in uplink	RO	0..100
dl_prb_total	total physical resource block usage in downlink	RO	0..100
ul_pream_total	total number of preambles	RO	0..4294967295
ul_pream_dedicated	number of dedicated preambles	RO	0..4294967295
ul_pream_low	number of preambles in the low range	RO	0..4294967295
ul_pream_high	number of preambles in the high range	RO	0..4294967295

Table 42. Global Measurement Values

Parameter*	**Description**	**Permissions R-read,W-write**	**Valid Value**
prb_usage x	physical resource blocks usage per traffic class for uplink DTCH or per QCI x	RO	0..100

pkt_loss x	packet loss rate in the uplink per QCI x for data radio bearers	RO	0..100
num_ue x	number of active UE's in uplink per QCI x. It refers to UEs for which there is buffered data for uplink in data radio bearers	RO	0..64
sched_tpt x	IP throughput in uplink per QCI x	RO	0..204800

Table 43. UL based Measurement Values

Parameter*	**Description**	**Permissions R-read,W-write**	**Valid Value**
prb_usage x	physical resource block usage per traffic class for downlink DTCH or per QCI x	RO	0..100
pkt_delay x	measures packet delay in downlink per QCI x for data radio bearers	RO	0..65535
pkt_loss x	packet loss rate in the downlink per QCI x for data radio bearers	RO	0..100
pkt_discard x	packet discard rate in the downlink per QCI x for data radio bearers	RO	0..100
num_ue x	number of active UE's in downlink per QCI x	RO	0..64
sched_tpt x	IP throughput in downlink per QCI x	RO	0..204800

Table 44. DL based Measurement Values

* Note that x refers to the QCI index for UL/DL based measurements. Range is 1-9 or ‘all’.

5.2.10.1 Display all measurements

show measurements cellNo

cellNo- cell entry

Operation	**Successful Outcome**
show measurements 0	measurements 0 selection global

	ul_prb_total :enable(1)
	    dl_prb_total :disable(0)
	    ul_pream :enable(1)
	  UL
	    prb_usage
	      qci 1 :enable(1)
	      qci 2 :disable(0)
	      ...
	     ...
	  DL
	    prb_usage
	      qci 1 :disable(0)
	      qci 2 :enable(1)
	      ...
	     ...
	values
	...

5.2.10.2 Display all selection flags or all values

show measurements cellNo selection/values

cellNo- cell entry

Operation	**Successful Outcome**

show measurements 0 selection	measurements 0 selection
	  global
	    ul_prb_total :enable(1)
	    dl_prb_total :disable(0)
	    ul_pream :enable(1)
	  UL
	    prb_usage
	      qci 1 :enable(1)
	      qci 2 :disable(0)
	      ...
	     ...
	  DL
	     prb_usage
	       qci 1 :disable(0)
	       qci 2 :enable(1)
	       ...
	     ...

show measurements 0 values	measurements 0 values
	  global
	    ul_prb_total :12
	    ul_pream :17
	  UL

	prb_usage qci 1 :44 qci 3 :43 ... DL prb_usage qci 2 :42 ...

5.2.10.3 Display a table of L2 measurements

show measurements cellNo selection/values tableName

tableName- any category in Table 38

cellNo- cell entry

Operation	Successful Outcome
show measurements 0 selection global	measurements 0 selection global ul_prb_total :enable(1) dl_prb_total :disable(0) ul_pream :enable(1)
show measurements 0 values DL	measurements 0 values DL prb_usage qci 1 :39 qci 2 :40 ... pkt_delay qci 1 :10 qci 2 :11 ... ...

5.2.10.4 Display a global L2 measurement

show measurements cellNo selection/values global paramName

cellNo- cell entry

paramName- any parameter in Table 39 or Table 42

Operation	Successful Outcome
show measurements 0 selection global ul_prb_total	measurements 0 selection global ul_prb_total: enable(1)
show measurements 0 values global ul_prb_total	measurements 0 values global ul_prb_total: 22

5.2.10.5 Display a UL/DL based L2 measurement for all QCI numbers

show measurements cellNo selection/values UL/DL paramName

cellNo- cell entry

paramName- any parameter in Table 40, Table 41, Table 43 or Table 44

For values, only the QCI numbers for which values are available will be displayed.

Operation	Successful Outcome
show measurements 0 selection UL num_ue	measurements 0 selection UL num_ue qci 1 :enable(1) qci 2 :disable(0) qci 3 :enable(1) …
show measurements 0 values UL prb_usage	measurements 0 values UL prb_usage qci 2 :35 qci 4 :40 qci 7 :39 …

5.2.10.6 Display a UL/DL based L2 measurement for a specific QCI number

show measurements cellNo selection/values UL/DL paramName qciNumber

cellNo- cell entry

paramName- any parameter in Table 40, Table 41, Table 43 or Table 44

qciNumber- integer in the range 1-9

Operation	Successful Outcome
show measurements 0 selection UL num_ue 4	measurements 0 selection UL num_ue 4 qci 4 :enable(1)
show measurements 0 values DL prb_usage 3	measurements 0 values DL prb_usage 3 qci 3 :27

5.2.10.7 Display measurements activation

show measurements cellNo active

cellNo- cell entry

Operation	Successful Outcome
show measurements 0 active	measurements 0 active: enable(1)

5.2.10.8 Set a global Selection flag

measurements cellNo selection global paramName value

cellNo- cell entry

paramName- any parameter in Table 39

value- enable/disable

Operation	Successful Outcome
measurements 0 selection global ul_prb_total disable	

5.2.10.9 Set a UL/DL based selection flag for all QCI numbers

measurements cellNo selection UL/DL paramName all value

cellNo- cell entry

paramName- any parameter in Table 40 or Table 41

all- wildcard to select all QCI's

value- enable/disable

Operation	Successful Outcome
measurements 0 selection DL num_ue all disable	

5.2.10.10 Set a UL/DL based selection flag for a specific QCI numbers

measurements cellNo selection UL/DL paramName qciNumber value

cellNo- cell entry

paramName- any parameter in Table 40 or Table 41

qciNumber- integer in the range 1-9

value- enable/disable

Operation	Successful Outcome
measurements 0 selection DL num_ue 5 enable	

5.2.10.11 Set measurements activation

measurements cellNo active value

cellNo- cell entry

value- enable/disable

Operation	Successful Outcome
measurements 0 active enable	

5.2.11 Handover configurations

Handover specific parameters for a cell are grouped together under the *show handover cell_no* and *handover cell_no* commands.

The *Presence* column indicates whether set commands expect a parameter to be given a value explicitly, or it can be omitted and given a default value.

Parameter	Description	Permissions R-read, W-write	Valid Values
x2_neigh	Specifies a list of configurations available for EUTRA X2 neighbor cells. See Table 46	RW-Lock	
s1_neigh	Specifies a list of configurations available for EUTRA S1 neighbor cells. See Table 47	RW-Lock	
utra_neigh	Specifies a list of configurations available for UTRA neighbor cells. See Table 48	RW-Lock	
report_cfg	Specifies a list of configurations available for measurement reports. See Table 49	RW-Lock	
meas_object_cfg	Specifies a list of configurations available for measurement objects. See Table 50	RW-Lock	
meas_id_cfg	Specifies a list of configurations available for measurement IDs. See Table 51	RW-Lock	
meas_cfg	Specifies a list of parameters related to the measurements. See Table 52	RW-Lock	

Table 45 Handover configurations

Parameter	Description	Permissions R-read, W-write	Valid Values	Presence
pci	Specifies Physical Cell ID which is an identification of a cell at physical layer.	RW-Restart	0..503	Mandatory

ip	Indicates IP address of the node required for socket creation for X2 interface.	RW-Restart	a.b.c.d
			a = 0..255
			b = 0..255
			c = 0..255
			d = 0..255
				Mandatory

x2ap_port	Specifies X2ap port of the node required for socket creation for X2 interface.	RW-Restart	0..65535	Mandatory

mac	Specifies MAC address of the node.	RW-Restart	aa:bb:cc:dd:ee:ff
			aa = 00.ff
			bb = 00..ff
			cc = 00..ff
			dd = 00..ff
			ee = 00..ff
			ff = 00..ff
				Mandatory

x2u_port	Indicates X2u port of the node.	RW-Restart	0..65535	Mandatory

Table 46 X2 Neighbors

Parameter	Description	Permissions R-read, W-write	Valid Values	Presence
pci	Specifies Physical Cell ID which is an identification of a cell at physical layer.	RW-Restart	0..503	Mandatory
type	Specifies the node type.	RW-Restart	macro, home	Mandatory

mcc	Specifies the mobile country code	RW-Restart	3 octets	Mandatory
mnc	Specifies the mobile network code.	RW-Restart	2 or 3 octets	Mandatory
cell_id	It is used to uniquely identify the cell within a network	RW-Restart	0..268435455	Mandatory

