Sun Netra CT900 Server

Software Developer's Guide

Part No.: 819-1178-16
February 2011

Contents

READ AND DELETE

1. To display the "Beta Draft" footer , show the BetaDraft conditional setting.

2. After Beta, hide the BetaDraft conditional setting.

Figures

READ AND DELETE

1. To display the "Beta Draft" footer , show the BetaDraft conditional setting.

2. After Beta, hide the BetaDraft conditional setting.

Tables

Preface

This document contains descriptions of the configuration and use of features of Oracle's Sun Netra CT900 server and the environment for writing applications.

This guide also provides information you need to access the platform compiler for:

- Writing applications that use the OpenHPI API (as described in Chapter 1)
- Writing applications that use the IPMI driver (as described in Chapter 3)

Using UNIX Commands

This document might not contain information about basic UNIX® commands and procedures such as shutting down the system, booting the system, and configuring devices. Refer to the following for this information:

- Software documentation that you received with your system
- Oracle Solaris Operating System documentation, which is at:

  `http://www.oracle.com/technetwork/indexes/documentation/index.html`

Shell Prompts

Shell	Prompt
C shell	*machine-name*%
C shell superuser	*machine-name*#
Bourne shell and Korn shell	$
Minix shell superuser	#

Related Documentation

The Sun Netra CT900 server documentation is listed in the following table. Except for the *Important Safety Information for Sun Hardware Systems*, all the documents listed are available online at:

http://www.oracle.com/pls/topic/lookup?ctx=ct900&id=homepage

Application	Title	Format	Location
Start Here	*Sun Netra CT900 Server Start Here*	Printed PDF	Shipping kit Online
Latest information	*Sun Netra CT900 Server Product Notes*	PDF HTML	Online
Overview	*Sun Netra CT900 Server Overview*	PDF HTML	Online
Installation	*Sun Netra CT900 Server Installation Guide*	PDF HTML	Online
Upgrade	*Sun Netra CT900 Server Upgrade Guide*	PDF HTML	Fan tray kit Online
Service	*Sun Netra CT900 Server Service Manua*	PDF HTML	Online

Application	Title	Format	Location
Reference	*Sun Netra CP3140 Switch Software Reference Manual*	PDF HTML	Online
Safety	*Sun Netra CT900 Server Safety and Compliance Manual*	PDF HTML	Online
Safety	*Important Safety Information for Sun Hardware Systems (printed version only)*	Printed	Shipping kit

You might want to refer to documentation on the following products for additional information: the Oracle Solaris OS, OpenBoot PROM firmware, the Sun Netra CP3010 board, Sun Netra CP3020 board, Sun Netra ATCA CP3220 blade server, Sun Netra ATCA CP3060 blade server, and Sun Netra ATCA CP3260 blade server. These documents are available online.

Documentation, Support, and Training

These web sites provide additional resources:

- Documentation `http://www.oracle.com/technetwork/indexes/documentation/index.html`
- Support `https://support.oracle.com`
- Training `https://education.oracle.com`

CHAPTER 1

Programming Environment

This chapter provides an overview of the software environment that forms the basis for developing applications for the Sun Netra CT900 server:

- "Sun Netra CT900 Server" on page 1
- "Hardware Descriptions" on page 2
- "Software Descriptions" on page 4
- "Management Framework" on page 6

Sun Netra CT900 Server

The Netra CT 900 server is an Advanced Telecom Computing Architecture (AdvancedTCA® or ATCA) packet-switching, backplane-based, rackmountable server.

The Netra CT 900 server complies with the following specifications:

- PICMG® 3.0 Revision 2.0 AdvancedTCA specifications
- PICMG 3.1 Revision 1.0 AdvancedTCA specifications

The hardware components for the Netra CT 900 server can be broken down into four sections:

- The shelf
- The shelf alarm panel
- The shelf management card
- The switch

Note – The Advanced Telecom Computing Architecture® (ATCA) has adopted the term *shelf* for alignment with typical practice in telecommunications Traditionally, the term *chassis* has been used with essentially the same meaning.

Hardware Descriptions

This section contains descriptions of the major components of the Sun Netra CT900 server.

The Shelf

The shelf features twelve node board slots and a redundant infrastructure (switch, management, power, and cooling), making it ideal for carrier-grade telecom and Internet applications. Beyond its high-availability features, the Netra CT 900 server is highly modular, scalable, and serviceable.

Hot-swappable system components provide built-in redundancy to simplify replacement and minimize service time. Redundant shelf management cards enable customers to manage multiple processor boards and conduct shelf diagnostics remotely for enhanced system reliability. Two 8U slots are reserved for PICMG 3.0/3.1 switches. The Netra CT 900 server routes Ethernet signals across the midplane without the use of cables, saving time in setup, maintenance, and repair, and eliminating the thermal challenges of traditional cabling methods.

The Shelf Alarm Panel

The shelf alarm panel (SAP) is a removable module mounted at the top right side of the shelf, above slots 9 through 14 in the shelf. It provides the connectors for the serial console interfaces of the shelf management cards, the telco alarm connector, the Telco Alarm LEDs, the user-definable LEDs, and the Alarm Silence push button.

The I^2C-bus devices on the shelf alarm panel are connected to the master-only I^2C-bus of both shelf management cards. Only the active shelf management card has access to the shelf alarm panel.

The Shelf Management Card

The Netra CT 900 server has two dedicated slots for the shelf management cards. Each shelf management card is a 78 mm by 280 mm form factor board with a SODIMM socket for the shelf management mezzanine (ShMM) device. The Netra CT 900 server has radial IPMBs and is designed to work with two redundant shelf management cards. The shelf management card also contains the fan controller for the three hot-swappable fan trays, and provides individual Ethernet connections to both switches.

The dual-IPMB interface from the ShMM is connected to the dual IPMBs on an ATCA node board through radial connections in the Netra CT 900 server midplane. Each shelf management card has an Ethernet port that is *not* available to the user; instead, Ethernet traffic from the shelf management card is routed to the Ethernet ports on the switches. Serial and telco alarm traffic from the shelf management card are routed to the ports and LEDs on the shelf alarm panel.

The shelf management card includes several on-board devices that enable different aspects of shelf management based on the ShMM. These facilities include I^2C-based hardware monitoring and control and General Purpose Input/Output (GPIO) expander devices.

The Switch

The switch for the Netra CT 900 server is an AdvancedTCA 3.0 and 3.1 Option 1 switch. This means that the switch implements two separate switched networks on a single printed circuit board (PCB). By separating the Base (3.0) and Extended Fabric (3.1) networks, the switch provides a separate control plane and data plane. It provides 10/100/1000BASE-T Ethernet switching on the 3.0 Base Fabric interface and on the 3.1 Extended Fabric interface it provides 1000BASE-X Ethernet switching. Both of these networks are fully managed and work with the robust FASTPATH management suite. Both networks support Layer 2 switching as well as Layer 3 routing. The switch also supports a rear transition module to expand connectivity with uplink ports.

Software Descriptions

The Sun Netra CT900 server software includes:

