VadaTech ATCA

Core SNMP Interface Reference Manual

May 20, 2009

Version 2.0

vadatech inc

THE POWER OF VISION

Notice

While reasonable efforts have been made to assure the accuracy of this document, VadaTech, Inc. assumes no liability resulting from any omissions in this document or from the use of the information obtained herein. VadaTech reserves the right to revise this document and to make changes periodically and the content hereof without obligation of VadaTech to notify any person of such revision or changes.

Electronic versions of this material may be read online, downloaded for personal use, or referenced in another document as a URL to the VadaTech Incorporated Web site. The text itself may not be published commercially in print or electronic form, edited, translated, or otherwise altered without the permission of VadaTech, Inc.

It is possible that this publication may contain reference to or information about VadaTech products (machines and programs), programming, or services that are not available in your country. Such references or information must not be construed to mean that VadaTech intends to announce such products, programming, or services in your country.

Trademarks

The VadaTech, Inc name and logo are registered trademarks of VadaTech Incorporated in the U.S.A. All other product or service names mentioned in this document are the property of their respective owners.

Revision History

Doc Rev	Description of Change	Revision Date
1.0	Document Created	02/23/2009
2.0	Updated for MCH release 1.4.0	05/20/2009

Table of Contents

Figures

Tables

1 Overview

This document describes the Core SNMP Interface, which defines the MIBs and interface elements that are common among all VadaTech platform SNMP Interfaces. For platform-specific SNMP Interfaces, refer to the platform's SNMP extension document.

1.1 Document References

- Intelligent Platform Management (IPMI) Interface Specification v2.0
- PICMG® 3.0 Revision 3.0 AdvancedTCA® Base Specification
- PICMG® AMC.0 R2.0 Advanced Mezzanine Card Base Specification
- RFC 1155 Structure and Identification of Management Information for TCP/IP-based Internets
- RFC 1157 Simple Network Management Protocol
- RFC 1212 Concise MIB Definitions
- RFC 1901 (SNMP v2c)
- RFC 3411 (SNMP v3)
- VadaTech MCH Software Management Manual
- VadaTech SNMP Trap Handler User Manual

1.2 Acronyms Used in this Document

Acronym	Description
API	Application Programming Interface
ATCA	Advanced Telecommunications Computing Architecture
MC	Management Controller
MIB	Management Information Base
SNMP	Simple Network Management Protocol

Table 1: Acronyms

2 SNMP Interface Overview

All platforms support v1, v2c, and v3 of the Simple Network Management Protocol (SNMP), with the capability for SNMP queries and SNMP traps.

Figure 1 describes the logical organization and entities associated with the platform SNMP interfaces.

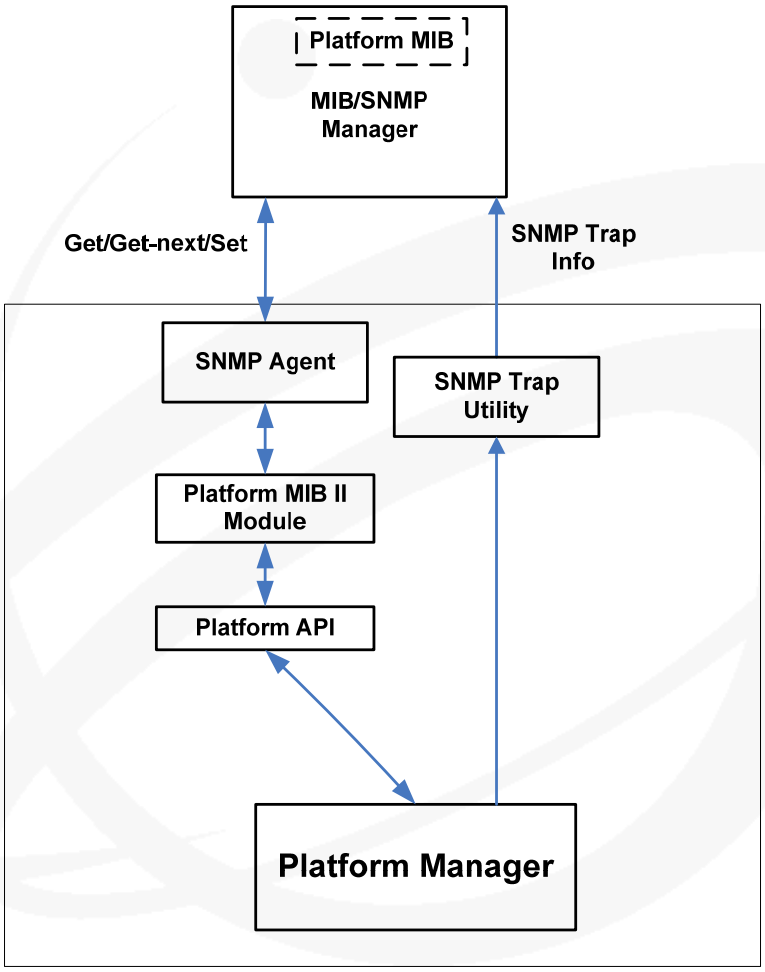

Figure 1: Platform Management Controller SNMP Agent / Sub-Agents

The SNMP agent listens to SNMP v1, v2c, and v3 queries (gets/sets) by default, evokes the corresponding MIB Module to process the request, and sends the SNMP response with return data to the SNMP/MIB manager. The SNMP agent is implemented to support SNMP-get, SNMP-get-next, and SNMP-set.

The elements of the VadaTech SNMP Interface are described in Table 2.

