

MiVoice MX-ONE

MiVoice MX-ONE SNMP Support Alarm Notification and Emergency Call Events

Release 7.4 SP1

April 22, 2022

Notice

The information contained in this document is believed to be accurate in all respects but is not warranted by **Mitel Networks™ Corporation (MITEL®)**. The information is subject to change without notice and should not be construed in any way as a commitment by Mitel or any of its affiliates or subsidiaries. Mitel and its affiliates and subsidiaries assume no responsibility for any errors or omissions in this document. Revisions of this document or new editions of it may be issued to incorporate such changes. No part of this document can be reproduced or transmitted in any form or by any means - electronic or mechanical - for any purpose without written permission from Mitel Networks Corporation.

Trademarks

The trademarks, service marks, logos and graphics (collectively "Trademarks") appearing on Mitel's Internet sites or in its publications are registered and unregistered trademarks of Mitel Networks Corporation (MNC) or its subsidiaries (collectively "Mitel") or others. Use of the Trademarks is prohibited without the express consent from Mitel. Please contact our legal department at legal@mitel.com for additional information. For a list of the worldwide Mitel Networks Corporation registered trademarks, please refer to the website: <http://www.mitel.com/trademarks>.

®, ™ Trademark of Mitel Networks Corporation
© Copyright 2022, Mitel Networks Corporation
All rights reserved

Contents

Chapter: 1	General	1
Chapter: 2	Glossary	2
Chapter: 3	Prerequisites	3
Chapter: 4	SNMP Introduction	4
	Trap	5
	Query	5
Chapter: 5	MiVoice MX-ONE SNMP	7
	Mivoice MX-ONE SNMP agent introduction	7
	MiVoice MX-ONE SNMP Implementation	8
	MIVOICE MX-ONE SNMP tsAlarmR1 OID IMPLEMENTATION . . .	8
	MIVOICE MX-ONE SNMP tsObjects OID IMPLEMENTATION	10
	MiVoice MX-ONE SNMP license mechanism	11
Chapter: 6	MIB Support	13
	Initial MIB support	13
	Partial Support of Ericsson MIBs	13
	Supported objects in deprecated Ericsson MD110 SNMP MIB . .	13
	Supported traps in deprecated Ericsson MD110 SNMP MIB . .	13
	Full Support of MX-ONE-TS-ALARM-MIBs	14
	Supported objects in MX-ONE-TS-ALARM-MIB	14
	Supported traps in MX-ONE-TS-ALARM-MIB	15
	MIVOICE MX-ONE SNMP MIB detail description	16
	System Status	16
	Interfaces	17
	Traps	26
	SUPPORT OF MITEL ERN MIB	28
	System Status	28

	Traps29
	SUPPORTED WRITEBLE OBJECTS IN MITEL ERN MIB30
Chapter: 7	Configuration	31
	Configuration of the SNMP Daemon31
	System Information31
	CONFIGURING USING <i>mxone_maintenance</i> TOOL FOR SNMP v2	31
	CONFIGURING USING <i>mxone_maintenance</i> TOOL FOR SNMP v3	32
	Additional SNMP Traps33
	Configuration of Alarm Notification Using E-mail and SMS34
	Enabling Mail from SNMP Traps34
	Proxy traps to the same destination34
	Disabling Trap Daemon34
	Restart Trap Daemon34
Chapter: 8	Verify the Installation	35
Chapter: 9	Using net-snmp Tools to Check Alarms	36
	Verifying the Deprecated Ericsson MIB36
	Verifying the Mitel ALARM and Status MIB36
	Verifying ALARM and Status trap sending36
	Example37
	VERIFYING EMERGENCY CALL EVENT TRAP SENDING37
	Example37

General

The MX-ONE Service Node can be supervised from a Simple Network Management Protocol (SNMP) management system and configured to send alarm notifications by snmp traps or optionally by e-mail or text messaging using Short Messaging Service (SMS).

This document describes the MX-ONE Service Node SNMP support and how the basic configuration of alarm notification is set up.

Note: MiVoice MX-ONE supports SNMPv1, SNMPv2c and SNMPv3 MIBs for monitoring and traps. The MX-ONE SNMP agent does not include provisioning or configuration from an NMS. Provisioning of MX-ONE shall be accomplished using Provisioning Manager and Service Node Manager or alternatively via an SSL CLI interface.

There are three snmp MIB implementations available in the Service Node; the old deprecated Ericsson MIB, the Mitel MIB for monitoring status and alarms, and the Mitel ERN MIB for emergency call events.

You can monitor MX-ONE Service Node from an SNMP management system, using one or all three of them.

The SNMP implementation in the MX-ONE Service Node is based on *net-snmp*. More documentation can be found in the `/usr/share/doc/packages/net-snmp` directory or at <http://www.net-snmp.org>.

Net-snmp is one of the packages installed along with the operating system on the MX-ONE Service Node. During installation, the net-snmp daemon is configured to use the AGENT_X protocol.

Note: The programs ALSNMP (Ericsson MIB) and AALSNMP (Mitel Status MIB) and ESNMP (Mitel emergency notification MIB) are part of the telephony system, and communicate with the snmp daemon via the AGENT_X protocol. The ALSNMP and ESNMP are not loaded by default, and needs to be loaded using the `pu_add` command, if the Ericsson MIB format or the Mitel emergency notifications are used.

The snmp daemon (snmpd) is installed in each LIM (node) of the MX-ONE. Consequently, the network management systems needs to get traps from all nodes individually to get the complete system view.

In the normal setup each node sends traps directly to network management systems but may be configured to route traps through a proxy to the network management systems. The snmpd daemon has a default setup and is operational after installation, but requires additional configuration for the traps to be sent to the management system.

The snmptrapd daemon is also be used for additional features like proxy agent for traps and to log traps to local disk. These features needs additional configuration.

The snmptrapd daemon program is installed, but is not activated as default, thus to enable e-mail, SMS or other features additional configuration is required.

Glossary

ERN

Emergency call Routing Notification (MIB variant)

ISO

International Standardization Organization

MIB

Management Information Base

NMS

Network Management Station

OID

(Managed) Object Identifier (of a managed object)

RFC

Request For Comments, IETF SIP Standards or drafts

SMI

Structure of Management Information

SNMP

Simple Network Management Protocol

For other abbreviations, see ACRONYMS, ABBREVIATIONS AND GLOSSARY.

Prerequisites

The MiVoice MX-ONE system must run.

Since the legacy Ericsson MIB SNMP format, and Mitel emergency notifications are regarded as an optional functions, the ALSNMP or ESNMP SW programs must be manually loaded before using the functionality desired. Only the Mitel status MIB (AALSNMP) is automatically loaded at installation.

See the ADMINISTRATOR USER'S GUIDE, and the INSTALLING AND CONFIG-URING MIVOICE MX-ONE, section Optional programs.

SNMP Introduction

The SNMP (Simple Network Management Protocol) is an Internet-standard protocol used to manage devices on IP networks.

As defined in the RFC 1157, "Network management stations execute management applications which monitor and control network elements. Network elements are devices such as hosts, gateways, terminal servers, and the like, which have management agents responsible for performing the network management functions requested by the network management stations. The Simple Network Management Protocol (SNMP) is used to communicate management information between the network management stations and the agents in the network elements."

SNMP is composed by three components:

- Managed devices or network elements
- Agent - software that runs on managed device that defines what can be monitored and controlled.
- Network Management Station (NMS) - software that runs on management system that are responsible for polling (query) and receiving traps from agents, process the received information and take the proper action.

SNMP protocol is used to exchange information between the monitored device and NMS.

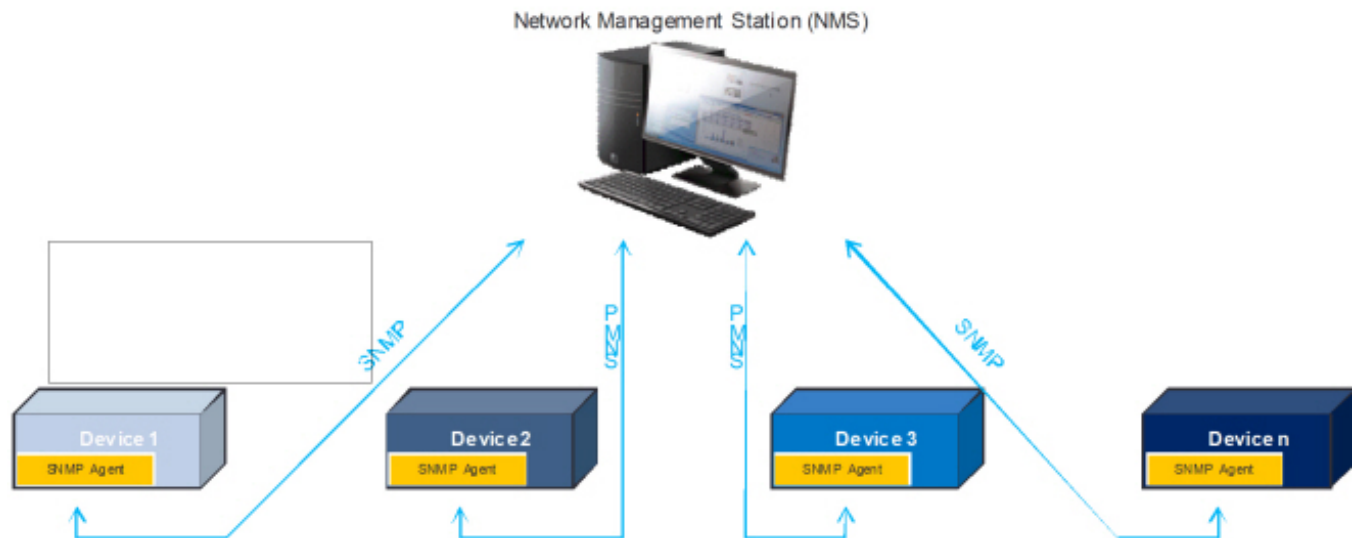


Figure 1: Network Management Station managing 4 devices

Basically, there are two methods that SNMP agents can communicate with Network Management Station, receiving a Trap or requesting information via Queries.