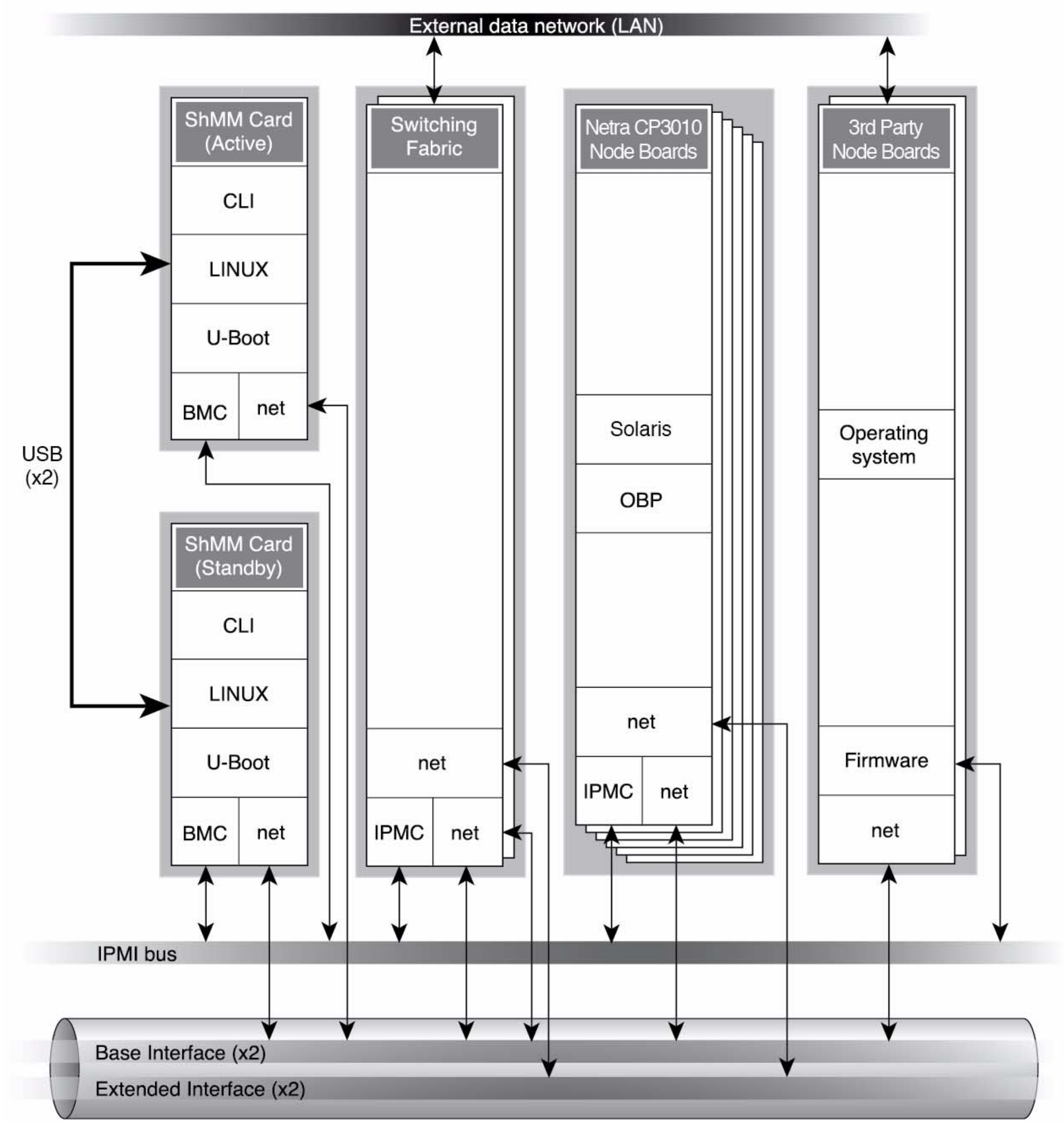
- Shelf Manager
- Operating systems and applications
- Firmware

The software is described in TABLE 1-1 and represented logically, with the hardware, in FIGURE 1-1.

TABLE 1-1 Sun Netra CT Server Software for System Administrators

Category	Name	Description
Shelf Management	IPM Sentry Shelf Manager	The Shelf Manager software runs on the shelf management card (ShMM) and is factory installed. It provides Remote Management Control Protocol (RMCP) and CLI access to IPMI for managing the server.
	Command-line interface (CLI)	The CLI is an onboard user interface to the Shelf Manager.
Operating Systems and Applications	Oracle Solaris Operating System (Solaris OS)	The Solaris OS runs on Sun-supported ATCA-compatible node boards, like the Sun Netra CP3010, Sun Netra CP3020, and CP3060 node boards. Solaris 10 is optionally preinstalled on the Sun Netra node boards. Solaris 10 and other versions of the Solaris OS can be downloaded and installed by the user.
	Monta Vista Carrier Grade Linux OS	The Sun Netra CP3020 can also run the Monta Vista Carrier Grade Linux OS.
Firmware	OpenBoot PROM firmware	Firmware on Sun-supported node boards, such as the Sun Netra CP3010 board, that controls booting. It includes diagnostics.
	U-Boot	Firmware on the shelf management cards that performs power-on self-test (POST) and controls booting of the shelf management card software.
	Intelligent Platform Management Controller (IPMC)	System management controller firmware that enables communication over the IPMI controller on a Sun-supported node board, such as the Sun Netra CP3010 board.

FIGURE 1-1 Logical Representation of Software and Hardware Interfaces in a *Sun Netra CT900 Server*



Management Framework

The Shelf Manager is a shelf-level management solution for ATCA products. The shelf management card provides the necessary hardware to run the Shelf Manager within an ATCA shelf. This overview focuses on aspects of the Shelf Manager and shelf management card that are common to any shelf management carrier used in an ATCA context.

Overview of Intelligent Platform Management in ATCA

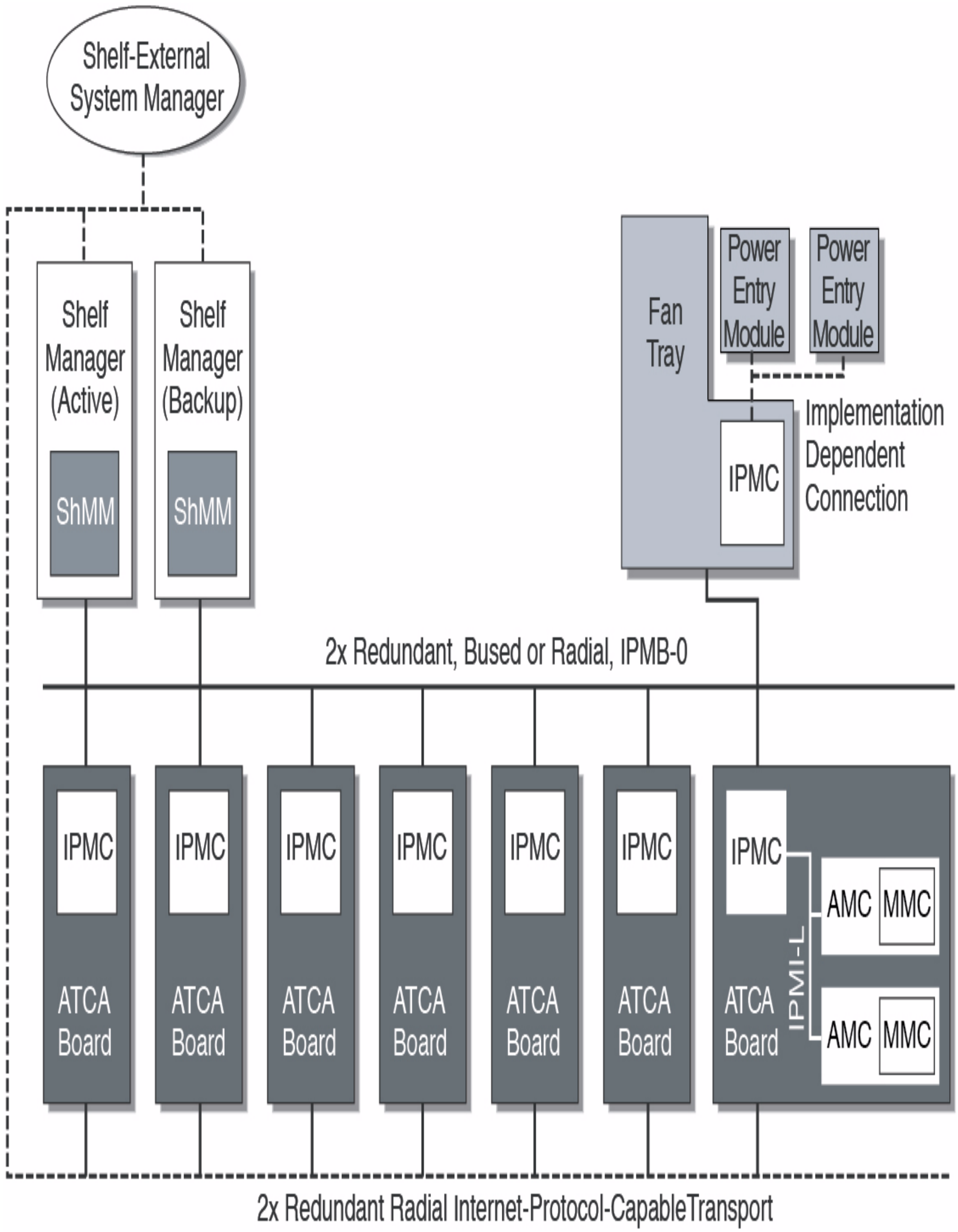
The Shelf Manager and shelf management card are Intelligent Platform Management (IPM) building blocks designed for modular platforms like ATCA, in which there is a strong focus on a dynamic population of FRUs and maximum service availability. The IPMI specification provides a solid foundation for the management of such platforms, but requires significant extension to support them well. PICMG 3.0, the ATCA specification, defines the necessary extensions to IPMI.