Table 47 S1 Neighbors

Parameter	**Description**	**Permissions** **R-read, W-write**	**Valid Values**	**Presence**
mcc	Specifies the mobile country code	RW-Restart	3 octets	Mandatory
mnc	Specifies the mobile network code	RW-Restart	2 or 3 octets	Mandatory
cell_id	Identifies a cell within an RNS (Radio Network Subsystem).	RW-Restart	0..65535	Mandatory
rnc_id	Identifies an RNC (Radio Network Controller) within a PLMN.	RW-Restart	0..65535	Mandatory
lac	Location area Code (LAC) is used to uniquely identify the cell within the location area.	RW-Restart	0..65535	Mandatory
rac	Routing Area Code (RAC) uniquely identifies the cell within the routing area.	RW-Restart	0..255	Mandatory
psc	Specifies primary scrambling code. It improves synchronization.	RW-Restart	0..511	Mandatory

Table 48 UTRA Neighbors

Parameter	Description	Permissions R-read, W-write	Valid Values	Presence
choice	Specifies that the report is for which type of cell.	RW-Restart	eutra, utra	Mandatory
trigger_type	Specifies how the report will be triggered	RW-Restart	event, periodic	Mandatory
event	Specifies the choice of E-UTRA or UTRA event triggered reporting criteria. EUTRA events are from a1 to a5 and UTRA events are b1 and b2. Value ab1 means A1 event for EUTRA cell or B1 event for UTRA cell. Same for ab2.	RW-Restart	ab1, ab2, a3, a4, a5	Optional Default: ab1
eutra_threshold_choice	Specifies threshold quantity to be used for measurement report triggering condition for events A1, A2, A4 and A5	RW-Restart	rsrp, rsrq	Optional Default: rsrp
threshold_rsrp	Threshold to be used in EUTRA measurement report triggering condition for events A1, A2, A3 and A5 provided threshold choice is RSRP	RW-Restart	0..97	Mandatory
threshold_rsrq	Threshold to be used in EUTRA measurement report triggering condition for events A1, A2, A3	RW-Restart	0..34	Mandatory

	and A5 provided threshold choice is RSRQ.			
a3_offset	Offset value to be used in EUTRA measurement report triggering condition for event A3	RW-Restart	-30..30	Mandatory
on_leave	Indicates whether or not the UE shall initiate the measurement reporting procedure when the leaving condition is met for a cell.	RW-Restart	false, true	Optional Default: false
eutra_threshold2_choice	Specifies threshold quantity to be used for measurement report triggering condition for event A5	RW-Restart	rsrp, rsrq	Optional Default: rsrp
threshold2_rsrp	Threshold2 to be used in EUTRA measurement report triggering condition for event A5 provided threshold choice is RSRP.	RW-Restart	0..97	Mandatory
threshold2_rsrq	Threshold2 to be used in EUTRA measurement report triggering condition for event A5 provided threshold choice is RSRQ.	RW-Restart	0..34	Mandatory
utra_threshold_choice	Specifies threshold quantity to be used for measurement report triggering condition for event B1 and B2	RW-Restart	rscp, ecno	Optional Default: rscp

threshold_rscp	Threshold to be used in UTRA measurement report triggering condition for event B1 and B2 provided utra_threshold_choice is rscp	RW-Restart	-5..91	Mandatory
threshold_ecno	EcNO threshold to be used in UTRA measurement report triggering condition for event B1 and B2 provided utra_threshold_choice is ecno.	RW-Restart	0..49	Mandatory
hysteresis	Hysteresis value applied to entry and leave condition of a report triggering event. It is expressed in dB.	RW-Restart	0..30	Optional Default: 0
time_to_trigger	Time period during which measurement report triggering condition needs to be met in order to trigger a measurement report.	RW-Restart	ms0, ms40, ms64, ms80, ms100, ms128, ms160, ms256, ms320, ms480, ms512, ms640, ms1024, ms1280, ms2560, ms5120	Optional Default: ms40
trigger_quantity	Quantities used to evaluate a measurement report triggering condition.	RW-Restart	rsrp, rsrq	Optional Default: rsrp
report_quantity	The quantities to be included in the measurement report.	RW-Restart	trigger, both	Optional default: both

max_report_cells	Maximum number of cells that can be included in a measurement report.	RW-Restart	1..8	Optional default: 8
interval	Interval between successive measurement reports.	RW-Restart	ms120, ms240, ms480, ms640, ms1024, ms2048, ms5120, ms10240, min1, min6, min12, min30, min60	Optional Default: ms480
amount	Number of times a measurement report is sent	RW-Restart	ra1, ra2, ra4, ra8, ra16, ra32, ra64, inf	Optional default: inf

Table 49 HO Report Configuration

Parameter	**Description**	**Permissions** **R-read, W-write**	**Valid Values**	**Presence**
choice	It is used to distinguish between measurement object of EUTRA and UTRA cell.	RW-Restart	eutra, utra	Mandatory
carrier_freq	Uniquely identifies the LTE band and carrier frequency.	RW-Restart	0..65535	Mandatory
rb	Indicates the maximum allowed measurement bandwidth on a carrier frequency as defined by the parameter Transmission Bandwidth	RW-Restart	mbw6, mbw15, mbw25, mbw50, mbw75, mbw100	Optional Default: mbw15

	Configuration "NRB”.			
freq_offset	Indicates a frequency specific offset of the frequency of the neighbor cell, specified in dB.	RW-Restart	dBMinus24, dBMinus22, dBMinus20, dBMinus18, dBMinus16, dBMinus14, dBMinus12, dBMinus10, dBMinus8, dBMinus6, dBMinus5, dBMinus4, dBMinus3, dBMinus2, dBMinus1, dB0, dB1, dB2, dB3, dB4, dB5, dB6, dB8, dB10, dB12, dB14, dB16, dB18, dB20, dB22, dB24	Optional Default: dBMinus1
cgi_cell	It indicates the PCI of that cell for which to report CGI.	RW-Restart	0..503	Optional default: 0

Table 50 HO Measurement Object Configuration

Parameter	**Description**	**Permissions** **R-read, W-write**	**Valid Values**	**Presence**
meas_object_id	It is used to identify a measurement object	RW-Restart	0..31	Mandatory

	configuration.			
report_id	It is used to identify the measurement report.	RW-Restart	0..31	Mandatory