Trap

The agent sends Traps when a status change occurs in the monitored device. The Trap is received by Network Management Station (NMS) that can present the information in a graphical format, if properly configured, as shown in the figure below.



Figure 2:Trap

Query

Query

A query can be requested from the NMS, for example from a MIB Browser to collect data from the monitored device.



Figure 3:Query

The SNMP protocol does not define which information a managed device should provide. Indeed, SNMP uses management information bases (MIBs) to define the available information. MIBs is described as a database of managed objects where monitored information such as interface status, etc is defined.

MIBs use a hierarchical tree-structured database. The Structure of Management Information Version 1 (SMIv1) defines how managed objects are named and specifies their associated information.

A unique name or object identifier (OID) is used to identify a managed object, where each has a numerical OID and an associated textual name, as the following example:

OID Name	aastraOidMxone (8)
Numerical	.1.3.6.1.4.1.11268.2.8
Textual	.iso.org.dod.internet.private.enterprises.aastra.aastraMibs.aastraOidMx-one or iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).aastra(11268).aastraMibs(2).aastraOidMx-one(8).

The figure below shows the SMI representation for the Mitel's aastraOidMxone OID.

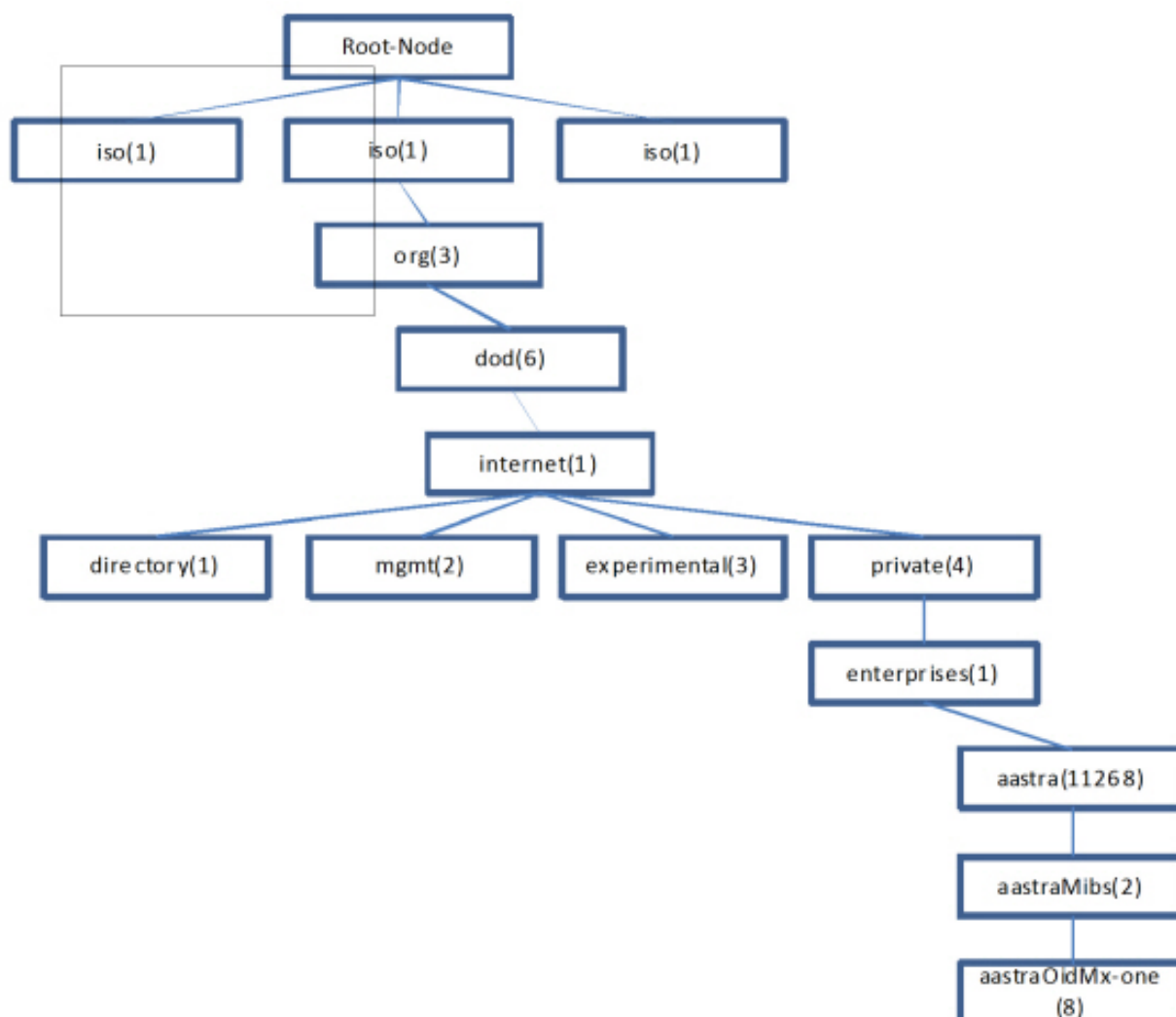


Figure 4: Structure of Management Information tree, aastraOidMxone

The supplier of the monitored device needs to provide a MIB (Management information base) file that defines what information is available from that specific device.

MiVoice MX-ONE SNMP

MiVoice MX-ONE SNMP

Mivoice MX-ONE SNMP agent introduction

MX-ONE is a device running on IP networks supports SNMP that means the MX-ONE Service Node is supervised by a SNMP network management station or network node manager.

MX-ONE Service Node can be configured to send alarm notifications by SNMP traps or optionally by e-mail or text messaging using Short Messaging Service (SMS).

The SNMP implementation in the MX-ONE Service Node is based on net-snmp. Net-snmp is one of the packages installed along with the operating system on the MX-ONE Service Node. During installation, the net-snmp daemon is configured to use the AGENT_X protocol. The program AALSNMP (Mitel MIB) is part of the telephony system and communicates with the SNMP daemon via the AGENT_X protocol. The SNMP daemon (snmpd) is installed in each server (node) of the MX-ONE. Consequently, the network management station needs to get traps from all nodes individually in order to get the complete system view.

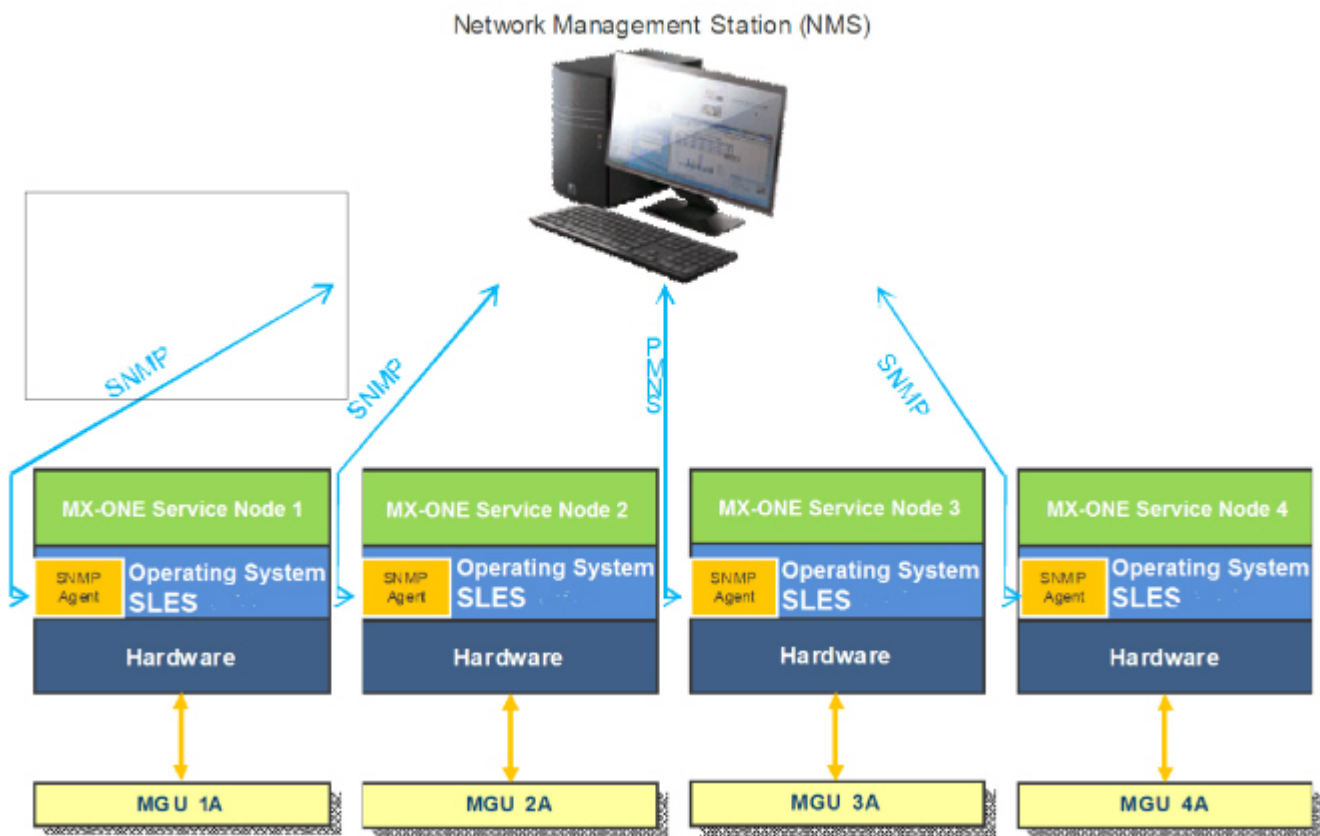


Figure 5:SNMP Agents in managed systems

MiVoice MX-ONE SNMP Implementation

MIVOICE MX-ONE SNMP tsAlarmR1 OID IMPLEMENTATION

The tsAlarmR1OID tree represents the following MX-ONE system alarms:

iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).astra(11268).astraMibs(2).astraOidMx-one(8).tsAlarm(1).tsAlarmR1(1)