Name	Description
MIB Manager	see SNMP Manager
Platform API	platform programming interface used by the SNMP module to communicate with the platform manager
Platform MIB	MIB module describing the tables and scalars specific to the platform
Platform MIB II Module	compiled platform manager-specific MIB II modules
Platform Manager	platform with which the SNMP Interface communicates (refer to the relevant Platform Management Manual for more information)
SNMP Traps	asynchronous event notifications (archaic 'reports')
SNMP Agent	SNMP kernel on the active platform manager's MC that handles the SNMP requests from the remote SNMP client, the MIB/SNMP Manager
SNMP Manager	client MIB Manager that interfaces with the SNMP agent for user-specified requests
SNMP Trap Utility	SNMP Trap interface used to notify external devices of user-configurable event notifications (refer to the VadaTech SNMP Trap Handler User Manual for more information)

Table 2: Elements of the SNMP Interface

The rest of this document will focus on the core ATCA tables and scalars common between all platforms. Also discussed are the generic SNMP commands available to the user, and how to use them.

3 Management Information Base Files

Each platform comes with a VadaTech Management Information Base (MIB) files. A platform-specific MIB file (**VT-<platform>.mib**) describes the set of platform objects to be managed, specific to the specified platform.

A remote application, such as an SNMP/MIB manager, can compile files (using a MIB compiler) and utilize this information to manage devices on the platform. The platform-specific MIB file(s) are located in the **/opt/vadatech/SNMP/mibs** directory. Users can utilize **scp** to retrieve these files from the platform Management Controller.

All VadaTech platforms supporting SNMP support SNMPv1, SNMPv2c and SNMPv3, as described in RFC 1157, RFC 1901, and RFC 3411, respectively.

3.1 MIB Tree Root OID

The platform MIBs are represented via hierarchal data model, where each variable is identified via an object identifier (OID). All the object identifiers have a common ROOT OID as shown in the tree described in Figure 2.

23858 is the unique private VadaTech enterprise number obtained from IANA. The root OID in the remainder of this section is denoted as <ROOT_OID>, and is described below:

<ROOT_OID> = 1.3.6.1.4.1.23858.2.1.1

A platform's unique MIB will have a platform OID tree that can be specified as shown below:

<PLATFORM_ROOT_OID> = 1.3.6.1.4.1.23858.2.1.<platform OID>.

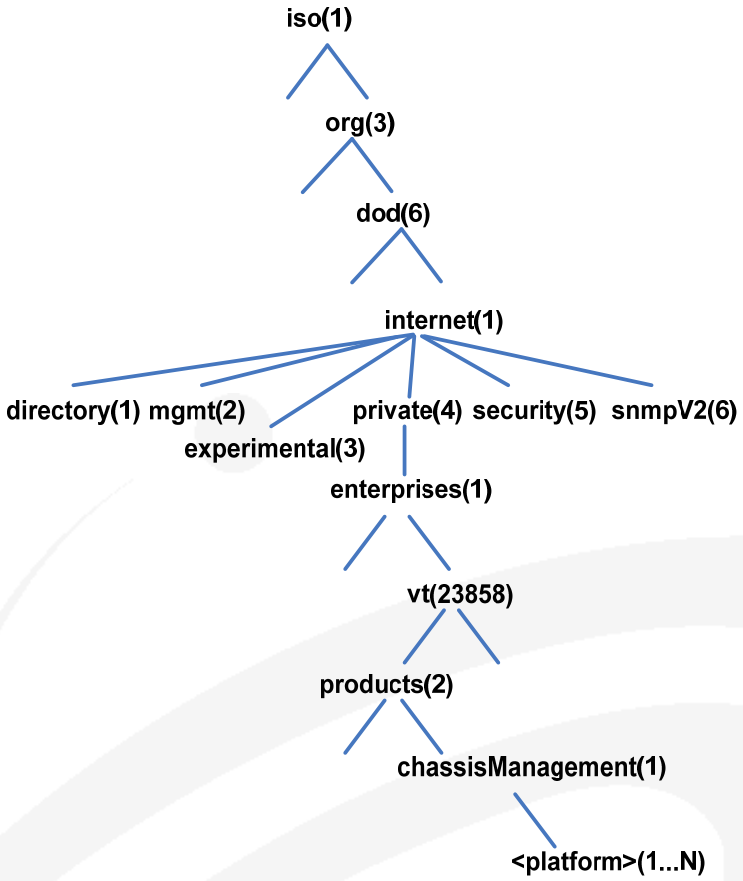

Figure 2: SNMP object identifier tree for VadaTech platform MIBs

3.2 Common MIB Objects

Table 3 is a summary of the supported groups of variables common among all platform MIBs:

Group	Object Name	Object Identifier	Description
Alarms	**alarmTable**	**<ROOT_OID>.1.1**	provide variables to view alarm status and to reset or clear alarms for all present Managers
FRU Device	**fruDeviceTable**	**<ROOT_OID>.2.1**	provide variables to obtain FRU information and to control a specific FRU payload using FRU control parameters
SDR	**sdrTable**	**<ROOT_OID>.3.1**	provide variables to obtain the Sensor Information (SDR) and sensor readings
SEL	**selTable**	**<ROOT_OID>.4.1**	provide variables to obtain the SEL entries and to delete entries or clear the SEL
LAN	**lanConfigTable**	**<ROOT_OID>.5.1**	provide variables to configure the LAN parameters
Event Filter	**eventFilterTable**	**<ROOT_OID>.6.1**	provide variables to configure the Event Filter Table
PEF	**pefConfig <parameterType>**	**<ROOT_OID>.7…15.0**	provide variables to configure the PEF parameters
Alert Policy	**alertPolicyTable**	**<ROOT_OID>.16.1**	provide variables to configure the Alert Policy Table
Alert String	**alertStringTable**	**<ROOT_OID>.17.1**	provide variables to configure the Alert String Table
Chassis Control	**chassisControl <controlType>**	**<ROOT_OID>.18…22.0**	provide variables for Chassis control
Manager Initialization Status	**initInProgress**	**<ROOT_OID>.23.0**	provide a variable to read manager initialization status
Management Controller	**mcTable**	**<ROOT_OID>.51.1**	provide variables to obtain information regarding a Management Controller and controls to reset and remove a Management Controller
Fan Tray	**fanTrayTable**	**<ROOT_OID>.52.1**	provide variables to obtain information regarding a Fan Tray and other controls to the fans
LED Properties	**ledPropTable**	**<ROOT_OID>.53.1**	provide variables to obtain information regarding LED properties
LED	**ledTable**	**<ROOT_OID>.54.1**	provide variables to obtain information regarding LEDs and configure LEDs