Table 51 HO Measurement ID Configuration

Parameter	**Description**	**Permissions** **R-read, W-write**	**Valid Values**	**Presence**
eutra_filter_coeff_rsrp	Filtering coefficient used for RSRP measurements.	RW-Restart	fc0, fc1, fc2, fc3, fc4, fc5, fc6, fc7, fc8, fc9, fc11, fc13, fc15, fc17, fc19	Optional default: fc0
eutra_filter_coeff_rsrq	Filtering coefficient used for RSRQ measurements.	RW-Restart	fc0, fc1, fc2, fc3, fc4, fc5, fc6, fc7, fc8, fc9, fc11, fc13, fc15, fc17, fc19	Optional default: fc0
utra_meas_quant_fdd	Measurement quantity used for UTRA measurements in FDD.	RW-Restart	cpich_RSCP, cpich_EcNO	Optional default: cpich_RSCP
utra_meas_quant_tdd	Measurement quantity used for UTRA measurements in TDD.	RW-Restart	pccpch-RSCP	Optional default: pccpch-RSCP
utra_filter_coeff	Filtering coefficient used for UTRA measurements.	RW-Restart	fc0, fc1, fc2, fc3, fc4, fc5, fc6, fc7, fc8, fc9, fc11, fc13, fc15, fc17, fc19	Optional default: fc4
gap_cfg_choice	It controls setup/ release of measurement gaps.	RW-Restart	setup, release	Optional default : release

gap_offset_choice	It indicates the gap pattern to be used for configuring measurement gaps	RW-Restart	gp0, gp1	Optional default : gp1
gap_offset	It indicates the value of gap offset to be used, specified in ms(milli second). If gap pattern ID 0 is used, then range is 0 to 39. For gap pattern ID 1, range is 0 to 79. Value of -1 indicates that measurement gaps are disabled.	RW-Restart	-1..79	Optional default: -1
hysteresis_rsrp	It indicates the value of threshold to be used in EUTRA measurement report triggering condition for periodic reports. Valid if triggeringQuantity is RSRP.	RW-Restart	0..30	Optional default: 9
hysteresis_rsrq	It indicates the value of threshold to be used in EUTRA measurement report triggering condition for periodic reports. Valid if triggeringQuantity is RSRQ.	RW-Restart	0..30	Optional default: 5

threshold_rscp	It indicates the value of threshold to be used in UTRA measurement report triggering condition for periodic reports. Valid if triggeringQuantity is RSCP.	RW-Restart	-5..91	Optional default: 60
threshold_ecno	It indicates the value of threshold to be used in UTRA measurement report triggering condition for periodic reports. Valid if triggeringQuantity is EcNO.	RW-Restart	0..49	Optional default: 43
threshold_rsrp	It indicates the value of threshold to be used in UTRA measurement report triggering condition for periodic reports. Valid if triggeringQuantity is RSRP.	RW-Restart	0..97	Optional default: 60
threshold_rsrq	It indicates the value of threshold to be used in UTRA measurement report triggering condition for periodic reports. Valid if triggeringQuantity is RSRQ	RW-Restart	0..34	Optional default: 28
threshold_rsrp_meas_gap_enable	RSRP threshold value below which the measurement gaps will be enabled	RW-Restart	0..97	Optional default: 60

threshold_rsrp_meas_gap_disable	RSRP threshold value above which the measurement gaps will be disabled	RW-Restart	0..97	Optional default: 69
threshold_rsrq_meas_gap_enable	RSRQ threshold value below which the measurement gaps will be enabled	RW-Restart	0..34	Optional default: 24
threshold_rsrq_meas_gap_disable	RSRQ threshold value above which the measurement gaps will be disabled	RW-Restart	0..34	Optional default: 29

Table 52 HO Measurement Configurations

5.2.11.1 X2 neighbor cells configurations

It covers neighbor cells configuration for Intra LTE handover performed by X2 interface.

5.2.11.1.1 Display all entries

The below command displays entries for all the X2 neighbor cells.

show handover cell_no x2_neigh

cell_no - cell entry.

Operation	Successful Outcome
show handover 0 x2_neigh	handover 0 x2_neigh 1 pci :118 ip :192.168.3.3 x2ap_port :36422 mac :c4:ed:ba:a5:bf:ff x2u_port :36100 handover 0 x2_neigh 2 pci :119 ip :192.168.3.4 x2ap_port :3642 mac :aa:bb:cc:dd:ee:ff x2u_port :36100

When cell_no is within range but instance is not available:

Operation	Successful Outcome
show handover 1 x2_neigh	>

5.2.11.1.2 Display a specific entry

The below command displays entry for a specific neighbor cell.

show handover cell_no x2_neigh x2_neigh_no

cell_no - cell entry.

x2_neigh_no – neighbor cell entry.

Operation	Successful Outcome
show handover 0 x2_neigh 1	handover 0 x2_neigh 1 pci :118 ip :192.168.3.3 x2ap_port :36422 mac :c4:ed:ba:a5:bf:ff x2u_port :36100

5.2.11.1.3 Display a specific parameter

Below command displays a specific parameter of X2 neighbor cell.

show cell cell_no x2_neigh x2_neigh_no paramName

cell_no - cell entry.

x2_neigh_no – x2 neighbor entry

paramName - any parameter from above mentioned list with read permission.

Operation	Successful Outcome
show handover 0 x2_neigh 1 pci	handover 0 x2_neigh 1 pci :118

5.2.11.1.4 Adding a new entry

Below command is used to add a new entry for X2 neighbor cells.

handover cell_no x2_neigh x2_neigh_no add <pci >< IP address> < x2ap port> <MAC address> <x2u port>

cell_no - cell entry.

x2_neigh_no – x2 neighbor entry

Operation	Successful Outcome

handover 0 x2_neigh 2 add 300 127.0.0.1 4000 aa:aa:aa:aa:aa:aa 5000	

If user tries to add an entry at the same index again, then error is issued.

Operation	Expected Outcome
handover 0 x2_neigh 2 add 300 127.0.0.1 4000 aa:aa:aa:aa:aa:aa 5000	entry already exists at this index, so add operation is not allowed

5.2.11.1.5 Updating an entry

Below command is used to update an existing entry of X2 neighbor cells.

handover cell_no x2_neigh x2_neigh_no update paramName paramValue

cell_no - cell entry.

x2_neigh_no – x2 neighbor entry

paramName - any parameter from above mentioned list with write permission.

value - value of the parameter to be set.

Operation	Successful Outcome
handover 0 x2_neigh 2 update pci 111	

If user tries to update a non-existent entry, then error is issued. Example considering entry does not exists at index 20:

Operation	Expected Outcome
handover 0 x2_neigh 20 update pci 111	no entry exists at this index, so update operation is not allowed

5.2.11.1.6 Deleting an entry

Below command is used to delete an entry of X2 neighbor cells.

handover cell_no x2_neigh x2_neigh_no delete

cell_no - cell entry.

x2_neigh_no – x2 neighbor entry

Operation	Successful Outcome
handover 0 x2_neigh 2 delete	

If user tries to delete a non-existent row, then error is issued. Example considering entry does not exists at index 20:

Operation	Expected Outcome
handover 0 x2_neigh 20 delete	no entry exists at this index, so delete operation is not allowed

5.2.11.2 S1 neighbor cells configurations

It covers neighbor cells configuration for Intra LTE handover performed by S1 interface.

5.2.11.2.1 Display all entries

The below command displays entries for all the S1 neighbor cells.

show handover cell_no s1_neigh

cell_no - cell entry.

Operation	Successful Outcome
show handover 0 s1_neigh	handover 0 s1_neigh 1 pci :111 type :macro(0) mcc :"111" mnc :"220" cell_id :321 handover 0 s1_neigh 2 pci :111 type :home(1) mcc :"222" mnc :"110" cell_id :1233

When cell_no is within range but instance is not available:

Operation	Successful Outcome
show handover 1 s1_neigh	>

5.2.11.2.2 Display a specific entry

The below command displays entry for a specific S1 neighbor cell.

show handover cell_no s1_neigh s1_neigh_no

cell_no - cell entry.

s1_neigh_no – neighbor cell entry.

Operation	Successful Outcome
show handover 0 s1_neigh 1	handover 0 s1_neigh 1 pci :111 type :macro(0) mcc :"111" mnc :"220" cell_id :321

5.2.11.2.3 Display a specific parameter

Below command displays a specific parameter of S1 neighbor cell.

show cell cell_no s1_neigh s1_neigh_no paramName

cell_no - cell entry.

s1_neigh_no – s1 neighbor entry

paramName - any parameter from above mentioned list with read permission.

Operation	Successful Outcome
show handover 0 s1_neigh 1 pci	handover 0 s1_neigh 1 pci :118

5.2.11.2.4 Adding a new entry

Below command is used to add a new entry for S1 neighbor cell.

handover cell_no s1_neigh s1_neigh_no add <pci value>< node type> < mcc> <mnc> <cell id>

cell_no - cell entry.

s1_neigh_no – s1neighbor entry

Operation	Successful Outcome
handover 0 s1_neigh 1 add 300 home 111 11 1234	

If user tries to add an entry at the same index again, then error is issued.