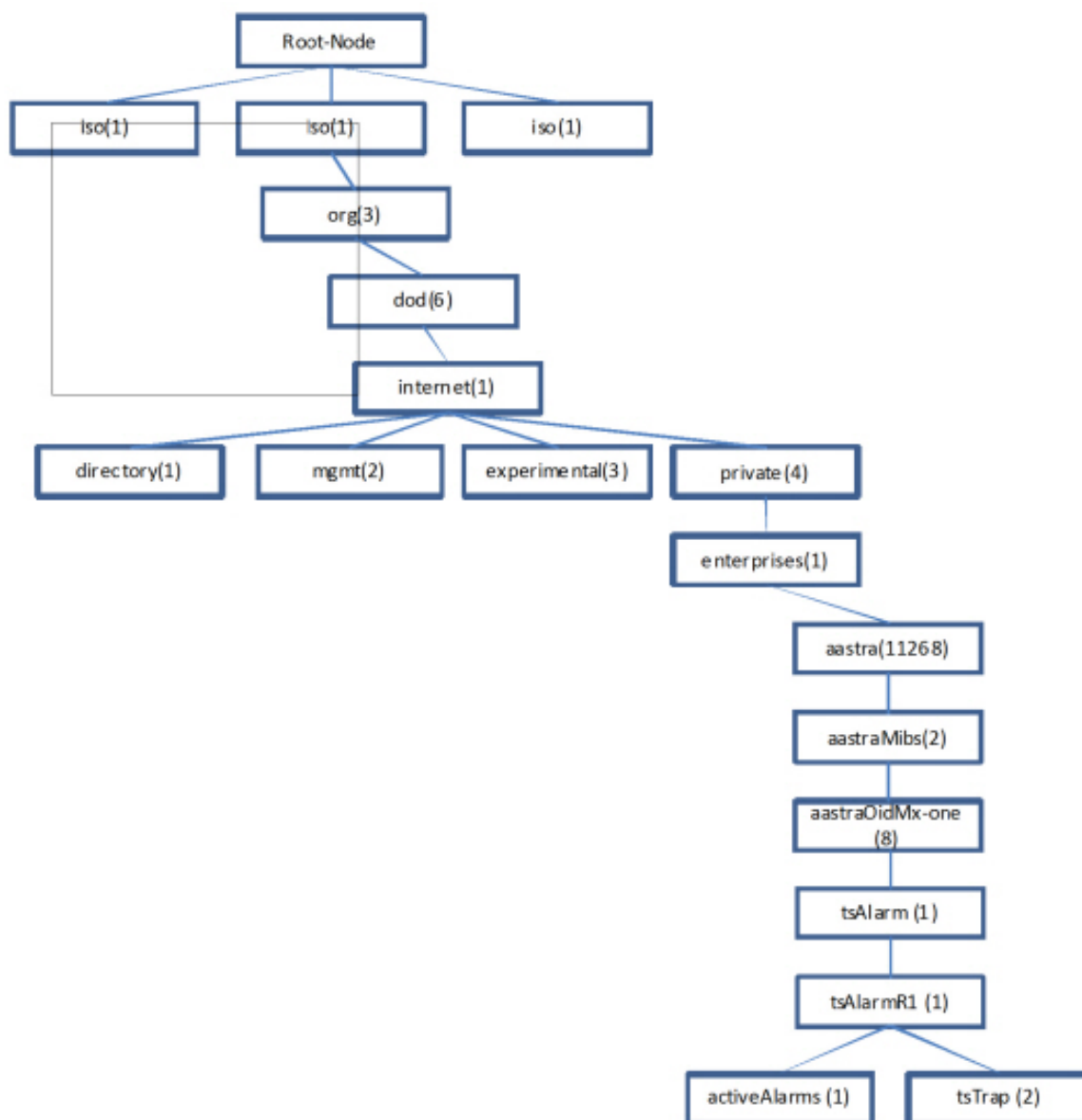


Figure 6: Structure of Management Information tree, TsAlarmR1

MX-ONE SNMP agent sends system alarms traps to a NMS when an alarm occurs in the system if it configured properly.

The Network Management System is customized to understand the information provided by MX-ONE SNMP agent.

You can use query to fetch alarm information from MX-ONE Service Node.



Figure 7: Trap

The figure below is an example of MX-ONE Alarms in one Network Management system that is customized to work with MX-ONE.

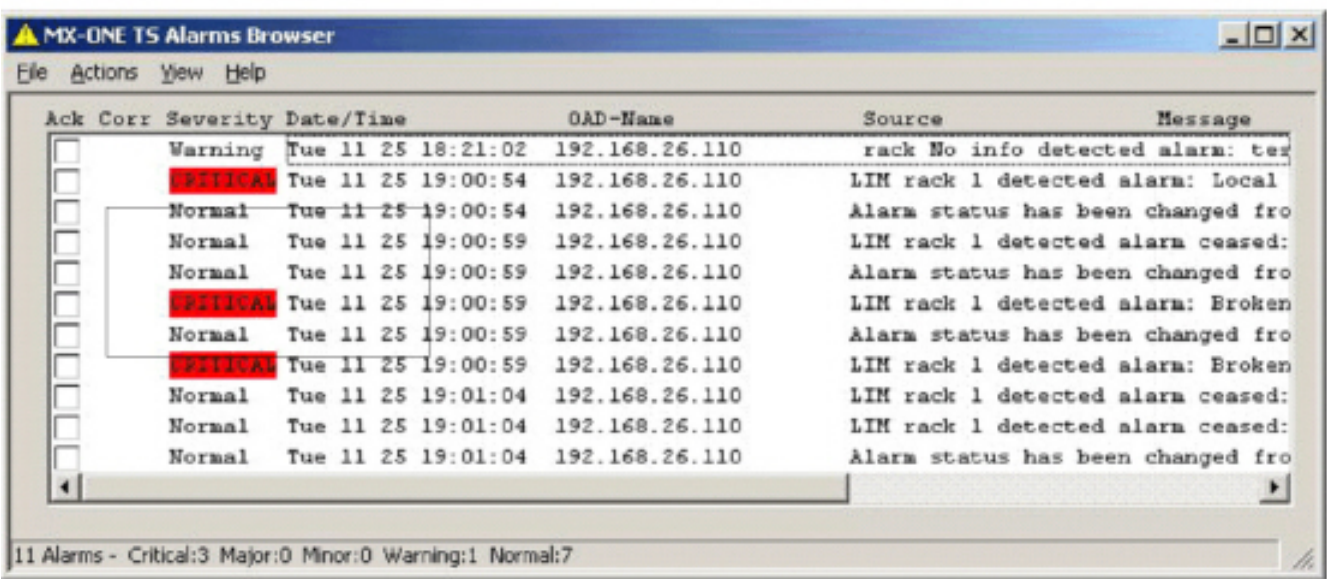


Figure 8: Structure of Management Information tree, TsAlarm

Note: NMS is customized to display MX-ONE information.

MIVOICE MX-ONE SNMP tsObjects OID IMPLEMENTATION

These news objects are represented by the tsObjects OID tree.

iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).astra(11268).astraMibs(2).astraOidMx-one(8).tsAlarm(1).tsObjects(2)