Table 3: Common Table and scalar objects described in the platform MIB files

3.2.1 Alarm Table

This table describes the alarm statuses for all software managers found on the system, and provides the ability to reset alarms for a specified amount of time.

MIB TABLE NAME: **alarmTable**
MIB TABLE OID: **1**
MIB TABLE ENTRY NAME: **alarmEntry**
MIB TABLE ENTRY OID: **1**

SYNTAX:

<ROOT_OID>.1.1.<var>.<address>

<var> variable name or index in the table described below
<address> 8-bit address representing a software manager's Management Controller

Var #	Name	Data Type	Access Mode	Description
1	aAddress	INTEGER	read-only	table index, as described by **<address>**; 8-bit address of the manager
2	aMinorStatus	INTEGER	read-only	1 – minor alarm is set 0 – minor alarm is not set
3	aMajorStatus	INTEGER	read-only	1 – major alarm is set 0 – major alarm is not set
4	aCriticalStatus	INTEGER	read-only	1 – critical alarm is set 0 – critical alarm is not set
5	aReset	INTEGER	read	current number of seconds to wait before resetting the alarm
			write	number of seconds to wait before resetting the alarm

Table 4: Variable descriptions for the Alarm Table

3.2.2 FRU Device

This table describes the FRU devices currently in the system.

MIB TABLE NAME: **fruDeviceTable**
MIB TABLE OID: **2**
MIB TABLE ENTRY NAME: **fruDeviceEntry**
MIB TABLE ENTRY OID: **1**

SYNTAX:

<ROOT_OID>.2.1.<var>.<address>.<fruId>

<var> variable name or index in the table described below
<address> 8-bit address representing a Management Controller
<fruId> FRU Identifier of the managed FRU

Var #	Name	Data Type	Access Mode	Description
1	fdAddress	INTEGER	read-only	table index component, as described by **<address>**; 8-bit IPMB address of the Management Controller representing the FRU
2	fdFruId	INTEGER	read-only	table index component, as described by **<fruId>**; FRU's ID 0 – Management Controller 1 .. N – Logical device
3	fdEntityId	INTEGER	read-only	FRU's entity ID
4	fdEntityInstance	INTEGER	read-only	FRU's entity instance
5	fdIdString	Display String	read-only	FRU's device ID string
6	fdSdrVersion	INTEGER	read-only	SDR version of the FRU or Management Controller device locator record for FRU
7	fdLogical	INTEGER	read-only	whether FRU is a logical device 0 – FRU is a physical device 1 – FRU is a logical device
8	fdChannel	INTEGER	read-only	FRU's channel number 0 – Management Controller 1 .. N – non-Management Controller devices
9	fdDeviceType	INTEGER	read-only	FRU's device type 10h (FRU Inventory Device) – FRU is a Management Controller
10	fdModifierType	INTEGER	read-only	FRU's modifier type FFh (Unspecified) – FRU is a Management Controller
11	fdRevision	INTEGER	read-only	FRU's device revision
12	fdFirmwareVersion	Display String	read-only	FRU's firmware revision
13	fdIpmiVersion	INTEGER	read-only	IPMI command specification version

14	fdSupportFlags	INTEGER	read-only	FRU's device support [7] Chassis Device [6] Bridge [5] IPMB Event Generator [4] IPMB Event Receiver [3] FRU Inventory Device [2] SEL Device [1] SDR Repository Device [0] Sensor Device
15	fdMfrId	INTEGER	read-only	FRU's manufacturer ID
16	fdProductId	INTEGER	read-only	FRU's product ID
17	fdHotSwapState	INTEGER	read-only	FRU's current hotswap state (M0-M7)
18	fdHealthy	INTEGER	read-only	FRU's presence/health status 0 – FRU is absent or unhealthy 1 – FRU is present and healthy
19	fdActivation	INTEGER	read	always 0
			write	0 – deactivates FRU 1 – activates FRU
20	fdPayloadControl	INTEGER	read	always 0
			write	0 – cold reset 1 – warm reset 2 – graceful reboot 3 – issue diagnostic interrupt
21	fdPowerLevel	INTEGER	read	current power level. Note, a -1 indicates the specified device's power level is not applicable
			write	0 – shutdown power 1..20 – change power level
22	fdProductAreaPresent	INTEGER	read-only	whether Product Info Area is present in FRU's inventory 0 – Product Info Area is absent 1 – Product Info Area is present
23	fdProductMfr	Display String	read-only	product manufacturer
24	fdProductName	Display String	read-only	product name
25	fdProductPartNumber	Display String	read-only	product part number
26	fdProductVersionNumber	Display String	read-only	product version
27	fdProductSerialNumber	Display String	read-only	product serial number
28	fdBoardAreaPresent	INTEGER	read-only	whether Board Info Area is present in FRU's inventory 0 – Board Info Area is absent 1 – Board Info Area is present
29	fdBoardMfr	Display String	read-only	board manufacturer
30	fdBoardProductName	Display String	read-only	board name
31	fdBoardSerialNumber	Display String	read-only	board serial number
32	fdBoardPartNumber	Display String	read-only	board part number

33	fdBoardMfgTime	Display String	read-only	board manufacturing date/time

Table 5: Variable descriptions for the FRU Device Table

3.2.3 SDR

This table describes the sensor data repository.