Operation	Expected Outcome
handover 0 s1_neigh 1 add 300 home 111 11 1234	entry already exists at this index, so add operation is not allowed

5.2.11.2.5 Updating an entry

Below command is used to update an existing entry of S1 neighbor cell.

handover cell_no s1_neigh s1_neigh_no update paramName paramValue

cell_no - cell entry.

s1_neigh_no - s1 neighbor entry

paramName - any parameter from above mentioned list with write permission.

paramValue - value of the parameter to be set.

Operation	Successful Outcome
handover 0 s1_neigh 1 update pci 112	

If user tries to update a non-existent row, then error is issued. Example considering entry does not exists at index 20:

Operation	Expected Outcome
handover 0 s1_neigh 20 update pci 111	no entry exists at this index, so update operation is not allowed

5.2.11.2.6 Deleting an entry

Below command is used to delete an entry of S1 neighbor cell.

handover cell_no s1_neigh s1_neigh_no delete

cell_no - cell entry.

s1_neigh_no – s1 neighbor entry

Operation	Successful Outcome
handover 0 s1_neigh 1 delete	

If user tries to delete a non-existent row, then error is issued. Example considering entry does not exists at index 20:

Operation	Expected Outcome
handover 0 s1_neigh 20 delete	no entry exists at this index, so delete operation is not allowed

5.2.11.3 Utra neighbor cells configurations

It covers neighbor cells configuration for inter RAT handover.

5.2.11.3.1 Display all entries

The below command displays entries for all the UTRA neighbor cells.

show handover cell_no utra_neigh

cell_no - cell entry.

Operation	Successful Outcome
show handover 0 utra_neigh	show handover 0 utra_neigh handover 0 utra_neigh 1 mcc :"321" mnc :"320" cell_id :4321 rnc_id :231 lac :321 rac :213 psc :22 handover 0 utra_neigh 2 mcc :"111" mnc :"110" cell_id :1234 rnc_id :12 lac : 31 rac :23 psc :21

When cell_no is within range but instance is not available:

Operation	Successful Outcome
show handover 1 utra_neigh	>

5.2.11.3.2 Display a specific entry

The below command displays entry for a specific neighbor cell.

show handover cell_no utra_neigh utra_neigh_no

cell_no - cell entry.

utra_neigh_no – neighbor cell entry.

Operation	Successful Outcome
show handover 0 utra_neigh 1	handover 0 utra_neigh 1

5.2.11.3.3 Display a specific parameter

Below command displays a specific parameter of UTRA neighbor cell.

show cell cell_no utra_neigh utra_neigh_no paramName

cell_no - cell entry.

utra_neigh_no – utra neighbor entry

paramName - any parameter from above mentioned list with read permission.

Operation	Successful Outcome
show handover 0 utra_neigh 1 pci	handover 0 utra_neigh 1 mcc :"321" mnc :"320" cell_id :4321 rnc_id :231 lac :321 rac :213 psc :22

5.2.11.3.4 Adding a new entry

Below command is used to add a new entry for utra neighbor cells.

handover cell_no utra_neigh utra_neigh_no add <mcc>< mnc> < cell id> <rnc id> <lac> <rac> <psc>

cell_no - cell entry.
utra_neigh_no – utra neighbor entry

Operation	Successful Outcome
handover 0 utra_neigh 1 add 111 11 1234 123 122 123 123	

If user tries to add an entry at the same index again, then error is issued.

Operation	Expected Outcome
handover 0 utra_neigh 1 add 111 11 1234 123 122 123 123	entry already exists at this index, so add operation is not allowed

5.2.11.3.5 Updating an entry

Below command is used to update an existing entry of UTRA neighbor cell.

handover cell_no utra_neigh utra_neigh_no update paramName paramValue

cell_no - cell entry.
utra_neigh_no – utra neighbor entry
paramName - any parameter from above mentioned list with write permission.
paramValue - value of the parameter to be set.

Operation	Successful Outcome
handover 0 utra_neigh 1 update mcc 112	

If user tries to update a non-existent row, then error is issued. Example considering entry does not exists at index 20:

Operation	Expected Outcome
handover 0 utra_neigh 20 update mcc 123	no entry exists at this index, so update operation is not allowed

5.2.11.3.6 Deleting an entry

Below command is used to delete an entry of UTRA neighbor cell.

handover cell_no utra_neigh utra_neigh_no delete

cell_no - cell entry.
utra_neigh_no – utra neighbor entry

Operation	Successful Outcome
handover 0 utra_neigh 1 delete	

If user tries to delete a non-existent row, then error is issued. Example considering entry does not exists at index 20:

Operation	Expected Outcome
handover 0 utra_neigh 20 delete	no entry exists at this index, so delete operation is not allowed

5.2.11.4 Measurement report configurations

It specifies criteria for triggering of an E-UTRA and UTRA measurement reporting event.

5.2.11.4.1 Display all entries

The below command displays all configurations of measurement report.

show handover cell_no report_cfg

cell_no - cell entry.

Operation	Successful Outcome
show handover 0 report_cfg	handover 0 report_cfg 1 choice :utra(2) trigger_type :event(1) event :ab1(1) eutra_threshold_choice :rsrp(0) threshold_rsrp :12 threshold_rsrq :21 a3_offset :-4 on_leave :false(0) eutra_threshold2_choice :rsrp(0) threshold2_rsrp :34 threshold2_rsrq :0 utra_threshold_choice :ecno(1) threshold_rscp :-3 threshold_ecno :23 hysteresis :23 time_to_trigger :ms5120(15) trigger_quantity :rsrp(0) report_quantity :both(1) max_report_cells :8 interval :ms480(2) amount :inf(7)

When cell_no is within range but instance is not available:

Operation	Successful Outcome
show handover 1 utra_neigh	>

5.2.11.4.2 Display a specific entry

The below command displays configuration of a specific measurement report.

show handover cell_no report_cfg report_cfg_no

cell_no - cell entry.
report_cfg_no – report configuration entry.

Operation	Successful Outcome
show handover 0 report_cfg 1	handover 0 report_cfg 1 choice :utra(2) trigger_type :event(1)

	event :ab1(1) eutra_threshold_choice :rsrp(0) threshold_rsrp :12 threshold_rsrq :21 a3_offset :-4 on_leave :false(0) eutra_threshold2_choice :rsrp(0) threshold2_rsrp :34 threshold2_rsrq :0 utra_threshold_choice :ecno(1) threshold_rscp :-3 threshold_ecno :23 hysteresis :23 time_to_trigger :ms5120(15) trigger_quantity :rsrp(0) report_quantity :both(1) max_report_cells :8 interval :ms480(2) amount :inf(7)

5.2.11.4.3 Display a specific parameter

Below command displays a specific parameter of the measurement report.

show handover cell_no report_cfg report_cfg_no paramName

cell_no - cell entry.
report_cfg_no –report configuration entry

paramName - any parameter from above mentioned list with read permission.

Operation	Successful Outcome
show handover 0 report_cfg 1 choice	handover 0 report_cfg 1 choice : utra(2)

5.2.11.4.4 Adding a new entry

User can add a new row by providing all the mandatory parameters and if required some/all the optional parameters.

Below command(s) can be used to add a new entry for report configuration.

handover cell_no report_cfg report_cfg_no add <choice> <trigger_type> <threshold_rsrp> <threshold_rsrq> <a3_offset> <threshold2_rsrp> <threshold2_rsrq> <threshold_rscp> <threshold_ecno>

handover cell_no report_cfg report_cfg_no add <choice> <trigger_type> <threshold_rsrp> <threshold_rsrq> <a3_offset> <threshold2_rsrp> <threshold2_rsrq> <threshold_rscp> <threshold_ecno> <event> <eutra_threshold_choice> ...

cell_no - cell entry.
report_cfg_no – report configuration entry

Operation	Successful Outcome
handover 0 report_cfg 1 add eutra event 11 12 13 14 15 16 17	

If user tries to add an entry at the same index again, then error is issued.

Operation	Expected Outcome
handover 0 report_cfg 1 add eutra event 11 12 13 14 15 16 17	entry already exists at this index, so add operation is not allowed

5.2.11.4.5 Updating an entry

Below command is used to update an existing entry of report configuration.

handover cell_no report_cfg report_cfg_no update paramName paramValue

cell_no - cell entry.
report_cfg_no – report configuration entry
paramName - any parameter from above mentioned list with write permission.
paramValue - value of the parameter to be set.