An AdvancedTCA Shelf Manager communicates inside the shelf with IPM Controllers, each of which is responsible for local management of one or more field replaceable units (FRUs), such as boards, fan trays or power entry modules. Management communication within a shelf occurs primarily over the Intelligent Platform Management Bus (IPMB), which is implemented on a dual-redundant basis in AdvancedTCA.

The PICMG Advanced Mezzanine Card (AdvancedMC or AMC) specification, AMC.0, defines a hot-swappable mezzanine form factor designed to fit smoothly into the physical and management architecture of AdvancedTCA.

FIGURE 1-2 includes an AMC carrier with an IPMC and two installed AMC modules, each with a Module Management Controller (MMC). On-carrier management communication occurs over IPMB-L ("L" for Local).

FIGURE 1-2 Example of ATCA Shelf



An overall system manager (typically external to the shelf) can coordinate the activities of multiple shelves. A system manager typically communicates with each Shelf Manager over an Ethernet or serial interface.

FIGURE 1-2 shows three levels of management: board, shelf, and system. The next section addresses the Shelf Manager software and shelf management card which implement an ATCA-compliant shelf manager and shelf management controller (ShMC).

Shelf Manager and Shelf Management Card

The Shelf Manager (consistent with ATCA Shelf Manager requirements) has two main responsibilities:

- Manage and track the FRU population and common infrastructure of a shelf, especially the power, cooling and interconnect resources and their usage. Within the shelf, this management and tracking primarily occurs through interactions between the Shelf Manager and the IPM controllers over Intelligent Platform Management bus 0 (IPMB-0).
- Enable the overall System Manager to join in that management and tracking through the System Manager interface, which is typically implemented over Ethernet.

Much of the Shelf Manager software is devoted to routine missions such as powering a shelf up or down and handling the arrival or departure of FRUs, including negotiating assignments of power and interconnect resources and monitoring the health status of each FRU. In addition, the Shelf Manager can take direct action when exceptions are raised in the shelf. For instance, in response to temperature exceptions the Shelf Manager can raise the fan levels or, if that step is not sufficient, even start powering down FRUs to reduce the heat load in the shelf.

Shelf Manager Features

The Shelf Manager software features include:

- Executes on the shelf management card, a compact SO-DIMM form-factor module, installed on a suitable carrier board for the shelf.
- Conforms to the ATCA specification.
- Monitors activities within the shelf via the ATCA-specified dual-redundant Intelligent Platform Management bus (IPMB).
- Accepts and logs events posted by any intelligent FRU in the shelf (reflecting exceptions in temperatures, voltages, etc.); posts alerts outside the shelf based on configurable IPMI Platform Event Filters.
- Supports hot-swapping of field-replaceable units (FRUs), while maintaining full management visibility.

- Interfaces to standard Telco Alarm infrastructures, via shelf management implemented dry contact relays.
- Supports redundant Shelf Manager instances for high availability.
- Integrates a watchdog timer, which resets the shelf management card if not periodically strobed; such resets automatically trigger a switchover to the backup shelf management card, if configured.
- Includes a battery-backed real-time clock for time stamping events.
- Implements a rich set of shelf-external interfaces accessible over Ethernet, including RMCP, required by ATCA, and CLI.

Hardware Resource Hierarchy

Each manageable component of the system is identified as a unique entity in the system. Every entity is uniquely named by an entity path that identifies the component in terms of its physical containment within the system.

An entity path consists of an ordered set of {Entity Type, Entity Location} pairs. The path defines the physical location of the entity in the system, in terms of which entity it is contained within and the entity that its container is contained in.

For more details, refer to the SAF-HPI-B.01.01 specification. You can obtain the specification at:

`http://saforum.org/`

Appendix A contains a presentation of the abbreviated resource table of a Sun Netra CT900 server, which contains two ShMM 500 shelf managers, two CT3140 switch blades, one CP3010 blade, one CP3020 blade, and one CP3060 blade.

Appendix B contains the resource data records for the 3.2 PICMG blades. The resource data records define the management instruments (sensors, controls, watchdog timers, inventory data repositories, or annunciators) associated with a resource.

System Administrator Interface Options

Another major subsystem of the Shelf Manager implements the System Administrator interface. The System Administrator is a logical concept that can include software as well as human operators in an operations center. The Shelf Manager provides two System Administrator interface options that provide different mechanisms of access to similar kinds of information and control regarding a shelf:

- IPMI Local Area Network (LAN) interface
- Command-line interface (CLI)

The IPMI LAN interface is used to maximize interoperability among independently implemented shelf products. This interface is required by the ATCA specification and supports IPMI messaging with the Shelf Manager through RMCP. A system administrator who uses RMCP to communicate with shelves should be able to interact with any ATCA-compliant Shelf Manager. This low-level interface provides access to the IPMI aspects of a shelf, including the ability for the system administrator to issue IPMI commands to IPMI controllers in the shelf, using the Shelf Manager as a proxy.

RMCP is a standard network interface to an IPMI controller through the LAN and is defined by the IPMI 1.5 specification.

The CLI provides a comprehensive set of textual commands that can be issued to the Shelf Manager through either a physical serial connection or a Telnet connection.

OpenHPI

The Open Hardware Platform Interface (OpenHPI) defines a C application programming interface to access platform management capabilities, such as:

- Configuration – The components in the system
- Inventory – The vendor, model, version, and serial number of the components
- Status – The temperature, voltage, fan speed, and state of the LEDs
- Control – The ability to power on, power off, and reset the system, along with setting the WDT

For a detailed description of the OpenHPI, along with supported return codes, refer to the OpenHPI specification at:

`http://www.openhpi.org/`

OpenHPI Overview

The Service Availability Forum (SAF) Hardware Platform Interface (HPI) specifies a generic mechanism to monitor and control highly available systems. The ability to monitor and control these systems is provided through a consistent, platform independent set of programmatic interfaces. The HPI specification provides data structures and functional definitions you can use to interact with manageable subsets of a platform or system. The HPI allows applications and middleware to access and manage hardware components through a standardized interface.

The HPI model includes four basic concepts: entities, resources, sessions, and domains. Each of these concepts is described briefly in this section.

Entities

Entities represent the physical components of the system. Each entity has a unique identifier, called an entity path, which is defined by the component's location in the physical containment hierarchy of the system.

Resources

Resources provide management access to the entities within the system. Frequently, resources represent functions performed by a local control processor used for management of the entity's hardware. Each resource is responsible for presenting a set of management instruments and management capabilities to the HPI User. Resources can be dynamically added and removed in a system as hot-swappable system components that include management capabilities are added and removed.

Sessions

Sessions provide all access to an HPI implementation by HPI users. An HPI session is opened on a single domain; one HPI user can have multiple sessions open at once, and there can be multiple sessions open on any given domain at once. Sessions also provide access to events created or forwarded by the domain accessed by the session. An HPI user accesses the system through sessions, where each session is opened on a domain. A session provides access to domain functions and to a set of resources that are accessible through the domain.

Domains

All HPI user functions are accessed through sessions, and each session is associated with a single domain. A domain provides access to zero or more resources and provides a set of associated services and capabilities. The latter are logically grouped into an abstraction called a domain controller. The resources that are accessible through a domain are listed in the domain's Resource Presence Table (RPT). The contents of this table can change over time, and the domain's session management capability rejects any attempt to access a resource that is not currently listed in the domain's Resource Presence Table.

Management Application Framework

As shown in FIGURE 1-3, the management application talks to the OpenHPI daemon through the OpenHPI library. The OpenHPI daemon talks to the platform (local or remote) through the plug-ins.