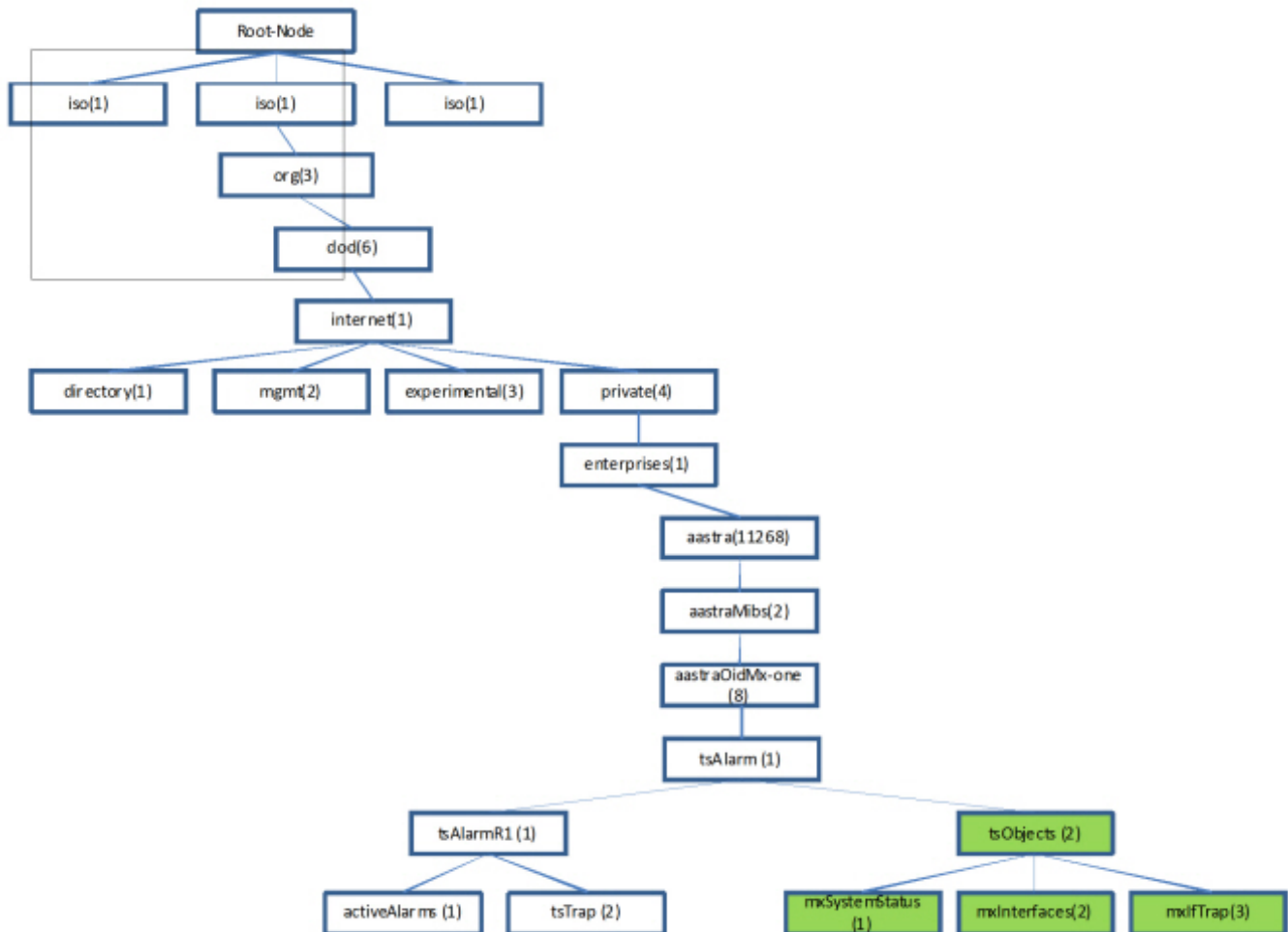


Figure 9: Structure of Management Information tree, TsObjects

MxSystemStatus and mxInterfaces, provides MX-ONE information regarding the items below. NNM can query this information. They are read-only.

mxSystemStatus(1)

- MX-ONE Release
- System Status

– Number of Servers, IPs, Backup Status and Info, GSM Sync info, MX-ONE Service Node Status.

mxInterfaces(2)

- Trunk

- Type, Description, Bearer Cap, Route number, Trunk Identification (TRU), EQU, Status.
- Operator
- Directory Number, Type, EQU, Operation and Busy Status.
- Call Information Logging
- Output, Type, Subtype, Dbname, Server, Operational Status.
- Computer Telephony Integration phase 3
- CTI Group, Server, IP, Operational Status.
- Computer Telephony Integration phase 1
- CTI Group, Server, IP, Operational Status.
- Interception Computer, ICU
- Individual, Type, Interface, Server Number, Information, Operational Status.
- Gateway
- Type, Identification, Description, Operational Status, IP.
- Media Gateways
- Type, Identification, Description, Operational Status, IP.
- Inter Server Signaling
- GJU side: Type, Identification, Remote Side, Description, Status.
- GSM side: Identification, Synchronism, Operational Status A and B, Active side.
- CAS Boards
- Type, Identification, Description, Operational Status.

MIB Trap, when there is a change on the status of the following items MX-ONE Service Node sends a trap to the NNM.

mxTrap(3)

- Trunk
- Operator
- Call Information Logging
- Computer Telephony Integration 1
- Computer Telephony Integration 3
- Interception Computer
- Gateway
- Media Gateway
- Inter Server Signaling
- Voice Cards
- CAS Boards

MiVoice MX-ONE SNMP license mechanism

The SNMP license in MX-ONE is divided into following two parts, they are:

Basic

It allows MX-ONE to send alarm information via traps to a NMS and the system administrator can query the system to get some alarm information (tsAlarmR1 in the screen below).

Advanced

It allows MX-ONE to send alarm and object information via trap to a NMS and the system administrator can query the system to get some alarm/object information (tsAlarmR1 and tsObjects in the screen below).

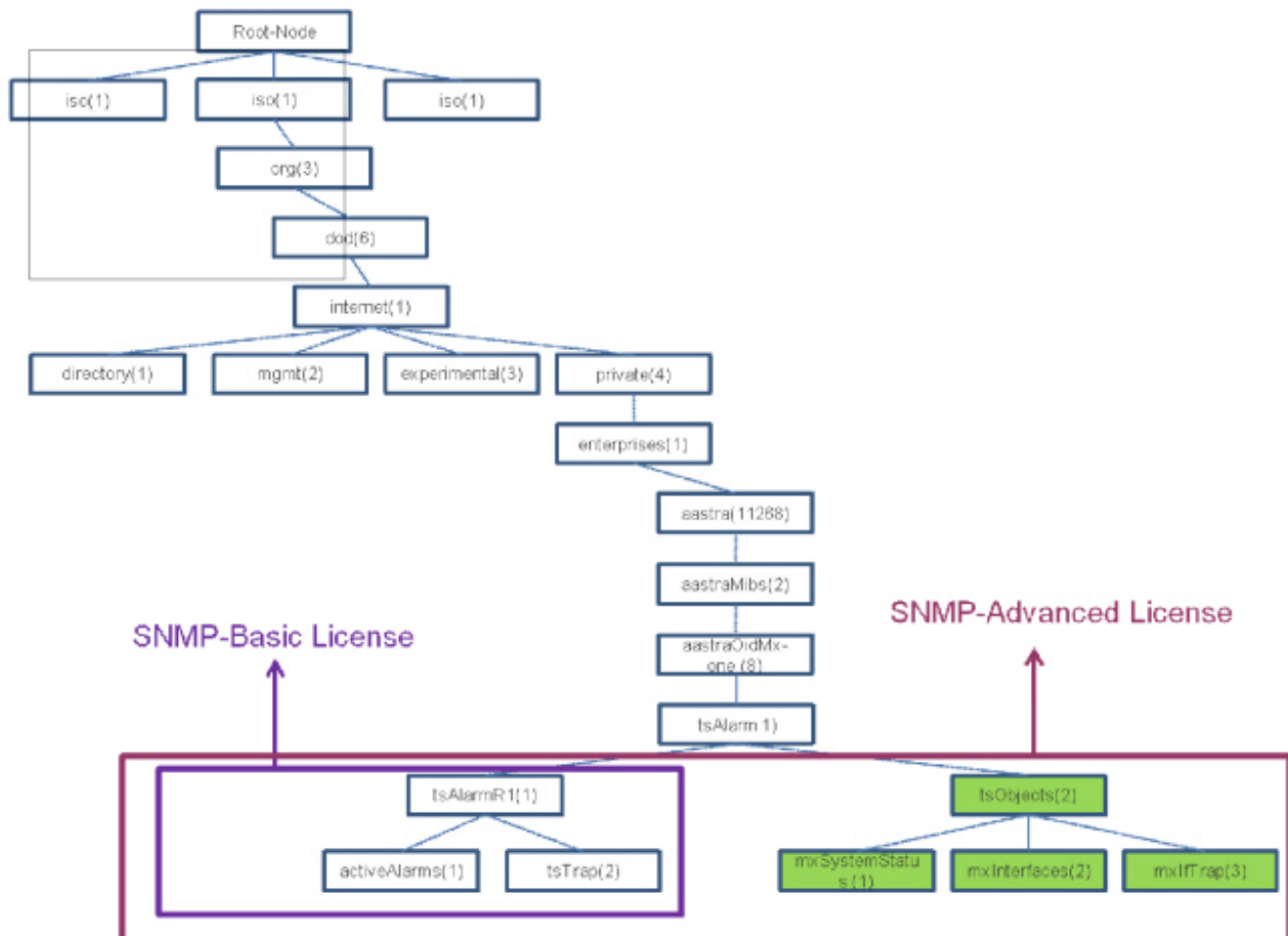


Figure 10: SNMP License mechanism

MIB Support

Initial MIB support

- MIB-2 – General network statistics (RFC 1213).
- Net-snmp agent extensions – Processes, disks, memory, load average, shell commands, and error handling.
- SNMPv3 MIBS (RFC 2571-6).
- Partial support of Ericsson-MD110-SNMP-MIB, using ALSNMP.
- Full support of Mitel MX-ONE-TS-ALARM-MIB, using AALSNMP.

The administrator can add additional MIBs, but it must be configured separately.

Partial Support of Ericsson MIBs

The definitions are found in an installed system at: /usr/share/snmp/mibs/Ericsson-DNA-SNMP-MIB.txt and /usr/share/snmp/mibs/Ericsson-MD110-SNMP-MIB.txt.

Implementation in program unit ASNMP.