Operation	Successful Outcome
handover 0 report_cfg 1 update choice eutra	

If user tries to update a non-existent row, then error is issued Example considering entry does not exists at index 20:

Operation	Expected Outcome
handover 0 report_cfg 20 update choice eutra	no entry exists at this index, so update operation is not allowed

5.2.11.4.6 Deleting an entry

Below command is used to delete an entry of report configurations.

handover cell_no report_cfg report_cfg_no delete

cell_no - cell entry.
report_cfg_no – report configuration entry

Operation	Successful Outcome
handover 0 report_cfg 1 delete	

If user tries to delete a non-existent row, then error is issued. Example considering entry does not exists at index 20:

Operation	Expected Outcome
handover 0 report_cfg 20 delete	no entry exists at this index, so delete operation is not allowed

5.2.11.5 Measurement object configurations

It specifies information applicable for intra-frequency or inter-frequency E-UTRA cells and inter-RAT UTRA neighboring cells.

5.2.11.5.1 Display all entries

The below command displays entries of all the available measurement object configurations.

show handover cell_no meas_object_cfg

cell_no - cell entry.

Operation	Successful Outcome
show handover 0 meas_object_cfg	handover 0 meas_object_cfg 1 choice :eutra(1) carrier_freq :100 rb :mbw15(1) freq_offset :dB0(15) cgi_cell :11 handover 0 meas_object_cfg 2 choice :eutra(1) carrier_freq :100 rb :mbw15(1) freq_offset :dB0(15) cgi_cell :0

When cell_no is within range but instance is not available:

Operation	Successful Outcome
show handover 1 meas_object_cfg	>

5.2.11.5.2 Display a specific entry

The below command displays entry for a specific measurement object configuration.

show handover cell_no meas_object_cfg meas_object_cfg_no

cell_no - cell entry.
meas_object_cfg_no – measurement object entry.

Operation	Successful Outcome
show handover 0 meas_object_cfg 1	handover 0 meas_object_cfg 1 choice :eutra(1) carrier_freq :100 rb :mbw15(1) freq_offset :dB1(16) cgi_cell :11

5.2.11.5.3 Display a specific parameter

Below command displays a specific parameter of measurement object configuration.

show cell cell_no meas_object_cfg meas_object_cfg_no paramName

cell_no - cell entry.
meas_object_cfg_no – measurement object configuration entry
paramName - any parameter from above mentioned list with read permission.

Operation	Successful Outcome
show handover 0 meas_object_cfg 1 choice	handover 0 meas_object_cfg 1 choice : eutra(1)

5.2.11.5.4 Adding a new entry

User can add a new entry by providing all the mandatory parameters and if required some/ all the optional parameters.

Below command(s) can be used to add a new entry for measurement object configuration.

handover cell_no meas_object_cfg meas_object_cfg_no add <choice> <carrier_freq>

handover cell_no meas_object_cfg meas_object_cfg_no add <choice> <carrier_freq> <rb> <freq_offset> …

cell_no - cell entry.
meas_object_cfg_no – measurement object configuration entry

Operation	Successful Outcome
handover 0 meas_object_cfg 1 add eutra 12345	

If user tries to add an entry at the same index again, then error is issued.

Operation	Expected Outcome
handover 0 meas_object_cfg 1 add utra 12345	entry already exists at this index, so add operation is not allowed

5.2.11.5.5 Updating an entry

Below command is used to update an existing entry of measurement object configuration.

handover cell_no meas_object_cfg meas_object_cfg_no update paramName paramValue

cell_no - cell entry.
meas_object_cfg_no – measurement object configuration entry
paramName - any parameter from above mentioned list with write permission.
paramValue - value of the parameter to be set.

Operation	Successful Outcome
handover 0 meas_object_cfg 1 update carrier_freq 1234	

If user tries to update a non-existent row, then error is issued Example considering entry does not exists at index 20:

Operation	Expected Outcome
handover 0 meas_object_cfg 20 update carrier_freq 1234	no entry exists at this index, so update operation is not allowed

5.2.11.5.6 Deleting an entry

Below command is used to delete an entry of measurement object configuration.

handover cell_no meas_object_cfg meas_object_cfg_no delete

cell_no - cell entry.
meas_object_cfg_no – measurement object configuration entry

Operation	Successful Outcome
handover 0 meas_object_cfg 1 delete	

If user tries to delete a non-existent row, then error is issued. Example considering entry does not exists at index 20:

Operation	Expected Outcome
handover 0 meas_object_cfg 20 delete	no entry exists at this index, so delete operation is not allowed

5.2.11.6 Measurement Id configurations

It is used to identify a measurement configuration, i.e., linking of a measurement object and a reporting configuration.

The meas id table is created by adding a reference of meas objects and report configs. So reference to a non-existent meas object ID or report ID will report an error.

On deleting a meas object or report entry, its corresponding entries in meas ID table referencing it will also be deleted. As a side effect, all meas id rows will be resynched with the persistency database, therefore the rows that are not yet active will be wiped even if referencing other existing active meas objects and reports. Conversely, active rows with valid references will still remain persistent.

5.2.11.6.1 Display all entries

The below command displays entries for all the available measurement ID configurations.

show handover cell_no meas_id_cfg

cell_no - cell entry.

Operation	Successful Outcome
show handover 0 meas_id_cfg	handover 0 meas_id_cfg 1 meas_object_id :30 report_id :5 handover 0 meas_id_cfg 2 meas_object_id :2 report_id :2 handover 0 meas_id_cfg 3 meas_object_id :7 report_id :7

When cell_no is within range but instance is not available:

Operation	Successful Outcome
show handover 1 meas_id_cfg	>

5.2.11.6.2 Display a specific entry

The below command displays entry for a specific measurement ID configuration.

show handover cell_no meas_id_cfg meas_id_cfg_no

cell_no - cell entry.
meas_id_cfg_no – measurement ID configuration entry.

Operation	Successful Outcome
show handover 0 meas_id_cfg 1	handover 0 meas_id_cfg 1 meas_object_id :30 report_id :5

5.2.11.6.3 Display a specific parameter

Below command displays a specific parameter of measurement ID configuration.

show cell cell_no meas_id_cfg meas_id_cfg_no paramName

cell_no - cell entry.
meas_id_cfg_no – measurement ID configuration entry
paramName - any parameter from above mentioned list with read permission.

Operation	Successful Outcome
show handover 0 meas_id_cfg 1 meas_object_id	handover 0 meas_id_cfg 1 meas_object_id :30

5.2.11.6.4 Adding a new entry

Below command is used to add a new entry for measurement ID configuration.

handover cell_no meas_id_cfg meas_id_cfg_no add <meas_object_id> <report_id>

cell_no - cell entry.
meas_id_cfg_no – measurement ID configuration entry

Operation	Successful Outcome
handover 0 meas_id_cfg 1 add 1 1	

If user tries to add an entry at the same index again, then error is issued.

Operation	Expected Outcome
handover 0 meas_id_cfg 1 add 1 1	entry already exists at this index, so add operation is not allowed

5.2.11.6.5 Updating an entry

Below command is used to update an existing entry of measurement ID configuration.

handover cell_no meas_id_cfg meas_id_cfg_no update paramName paramValue

cell_no - cell entry.
meas_id_cfg_no – measurement ID configuration entry
paramName - any parameter from above mentioned list with write permission.
paramValue - value of the parameter to be set.

Operation	Successful Outcome
handover 0 meas_id_cfg 1 update meas_object_id 2	

If user tries to update a non-existent row, then error is issued Example considering entry does not exists at index 20:

Operation	Expected Outcome
handover 0 meas_id_cfg 20 update meas_object_id 2	no entry exists at this index, so update operation is not allowed

5.2.11.6.6 Deleting an entry

Below command is used to delete an entry of measurement ID configuration.

handover cell_no meas_id_cfg meas_id_cfg_no delete

cell_no - cell entry.
meas_id_cfg_no – measurement ID configuration entry

Operation	Successful Outcome
handover 0 meas_id_cfg 1 delete	

If user tries to delete a non-existent row, then error is issued. Example considering entry does not exists at index 20:

Operation	Expected Outcome
handover 0 meas_id_cfg 20 delete	no entry exists at this index, so delete operation is not allowed

5.2.11.7 Measurement configurations

It specifies measurements to be performed by the UE, and covers intra-frequency, inter-frequency and inter-RAT mobility as well as configuration of measurement gaps.