FIGURE 1-3 OpenHPI Architecure

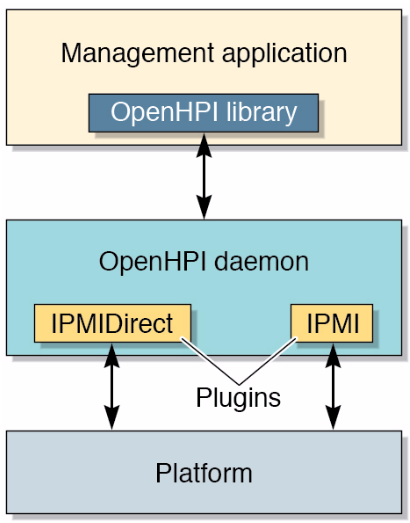
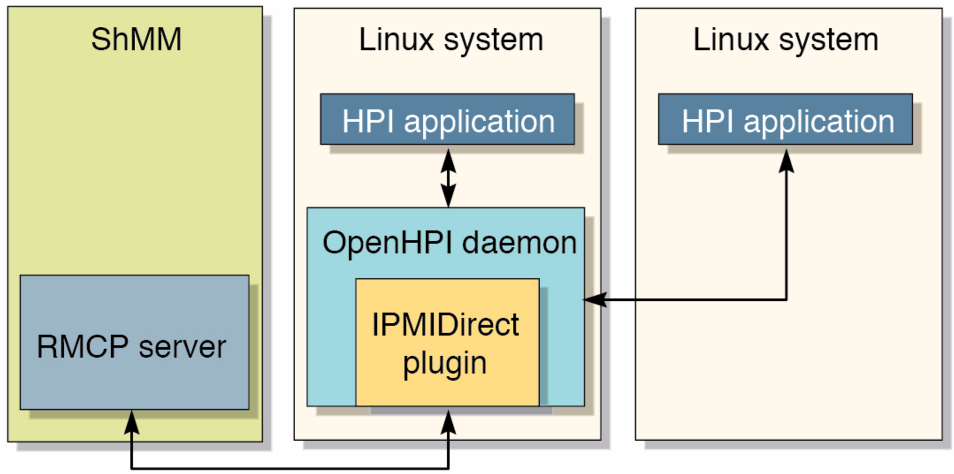


FIGURE 1-4 shows a Linux OS system running the OpenHPI daemon (IPMI direct plug-in), communicating with the ShMM over RMCP for shelf management.

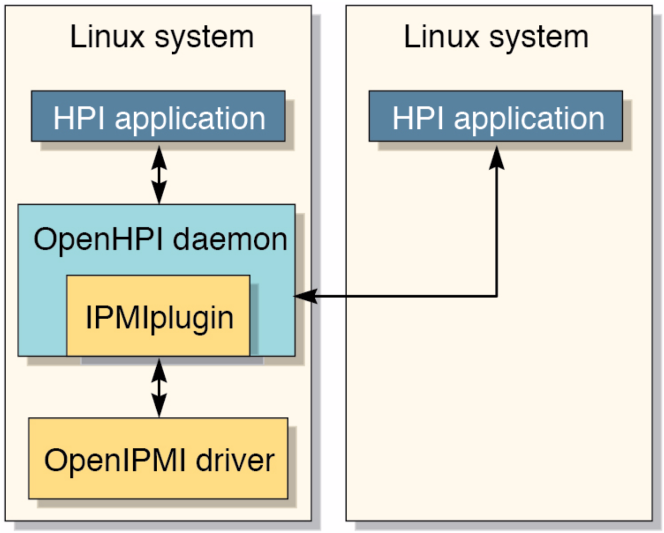
FIGURE 1-4 HPI Applications, OpenHPI Daemon, and RMCP Server Relationships



The SAF HPI draws heavily on the concepts set forth by the Intelligent Platform Management Interface (IPMI) specification to define platform-independent capabilities and data formats. Thus, an implementation of the HPI interface on a platform that uses IPMI as a platform management infrastructure can be very straightforward. However, because HPI is a generic interface specification, it can be implemented on any platform with sufficient underlying platform management technology.

FIGURE 1-5 shows the OpenHPI daemon (IPMI plug-in) running on a system with an OpenIPMI driver for local management.

FIGURE 1-5 HPI Application and OpenIPMI Driver Relationships



CHAPTER 2

Simple Network Management Protocol

The simple network management protocol (SNMP) forms part of the internet protocol suite, as defined by the Internet Engineering Task Force (IETF). SNMP is used by network management systems to monitor network-attached devices for conditions that warrant administrative attention. SNMP consists of a set of standards for network management, including an Application Layer protocol, a database schema, and a set of data objects.

This chapter includes descriptions of the SNMP network protocol and instructions on how to use the protocol.

This chapter contains the following topics:

For more information about SNMP, go to:

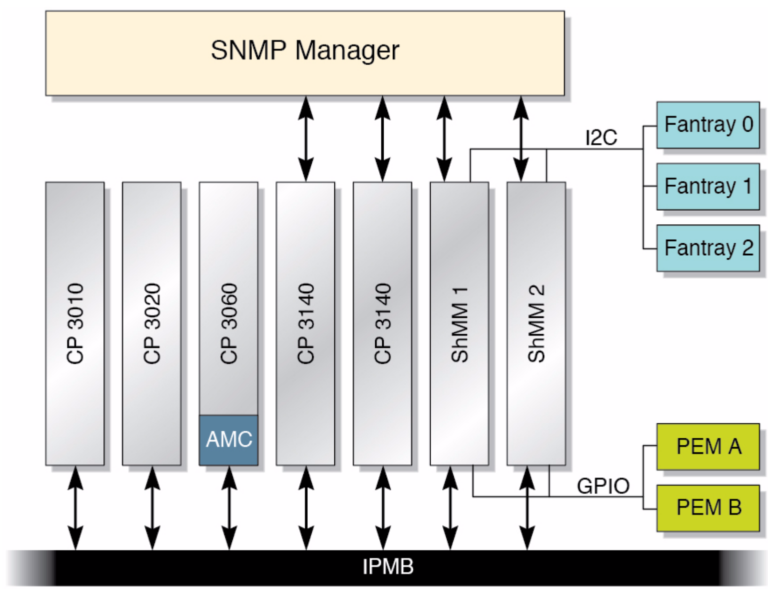
`http://net-snmp.sourceforge.net/`

SNMP Overview

To be managed, a device must have an SNMP agent associated with it. The agent receives requests for data representing the state of the device and provides an appropriate response. The agent can also control the state of the device. Additionally, the agent can generate SNMP traps, which are unsolicited messages sent to selected NMSs to signal significant events relating to the device.

FIGURE 2-1 shows a high-level overview of the Sun Netra CT900 server from the SNMP manager's perspective. Fan trays and power entry modules (PEMs) are just a couple examples of resources that are manageable through the ShMM.

FIGURE 2-1 Overview of the SNMP Management Relationships



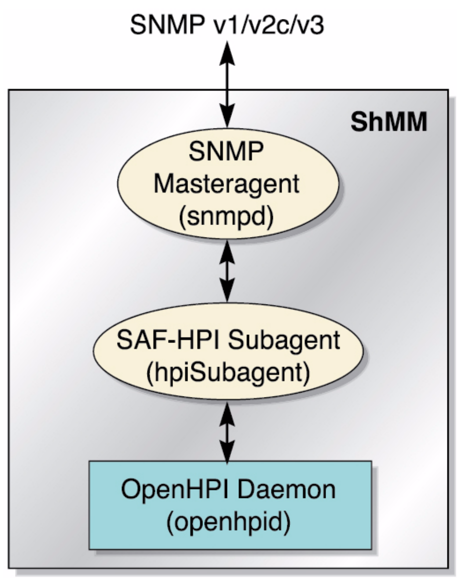
ShMM SNMP Architecture

System level SNMP support is provided through the ShMM, which oversees the status of each FRU in the system. SNMP support on the ShMM is implemented in a master- and sub-agent architecture. The `hpiSubagent` is an open source OpenHPI SNMP subagent, based on the *Service Availability Forum Hardware Platform Interface* specification (SAI-HPI-B.01.01).

The design of the master agent, as well as the communication protocol between the master agent and subagent, is beyond the scope of this document.

The following diagram illustrates the SNMP agent architecture on the ShMM.

FIGURE 2-2 SNMP Architecture



Blades, such as the CP3140 switch blade, can provide support for additional blade specific features locally through the SNMP agent on the blade.

ShMM SNMP Agent Configuration

The SNMP agent on the ShMM can be configured by modifying the `snmpd.conf` and `hpiSubagent.conf` files, both of which reside in the `/etc` directory.

The `hpiSubagent.conf` file contains parameters for configuring the HPI check interval, event rows, and event overflow action. The parameters in the `snmpd.conf` file are documented in the `snmpd.conf` manual page.