MIB TABLE NAME: **sdrTable**
MIB TABLE OID: **3**
MIB TABLE ENTRY NAME: **sdrEntry**
MIB TABLE ENTRY OID: **1**

SYNTAX:

<ROOT_OID>.3.1.<var>.<address>.<index>

<var> variable name or index in the table described below
<address> 8-bit address representing a Software Manager's Management Controller
<index> index of the sensor on the Management Controller, in sequential order

Var #	Name	Data Type	Access Mode	Description
1	sdrAddress	INTEGER	read-only	table index component, as described by **<address>**; 8-bit IPMB address of the Management Controller on which the SDR belongs
2	sdrIndex	INTEGER	read-only	table index component, as described by **<index>**; index of the sensor on the Management Controller, in sequential order
3	sdrVersion	INTEGER	read-only	SDR version
4	sdrType	INTEGER	read-only	SDR type 1 – full sensor record 2 – compact sensor record
5	sdrOwnerId	INTEGER	read-only	SDR owner ID
6	sdrOwnerLun	INTEGER	read-only	sensor owner LUN
7	sdrSensorNumber	INTEGER	read-only	sensor number
8	sdrSensorEntityId	INTEGER	read-only	entity ID of device monitored by sensor
9	sdrSensorEntityInstance	INTEGER	read-only	entity instance of device monitored by sensor
10	sdrSensorInitialization	INTEGER	read-only	sensor initialization
11	sdrSensorCapabilities	INTEGER	read-only	sensor capabilities
12	sdrSensorType	INTEGER	read-only	sensor type
13	sdrSensorEventReadingTypeCode	INTEGER	read-only	event/reading type
14	sdrSensorAssertionEventMask	INTEGER	read-only	assertion event mask
15	sdrSensorLowerThresholdReadingMask	INTEGER	read-only	lower threshold reading mask
16	sdrSensorDeassertionEventMask	INTEGER	read-only	de-assertion event mask
17	sdrSensorUpperThresholdReadingMask	INTEGER	read-only	upper threshold reading mask

18	sdrSensorMask	INTEGER	read-only	reading mask: non-threshold sensors settable/readable threshold mask: threshold sensors
19	sdrSensorUnits1	INTEGER	read-only	sensor units 1
20	sdrSensorUnits2	INTEGER	read-only	base unit
21	sdrSensorUnits3	INTEGER	read-only	modifier unit
22	sdrSensorLinearization	INTEGER	read-only	linearization – full sensor record 0 – compact sensor record
23	sdrSensorM	INTEGER	read-only	M value – full sensor record 0 – compact sensor record
24	sdrSensorTolerance	INTEGER	read-only	tolerance – full sensor record 0 – compact sensor record
25	sdrSensorB	INTEGER	read-only	B value – full sensor record 0 – compact sensor record
26	sdrSensorAccuracy	INTEGER	read-only	accuracy – full sensor record 0 – compact sensor record
27	sdrSensorAccuracyExp	INTEGER	read-only	accuracy exponent – full sensor record 0 – compact sensor record
28	sdrSensorDirection	INTEGER	read-only	direction
29	sdrSensorRExp	INTEGER	read-only	result exponent 0 – compact sensor record
30	sdrSensorBExp	INTEGER	read-only	B exponent 0 – compact sensor record
31	sdrSensorAnalogCharacteristicFlags	INTEGER	read-only	analog characteristic flag 0 – compact sensor record
32	sdrSensorNominalReading	INTEGER	read-only	nominal reading 0 – compact sensor record
33	sdrSensorNormalMaximum	INTEGER	read-only	normal maximum 0 – compact sensor record
34	sdrSensorNormalMinimum	INTEGER	read-only	normal minimum 0 – compact sensor record
35	sdrSensorMaximumReading	INTEGER	read-only	sensor maximum reading 0 – compact sensor record
36	sdrSensorMinimumReading	INTEGER	read-only	sensor minimum reading 0 – compact sensor record
37	sdrSensorUpperNonRecoverableThreshold	INTEGER	read-only	upper non-recoverable threshold 0 – compact sensor record
38	sdrSensorUpperCriticalThreshold	INTEGER	read-only	upper critical threshold 0 – compact sensor record
39	sdrSensorUpperNonCriticalThreshold	INTEGER	read-only	upper non-critical threshold 0 – compact sensor record
40	sdrSensorLowerNonRecoverableThreshold	INTEGER	read-only	lower non-recoverable threshold 0 – compact sensor record
41	sdrSensorLowerCriticalThreshold	INTEGER	read-only	lower critical threshold 0 – compact sensor record
42	sdrSensorLowerNonCriticalThreshold	INTEGER	read-only	lower non-critical threshold 0 – compact sensor record
43	sdrSensorPositiveGoingThresholdHysteresis	INTEGER	read-only	positive-going threshold hysteresis
44	sdrSensorNegativeGoingThresholdHysteresis	INTEGER	read-only	negative-going threshold hysteresis
45	sdrSensorDeviceIdString	Display String	read-only	sensor ID string
46	sdrSensorReading	INTEGER	read-only	sensor reading

47	sdrSensorReadingProcessed	Display String	read-only	processed sensor reading after applying conversion formula N/A – compact sensor record

Table 6: Variable descriptions for the SDR Table

3.2.4 SEL

This table describes the events in the System Event Log.