In order to enable measurement gaps, *gap_offset* must be set to any valid value in range (0..79). To disable the measurement gaps, *gap_offset* must be set to -1.

5.2.11.7.1 Display all entries

The below command displays the measurement configuration.

show handover cell_no meas_cfg

cell_no - cell entry.

Operation	Successful Outcome
show handover 0 meas_cfg	handover 0 meas_cfg eutra_filter_coeff_rsrp : fc9(9) eutra_filter_coeff_rsrq : fc9(9) utra_meas_quant_fdd : cpichRSCP(0) utra_meas_quant_tdd : pccpchRSCP(0) utra_filter_coeff : fc9(9) gap_cfg_choice : setup(2) gap_offset_choice : gp0(0) gap_offset : 11 hysteresis_rsrp : 3 hysteresis_rsrq : 23 threshold_rscp : -5 threshold_ecno : 32 threshold_rsrp : 45 threshold_rsrq : 32 threshold_rsrp_meas_gap_enable : 23 threshold_rsrp_meas_gap_disable : 24 threshold_rsrq_meas_gap_enable : 13 threshold_rsrq_meas_gap_disable : 14

When cell_no is within range but instance is not available:

Operation	Successful Outcome
show handover 1 meas_cfg	>

5.2.11.7.2 Display a specific parameter

Below command displays a specific parameter of measurement configuration.

show cell cell_no meas_cfg paramName

cell_no - cell entry.
paramName - any parameter from above mentioned list with read permission.

Operation	Successful Outcome
show handover 0 meas_cfg eutra_filter_coeff_rsrp	handover 0 meas_cfg eutra_filter_coeff_rsrp :fc9(9)

5.2.11.7.3 Updating a parameter

Below command is used to set a parameter of measurement configuration.

handover cell_no meas_cfg paramName paramValue

cell_no - cell entry.
paramName - any parameter from above mentioned list with write permission.
paramValue - value of the parameter to be set.

Operation	Successful Outcome
handover 0 meas_cfg eutra_filter_coeff_rsrp fc9	

5.2.12 Statistics

The following statistics are available for a single cell object via the *show statistics cell_no* command.

Note that querying over the statistics before the source subsystems have completely started will return an empty data set.

Parameter	Description	Permissions R-read, W-write	Valid Values
mac [rnti X]	specifies mac statistics data aggregated/averaged (except SNR) over all UEs as well per rnti basis. See Table 54	RO	
rlc [rnti X]	specifies rlc statistics data aggregated over all UEs as well per rnti basis. See Table 55	RO	

pdcp [rnti X]	specifies pdcp statistics data aggregated over all UEs as well per rnti basis. See Table 56	RO	
s1u [rnti X]	specifies s1u statistics data aggregated over all UEs as well per rnti basis. See Table 57	RO	
rrc	specifies rrc statistics present for available rntis. See Table 58	RO	
global	specifies global statistics present for available rntis. See Table 59	RO	
rnti_list	specifies rrc rnti statistics present for available rntis. See Table 60	RO	
phy [ue X]	specifies physical layer statistics for available UEs. See Table 61	RO	
dsp [num]	specifies DSP memory and packets heap stats See Table 62	RO	

Table 53 Statistics Categories

Parameter	**Description**	**Permissions** **R-read, W-write**	**Valid Values**
ul_snr_pusch	SNR calculated in UL over PUSCH	RO	0..255
ul_snr_pucch	SNR calculated in UL over PUCCH	RO	0..255
dl_snr	SNR calculated in DL	RO	0..15
dl_tpt	DL throughput calculated at MAC	RO	0..268435455
ul_tpt	UL throughput calculated at MAC	RO	0..268435455

dl_bler_tb1	DL Bler calculated for TB1 (Percentage of NACKs received in DL for TB1)	RO	0..100
dl_bler_tb2	DL Bler calculated for TB2 (Percentage of NACKs received in DL for TB2)	RO	0..100
ul_bler	Bler calculated in UL (Percentage of NACKs sent for UL data)	RO	0..100

Table 54 MAC Statistics

Parameter	**Description**	**Permissions** **R-read, W-write**	**Valid Values**
dl_tpt	DL throughput at RLC	RO	0..268435455
ul_tpt	UL throughput at RLC	RO	0..268435455
pkt_whole_nack_recv	number of NACKs received for Complete RLC PDU	RO	0..268435455
pkt_seg_nack_recv	number of NACKs received for Segments of RLC PDUs	RO	0..268435455

Table 55 RLC Statistics

Parameter	**Description**	**Permissions** **R-read, W-write**	**Valid Values**
dl_tpt	DL throughput at PDCP	RO	0..268435455
ul_tpt	UL throughput at PDCP	RO	0..268435455
ul_pkt_lost	packet loss in UL at PDCP	RO	0..268435455
dl_pkt_lost	packet loss in DL at PDCP due to Queue Full	RO	0..268435455

Table 56 PDCP Statistics

Parameter	**Description**	**Permissions** **R-read, W-write**	**Valid Values**
dl_tpt	S-GW UL throughput	RO	0..268435455
ul_tpt	S-GW DL throughput	RO	0..268435455

Table 57 S1U Statistics

Parameter	Description	Permissions R-read, W-write	Valid Values
conn_req_recv	number of RRC Connection Requests received	RO	0..268435455
conn_setup_comp_nt_recv	number of RRC Connnection Setup Complete not received	RO	0..268435455
conn_reconf_comp_nt_recv	number of RRC Connection Reconfiguration Complete Not Received	RO	0..268435455
conn_reestablishment_recv	number of RRCConnectionReestablishments received	RO	0..268435455
rlf_reported_l2	number of RLFs reported by Layer-2	RO	0..268435455
ues_configured	total number of UEs configured at Layer-3	RO	0..268435455
ues_released	total number of UEs released at Layer-3	RO	0..268435455

Table 58 RRC Statistics

Parameter	Description	Permissions R-read, W-write	Valid Values
num_rach	number of RACH received	RO	0..268435455
num_msg2	number of msg2 sent	RO	0..268435455
num_msg3	number of msg3 received	RO	0..268435455
num_pdcp_inactivity_timer	number of times PDCP Inactivity Timer Expired Called	RO	0..268435455
num_rlf_at_mac	number of RLFs triggered at MAC	RO	0..268435455
num_rel_triggrd_at_mac	number of releases triggered at MAC	RO	0..268435455
ul_snr_pucch_ue_rel	number of UEs released based on low UL SNR over PUCCH	RO	0..268435455

ul_snr_pusch_ue_rel	number of UEs released based on low UL SNR over PUSCH	RO	0..268435455
dl_crc_ue_rel	number of UEs released based on High DL Bler	RO	0..268435455
rlc_num_rlf	number of RLFs triggered at RLC	RO	0..268435455

Table 59 Global Statistics

Parameter	**Description**	**Permissions** **R-read, W-write**	**Valid Values**
rnti	RNTI present at Layer-3	RO	10..65523

Table 60 RRC Rnti Statistics

Parameter	**Description**	**Permissions** **R-read, W-write**	**Valid Values**
ul_cqi_ant0	UL CQI on the first configured antenna in dB	RO	–128..127
ul_cqi_ant1	UL CQI on the second configured antenna in dB	RO	–128..127
noise_var_ant0	Noise variance on the first configured antenna in dBm	RO	-32768..32767
noise_var_ant1	Noise variance on the second configured antenna in dBm	RO	-32768..32767
rssi_ant0	RSSI on the first configured antenna	RO	-32768..32767
rssi_ant1	RSSI on the second configured antenna	RO	-32768..32767