Understanding the MIB Variable Descriptions

The management information base (MIB) defines a virtual datastore accessible through the SNMP software (the content being provided either by corresponding data maintained by the agent) or through the agent obtaining the required data from the managed device. For data written to the virtual datastore by the network manager, the agent performs an action that affects the state of itself or the managed device.

In the Sun Netra CT900 server, SNMP support is provided through a master agent and subagent architecture, with the master agent (as of R3.0, provided by the PPS) handling support for the non-Sun Netra CT900 server specific SNMP objects (that is, MIB2). The Sun Netra CT900 server level SNMP support is defined primarily by the HPI MIB for SAF-HPI B-01-01 specification and is implemented by the `hpiSubagent` on the ShMM. Each CP3140 switch blade can also provide additional support for local objects of interest through its MIBs, which are accessed directly through the SNMP agent running on the blade.

SAF-HPI MIB

This MIB defines the HPI instrumentation based on the SAI-HPI-B.01.01 specification, which views a hardware platform as a collection of physical entities that can be managed individually.

A logical collection of entities comprises a management domain. Each entity has a common set of attributes reflected in the entity table and might have additional categorical attributes that might exist in one or more of the tables. Every entity is uniquely named by an entity path that identifies the component in terms of its physical containment within the system. Refer to the HPI specification for more details on the HPI model. You can obtain a copy of the HPI specification at:

`http://www.saforum.org/`

SAF-HPI MIB Table Hierarchy

The data associated with each managed entity is stored in the various tables defined by the SAF-HPI MIB. The relationship between these tables closely resembles the concepts specified in the HPI specification (domains->resources->entities). To traverse the tables, the proper index must first be constructed based on the `saHpiDomainId`, the `saHpiResourceEntryId`, the `saHpiResourceIsHistorical`, and the `saHpiRdrEntryId` when applicable.

The following sections contain diagrams which illustrate the relationship between some of the commonly used tables defined in the SAF-HPI MIB. For the description and the list of objects contained in each table, refer to the MIB file.

You can obtain copies of the standard MIBs, at:

`http://www.faqs.org/`

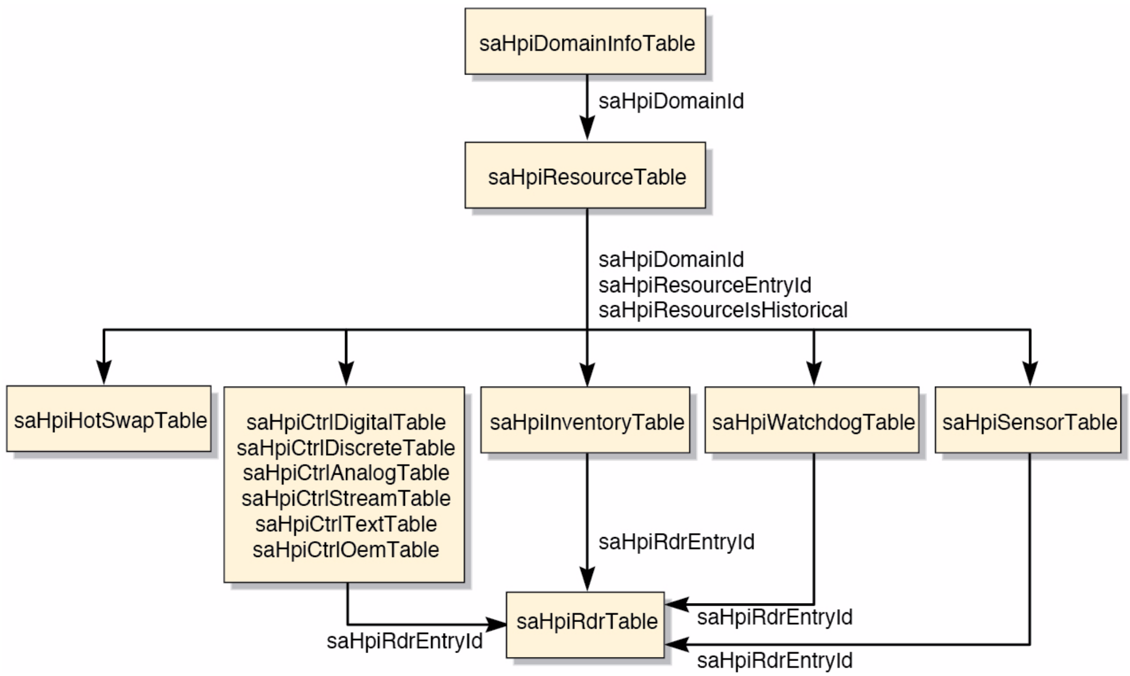
Entity Tables

Each entity has a common set of attributes that are reflected in an entity table. Entities can have categorical attributes that exist in one or more of the following tables:

- The hot-swap table defines the management attributes for an entity that supports hotswap (generally referred to as a FRU).
- The controls table defines the variables for reading and setting controls associated with an entity.
- The sensors table defines the variables for reading sensors associated with an entity, as well as controlling event generation for that sensor.
- The watchdog table defines the variables for reading watchdog events associated with an entitry.
- The inventory control table defines the variables for reading inventory resources and changing the settings.

FIGURE 2-3 illustrates the relationship between the entity tables.

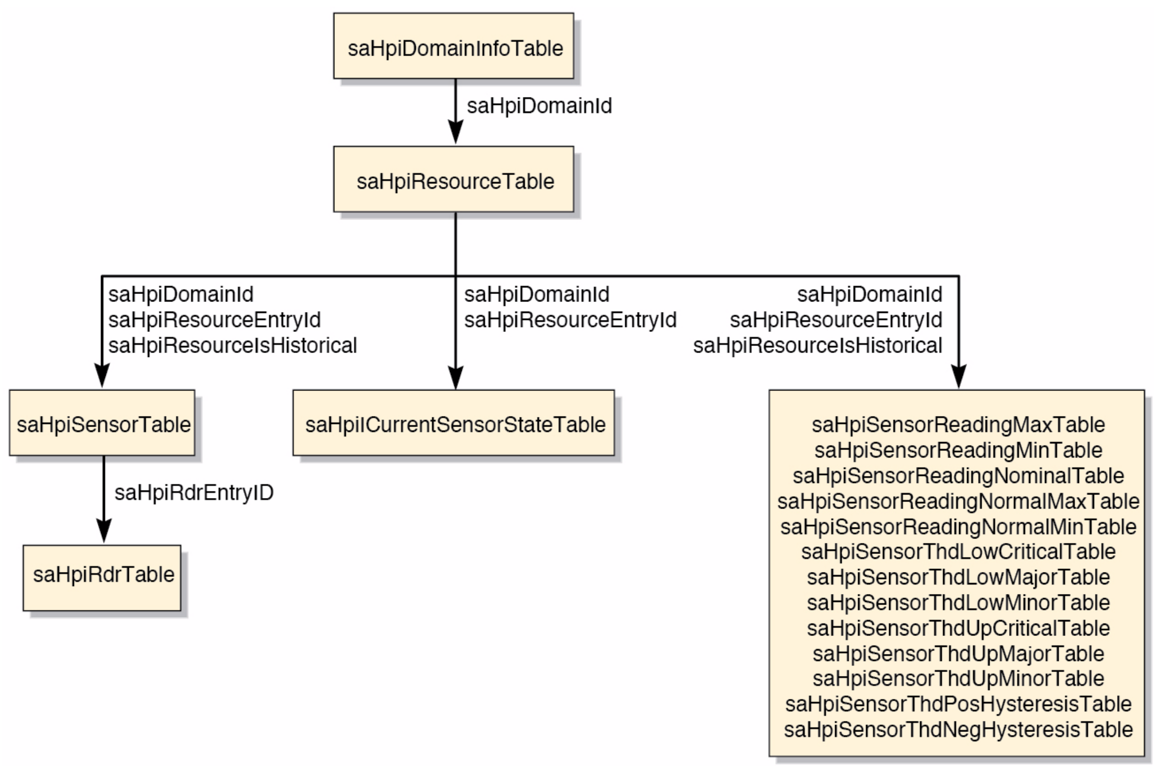
FIGURE 2-3 Enitity Table Relationships



Sensor Tables

FIGURE 2-4 illustrates the relationship between the sensor tables.