MIB TABLE NAME: **selTable**
MIB TABLE OID: **4**
MIB TABLE ENTRY NAME: **selEntry**
MIB TABLE ENTRY OID: **1**

SYNTAX:

<ROOT_OID>.4.1.<var>.<index>

<var> variable name or index in the table described below
<index> SEL table entry index

Var #	Name	Data Type	Access Mode	Description
1	selIndex	INTEGER	read-only	table index, as described by **<index>**
2	selSensorType	INTEGER	read-only	sensor type code of sensor that generated event.
3	selSensorNumber	INTEGER	read-only	sensor number of sensor that generated event
4	selEventDirection	INTEGER	read-only	event direction 0 – assertion 1 – de-assertion
5	selTimestamp	Display String	read-only	time event was logged
6	selEventGeratorSiteType	INTEGER	read-only	event generator's site type
7	selEventGeneratorSiteNumber	INTEGER	read-only	event generator's site number
8	selEventGeneratorAddress	INTEGER	read-only	event generator's IPMB address
9	selEventGeneratorFruId	INTEGER	read-only	event generator's FRU identifier
10	selSensorSpecificOffset	INTEGER	read-only	event type of sensor that generated event
11	selClear	INTEGER	read	always 0
			write	1 – clear all entries in SEL

Table 7: Variable descriptions for the SEL Table

3.2.5 LAN

This table describes the LAN configuration for all channels in the system.

MIB TABLE NAME: **lanConfigTable**
MIB TABLE OID: **5**
MIB TABLE ENTRY NAME: **lanConfigChannelEntry**
MIB TABLE ENTRY OID: **1**

SYNTAX:

<ROOT_OID>.5.1.<var>.<channelIndex>

<var> variable name or index in the table described below
<channelIndex> channel number

Var #	Name	Data Type	Access Mode	Description
1	lcChannelIndex	INTEGER	read-only	table index, as described by **<channelIndex>**; channel number
2	lcSetInProgress	INTEGER	read-only	LAN channel's set in-progress parameter
3	lcAuthenticationTypeSupport	INTEGER	read-only	LAN channel's authentication type support
4	lcAuthenticationTypeEnable	OCTET STRING (size 5)	read-only	LAN channel's authentication type enable
5	lcIpAddress	IP Address	read-only	LAN channel's IP address
6	lcIpAddressSource	INTEGER	read-only	LAN channel's IP address
7	lcMacAddress	OCTET STRING (size 6)	read-only	LAN channel's MAC address
8	lcSubnetMask	IP Address	read-only	LAN channel's subnet mask
9	lcIpv4HeaderParameters	OCTET STRING (size 3)	read-only	LAN channel's IPV4 header
10	lcPrimaryRmcpPortNumber	INTEGER	read-only	LAN channel's primary RMCP port number
11	lcSecondaryRmcpPortNumber	INTEGER	read-only	LAN channel's secondary RMCP port number
12	lcBmcGeneratedArpControl	INTEGER	read-only	LAN channel's BMC-generated ARP control
13	lcGratuitousArpInterval	INTEGER	read-only	LAN channel's gratuitous ARP interval
14	lcDefaultGatewayAddress	IP Address	read-only	LAN channel's default gateway address
15	lcDefaultGatewayMacAddress	OCTET STRING (size 6)	read-only	LAN channel's default gateway MAC address
16	lcBackupGatewayAddress	IP Address	read-only	LAN channel's backup gateway MAC address

17	lcBackupGatewayMacAddress	OCTET STRING (size 6)	read-only	LAN channel's backup default gateway MAC address
18	lcCommunityString	OCTET STRING (size 0..18)	read-only	LAN channel's community string
19	lcNumberOfDestinations	INTEGER	read-only	Number of destination available on LAN channel
20	lcDestinationType0	OCTET STRING (size 3)	read-only	LAN channel's destination type with selector set to 0
...	...	...	...	...
35	lcDestinationType15	OCTET STRING (size 3)	read-only	LAN channel's Destination Type with selector set to 15
36	lcDestinationAddress0	OCTET STRING (size 12)	read-only	LAN channel's destination address with destination selector 0
...	...	...	...	...
51	lcDestinationAddress15	OCTET STRING (size 12)	read-only	LAN channel's destination address with destination selector 15
52	lcVlanId	OCTET STRING (size 2)	read-only	LAN channel's 802.1q VLAND ID
53	lcVlanPriority	INTEGER	read-only	LAN channel's 802.q VLAN priority
54	lcRmcpCipherSuiteEntrySupport	INTEGER	read-only	Number of cipher suites available to be enabled on LAN channel
55	lcRmcpCipherSuiteEntries	OCTET STRING (size 17)	read-only	LAN channel's cipher suite IDs supported by LAN channel
55	lcRmcpCipherSuitePrivilegeLevels	OCTET STRING (size 9)	read-only	Privilege levels associated with each cipher suite for LAN channel
57	lcDestinationAddressVlanTag0	OCTET STRING (size 0 | 3)	read-only	LAN channel's destination address VLAN tags with destination selector 0
...	...	...	...	...
72	lcDestinationAddressVlanTag15	OCTET STRING (size 0 | 3)	read-only	LAN channel's destination address VLAN tags with destination selector 15

Table 8: Variable descriptions for the LAN Configuration Table

3.2.6 Event Filter

This table describes the currently enabled event filters.