Supported objects in deprecated Ericsson MD110 SNMP MIB

In the MX-ONE Service Node, only the following objects in the MD110 SNMP MIB are supported with read-only access:

	OID	Description
md110Release	.1.3.6.1.4.1.193.8.1.1	
mdAlarmStatus	.1.3.6.1.4.1.193.8.1.3.1.2	
activeAlarms	.1.3.6.1.4.1.193.8.1.3.2.1.1.1-10000	

Supported traps in deprecated Ericsson MD110 SNMP MIB

	OID	Description
mdAlarmCritical	.1.3.6.1.4.1.193.8.1.0.1	
mdAlarmMajor	.1.3.6.1.4.1.193.8.1.0.2	

	OID	Description
mdAlarmMinor	.1.3.6.1.4.1.193.8.1.0.3	
mdAlarmWarning	.1.3.6.1.4.1.193.8.1.0.4	
mdAlarmCease	.1.3.6.1.4.1.193.8.1.0.5	
mdAlarmStatusChange	.1.3.6.1.4.1.193.8.1.0.2000	

Full Support of MX-ONE-TS-ALARM-MIBs

The definitions are found in an installed system at: /usr/share/snmp/mibs/MX-ONE-TS-ALARM-MIB.txt or in Operational directions MX-ONE TS Alarm MIB.

Implementation in program unit AASNMP.

Supported objects in MX-ONE-TS-ALARM-MIB

Table 1Alarm OIDs

	OID	Description
tsAlarmR1	.1.3.6.1.4.1.11268.2.8.1.1	
activeAlarms	.1.3.6.1.4.1.11268.2.8.1.1.1	
mxalStatus	.1.3.6.1.4.1.11268.2.8.1.1.1.1	See below *)
activeAlarmTable	.1.3.6.1.4.1.11268.2.8.1.1.1.2	A table of active alarms in the MX-ONE node.
ActiveAlarmEntry	.1.3.6.1.4.1.11268.2.8.1.1.1.2.1	
mxalHandle	.1.3.6.1.4.1.11268.2.8.1.1.1.2.1.1	Handle of an MX-ONE alarm. An alarm instance is identified by its handle and the lim number.
mxalFrom	.1.3.6.1.4.1.11268.2.8.1.1.1.2.1.2	Sending lim (and unit) related to the alarm.
mxalFaultCode	.1.3.6.1.4.1.11268.2.8.1.1.1.2.1.3	Fault Code of an MX-ONE alarm. Indicates type of alarm. Fault code is domain:code (within domain). Example:Fault code 5:8.
mxalSeverity	.1.3.6.1.4.1.11268.2.8.1.1.1.2.1.4	See below *)

	OID	Description
mxalWhere	.1.3.6.1.4.1.11268.2.8.1.1.2.1.5	Faulty unit etc related to the alarm.
mxalExplanation	.1.3.6.1.4.1.11268.2.8.1.1.2.1.6	Textual description associated with the alarm.
mxalNoticed	.1.3.6.1.4.1.11268.2.8.1.1.2.1.7	Indicates if the alarm is noticed. May indicate that someone is working on the problem.
mxalNoticedNote	.1.3.6.1.4.1.11268.2.8.1.1.2.1.8	Comment added to the alarm when noticed.

*) The current alarm status of the MX-ONE node

- Indeterminate - Status unknow
- Critical - Any active alarms with severity 4,
- Major - Any active alarms with severity 3,
- Minor - Any active alarms with severity 2,
- Warning - Any active alarms with severity 1,
- Normal - Any active alarms with severity 0, or no active alarms.

Supported traps in MX-ONE-TS-ALARM-MIB

Table 2Trap OIDs

	OID	Description
tsTrap	.1.3.6.1.4.1.11268.2.8.1.1.2	
tsTrapV2	.1.3.6.1.4.1.11268.2.8.1.1.2.0	
mxAlarmCritical	.1.3.6.1.4.1.11268.2.8.1.1.2.0.1	This trap is sent when an MX-ONE alarm with severity 4 is detected.
mxAlarmMajor	.1.3.6.1.4.1.11268.2.8.1.1.2.0.2	This trap is sent when an MX-ONE alarm with severity 3 is detected.
mxAlarmMinor	.1.3.6.1.4.1.11268.2.8.1.1.2.0.3	This trap is sent when an MX-ONE alarm with severity 2 is detected.

	OID	Description
mxAlarmWarning	.1.3.6.1.4.1.11268.2.8.1.1.2.0.4	This trap is sent when an MX-ONE alarm with severity 1 is detected.
mxAlarmSysClear	.1.3.6.1.4.1.11268.2.8.1.1.2.0.5	This trap is sent when an MX-ONE alarm is cleared by the system.
mxAlarmOpClear	.1.3.6.1.4.1.11268.2.8.1.1.2.0.6	This trap is sent when an MX-ONE alarm is cleared by the operator.
mxAlarmStatusChange	.1.3.6.1.4.1.11268.2.8.1.1.2.0.2000	This trap is sent when the alarm status of the lim is changed.

MIVoice MX-ONE SNMP MIB detail description

The definitions are found in an installed system at: /usr/share/snmp/mibs/MX-ONE-TS-ALARM-MIB.txt.
Implementation in program unit ESNMP.

System Status

System Status

System Status Information

iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).astra(11268).astra-Mibs(2).astraOidMx-one(8).tsAlarm(1).tsObjects(2).mxSystemStatus(1)

System Status (mxSystemStatus)

	OID	Description
mxSystemStatus	.1.3.6.1.4.1.11268.2.8.1.2.1	
mxRelease	.1.3.6.1.4.1.11268.2.8.1.2.1.1.0	Indicates MX-ONE software release. This data is provided from the software interface and cannot be changed.
mxLimTable	.1.3.6.1.4.1.11268.2.8.1.2.1.2	A table with data for the interfaces.
mxLimEntry	.1.3.6.1.4.1.11268.2.8.1.2.1.2.1	Internal use. No information is provided.

	OID	Description
mxLimIndex	.1.3.6.1.4.1.11268.2.8.1.2.1.2.1.1	Defines the MX-ONE Service Node number.
mxLimWhere	.1.3.6.1.4.1.11268.2.8.1.2.1.2.1.2	Presents MX-ONE Service Node host name, if defined in the DNS.
mxLimIp	.1.3.6.1.4.1.11268.2.8.1.2.1.2.1.3	The currently IP address in use by the MX-ONE Service Node.
mxLimBackupStatus	.1.3.6.1.4.1.11268.2.8.1.2.1.2.1.4	MX-ONE Service Node backup operational status.
mxLimBackupInfo	.1.3.6.1.4.1.11268.2.8.1.2.1.2.1.5	Additional information regarding MX-ONE Service Node backup. Date and hour of the current backup, if it is valid.
mxLimStatus	.1.3.6.1.4.1.11268.2.8.1.2.1.2.1.7	MX-ONE Service Node operational state (running, manual-blocked, system-blocked, blocked, unknown). Unknown result if the MX-ONE Service Node connected to is isolated.
mxObjectStatus	.1.3.6.1.4.1.11268.2.8.1.2.1.3.0	The traffic state of the object.

Interfaces

Route and Trunk information

iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).astra(11268).astra-Mibs(2).astraOidMx-one(8) tsAlarm(1).tsObjects(2).mxInterfaces(2).mxIfTrunk(2)

Interfaces (mxInterfaces)

	OID	Description
mxInterfaces	.1.3.6.1.4.1.11268.2.8.1.2.1	
mxIfTrunk	.1.3.6.1.4.1.11268.2.8.1.2.2.2	Indicates MX-ONE software release. This data is provided from the software interface and cannot be changed.

	OID	Description
mxRouteTable	.1.3.6.1.4.1.11268.2.8.1.2.2.2.1	A table with Route information.
mxRouteEntry	.1.3.6.1.4.1.11268.2.8.1.2.2.2.1.1	Internal use. No information is provided.
mxRouteIndex	.1.3.6.1.4.1.11268.2.8.1.2.2.2.1.1.1	The route number.
mxRouteType	.1.3.6.1.4.1.11268.2.8.1.2.2.2.1.1.2	Type of route, public or tie-line. Signalling system used. Example: Public route, signal system = ISDN.
mxRouteDescr	.1.3.6.1.4.1.11268.2.8.1.2.2.2.1.1.3	Route type. See MML TYPE of interface.
mxRouteBearerCap	.1.3.6.1.4.1.11268.2.8.1.2.2.2.1.1.4	Bearer capabilities of the route Corresponds to the route parameter BCAP. The positions is a string of 0 and 1. Each position of the string corresponds to a Bearer capability attribute, A position set to 1 means that the corresponding capability is provided. pos 1 :64 kbps Unrestricted pos 2 :64 kbps Restricted pos 3 :3.1 kHz Audio pos 4 :Speech pos 5 :7 kHz Audio pos 6 :16 kbit/s unrestr. digital
mxTrunkTable	.1.3.6.1.4.1.11268.2.8.1.2.2.2.2	A table with trunk interfaces data.
mxTrunkEntry	.1.3.6.1.4.1.11268.2.8.1.2.2.2.2.1	Internal use. No information is provided.

	OID	Description
mxTrunkIndex	.1.3.6.1.4.1.11268.2.8.1.2.2.2.2.1.1	A unique value for each interface. Its value is calculated as: $X = \text{route number}$, $Y = \text{lim number}$, $Z = \text{trunk individual within lim and route}$. $\text{mxTrunkIndex} = (X * 2097152) + (Y * 4096) + Z$
mxTrunkId	.1.3.6.1.4.1.11268.2.8.1.2.2.2.2.1.2	Trunk line number. Server and sequence number for the external line.
mxTrunkRouteNo	.1.3.6.1.4.1.11268.2.8.1.2.2.2.2.1.3	The route number. Linkin to mxRouteTable.
mxTrunkWhere	.1.3.6.1.4.1.11268.2.8.1.2.2.2.2.1.4	EQU position. Note that EQU is not presented to SIP trunks.
mxTrunkDescr	.1.3.6.1.4.1.11268.2.8.1.2.2.2.2.1.5	Trunk type. See MML TYPE of interface.
mxTrunkAddDescr	.1.3.6.1.4.1.11268.2.8.1.2.2.2.2.1.6	Future development. A textual string containing information about the remote end.
mxTrunkOperStatus	.1.3.6.1.4.1.11268.2.8.1.2.2.2.2.1.7	The traffic state of the interface (idle, busy, blocked, unknown).