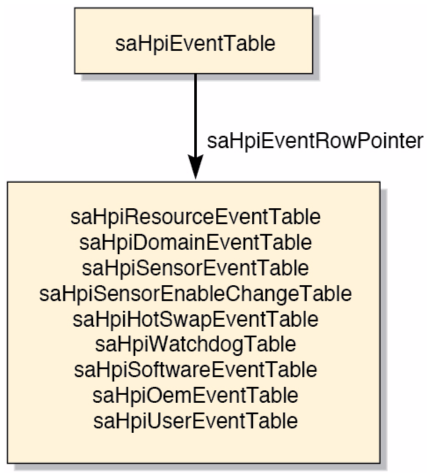
FIGURE 2-4 Sensor Table Relationships



Event Tables

The `saHpiEventTable` presents the list of all events that are present in the HPI system. This table is used as a master event table with an index that points to the specific subtable that contains more details on the event. FIGURE 2-5 illustrates the relationship between the `saHpiEventTable` and the event subtables.

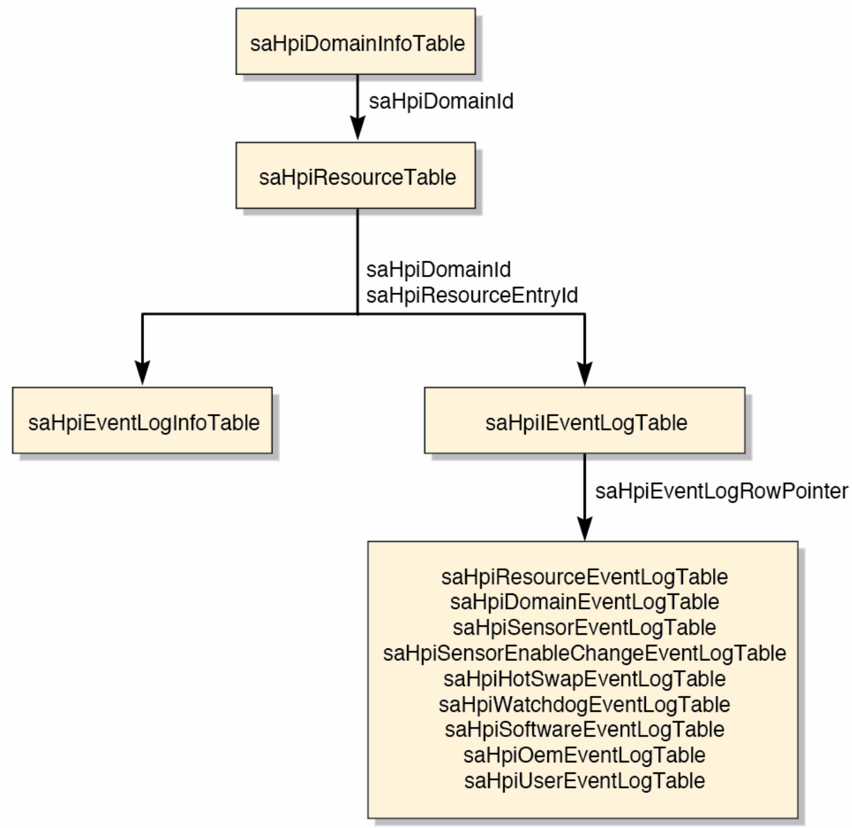
FIGURE 2-5 Event Table Relationships



Event Log Tables

FIGURE 2-6 illustrates the relationship between the event log tables.

FIGURE 2-6 Event Log Table Relationships



Configuring the `hpiSubagent`

The SNMP subagent configuration file, /etc/snmpd.conf, defines how the SNMP subagent operates and includes directives for access control. The following procedures describe how to set the access control and to enable SNMP version 3 usage.

▼ To Enable Read-Write Access

By default, the `hpiSubagent` is configured for read-only access in the access control section in the `snmpd.conf` file:

```
# # Enable read-only access for the "public" community.
rocommunity public
```

1. **Replace the `rocommunity` value with `rwcommunity`:**

```
# # Enable read-write access for the "public" community.
rwcommunity public
```

2. **Restart the `hpiSubagent`:**

```
# reboot
```

▼ To Enable SNMP Version 3 Usage of Subagent

Note – You must make the following changes in the `snmpd.conf` file on both ShMM cards.

---

1. **Configure the `engineID` in the `snmpd.conf` file:**

```
engineID string
```

You must configure the subagent with an `engineID` to be able to respond to SNMP version 3 messages. The default value of the `engineID` is the first IP address found for the host name of the machine.

2. **Configure the** `username`**, authentication type, and** `authpassphrase` **for the user.**

```
createUser username MD5|SHA authpassphrase DES privpassphrase
```

MD5 and SHA are authentication types. To use SHA, you must have already built the package with OpenSSL and installed it on the ShMMs. DES is the privacy protocol. If the *privpassphrase* is not specified, it is assumed to be the same as *authpassphrase*. As of release 3, SHA and DES are not supported. They are included in the command syntax for reference only.

The following is an example of the supported command syntax:

```
createUser admin MD5 admin123
```

This line creates the user named `admin` with the authentication type as `MD5` and the *authpassphrase* as `admin123`.

Note – As of release 3.0, SHA and DES authentication are not supported.

---

3. **Configure the access control of user.**

```
rouser admin
```

This configuration line provides read-only access to the `admin` user.

```
rwuser admin
```

This configuration line provides read-write access to the `admin` user.

4. **Restart the** `hpiSubagent`**:**

```
# reset
```

5. **Check the SNMP usage with the** `snmpwalk` **command:**

```
snmpwalk -v3 -u admin -l authNoPriv -a MD5 -A admin123 ShMMIP
HPI-B0101-MIB::saHpiResourceTable.1
```

This is an example of `snmpwalk` on the `saHpiResourceTable` using SNMP version 3. The user is `admin`; the authentication type is `MD5`; and, the `authpassphrase` is `admin123`. *ShMMIP* is the IP address of the Shelf Manager.

Note – As of release 3.0, `authPriv` is not supported.

SNMP Usage Examples

The following sections include examples of how to use the `snmpwalk` command to view the contents of the HPI subagent MIB tables. All of the examples are based on a shelf with the following configuration:

- One active ShMM-500
- Three fan trays
- Two PEMs
- Two switches
- One CP3010 blade
- One CP3020 blade
- One CP3060 blade with AMC installed

All of the examples include *ShMMIP*, which is the IP address of the Shelf Manager.

Getting Information on Resources

The `saHpiResourceTable` contains the information on all of the resources in the ATCA shelf. Resources include slots, ATCA blades, switches, and ShMM cards. The information includes `ResourceId`, `ResourceTag`, `ResourceEntityPath`, and `ResourceCapabilities`. The index to the table is `domainID.resourceID.isHistorical`.