MIB TABLE NAME: **eventFilterTable**
MIB TABLE OID: **6**
MIB TABLE ENTRY NAME: **eventFilterEntry**
MIB TABLE ENTRY OID: **1**

SYNTAX:

<ROOT_OID>.6.1.<var>.<filterNumber>

<var> variable name or index in the table described below
<filterNumber> index of the entry in the event filter table

Var #	Name	Data Type	Access Mode	Description
1	efFilterNumber	INTEGER	read-only	table index, as described by **<filterNumber>**; index of the entry in the event filter table
2	efSoftwareConfigurable	INTEGER	read-only	whether event filter is available for configuration by system management software 0 – filter not available for configuration 1 – filter available for configuration
3	efEnable	INTEGER	read-only	whether event filter is enabled 0 – alert policy is disabled 1 – alert policy is enabled
4	efAlert	INTEGER	read-only	whether alerting is enabled 0 – alerting is disabled 1 – alerting is enabled
5	efPowerOff	INTEGER	read-only	whether powering off is enabled 0 – powering off is disabled 1 – Powering off is enabled
6	efReset	INTEGER	read-only	whether resetting is enabled 0 – resetting is disabled 1 – resetting is enabled
7	efPowerCycle	INTEGER	read-only	whether power cycling is enabled 0 – power cycling is disabled 1 – power cycling is enabled
8	efOemAction	INTEGER	read-only	whether OEM action is enabled 0 – OEM action is disabled 1 – OEM action is enabled
9	efDiagnosticInterruptAction	INTEGER	read-only	whether diagnostic interrupting is enabled 0 – diagnostic interrupting is disabled 1 – diagnostic interrupting is enabled
10	efGroupControlAction	INTEGER	read-only	whether group control is enabled 0 – group control is disabled 1 – group control is enabled
11	efPolicyNumber	INTEGER	read-only	policy number to use

12	efGroupControlSelector	INTEGER	read-only	group control selector
13	efSeverity	INTEGER	read-only	event severity
14	efGeneratorID1	INTEGER	read-only	slave address or software ID to match FFh – match any
15	efGeneratorID2	INTEGER	read-only	channel/LUN to match FFh – match any
16	efSensorType	INTEGER	read-only	type of sensor FFh – match any
17	efSensorNumber	INTEGER	read-only	sensor number FFh – match any
18	efTrigger	INTEGER	read-only	direction and event/reading type FFh – match any
19	efOffsetMask1	INTEGER	read-only	event data 1 event offset mask, bit position 7 to 0
20	efOffsetMask2	INTEGER	read-only	event data 1 event offset mask, bit position 15 to 8
21	efDataMask1ANDMask	INTEGER	read-only	event data 1 AND mask
22	efDataMask1Compare1Mask	INTEGER	read-only	event data 1 compare 1
23	efDataMask1Compare2Mask	INTEGER	read-only	event data 1 compare 2
24	efDataMask2ANDMask	INTEGER	read-only	event data 2 AND mask
25	efDataMask2Compare1Mask	INTEGER	read-only	event data 2 compare 1
26	efDataMask2Compare2Mask	INTEGER	read-only	event data 2 compare 2
27	efDataMask3ANDMask	INTEGER	read-only	event data 3 AND mask
28	efData3Compare1Mask	INTEGER	read-only	event data 3 compare 1
29	efData3Compare2Mask	INTEGER	read-only	event data 3 compare 2
30	efData	OCTET STRING (size 20)	read-write	event filter data

Table 9: Variable descriptions for the Event Filter Table

3.2.7 PEF

This table describes the scalars relevant to the Platform Event Filter configuration information. The index is always 0.

- **Note**, bracket notation describes bit masks. For additional information on configurable PEF parameters, refer to the PEF Configuration Parameters Table in the Intelligent Platform Management (IPMI) Interface Specification.

SYNTAX: **<ROOT_OID>.<var>.0**

<var> variable name or index in the table described below

Var #	Name	Data Type	Access Mode	Description
7	pefConfigSetInProgress	INTEGER	read-only	set in progress parameter
8	pefConfigControl	INTEGER	read-only	[3] whether PEF alert startup delay is enabled [2] whether PEF startup delay enable [1] event messages are enabled for PEF actions [0] whether PEF is enabled
9	pefConfigActionGlobalControl	INTEGER	read-only	global control actions [5] diagnostic interrupt [4] OEM action [3] power cycle [2] reset [1] power down [0] alert
10	pefConfigStartupDelay	INTEGER	read-only	time to delay PEF after a system powers up or resets, in seconds 0 – no delay
11	pefConfigAlertStartupDelay	INTEGER	read-only	time to delay alerts after system powers up or resets, in seconds 0 – no delay
12	pefConfigNumberOfEventFilters	INTEGER	read-only	number of event filters
13	pefConfigNumberOfAlertPolicyEntries	INTEGER	read-only	number of alert policies
14	pefConfigNumberOfAlertStringEntries	INTEGER	read-only	number of alert strings
15	pefConfigSystemGuid	OCTET STRING (Size 17)	read-write	system GUID

Table 10: PEF Configuration Scalar Objects

3.2.8 Alert Policy

This table displays the system's alert policies.

MIB TABLE NAME: **alertPolicyTable**
MIB TABLE OID: **16**
MIB TABLE ENTRY NAME: **alertPolicyEntry**
MIB TABLE ENTRY OID: **1**

SYNTAX:

<ROOT_OID>.16.1.<var>.<entryNumber>

<var> variable name or index in the table described below
<entryNumber> index of the entry in the alert policy table

Var #	Name	Data Type	Access Mode	Description
1	apEntryNumber	INTEGER	read-only	table index, as described by **<entryNumber>**; index of the entry in the alert policy table
2	apAlertPolicy	INTEGER	read-only	alert policy
3	apEnabled	INTEGER	read-only	whether alert policy is enabled 0 – alert policy is disabled 1 – alert policy is enabled
4	apPolicyNumber	INTEGER	read-only	alert policy's number
5	apDestination	INTEGER	read-only	selector of destination to use to send alert
6	apChannel	INTEGER	read-only	channel the alert is to be sent over
7	apAlertString	INTEGER	read-only	selector of alert string to use in alert
8	apEventSpecific	INTEGER	read-only	whether alert string look-up is event specific 0 – not event specific 1 – event specific
9	apData	OCTET STRING (size 3)	read-write	alert policy data

Table 11: Variable descriptions for the Alert Policy Table

3.2.9 Alert String

This table displays the alert strings configured for the platform.