Operator or attendant information

iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).astra(11268).astraMibs(2).astraOidMx-one(8).tsAlarm(1).tsObjects(2).mxInterfaces(2).mxIfOperator(3)

Interfaces (mxInterfaces)

	OID	Description
mxIfOperator	.1.3.6.1.4.1.11268.2.8.1.2.2.3	
mxOpiTable	.1.3.6.1.4.1.11268.2.8.1.2.2.3.1	A table with attendant / operator data.
mxOpiEntry	.1.3.6.1.4.1.11268.2.8.1.2.2.3.1.1	Internal use. No information is provided.

	OID	Description
mxOpiIndex	.1.3.6.1.4.1.11268.2.8.1.2.2.3.1.1.1	A unique value for each Attendant / Operator.
mxOpiDirno	.1.3.6.1.4.1.11268.2.8.1.2.2.3.1.1.2	A textual string displaying the directory number.
mxOpiType	.1.3.6.1.4.1.11268.2.8.1.2.2.3.1.1.3	Operator console type, see Type Parameter in CPI PBX Operator Traffic, OP document.
mxOpiWhere	.1.3.6.1.4.1.11268.2.8.1.2.2.3.1.1.4	Port identity, EQU position or IP address.
mxOpiDescr	.1.3.6.1.4.1.11268.2.8.1.2.2.3.1.1.5	Future development. A textual string containing information about the interface.
mxOpiAddDescr	.1.3.6.1.4.1.11268.2.8.1.2.2.3.1.1.6	Future development. A textual string containing information about the interface.
mxOpiOperStatus	.1.3.6.1.4.1.11268.2.8.1.2.2.3.1.1.7	The operational state of the attendant (manual-absent(1), automatic-absent(2), present(3), blocked(4), unknown(5))
mxOpiBusyStatus	.1.3.6.1.4.1.11268.2.8.1.2.2.3.1.1.8	The traffic state of the attendant (idle, busy or unknown).

Call Information Logging

iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).astra(11268).astra-Mibs(2).astraOidMx-one(8).tsAlarm(1).tsObjects(2).mxInterfaces(2).mxIfCil(5)

Interfaces (mxInterfaces)

	OID	Description
mxIfCil	.1.3.6.1.4.1.11268.2.8.1.2.2.5	
mxCilTable	.1.3.6.1.4.1.11268.2.8.1.2.2.5.1	A table with CIL output data.
mxCilEntry	.1.3.6.1.4.1.11268.2.8.1.2.2.5.1.1	Internal use. No information is provided.

	OID	Description
mxCilIndex	.1.3.6.1.4.1.11268.2.8.1.2.2.5.1.1.1	A unique value for each output.
mxCilOutput	.1.3.6.1.4.1.11268.2.8.1.2.2.5.1.1.2	A unique value for each output. Its value is a string in format XXX-Y. XXX = MX-ONE Service Node number, Y =output number.
mxCilType	.1.3.6.1.4.1.11268.2.8.1.2.2.5.1.1.3	Type of storage.
mxCilSubtype	.1.3.6.1.4.1.11268.2.8.1.2.2.5.1.1.4	Subtype of storage.
mxCilDbName	.1.3.6.1.4.1.11268.2.8.1.2.2.5.1.1.5	Path or device for the storage.
mxCilServer	.1.3.6.1.4.1.11268.2.8.1.2.2.5.1.1.6	Storage server information.
mxCilOperStatus	.1.3.6.1.4.1.11268.2.8.1.2.2.5.1.1.7	The operational state of the interface (up, down or unknown). The unknown(3) state indicates that the agent can't get the status of the interface. The interface can then be either up or down.

CSTA I and CSTA III information

iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).astra(11268).astra-Mibs(2).astraOidMx-one(8) tsAlarm(1).tsObjects(2).mxInterfaces(2).mxIfCsta(6)

Interfaces (mxInterfaces)

	OID	Description
mxIfCsta	.1.3.6.1.4.1.11268.2.8.1.2.2.6	
mxCti1Table	.1.3.6.1.4.1.11268.2.8.1.2.2.6.1	A table with CSTA I data.

	OID	Description
mxCti1Entry	.1.3.6.1.4.1.11268.2.8.1.2.2.6.1.1	Internal use. No information is provided
mxCti1Index	.1.3.6.1.4.1.11268.2.8.1.2.2.6.1.1.1	A unique value for each link group.
mxCti1Group	.1.3.6.1.4.1.11268.2.8.1.2.2.6.1.1.2	CSTA group name.
mxCti1Lim	.1.3.6.1.4.1.11268.2.8.1.2.2.6.1.1.3	MX-ONE Service Node number where the interface is initiated.
mxCti1Ip	.1.3.6.1.4.1.11268.2.8.1.2.2.6.1.1.4	MX-ONE Service Node IP address where CSTA is initiated.
mxCti1OperStatus	.1.3.6.1.4.1.11268.2.8.1.2.2.6.1.1.5	The operational state of the interface (up, down, faulty, unknown). The unknown state indicates that the agent can't get the status of the interface. The interface can then be either up or down
mxCti3Table	.1.3.6.1.4.1.11268.2.8.1.2.2.6.2	A table with CSTA III data
mxCti3Entry	.1.3.6.1.4.1.11268.2.8.1.2.2.6.2.1	Internal use. No information is provided.
mxCti3Index	1.3.6.1.4.1.11268.2.8.1.2.2.6.2.1.1	Internal use. No information is provided.
mxCti3Type	1.3.6.1.4.1.11268.2.8.1.2.2.6.2.1.2	Information of type of protocol used for this interface. ASN1 or XML.
mxCti3Lim	1.3.6.1.4.1.11268.2.8.1.2.2.6.2.1.3	MX-ONE Service Node number where the interface is initiated.
mxCti3Ip	1.3.6.1.4.1.11268.2.8.1.2.2.6.2.1.4	IP address, only presented if the Service is up.

	OID	Description
mxCti3Port	1.3.6.1.4.1.11268.2.8.1.2.2.6.2.1.5	Port number.
mxCti3OperStatus	1.3.6.1.4.1.11268.2.8.1.2.2.6.2.1.6	The operational state of the interface (UnInitialized, Initialized, Enabled, Disabled, NotExist) . The unknown state indicates that the agent can't get the status of the interface. The interface can then be either up or down

System Computer information (ICU):

iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).astra(11268).astra-Mibs(2).astraOidMx-one(8) .tsAlarm(1).tsObjects(2).mxInterfaces(2).mxIfGici(7)

Interfaces (mxInterfaces)

	OID	Description
mxIfGici	.1.3.6.1.4.1.11268.2.8.1.2.2.7	
mxIcuTable	.1.3.6.1.4.1.11268.2.8.1.2.2.7.1	A table with Information System Computer connections.
mxIcuEntry	.1.3.6.1.4.1.11268.2.8.1.2.2.7.1.1	Internal use. No information is provided.
mxIcuIndex	.1.3.6.1.4.1.11268.2.8.1.2.2.7.1.1.1	A unique value for each information system.
mxIculfcInd	.1.3.6.1.4.1.11268.2.8.1.2.2.7.1.1.2	Information computer individual (Ifcind).
mxIcuType	.1.3.6.1.4.1.11268.2.8.1.2.2.7.1.1.3	Information of type of interface (IFC-initiated, ICS, VS-F, EM, ACD-MIS, ANCD, Unknown).
mxIcuInterface	.1.3.6.1.4.1.11268.2.8.1.2.2.7.1.1.4	Type of interface. e.g. V.24 or Ethernet.

	OID	Description
mxlcuLim	.1.3.6.1.4.1.11268.2.8.1.2.2.7.1.1.5	MX-ONE Service Node number where the information system computer is connected. Not used =0.
mxlcuInfo	.1.3.6.1.4.1.11268.2.8.1.2.2.7.1.1.6	IP address, port number or V.24 data.
mxlcuOperStatus	.1.3.6.1.4.1.11268.2.8.1.2.2.7.1.1.7	The operational state of the interface. The unknown(3) state indicates that the agent can't get the status of the interface. The interface can then be either up or down, also used for generic type.

Gateways Information

iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).aastra(11268).aastra-Mibs(2).aastraOidMx-one(8).tsAlarm(1).tsObjects(2).mxInterfaces(2).mxIfSwitch(9)

Interfaces (mxInterfaces)

	OID	Description
mxIfSwitch	.1.3.6.1.4.1.11268.2.8.1.2.2.9	
mxMgwTable	.1.3.6.1.4.1.11268.2.8.1.2.2.9.1	A table with Information System Computer connections.
mxMgwEntry	.1.3.6.1.4.1.11268.2.8.1.2.2.9.1.1	Internal use. No information is provided.
mxMgwIndex	.1.3.6.1.4.1.11268.2.8.1.2.2.9.1.1.1	A unique value for each information system.
mxMgwWhere	.1.3.6.1.4.1.11268.2.8.1.2.2.9.1.1.2	Information computer individual (Ifcind).
mxMgwType	.1.3.6.1.4.1.11268.2.8.1.2.2.9.1.1.3	Information of type of interface (IFC-initiated, ICS, VS-F, EM, ACD-MIS, ANCD, Unknown).
mxMgwDescr	.1.3.6.1.4.1.11268.2.8.1.2.2.9.1.1.4	Type of interface. e.g. V.24 or Ethernet.