▼ To View All of the Information for All of the Resources of a Domain

● **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiResourceTable.1
```

where *ShMMIP* is the IP address of active Shelf Manager and `1` is domainID.

▼ To View a Column of Data for All of the Resources of a Domain

The following example shows how to view the `ResourceTag` for all of the resources on a shelf with one ShMM.

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiResourceTag.1

HPI-B0101-MIB::saHpiResourceTag.1.1.false = STRING: "Shelf Resource"
HPI-B0101-MIB::saHpiResourceTag.1.2.false = STRING: "OEM Slot 1"
HPI-B0101-MIB::saHpiResourceTag.1.3.false = STRING: "ATCA Board Slot 1"
HPI-B0101-MIB::saHpiResourceTag.1.4.false = STRING: "ATCA Board Slot 2"
HPI-B0101-MIB::saHpiResourceTag.1.5.false = STRING: "ATCA Board Slot 3"
HPI-B0101-MIB::saHpiResourceTag.1.6.false = STRING: "ATCA Board Slot 4"
HPI-B0101-MIB::saHpiResourceTag.1.7.false = STRING: "ATCA Board Slot 5"
HPI-B0101-MIB::saHpiResourceTag.1.8.false = STRING: "ATCA Board Slot 6"
HPI-B0101-MIB::saHpiResourceTag.1.9.false = STRING: "ATCA Board Slot 7"
HPI-B0101-MIB::saHpiResourceTag.1.10.false = STRING: "ATCA Board Slot 8"
HPI-B0101-MIB::saHpiResourceTag.1.11.false = STRING: "ATCA Board Slot 9"
HPI-B0101-MIB::saHpiResourceTag.1.12.false = STRING: "ATCA Board Slot 10"
HPI-B0101-MIB::saHpiResourceTag.1.13.false = STRING: "ATCA Board Slot 11"
HPI-B0101-MIB::saHpiResourceTag.1.14.false = STRING: "ATCA Board Slot 12"
HPI-B0101-MIB::saHpiResourceTag.1.15.false = STRING: "ATCA Board Slot 13"
HPI-B0101-MIB::saHpiResourceTag.1.16.false = STRING: "ATCA Board Slot 14"
HPI-B0101-MIB::saHpiResourceTag.1.17.false = STRING: "Power Entry Module Slot 1"
HPI-B0101-MIB::saHpiResourceTag.1.18.false = STRING: "Power Entry Module Slot 2"
HPI-B0101-MIB::saHpiResourceTag.1.19.false = STRING: "Shelf FRU Information Slot
1"
HPI-B0101-MIB::saHpiResourceTag.1.20.false = STRING: "Shelf FRU Information Slot
2"
HPI-B0101-MIB::saHpiResourceTag.1.21.false = STRING: "Dedicated ShMc Slot 1"
HPI-B0101-MIB::saHpiResourceTag.1.22.false = STRING: "Dedicated ShMc Slot 2"
HPI-B0101-MIB::saHpiResourceTag.1.23.false = STRING: "Fan Tray Slot 1"
HPI-B0101-MIB::saHpiResourceTag.1.24.false = STRING: "Fan Tray Slot 2"
HPI-B0101-MIB::saHpiResourceTag.1.25.false = STRING: "Fan Tray Slot 3"
HPI-B0101-MIB::saHpiResourceTag.1.26.false = STRING: "Alarm Slot 1"
HPI-B0101-MIB::saHpiResourceTag.1.27.false = STRING: "PPS BMC"
HPI-B0101-MIB::saHpiResourceTag.1.28.false = STRING: "Shelf EEPROM 1"
HPI-B0101-MIB::saHpiResourceTag.1.29.false = STRING: "Shelf EEPROM 2"
HPI-B0101-MIB::saHpiResourceTag.1.30.false = STRING: "SAP Board"
HPI-B0101-MIB::saHpiResourceTag.1.31.false = STRING: "Fan Tray 0"
HPI-B0101-MIB::saHpiResourceTag.1.32.false = STRING: "Fan Tray 1"
HPI-B0101-MIB::saHpiResourceTag.1.33.false = STRING: "Fan Tray 2"
HPI-B0101-MIB::saHpiResourceTag.1.34.false = STRING: "PEM A"
HPI-B0101-MIB::saHpiResourceTag.1.35.false = STRING: "PEM B"
```

```
HPI-B0101-MIB::saHpiResourceTag.1.36.false = STRING: "ATS1460"
HPI-B0101-MIB::saHpiResourceTag.1.37.false = STRING: "ShMM-500"
HPI-B0101-MIB::saHpiResourceTag.1.38.false = STRING: "ATS1160"
HPI-B0101-MIB::saHpiResourceTag.1.39.false = STRING: "NetraCP-3010"
HPI-B0101-MIB::saHpiResourceTag.1.40.false = ""
HPI-B0101-MIB::saHpiResourceTag.1.41.false = ""
HPI-B0101-MIB::saHpiResourceTag.1.42.false = STRING: "NetraCP-3020"
HPI-B0101-MIB::saHpiResourceTag.1.43.false = ""
HPI-B0101-MIB::saHpiResourceTag.1.44.false = ""
HPI-B0101-MIB::saHpiResourceTag.1.45.false = STRING: "NetraCP-3060"
HPI-B0101-MIB::saHpiResourceTag.1.46.false = STRING: "AMC Slot 1"
HPI-B0101-MIB::saHpiResourceTag.1.47.false = ""
HPI-B0101-MIB::saHpiResourceTag.1.48.false = ""
HPI-B0101-MIB::saHpiResourceTag.1.49.false = STRING: "SB-AMC-HD-A-40"
......
```

The following example shows how to view the `ResourceTag` for all of the resources on a shelf with two ShMMs.

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiResourceTag.1

HPI-B0101-MIB::saHpiResourceTag.1.1.false = STRING: "Shelf Resource"
HPI-B0101-MIB::saHpiResourceTag.1.2.false = STRING: "OEM Slot 1"
HPI-B0101-MIB::saHpiResourceTag.1.3.false = STRING: "ATCA Board Slot 1"
HPI-B0101-MIB::saHpiResourceTag.1.4.false = STRING: "ATCA Board Slot 2"
HPI-B0101-MIB::saHpiResourceTag.1.5.false = STRING: "ATCA Board Slot 3"
HPI-B0101-MIB::saHpiResourceTag.1.6.false = STRING: "ATCA Board Slot 4"
HPI-B0101-MIB::saHpiResourceTag.1.7.false = STRING: "ATCA Board Slot 5"
HPI-B0101-MIB::saHpiResourceTag.1.8.false = STRING: "ATCA Board Slot 6"
HPI-B0101-MIB::saHpiResourceTag.1.9.false = STRING: "ATCA Board Slot 7"
HPI-B0101-MIB::saHpiResourceTag.1.10.false = STRING: "ATCA Board Slot 8"
HPI-B0101-MIB::saHpiResourceTag.1.11.false = STRING: "ATCA Board Slot 9"
HPI-B0101-MIB::saHpiResourceTag.1.12.false = STRING: "ATCA Board Slot 10"
HPI-B0101-MIB::saHpiResourceTag.1.13.false = STRING: "ATCA Board Slot 11"
HPI-B0101-MIB::saHpiResourceTag.1.14.false = STRING: "ATCA Board Slot 12"
HPI-B0101-MIB::saHpiResourceTag.1.15.false = STRING: "ATCA Board Slot 13"
HPI-B0101-MIB::saHpiResourceTag.1.16.false = STRING: "ATCA Board Slot 14"
HPI-B0101-MIB::saHpiResourceTag.1.17.false = STRING: "Power Entry Module Slot 1"
HPI-B0101-MIB::saHpiResourceTag.1.18.false = STRING: "Power Entry Module Slot 2"
HPI-B0101-MIB::saHpiResourceTag.1.19.false = STRING: "Shelf FRU Information
Slot 1"
HPI-B0101-MIB::saHpiResourceTag.1.20.false = STRING: "Shelf FRU Information
Slot 2"
HPI-B0101-MIB::saHpiResourceTag.1.21.false = STRING: "Dedicated ShMc Slot 1"
HPI-B0101-MIB::saHpiResourceTag.1.22.false = STRING: "Dedicated ShMc Slot 2"
HPI-B0101-MIB::saHpiResourceTag.1.23.false = STRING: "Fan Tray Slot 1"
```

```
HPI-B0101-MIB::saHpiResourceTag.1.24.false = STRING: "Fan Tray Slot 2"
HPI-B0101-MIB::saHpiResourceTag.1.25.false = STRING: "Fan Tray Slot 3"
HPI-B0101-MIB::saHpiResourceTag.1.26.false = STRING: "Alarm Slot 1"
HPI-B0101-MIB::saHpiResourceTag.1.27.false = STRING: "PPS BMC"
HPI-B0101-MIB::saHpiResourceTag.1.28.false = STRING: "Shelf EEPROM 1"
HPI-B0101-MIB::saHpiResourceTag.1.29.false = STRING: "Shelf EEPROM 2"
HPI-B0101-MIB::saHpiResourceTag.1.30.false = STRING: "SAP Board"
HPI-B0101-MIB::saHpiResourceTag.1.31.false = STRING: "Fan Tray 0"
HPI-B0101-MIB::saHpiResourceTag.1.32.false = STRING: "Fan Tray 1"
HPI-B0101-MIB::saHpiResourceTag.1.33.false = STRING: "Fan Tray 2"
HPI-B0101-MIB::saHpiResourceTag.1.34.false = STRING: "PEM A"
HPI-B0101-MIB::saHpiResourceTag.1.35.false = STRING: "PEM B"
HPI-B0101-MIB::saHpiResourceTag.1.36.false = STRING: "ATS1460"
HPI-B0101-MIB::saHpiResourceTag.1.37.false = STRING: "ATS1160"
HPI-B0101-MIB::saHpiResourceTag.1.38.false = STRING: "ShMM-500"
HPI-B0101-MIB::saHpiResourceTag.1.39.false = STRING: "ShMM-500"
HPI-B0101-MIB::saHpiResourceTag.1.40.false = STRING: "NetraCP-3010"
HPI-B0101-MIB::saHpiResourceTag.1.41.false = ""
HPI-B0101-MIB::saHpiResourceTag.1.42.false = ""
HPI-B0101-MIB::saHpiResourceTag.1.43.false = STRING: "NetraCP-3020"
HPI-B0101-MIB::saHpiResourceTag.1.44.false = ""
HPI-B0101-MIB::saHpiResourceTag.1.45.false = ""
```

Note – Resource IDs in both examples are not fixed or static. The same `snmpwalk` command could result in different resource IDs on different shelves. Even on the same shelf with a new instance of the HPI subagent, the assigned resource IDs could be different.

▼ To View a Specific Resource of a Domain

● **Type:**