MIB TABLE NAME: **alertStringTable**
MIB TABLE OID: **17**
MIB TABLE ENTRY NAME: **alertStringEntry**
MIB TABLE ENTRY OID: **1**

SYNTAX:

<ROOT_OID>.17.1.<var>.<index>

<var> variable name or index in the table described below
<alertStringIndex> index of the entry in the alert policy table

Var #	Name	Data Type	Access Mode	Description
1	asIndex	INTEGER	read-only	table index, as described by **<alertStringIndex>**; index of the entry in the alert string table
2	asFilterNumber	INTEGER	read-only	corresponding event filter for event specific alert strings
3	asSet	INTEGER	read-only	alert string's set selector
4	asKey	OCTET STRING (size 2)	read-write	alert string key data
5	asData	Display String	read-write	alert string

Table 12: Variable descriptions for the Alert String Table

3.2.10 Chassis Control

This table describes the scalars relevant to controlling the Chassis. The index is always 0.

SYNTAX: **<ROOT_OID>.<var>.0**

<var> variable name or index in the table described below

Var #	Name	Data Type	Access Mode	Description
18	chassisDown	INTEGER	read	always 0
			write	8-bit address of the Management Controller
19	chassisUp	INTEGER	read	always 0
			write	8-bit address of the Management Controller
20	chassisReset	INTEGER	read	always 0
			write	8-bit address of the Management Controller
21	chassisInterrupt	INTEGER	read	always 0
			write	8-bit address of the Management Controller
22	chassisSoftShutdown	INTEGER	read	always 0
			write	8-bit address of the Management Controller

Table 13: Chassis Control Scalar Objects

3.2.11 Manager Initialization Status

This table describes the scalar to determine the status of the platform manager. The index is always 0.

SYNTAX: **<ROOT_OID>.<var>.0**

<var> variable name or index in the table described below

Var #	Name	Data Type	Access Mode	Description
23	initInProgress	INTEGER	read-only	1 – manager initialization is in progress 0 – manager is initialized

Table 14: Initialization In Progress Scalar Object

3.2.12 Management Controller

This table describes the Management Controllers currently in the system.

MIB TABLE NAME: **mcTable**
MIB TABLE OID: **51**
MIB TABLE ENTRY NAME: **mcEntry**
MIB TABLE ENTRY OID: **1**

SYNTAX:

<ROOT_OID>.51.1.<var>.<address>

<var> variable name or index in the table described below
<address> 8-bit address representing a Management Controller

Var #	Name	Data Type	Access Mode	Description
1	mcAddress	INTEGER	read-only	table index, as described by **<address>**; 8-bit IPMB address of the Management Controller
2	mcMaxFruId	INTEGER	read-only	numerically largest FRU ID managed by the Management Controller, excluding reserved FRU IDs
3	mcHotSwapState	INTEGER	read-only	Management Controller's current hotswap state (M0-M7)
4	mcHealthy	INTEGER	read-only	Management Controller's presence/health status 0 – MC is absent or unhealthy 1 – MC is present and healthy
5	mcColdReset	INTEGER	read	always 0
			write	1 – trigger cold reset of the Management Controller
6	mcWarmReset	INTEGER	read	always 0
			write	1 – trigger warm reset of the Management Controller
7	mcExtract	INTEGER	read	always 0
			write	1 – set the Management Controller as extracted

Table 15: Variable descriptions for the Management Controller Table

3.2.13 Fan Tray

This table describes the Fan Trays currently in the system.

MIB TABLE NAME: **fanTrayTable**
MIB TABLE OID: **52**
MIB TABLE ENTRY NAME: **fanTrayEntry**
MIB TABLE ENTRY OID: **1**

SYNTAX:

<ROOT_OID>.52.1.<var>.<address>.<fruId>

<var> variable name or index in the table described below
<address> 8-bit address representing a software manager's Management Controller
<fruId> FRU ID of the Management Controller representing the Fan Tray

Var #	Name	Data Type	Access Mode	Description
1	ftAddress	INTEGER	read-only	table index component, as described by **<address>**; 8-bit IPMB address of the Management Controller representing the Fan Tray
2	ftFruId	INTEGER	read-only	table index component, as described by **<fruId>**; FRU ID of the Management Controller representing the Fan Tray
3	ftHotSwapState	INTEGER	read-only	Fan Tray's Management Controller current hot-swap state (M0-M7)
4	ftHealthy	INTEGER	read-only	Fan Tray's Management Controller presence/health status 0 – Fan Tray is absent or unhealthy 1 – Fan Tray is present and healthy
5	ftOverrideState	INTEGER	read-only	whether Fan Tray is at local or override state 0 – Fan Tray at local control 1 – Fan Tray at override control
6	ftMinimumSpeedLevel	INTEGER	read-only	minimum speed level supported by Fan Tray 0 – Fan Tray not present
7	ftMaximumSpeedLevel	INTEGER	read-only	maximum speed level supported by Fan Tray 0 – Fan Tray not present
8	ftNormalOperatingLevel	INTEGER	read-only	normal operating speed level for Fan Tray. 0 – Fan tray not present
9	ftCurrentLevel	INTEGER	read	current operating level FEh – Emergency shutdown
			write	0 .. maximum – change speed level 0xFE – shut down

Table 16: Variable descriptions for the Fan Tray Table

3.2.14 LED Properties

This table describes the LEDs in the system.

