VadaTech ATCA

SNMP Trap Handler User Manual

May 20, 2009

Version 2.0

vadatech inc
THE POWER OF VISION

Copyright

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 The text itself may not be published commercially in print or electronic form, edited, translated, or otherwise altered without the permission of VadaTech, Inc.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 VadaTech SNMP Trap Handler, available to all platforms supporting SNMP.

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)

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
PEF	Platform Event filter
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.

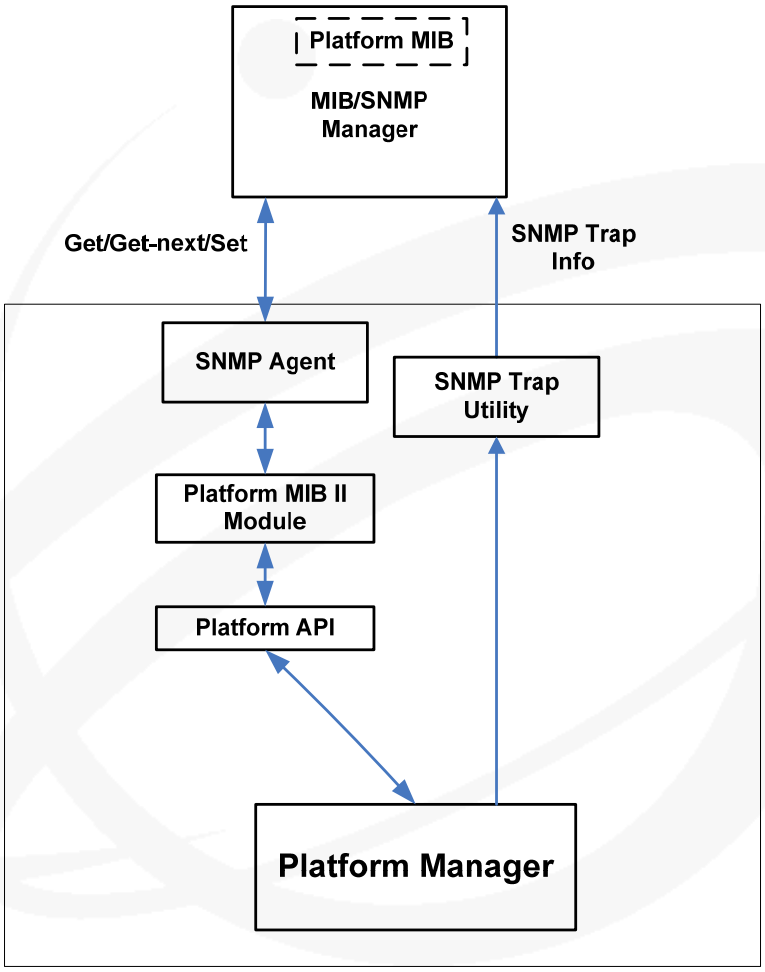

Figure 1: Platform Management Controller SNMP Agent / Sub-Agents

The platform manager can be configured, as described in Section 5: SNMP Trap Configuration, to send event notifications through the SNMP Trap Handler Utility to an SNMP client.

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

Name	Description
ATCA Core MIB	MIB module describing the tables and scalars common to all ATCA-based SNMP interfaces (see the VadaTech ATCA Core SNMP Interface Reference Manual for more information)
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 (see the VadaTech ATCA Core SNMP Interface Reference Manual for more information)
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 (see the VadaTech ATCA Core SNMP Interface Reference Manual for more information)
SNMP Manager	client MIB Manager that interfaces with the SNMP agent for user-specified requests, and / or listens for event notifications from the SNMP Trap utility
SNMP Trap Utility	SNMP Trap interface used to notify external devices of user-configurable event notifications

Table 2: Elements of the SNMP Interface

3 Default PEF Configuration

A VadaTech platform comes preconfigured with a default PEF configuration with channel alerting disabled for all channels (see Section 5.1: Enable Channel Alerting), SNMP traps disabled for all channels (see Section 5.2: Enabling SNMP Trap Handling), default destinations (destination 0) for each channel set to 192.168.0.250 (see Section 5.3: Configuring a Trap Destination Address), and event filters enabled for a number of common events (see Section 5.4: Enabling and Configuring Platform Event Filters).

4 SNMP Trap Clients

SNMP traps are received and logged by a service, like the **net-snmp SNMP trap daemon**.

The VadaTech SNMP Agent is based on **net-snmp 5.3.1**. The **net-snmp** package provides some basic management tools that include an SNMP trap daemon. This application, or any other used for SNMP trap handling, is beyond the scope of this document. Refer to the application's documentation for information on the application's installation and configuration.

5 SNMP Trap Configuration

Each platform comes with a Command Line Interface (CLI) that provides the user with the ability to configure SNMP trap handling. The configuration process described in this section is a summary of the information found in the corresponding platform's reference manual. For more information about any of the commands used in this document, refer to the corresponding CLI reference manual.

To setup the platform for handling SNMP traps, a user will need to enable channel alerting, enable traps, add a destination IP address to send the traps, and configure the Platform Event Filtering (PEF) to provide the traps. These steps are outlined in the following sections and can be accomplished in any order.

5.1 Enable Channel Alerting

To allow the channel to forward event notifications, the channel must have alerting enabled. To enable alerting, use the **set_channel_access** command. The following example enables channel alerting for channel 2:

```
set_channel_access –c 2 –A 1
```

To verify that the channel access is set correctly, use the **get_channel_access** command.

5.2 Enabling SNMP Trap Handling

To enable trap handling for a specific channel, use the **snmp_trap_enable** command. The following will enable trap handling for channel 2:

```
# snmp_trap_enable –c 2
```

To verify that the traps have been enabled for a specified channel, use the **get_snmp_trap_info** command.

5.3 Configuring a Trap Destination Address

A trap destination address is set using the **snmp_trap_set_address** command. The following example will set a destination address of 192.168.0.114 as the recipient for event notifications sent on channel 2:

```
# snmp_trap_set_address –c 2 –d 2 –i 192.168.0.114
```

To verify that the destination has been set correctly, use the **get_snmp_trap_info** command.

To remove a destination address, use the **snmp_trap_remove_address** command.

5.4 Enabling and Configuring Platform Event Filters

Platform Event Filtering allows a user to specify the following:

- what kind of event to 'trap'
- where to send the event notification of the trapped event
- how to filter the information sent in the notification
- when to send the event

PEF configuration is outside the scope of this document. Detailed information on how to use the PEF can be found in the Intelligent Platform Management (IPMI) Interface Specification and VadaTech PEF and LAN Configuration User Manual.

To enable a configured Event Filter, use the **set_pef_config_parameters** command to configure parameter 9 to enable the alert policy, and set the channel to the known SNMP trap channel, as discussed in Section 5.2: Enabling SNMP Trap Handling. The following enables the alert policy for channel 2, specifying alert policy number 1, policy number 1, without configuring an alert string:

```
# set_pef_config_parameters –p 9 –d "01 18 21 00"
```

6 Testing SNMP Traps

To test an SNMP trap configuration, use the **set_snmp_trap_test** command. Once all items are configured, as discussed in Section 5: SNMP Trap Configuration, send a test trap to the desired channel and destination numbers to verify a proper configuration. The following example will send a test event notification to destination 1, channel 2:

```
# set_snmp_trap_test –c 2 –d 1
```

When executed, if traps were configured correctly on both the client and the server, the client should receive an event notification from the server.

Index