Table 61 PHY Statistics

Parameter	**Description**	**Permissions** **R-read, W-write**	**Valid Values**
memory [heap]	DSP stats about memory heaps. See Table 63	RO	

packets [heap]	DSP stats about packets heaps. See Table 64	RO	

Table 62 DSP Statistics

Parameter	**Description**	**Permissions** **R-read, W-write**	**Valid Values**
heap_id	A numeric identifier of the memory heap	RO	0..65535
total_size	Total memory available in the heap	RO	0..65535
total_free_size	The current size of free memory in the heap	RO	0..65535
largest_free_size	The current largest contiguous free memory block in the heap	RO	0..65535
max_allocated_blocks	The maximum number of blocks allocated from this heap at any single point in time, during its lifetime	RO	0..65535
num_allocated_blocks	The total number of blocks currently allocated in the heap	RO	0..65535

Table 63 DSP Memory Statistics

Parameter	**Description**	**Permissions** **R-read, W-write**	**Valid Values**
heap_name	A string identifier of the memory heap	RO	String size 0..35
free_data_packets	The current number of free data packets which are available in the heap	RO	0..65535
zero_buffer_packets	The current number of free packets (with no buffers) which are available in the heap	RO	0..65535

packets_in_garbage	The current number of packets (with and without) data buffers which are in the garbage queue of the heap	RO	0..65535
data_buffer_thr_status	This field is set to indicate that the heap has hit the specified data buffer threshold. It can take values *clear* and *starving*. It is applicable only if the 'starvation data threshold' was specified during heap creation in stack code	RO	Enum (clear, starving)
data_buffer_starvation_cnt	Indicates the number of times the data buffer queue was starved of packets. This is set only if the heap was configured to use starvation queues in stack code	RO	0..65535
zero_data_buffer_thr_status	This field is set to indicate that the heap has hit the specified zero data buffer threshold. It can take values *clear* and *starving*. It is applicable only if the 'starvation zero buffer threshold' was specified during heap creation in stack code	RO	Enum (clear, starving)
zero_data_buffer_starvation_cnt	Indicates the number of times the zero buffer queue was starved of packets. This is set only if the heap was configured to use starvation queues in stack code	RO	0..65535

Table 64 DSP Packets Statistics

5.2.12.1 Display all statistics

show statistics cellNo

cellNo – cell entry

Operation	Successful Outcome
show statistics 0	statistics 0 mac aggregated dl_tpt : 554 kbps ul_tpt : 556 kbps dl_bler_tb1 : 558 ratio/100 dl_bler_tb2 : 560 ratio/100 ul_bler : 562 ratio/100 rnti 10 ul_snr_pusch : 273 ul_snr_pucch : 274 dl_snr : 275 dl_tpt : 276 kbps ul_tpt : 277 kbps dl_bler_tb1 : 278 ratio/100 dl_bler_tb2 : 279 ratio/100 ul_bler : 280 ratio/100 rnti 11 .... rlc aggregated dl_tpt : 580 kbps ul_tpt : 582 kbps pkt_whole_nack_recv : 584 pkt_seg_nack_recv : 588 rnti 10 dl_tpt : 290 kbps ul_tpt : 291 kbps pkt_whole_nack_recv : 292

```
        pkt_seg_nack_recv       : 293
    rnti 11
        ....
pdcp
    aggregated
        dl_tpt              : 574 kbps
        ul_tpt              : 576 kbps
        ul_pkt_lost         : 578
        dl_pkt_lost         : 580
    rnti 10
        dl_tpt          : 286 kbps
        ul_tpt          : 287 kbps
        ul_pkt_lost : 288
        dl_pkt_lost : 289
    rnti 11
        ....
s1u
    aggregated
        dl_tpt      : 572 kbps
        ul_tpt      : 570 kbps
    rnti 10
        dl_tpt : 285 kbps
        ul_tpt : 284 kbps
    rnti 11
        ....
rrc
    conn_req_recv                   : 268
    conn_setup_comp_nt_recv   : 266
    conn_reconf_comp_nt_recv  : 267
    conn_reestablishment_recv : 265
    rlf_reported_l2                 : 265
    ues_configured                  : 275
    ues_released                    : 274
```

```
global
   num_rach                        : 286
   num_msg2                        : 287
   num_msg3                        : 288
   num_pdcp_inactivity_timer : 289
   num_rlf_at_mac                  : 290
   num_rel_triggrd_at_mac       : 291
   ul_snr_pucch_ue_rel           : 292
   ul_snr_pusch_ue_rel           : 293
   dl_crc_ue_rel                   : 294
   rlc_num_rlf                      : 295
rnti_list
   ue 1
     rnti : 10
   ue 2
     rnti : 11
phy
   ue 1
      ul_cqi_ant0        : 4 dB
      ul_cqi_ant1        : 5 dB
      noise_var_ant0 : 6 dBm
      noise_var_ant1 : 7 dBm
      rssi_ant0           : 8 dBm
      rssi_ant1           : 9 dBm
   ue 2
      ....
dsp 0
   memory
      heap 0
         heap_id : 1
         total_size : 32768
         total_free_size : 32768
         largest_free_size : 32768
```

	max_allocated_blocks : 10
	num_allocated_blocks : 2
	   heap 1
	      ....
	  packets
	    heap 0
	      heap_name : “pktHeap_001”
	      free_data_packets : 99
	      zero_buffer_packets : 0
	      packets_in_garbage : 1
	      data_buffer_thr_status : clear(0)
	      data_buffer_starvation_cnt : 0
	      zero_data_buffer_thr_status : starving(1)
	      zero_data_buffer_starvation_cnt : 2
	    heap 1
	      ....
	dsp 1
	  memory
	    ....
	  packets
	    ....

5.2.12.2 Display specific category statistics

show statistics cellNo category

cellNo – cell entry

category- any stats category from Table 53

Operation	Successful Outcome

show statistics 0 mac	statistics 0 mac
	  aggregated
	    dl_tpt : 348 kbps
	    ul_tpt : 350 kbps
	    dl_bler_tb1 : 352 ratio/100
	    dl_bler_tb2 : 354 ratio/100

	ul_bler : 356 ratio/100 rnti 10 ul_snr_pusch : 170 ul_snr_pucch : 171 dl_snr : 172 dl_tpt : 173 kbps ul_tpt : 174 kbps dl_bler_tb1 : 175 ratio/100 dl_bler_tb2 : 176 ratio/100 ul_bler : 177 ratio/100 rnti 11 ul_snr_pusch : 172 ul_snr_pucch : 173 dl_snr : 174 dl_tpt : 175 kbps ul_tpt : 176 kbps dl_bler_tb1 : 177 ratio/100 dl_bler_tb2 : 178 ratio/100 ul_bler : 179 ratio/100

5.2.12.3 Display category wise aggregated statistics

show statistics cellNo category aggregated

cellNo – cell entry

category - any stats category from Table 53 except rnti_list, rrc, global,phy

Operation	**Successful Outcome**
show statistics 0 rlc aggregated	statistics 0 rlc aggregated dl_tpt : 412 kbps ul_tpt : 414 kbps pkt_whole_nack_recv : 416 pkt_seg_nack_recv : 418

5.2.12.4 Display category statistics for specific rnti

show statistics cellNo category rnti rntiValue cellNo – cell entry

category - any stats category from Table 53 except rnti_list, rrc, global

rntiValue- valid rnti present in the system

Operation	Successful Outcome
show statistics 0 rlc rnti 10	statistics 0 rlc rnti 10 dl_tpt : 18448 ul_tpt : 11392

5.2.12.5 Display a specific aggregated statistic in specific category

show statistics cellNo category aggregated statName

cellNo – cell entry

category - any stats category from Table 53 except rnti_list, rrc, global,phy

statName - any stat from Table 54(except SNR), Table 55, Table 56, Table 57

Operation	Successful Outcome
show statistics 0 pdcp aggregated ul_tpt	statistics 0 pdcp aggregated ul_tpt : 11384

5.2.12.6 Display a specific statistic in specific category for specific rnti

show statistics cellNo category rnti rntiValue statName

cellNo – cell entry

category - any stats category from Table 53 except rnti_list, rrc, global

rntiValue - valid rnti present in system

statName - any stat from Table 54, Table 55, Table 56, Table 57

Operation	Successful Outcome
show statistics 0 pdcp rnti 10 ul_tpt	statistics 0 pdcp rnti 10 ul_tpt : 11384

An example showing when no rnti 11 exists in the system.