MIB TABLE NAME: **ledPropTable**
MIB TABLE OID: **53**
MIB TABLE ENTRY NAME: **ledPropTable**
MIB TABLE ENTRY OID: **1**

SYNTAX:

<ROOT_OID>.53.1.<var>.<address>.<fruId>

<var> variable name or index in the table described below
<address> IPMB address of the Management Controller managing the FRU
<fruId> FRU identifier

Var #	Name	Data Type	Access Mode	Description
1	ledpAddress	INTEGER	read-only	table index component, as described by **<address>**; 8-bit IPMB address of the Management Controller representing the FRU
2	ledpFruId	INTEGER	read-only	table index component, as described by **<fruId>**; FRU identifier
3	ledpGeneralStatus	INTEGER	read-only	bitmask indicating FRU's ability to control the four general status LEDs [3] LED3 can be controlled [2] LED2 can be controlled [1] LED1 can be controlled [0] LED0 can be controlled
4	ledpApplicationLedCount	INTEGER	read-only	number of application-specific LEDs

Table 17: Variable descriptions for the LED Properties Table

3.2.15 LED

This table describes the state of the LEDs in the system.

MIB TABLE NAME: **ledTable**
MIB TABLE OID: **54**
MIB TABLE ENTRY NAME: **ledEntry**
MIB TABLE ENTRY OID: **1**

SYNTAX:

<ROOT_OID>.54.1.<var>.<address>.<fruId>.<ledId>

<var> variable name or index in the table described below
<address> 8-bit IPMB address of the Management Controller representing the FRU
<fruId> FRU identifier
<ledId> LED identifier

Var #	Name	Data Type	Access Mode	Description
1	ledAddress	INTEGER	read-only	table index component, as described by **<address>**; 8-bit IPMB address of the Management Controller representing the Fan Tray
2	ledFruId	INTEGER	read-only	table index component, as described by **<fruId>**; FRU identifier
3	ledId	INTEGER	read-only	table index component, as described by **<ledId>**; LED identifier
4	ledPresentFunction	INTEGER	read-only	0 – LED is off 1 – LED is on 2 – LED is blinking
5	ledState	INTEGER	read-only	0 – lamp test is enabled 1 – override state is enabled 2 – IPM Controller has a local control
6	ledSetFunction	INTEGER	read	0 – LED is off 1 – LED is on <onDuration<<8|offDuration> LED is blinking, in 10s of milliseconds
			write	0 – turn LED off 1 – turn LED on 2 - restore to local state <onDuration<<8|offDuration> – blink LED, in 10s of milliseconds
7	ledLampTest	INTEGER	read	duration in 100s of milliseconds 0 – Lamp test is currently not on
			write	duration to run lamp test, in 100s of milliseconds

Table 18: Variable descriptions for the LED Table

4 SNMP Client Implementation

Any SNMP client implementation should be able to access the platform manager-defined variables. The VadaTech SNMP agent is based on **net-snmp 5.3.1**. The **net-snmp** package provides some basic management tools to access the platform SNMP agent, such as the **snmpget**, **snmpset**, and **snmpwalk** commands, which are used by this document to instruct how to access VadaTech objects via the SNMP interface. More information on these, and other commands provided by the **net-snmp** package, can be found at the following locations:

- http://www.net-snmp.org/docs/man/
- http://net-snmp.sourceforge.net/docs/readmefiles.html

To install a MIB file on a client SNMP implementation, follow the instructions supplied with the package. The MIB file on the client-side of the SNMP will be used to decode objects and values returned to and from the SNMP Interface. To decode a VadaTech SNMP module object, the SNMP client implementation must be made aware of its corresponding MIB file, otherwise the output will be the numbered address and output of the variable.

If you are using **net-snmp**, copy the **VT-<PLATFORM>.mib** file (see the corresponding platform extension to this document), to **net-snmp**'s **mibs** directory, as directed by the SNMP client implementation's documentation. Then, for example, to execute **snmpwalk**, use the **-m** option to specify the appropriate MIB module, defined in the corresponding platform extension to this, document. The MIB module name can also be found in the corresponding MIB file.

For example, to read the Sensor Event Log Table, common to all platforms:

```
snmpwalk –t 5 -v 1 –m +VT-<PLATFORM>-MIB <platform IP address> -c
      VadaTech selTable
```

By default, the community name for the SNMP agent on the platform manager is '**VadaTech**', as shown in the above example, for both read and write requests. This can be changed by editing the **/opt/vadatech/SNMP/etc/snmpd.conf** file on the platform Management Controller. The SNMP MIB browser needs to match the community name for both read and write requests. For example, the following is a way to read the number of alert string entries that are in the alert policy table in the PEF configuration:

```
snmpget –t 5 -v 1 –m +VT-<PLATFORM>-MIB <platform IP address> -c
      VadaTech .iso.3.6.1.4.1.23858.2.1.<platform OID>.16.1.1.1
```

To set a variable (for read-write variables only), **snmpset** can be used. The following will set the time to delay a PEF, after a system powers up or resets, to 15 seconds (using 'i' to specify that the field is of INTEGER type):

```
snmpset -v 1 –m +VT-<PLATFORM>-MIB <platform IP address> –c VadaTech
      .iso.3.6.1.4.1.23858.2.1. <platform OID>.10.0 i 15
```

From Figure 2, a user can also replace each of the numbers with the corresponding alpha-numeric name from the OID tree. This document uses the numeric values for brevity.

For more information on the SNMP management tools available to a specific SNMP client implementation, refer to the client's documentation.

Index

L

M

P

R

S