	OID	Description
mxMgwOperStatus	.1.3.6.1.4.1.11268.2.8.1.2.2.9.1.1.5	The operational state of the interface. The unknown(3) state indicates that the agent can't get the status of the interface. The interface can then be either up or down.
mxGateWayTable	.1.3.6.1.4.1.11268.2.8.1.2.2.9.2	A table with media gateway data.
mxGateWayEntry	.1.3.6.1.4.1.11268.2.8.1.2.2.9.2.1	Internal use. No information is provided.
mxGateWayIndex	.1.3.6.1.4.1.11268.2.8.1.2.2.9.2.1.1	A unique value for each interface.
mxGateWayType	.1.3.6.1.4.1.11268.2.8.1.2.2.9.2.1.2	Information of type of interface for this interface, e.g. MGU, LSU_E or MS.
mxGateWayDescr	.1.3.6.1.4.1.11268.2.8.1.2.2.9.2.1.4	IP address of the gateway board.
mxGateWayOperStatus	.1.3.6.1.4.1.11268.2.8.1.2.2.9.2.1.5	The operational state of the interface. The unknown(3) state indicates that the agent can't get the status of the interface. The interface can then be either up or down.

CAS Boards information

iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).astra(11268).astraMibs(2).astraOidMx-one(8).tsAlarm(1).tsObjects(2).mxInterfaces(2).mxIfCasBoards(12)

Interfaces (mxInterfaces)

	OID	Description
mxIfCasboards	.1.3.6.1.4.1.11268.2.8.1.2.2.12	
mxCasBoardTable	.1.3.6.1.4.1.11268.2.8.1.2.2.12.1	A table with CAS boards data.
mxCasBoardEntry	.1.3.6.1.4.1.11268.2.8.1.2.2.12.1.1	Internal use. No information is provided.

	OID	Description
mxCasBoardIndex	.1.3.6.1.4.1.11268.2.8.1.2.2.12.1.1.1	A unique value for each interface.
mxCasBoardData	.1.3.6.1.4.1.11268.2.8.1.2.2.12.1.1.2	Type of configuration, corresponds to the EL7 ICAT configuration see command EXTEI.
mxCasBoardWhere	.1.3.6.1.4.1.11268.2.8.1.2.2.12.1.1.3	The equipment position of the CAS board.
mxCasBoardDirno	.1.3.6.1.4.1.11268.2.8.1.2.2.12.1.1.4	A textual string directory number.
mxCasBoardOperStatus	.1.3.6.1.4.1.11268.2.8.1.2.2.12.1.1.5	The operational state of the interface (idle, busy, blocked, unknown). The unknown(3) state indicates that the agent can't get the status of the interface. The interface can then be either up or down.

Traps

Traps

The table below shows the IODs that are available in the mxTraps.

One trap is sent when MX-ONE status for one of these objects is changed.

	OID	Description
mxIfTrap	.1.3.6.1.4.1.11268.2.8.1.2.3	
mxObjectBackupStatusChange	.1.3.6.1.4.1.11268.2.8.1.2.3.1	This trap is sent when MX-ONE status for this object is changed. The following information is presented: mxLimBackupStatus, *)
mxObjectTrunkChange	.1.3.6.1.4.1.11268.2.8.1.2.3.2	This trap is sent when MX-ONE status for this object is changed. The following information is presented: mxTrunkIndex, *)

	OID	Description
mxObjectOpiChange	.1.3.6.1.4.1.11268.2.8.1.2.3.3	This trap is sent when MX-ONE status for this object is changed. The following information is presented: mxOpilIndex, *)
mxObjectCilChange	.1.3.6.1.4.1.11268.2.8.1.2.3.4	This trap is sent when MX-ONE status for this object is changed. The following information is presented: mxCilIndex, *)
mxObjectCti1Change	.1.3.6.1.4.1.11268.2.8.1.2.3.5	This trap is sent when MX-ONE status for this object is changed. The following information is presented: mxCti1Index, *)
mxObjectCti3Change	.1.3.6.1.4.1.11268.2.8.1.2.3.6	This trap is sent when MX-ONE status for this object is changed. The following information is presented: mxCti3Index, *)
mxObjectIcuChange	.1.3.6.1.4.1.11268.2.8.1.2.3.7	This trap is sent when MX-ONE status for this object is changed. The following information is presented: mxIcuIndex, *)
mxObjectMgwChange	.1.3.6.1.4.1.11268.2.8.1.2.3.8	This trap is sent when MX-ONE status for this object is changed. The following information is presented: mxMgwIndex, *)
mxObjectGatewayChange	.1.3.6.1.4.1.11268.2.8.1.2.3.9	This trap is sent when MX-ONE status for this object is changed. The following information is presented: mxGateWayIndex, *)
mxObjectGjuChange	.1.3.6.1.4.1.11268.2.8.1.2.3.10	This trap is sent when MX-ONE status for this object is changed. The following information is presented: mxGjuIndex, *)

	OID	Description
mxObjectGsmChange	.1.3.6.1.4.1.11268.2.8.1.2.3.11	This trap is sent when MX-ONE status for this object is changed. The following information is presented: mxGsmIndex, *)
mxObjectGsmSideChange	.1.3.6.1.4.1.11268.2.8.1.2.3.12	This trap is sent when MX-ONE status for this object is changed. The following information is presented: mxGsmActiveSide, *)
mxObjectVcuChange	.1.3.6.1.4.1.11268.2.8.1.2.3.13	Obsolete, VCU is not supported. This trap was sent when MX-ONE status for this object was changed. The following information was presented: mxVcuIndex, *)
mxObjectCasBoardChange	.1.3.6.1.4.1.11268.2.8.1.2.3.14	This trap is sent when MX-ONE status for this object is changed. The following information is presented: mxCasBoardIndex, *)

*) Additionally, the following information is also presented mxObjectStatus, mxal-Handle, mxalFrom, mxalFaultCode, mxalSeverity, mxalWhere, mxalExplanation, mxal-Noticed, mxalNoticedNote.

SUPPORT OF MITEL ERN MIB

The definitions are found in an installed system at: /usr/share/snmp/mibs/MITEL-MIB.txt and /usr/share/snmp/mibs/MITEL-ERN-MIB.txt.

Implementation in program unit ESNMP.

System Status

System Status

Supported Objects in MITEL ERN MIB

iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).mitel(1027). mitelProprietary (4). mitelPropApplications (1). mitelAppCallServer(1)

System Status (mxSystemStatus)

	OID	Description
mitelCsEmergencyResponse	1.3.6.1.4.1.1027.4.1.1.3	The branch to support ER Adviser.
mitelCsErSeqNumber	.1.3.6.1.4.1.1027.4.1.1.3.1	Sequence number, assigned for each new emergency call since last (re-)start.
mitelCsErCallType	.1.3.6.1.4.1.1027.4.1.1.3.2	Always 1.
mitelCsErDetectTime	.1.3.6.1.4.1.1027.4.1.1.3.3	Time of the emergency call in local time.
mitelCsErCallingDN	.1.3.6.1.4.1.1027.4.1.1.3.4	The DN of the device used to place the emergency call.
mitelCsErCallingPNI	.1.3.6.1.4.1.1027.4.1.1.3.5	The Primary Node ID for the caller. E.g "Lim x", where x is the service node number.
mitelCsErCesidDigits	.1.3.6.1.4.1.1027.4.1.1.3.6	The ELIN/CESID from the calling device if applicable.
mitelCsErDialledDigits	.1.3.6.1.4.1.1027.4.1.1.3.7	The number dialed, typically 911 is the US.
mitelCsErRegistrationDN	.1.3.6.1.4.1.1027.4.1.1.3.8	Extension number of the device that made the call.
mitelCsErUnackTableIndex	.1.3.6.1.4.1.1027.4.1.1.3.9.1	The index of the row for the Manager to acknowledge the notification.
mitelCsErUnackTableToken	.1.3.6.1.4.1.1027.4.1.1.3.9.2	The status of this row. Normally 1.

Traps

Traps

	OID	Description
mitelCsErNotification	.1.3.6.1.4.1.1027.4.1.1.3.401	A new trap is sent for each new emergency call, and repeated every two minutes, for two hours, if not cleared. In the trap following data is presented: mitelCsErSeqNumber, mitelCsErCallType, mitelCsErDetectTime, mitelCsErCallingDN, mitelCsErCallingPNI, mitelCsErCesidDigits, mitelCsErDialledDigits, mitelCsErRegistrationDN, mitelCsErUnackTableIndex, mitelCsErUnackTableToken.

SUPPORTED WRITEBLE OBJECTS IN MITEL ERN MIB

SUPPORTED WRITEBLE OBJECTS IN MITEL ERN MIB

In the MX-ONE Service Node, following objects in the MITEL ERN MIB are supported with write-only access.

	OID	Description
mitelCsErUnackTable	.1.3.6.1.4.1.1027.4.1.1.3.9.1.2.0	
mitelCsErUnackTableToken	.1.3.6.1.4.1.1027.4.1.1.3.9.1.2.0-63	Remove repetitive sending of emergency traps, by writing the integer received by the NMS in the mitelCsErSeqNumber in the mitelCsErNotification trap.