Operation	Successful Outcome
show statistics 0 pdcp rnti 11 ul_tpt	>

5.2.12.7 Display a specific statistic in specific category (rrc, global)

show statistics cellNo category statName

cellNo – cell entry

category- rrc,global stats category from Table 53

statName-any stat from Table 58, Table 59

Operation	Successful Outcome
show statistics 0 rrc conn_setup_comp_nt_recv	statistics 0 rrc conn_setup_comp_nt_recv : 1

5.2.12.8 Display statistics in rnti_list category

show statistics cellNo rnti_list ue ueNum

cellNo – cell entry

ueNum- ue entry

Operation	Successful Outcome
show statistics 0 rnti_list ue 1	statistics 0 rnti_list ue 1 rnti : 22

5.2.12.9 Display a specific statistic in rnti_list category

show statistics cellNo rnti_list ue ueNum statName

cellNo – cell entry

ueNum - ue entry

statName - any stat from Table 60

Operation	Successful Outcome
show statistics 0 rnti_list ue 1 rnti	statistics 0 rnti_list ue 1 rnti : 22

5.2.12.10 Display full DSP stats

show statistics cellNo dsp

cellNo – cell entry

Operation	Successful Outcome

<table>
<tr>
<td>show statistics 0 dsp</td>
<td><pre>
statistics 0 dsp
  dsp 0
    memory
      heap 0
        heap_id : 1
        total_size : 32768
        total_free_size : 32768
        largest_free_size : 32768
        max_allocated_blocks : 10
        num_allocated_blocks : 2
      heap 1
        ....
    packets
      heap 0
        heap_name : “pktHeap_001”
        free_data_packets : 99
        zero_buffer_packets : 0
        packets_in_garbage : 1
        data_buffer_thr_status : clear(0)
        data_buffer_starvation_cnt : 0
        zero_data_buffer_thr_status : starving(1)
        zero_data_buffer_starvation_cnt : 2
      heap 1
        ....
  dsp 1
    memory
      ....
    packets
      ....
</pre></td>
</tr>
</table>

5.2.12.11 Display full stats in a specific DSP

show statistics cellNo dsp dspNum

cellNo – cell entry

dspNum – DSP number

Operation	Successful Outcome
show statistics 0 dsp 1	statistics 0 dsp 1 memory heap 0 heap_id : 1 total_size : 32768 total_free_size : 32768 largest_free_size : 32768 max_allocated_blocks : 10 num_allocated_blocks : 2 heap 1 .... packets heap 0 heap_name : “pktHeap_001” free_data_packets : 99 zero_buffer_packets : 0 packets_in_garbage : 1 data_buffer_thr_status : clear(0) data_buffer_starvation_cnt : 0 zero_data_buffer_thr_status : starving(1) zero_data_buffer_starvation_cnt : 2 heap 1 ....

5.2.12.12 Display specific category stats in a DSP

show statistics cellNo dsp dspNum category

cellNo – cell entry

dspNum – DSP number

category – memory, packets

Operation	Successful Outcome
show statistics 0 dsp 1	statistics 0 dsp 1 memory

memory	heap 0 heap_id : 1 total_size : 32768 total_free_size : 32768 largest_free_size : 32768 max_allocated_blocks : 10 num_allocated_blocks : 2 heap 1 ....

5.2.12.13 Display specific heap stats in specific category in a DSP

show statistics cellNo dsp dspNum category heapNo

cellNo – cell entry

dspNum – DSP number

category – memory, packets

heapNo – heap number

Operation	Successful Outcome
show statistics 0 dsp 1 memory 1	statistics 0 dsp 1 memory 1 heap 1 heap_id : 2 total_size : 32768 total_free_size : 32768 largest_free_size : 32768 max_allocated_blocks : 10 num_allocated_blocks : 2

5.2.12.14 Display specific statistic in heap in specific category in a DSP

show statistics cellNo dsp dspNum category heapNo statName

cellNo – cell entry

dspNum – DSP number

category – memory, packets

heapNo – heap number

statName – any stat in Table 63, Table 64

Operation	Successful Outcome
show statistics 0 dsp 1 memory 1 total_size	statistics 0 dsp 1 memory 1 total_size : 32768

5.2.13 Alarms

The klish interface is not forced to obey to a standard MIB, filled with default values for unsupported parameters, therefore it exposes only relevant information. This makes it much simpler than the SNMP interface.

The *alarm* command groups the following contexts:

Parameter	Description	Permissions R-read, W-write	Valid Values
model	Specifies the available alarm states models. See Table 66	RO	255 octets
active	Specifies active alarms. See Table 67	RO	255 octets
stats	Specifies active alarm statistics. See Table 68	RO	

Table 65 Alarm Categories

Parameter	Description	Permissions R-read, W-write	Valid Values
desc	Describes the alarm state in readable format.	RO	String size 0..255

Table 66 Alarm Model Table

Parameter	Description	Permissions R-read, W-write	Valid Values
desc	Provides a textual description of the active alarm.	RO	String size 0..255

Table 67 Active Alarm Table

Parameter	Description	Permissions R-read, W-write	Valid Values
current_active	Total number of currently active alarms on the system.	RO	0..4294967295
total_active	Total number of active alarms since system restarted.	RO	0..4294967295
last_raise	Value of *sysUpTime* at the time of the last alarm raise for this alarm list.	RO	0..4294967295
last_clear	Value of *sysUpTime* at the time of the last alarm clear for this alarm list.	RO	0..4294967295

Table 68 Active Alarm Stats

5.2.13.1 Alarm models table

5.2.13.1.1 Display alarms model

show alarms model enb

Operation	Successful Outcome

show alarms model enb	alarms model enb
	id 1
	  state 1
	    desc: sctp clear
	  state 2
	    desc: sctp alarmed
	id 2
	  state 1
	    desc: x2ap setup failure clear
	  state 2
	    desc: x2ap setup failure alarmed
	id 3
	  state 1
	    desc: x2ap unknown neighbour clear
	  state 2

	desc: x2ap unknown neighbour alarmed id 4 state 1 desc: max cell capacity clear state 2 desc: max cell capacity alarmed

5.2.13.1.2 Display alarm model parameter

show alarms model enb at index index_no

index_no : alarm model index (Range: 1..4)

Operation	Successful Outcome
show alarms model enb 1	show alarms model enb 1 alarms model enb 1 state 1 desc: sctp clear state 2 desc: sctp alarmed

show alarms model enb at index index_no and state state_no.

index_no : alarm model index (Range: 1..4)

state_no : alarm model state (Range: 1..2)

Operation	Successful Outcome
show alarms model enb 1 1	alarms model enb 1 1 desc: sctp clear

show alarms model enb at index index_no and state state_no paramName

index_no : alarm model index (Range: 1..4)

state_no : alarm model state (Range: 1..2)

paramName - any parameter from Table 66 with read permission.

Operation	Successful Outcome
show alarms model enb 1 1 desc	alarms model enb 1 1 desc: sctp clear

5.2.13.2 Active alarms

5.2.13.2.1 Display active alarm

show alarms active enb

Assuming that currently x2ap unknown neighbor and x2ap setup failure alarm is raised.

Operation	Successful Outcome
show alarms active enb	alarms active enb x2ap unknown neighbour alarmed x2ap setup failure alarmed

5.2.13.2.2 Display active alarm parameter

show alarms active enb paramName

paramName - any parameter from Table 67 with read permission.

Assuming that currently x2ap unknown neighbor and x2ap setup failure alarm is raised.

Operation	Successful Outcome
show alarms active enb desc	alarms active enb x2ap unknown neighbour alarmed x2ap setup failure alarmed

5.2.13.3 Active alarm stats

5.2.13.3.1 Display active alarm stats

show alarms stats enb

Operation	Successful Outcome
show alarms stats enb	alarms stats enb current_active : 3 total_active : 4 last_raise : (192) 0:00:01.92

	last_clear : (36) 0:00:00.36

5.2.13.3.2 Display a specific active alarm stat

show alarms stats enb alarmStatName

alarmStatName specifies any parameter from Table 68

Operation	**Successful Outcome**
show alarms stats enb current_active	alarms stats enb current_active : 2