Configuration

SNMP is pre-configured at installation and can be reconfigured by using the *mxone_maintenance* script, however some part needs to be manually added by editing the configuration files in the */etc/snmp/* directory, or by the tools provided in the (*/usr/bin/snmpconf*) distribution.

Configurations file are found at path: */etc/snmp/*

Example files are found at path: */opt/eri_sn/etc/templates*

Other binary files: */opt/eri_sn/bin/snmpAction.pl*

Optional binary file: */usr/bin/traptoemail*

Configuration of the SNMP Daemon

The SNMP agent used in the MX-ONE Service Node is a daemon called *snmpd*, the associated configuration file is */etc/snmp/snmpd.conf*.

System Information

The agent is configured to fit the installation by editing the configuration file. At a minimum, the following entries should be updated in every LIM:

- *syslocation* – Physical location of the system
- *syscontact* – Contact information for the administrator of the system

NOTE: The above information must be manually edited in the */etc/snmp/snmpd.conf* file.

CONFIGURING USING *mxone_maintenance* TOOL FOR SNMP v2

Start the *mxone_maintenance* tool with command:

```
sudo -H mxone_maintenance
```

Navigate to SNMP section:

```
more_configurations -> snmp > V2 -> add
```

[illegible]

Configure according to the specific information:

snmp manager ip address	trap2sink - More than one trap monitoring system can be defined (one per line)
snmp readonly community string	rocommunity - The community name to allow read access
snmp readwrite community string	rwcommunity - The community name to allow write access
snmp trap community string	trapcommunityname - The community name that will accompany the sent trap
snmp proxy address	common trap sender set the trap2sink

At installation the default trap2sink is defined to *localhost*, thus, sending all traps to the *snmptrapd* running on the same MX-ONE node.

Additional trap2sinks may be added to send information to several trap monitors or management systems. This is done by adding one line for each new trap monitor or management system.

If the system needs to have a common trap sender set the trap2sink in all lims to the same address, e.g.: lim 1. Then let the snmptrapd daemon in lim 1 forward (proxv) the traps to network management centre.

CONFIGURING USING *mxone maintenance* TOOL FOR SNMP v3

Start the *mxone maintenance* tool with command:

```
sudo -H mxone maintenance
```

Navigate to SNMP section:

more configurations -> snmp > V3 -> add

NOTE: The SNMPv3 can be defined for the whole system or a specific server. The latter can be useful for adding a server or repairing of a server.

The following data must be entered:

Field Name	Description
SNMPv3 manager IP address	Address of the monitoring system
SNMPv3user name	User name in the monitoring system
Select System	<ul style="list-style-type: none"> • MPA System • Emergency system (e.g. Revolution) • Standard System
Select security level	<ul style="list-style-type: none"> • noAuthNoPriv • authNoPriv • authPriv
Select authority protocol	<ul style="list-style-type: none"> • MD5 • SHA-1
Enter password	must be at least 8 characters long
Select privacy protocol	<ul style="list-style-type: none"> • DES • AES-128
Enter password	must be at least 8 characters long

NOTE: Depending on selected security level, one or more fields are not necessary.

Additional traps like disk monitoring can be enabled by changing the configuration by editing the `/etc/snmp/snmpd.conf` file and restart of the `net-snmp` daemon with the `systemctl restart snmpd` command, see manual pages `man snmpd.conf`.

Configuration of Alarm Notification Using E-mail and SMS

The MX-ONE Service Node can be configured to send alarm notifications by e-mail or directly or via Short Messaging Service (SMS). The notices are based on the SNMP daemon **snmptrapd**. The daemon receives and logs SNMP TRAP messages, then translates them into an e-mail format. A public e-mail to SMS server can be used to forward the notification to a mobile phone using SMS.

Enabling Mail from SNMP Traps

Mail notification is configured by editing the `/etc/snmp/snmptrapd.conf` file. Enter the destination e-mail address(es) for the notifications after uncommenting the desired trap(s).

For additional information regarding configuration of this feature, type: **man snmptrapd.conf**.

To enable service to be started at next boot, run `systemctl` and enable `snmptrapd`. To start the daemon after configuration is complete, run `systemctl` and start `snmptrapd`.

Proxy traps to the same destination

If network management centre requires only one trapsender from the system use the `snmptrapd` proxy functionality.

Set the `snmp` proxy address to server 1's IP-address in the `mxone_maintenace` tool.

Disabling Trap Daemon

If the `snmptrapd` functionality to send mail or SMS is not required, then delete any line in the `trap2sink` section, containing `localhost` in the `/etc/snmp/snmptrapd.conf` file.

Stop the service by `systemctl stop snmptrapd`, and to disable the service type `systemctl disable snmptrapd`.

Restart Trap Daemon

Use command `systemctl restart snmptrapd` to restart the service when reconfiguration is done.

Verify the Installation

1. Verify that these files exist depending on what MIBs are used:

2. Alarms using Ericsson MIBs.

`/usr/share/snmp/mibs/Ericsson-DNA-SNMP-MIB.txt`

`/usr/share/snmp/mibs/Ericsson-MD110-SNMP-MIB.txt`

3. Alarm and status using Mitel (Aastra) alarm MIBs.

`/usr/share/snmp/mibs/MX-ONE-TS-ALARM-MIB.txt`

4. Emergency call events using Mitel ERN MIBs.

`/usr/share/snmp/mibs/MITEL-MIB.txt`

`/usr/share/snmp/mibs/MITEL-ERN-MIB.txt`

5. Verify the `/etc/snmp/snmp.conf` file:

Check that the following entries are present.

`mibs +Ericsson-DNA-SNMP-MIB:Ericsson-MD110-SNMP-MIB`

`mibs +MX-ONE-TS-ALARM-MIB`

`mibs +MITEL-MIB`

`mibs +MITEL-ERN`

6. Verify started net-snmp daemons.

Check status of the daemons

`# chkconfig snmp`

`# chkconfig snmptrap`

If a daemon is not running use `commandchkconfig xxxx on` to enable service to be started at next boot, and use `command /etc/init.d/xxxx start` to start the service.

Using net-snmp Tools to Check Alarms

By use of command **snmpwalk**, you can perform a check to see if the daemon is running and in contact with the ALSNMP/AALSNMP program(s).

Verifying the Deprecated Ericsson MIB

Use command **alarm -i** to create a test alarms with different severities.

E.g. **alarm -i -C 666 -D 1 --alarm-severity 4 --alarm-text "Test alarm"\
--faulty-lim 1 --add-text "I did this"**

snmpwalk -v 2c -c public -m all localhost 1.3.6.1.4.1.193.8

At least the *md110Release* and *mdAlarmStatus* objects should be printed, and option-ally any *activeAlarms* object.

Use command **alarm -e** to remove alarm when ready.

E.g. **alarm -e -C 666 -D 1**

Verifying the Mitel ALARM and Status MIB

Use command **alarm -i** to create a test alarms with different severities.

E.g. **alarm -i -C 666 -D 1 --alarm-severity 4 --alarm-text "Test alarm"\
--faulty-lim 1 --add-text "I did this"**

Use command **alarm -e -C 666 -D 1**

snmpwalk -v 2c -c public -m all localhost 1.3.6.1.4.1.11268.2.8.1.1.1 will only print active alarms.

Use command **alarm -e** to remove alarm when ready.

E.g. **alarm -e -C 666 -D 1**

See also the commands: **snmpget**, **snmpgetnext**, and **snmptable**.

Verifying ALARM and Status trap sending

To verify that the alarms are forwarded you may use alarm commands to create, change and remove alarm.

Hints where to check: **/var/log/messages**

/var/log/net-snmp.log

/var/log/mail

Example

Use command `alarm -i` to create a test alarms with different severities.

```
alarm -i -C 666 -D 1 --alarm-severity 4 --alarm-text "Test alarm"\
--faulty-lim 1 --add-text "I did this"
```

Use command `alarm -p` to verify that the alarm exist, and to get the alarm handle.

```
alarm -p -C 666 -D 1
```

Check that a trap was sent to management system and/or mail system.

Use command `alarm_noticed --alarm-handle XX` to change noticed state of the alarm.

```
alarm_noticed --alarm-handle XX
```

Check that a trap was sent to management system and/or mail system.

Use command `alarm -e --alarm-handle XX` to remove alarm.

```
alarm -e --alarm-handle XX
```

Check that a trap was sent to management system and/or mail system.

VERIFYING EMERGENCY CALL EVENT TRAP SENDING

You may need to start the `snmptrapd` daemon to get the traps printed to the `net-snmp.log`.

Hints where to check: `/var/log/net-snmp.log`

Example

Make an emergency call (E.g. 112 or 911 call).

You may need to initiate an alternative destination with a different number to test the feature.

Be sure that the destination is initiated with the ADC D26=1 set to trigger the emergency call event trap sending.

Check that a trap was received at the management system and/or mail system.

Use command: `“diagnostic_print -lim ALL -unit ESNMP -request 2”` to print the list of emergency events to verify that the call is in the list.

Use command: `“snmpset -v1 -c <auth> <pbx-ip> .1.3.6.1.4.1.1027.4.1.1.3.9.1.2.<mitelCsErUnackTableIndex> i <mitelCsErSe-qNumber>”` to remove one entry from the list.

Use command: `“diagnostic_print -lim ALL -unit ESNMP -request 2”` to print the list of emergency events to verify the removed entry is gone.

Additional tests with calls should be done per lim.

Also check that the repetition is done by delaying the `snmpset` command.

Remove the entry from the list by procedure in the NMS application.