```
snmpwalk -v 2c -c public ShMMIP HP-B0101-MIB::saHpiResourceTag.1.40
```

In this command example, the domainID is `1`, and the resource ID is `40`.

Getting Information on Properties

The `saHpiRdrTable` contains the resource data records for all resources. The information includes `RdrType` (where `Rdr` is a sensor, a control, or the watchdog), `RdrEntityPath`, and `RdrRowPointer` (which is a pointer to another table based on `RdrType`). If `RdrType` is a sensor, then the entry is a pointer to an entry in the sensor table. If `RdrType` is a control, then the entry is a pointer to an entry in the control table. The index to the table is `domainID.resourceID.isHistorical.RDRID`.

▼ To View the RDR Entries for All of the Resources of a Domain

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiRdrTable.1
```

where `1` is domain ID.

▼ To View a Column From the RDR Table for All of the Resources of a Domain

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiRdrType.1

HPI-B0101-MIB::saHpiRdrType.1.1.false.70416 = INTEGER: ctrlRdr(2)
HPI-B0101-MIB::saHpiRdrType.1.1.false.70417 = INTEGER: ctrlRdr(2)
HPI-B0101-MIB::saHpiRdrType.1.1.false.135168 = INTEGER: sensorRdr(3)
HPI-B0101-MIB::saHpiRdrType.1.1.false.135936 = INTEGER: sensorRdr(3)
HPI-B0101-MIB::saHpiRdrType.1.1.false.196608 = INTEGER: inventoryRdr(4)
HPI-B0101-MIB::saHpiRdrType.1.2.false.69664 = INTEGER: ctrlRdr(2)
HPI-B0101-MIB::saHpiRdrType.1.2.false.135184 = INTEGER: sensorRdr(3)
HPI-B0101-MIB::saHpiRdrType.1.2.false.135185 = INTEGER: sensorRdr(3)
HPI-B0101-MIB::saHpiRdrType.1.2.false.135186 = INTEGER: sensorRdr(3)
HPI-B0101-MIB::saHpiRdrType.1.3.false.69664 = INTEGER: ctrlRdr(2)
HPI-B0101-MIB::saHpiRdrType.1.3.false.135184 = INTEGER: sensorRdr(3)
HPI-B0101-MIB::saHpiRdrType.1.3.false.135185 = INTEGER: sensorRdr(3)
HPI-B0101-MIB::saHpiRdrType.1.3.false.135186 = INTEGER: sensorRdr(3)
HPI-B0101-MIB::saHpiRdrType.1.4.false.69664 = INTEGER: ctrlRdr(2)
HPI-B0101-MIB::saHpiRdrType.1.4.false.135184 = INTEGER: sensorRdr(3)
HPI-B0101-MIB::saHpiRdrType.1.4.false.135185 = INTEGER: sensorRdr(3)
HPI-B0101-MIB::saHpiRdrType.1.4.false.135186 = INTEGER: sensorRdr(3)
HPI-B0101-MIB::saHpiRdrType.1.5.false.69664 = INTEGER: ctrlRdr(2)
......
```

This command searches on the `RdrType` for all of the data records. The output shows the string `ctrlRdr` and an integer. The `2` represents a control RDR. The `3` represents a sensor RDR, and the `4` represents an inventory RDR.

▼ To View a Column From the RDR Table for a Resource of a Domain

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiRdrType.1.40

HPI-B0101-MIB::saHpiRdrType.1.40.false.131075 = INTEGER: sensorRdr(3)
HPI-B0101-MIB::saHpiRdrType.1.40.false.131076 = INTEGER: sensorRdr(3)
HPI-B0101-MIB::saHpiRdrType.1.40.false.131077 = INTEGER: sensorRdr(3)
HPI-B0101-MIB::saHpiRdrType.1.40.false.131078 = INTEGER: sensorRdr(3)
```

where `1` is domain ID and `40` is the resource ID.

▼ To View What Each of the RDR Entries Represent

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiRdrIdString.1.40

HPI-B0101-MIB::saHpiRdrIdString.1.40.false.131075 = STRING: "BMC Watchdog"
HPI-B0101-MIB::saHpiRdrIdString.1.40.false.131076 = STRING: "CPU1 Temp"
HPI-B0101-MIB::saHpiRdrIdString.1.40.false.131077 = STRING: "CPU2 Temp"
HPI-B0101-MIB::saHpiRdrIdString.1.40.false.131078 = STRING: "Inlet Temp"
```

▼ To View a Column From the RDR Table for an RDR EntryID

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiRdrType.1.40.false.131076

HPI-B0101-MIB::saHpiRdrType.1.40.false.131076 = INTEGER: sensorRdr(3)
```

where `1` is domain ID, `40` is resource ID, `false` is the `isHistorical` value, and `131076` is the RDR entry ID.

Getting Information on Sensors

The `saHpiSensorTable` has information on all of the sensors for all of the resources. The information includes `SensorType` (for instance, temperature or voltage), `SensorCategory` (for instance, threshold, presence, or enable), and `SensorBaseUnits` (for instance, volts or degrees in Celsius).

The `saHpiCurrentSensorStateTable` contains information on the current state of all sensors for all of the resources, including:

- Current value
- Event state of sensor
- Whether or not the sensor is enabled
- Whether or not the event generation was from an enabled sensor

The index to the `saHpiSensorTable` is `domainID.resourceID.isHistorical.sensorNum`.

The index to `saHpiCurrentSensorStateTable` is `domainID.resourceID.sensorNum`.

▼ To View Information on All of the Sensors for All of the Resources in a Domain

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiSensorTable.1
```

where `1` is domainID.

▼ To View a Column From the Sensor Table for All of the Resources on a Domain

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiSensorType.1

HPI-B0101-MIB::saHpiSensorType.1.1.false.4096 = INTEGER: operational(161)
HPI-B0101-MIB::saHpiSensorType.1.1.false.4864 = INTEGER: oemSensor(193)
HPI-B0101-MIB::saHpiSensorType.1.2.false.4112 = INTEGER: entityPresence(38)
HPI-B0101-MIB::saHpiSensorType.1.2.false.4113 = INTEGER:
otherUnitsBasedSensor(12)
HPI-B0101-MIB::saHpiSensorType.1.2.false.4114 = INTEGER:
otherUnitsBasedSensor(12)
HPI-B0101-MIB::saHpiSensorType.1.3.false.4112 = INTEGER: entityPresence(38)
HPI-B0101-MIB::saHpiSensorType.1.3.false.4113 = INTEGER:
otherUnitsBasedSensor(12)
HPI-B0101-MIB::saHpiSensorType.1.3.false.4114 = INTEGER:
otherUnitsBasedSensor(12)
HPI-B0101-MIB::saHpiSensorType.1.4.false.4112 = INTEGER: entityPresence(38)
......
```

In this example, the command returns the sensor type information for all of the sensors for all of the resources.

▼ To View a Column From Sensor Table for a Resource

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiSensorType.1.40

HPI-B0101-MIB::saHpiSensorType.1.40.false.3 = INTEGER: reserved2(36)
HPI-B0101-MIB::saHpiSensorType.1.40.false.4 = INTEGER: temperature(2)
HPI-B0101-MIB::saHpiSensorType.1.40.false.5 = INTEGER: temperature(2)
HPI-B0101-MIB::saHpiSensorType.1.40.false.6 = INTEGER: temperature(2)
```

This command returns the sensor type information for a specific resource, where `1` is domain ID and `40` is the resource ID.

There are three temperature sensors for resource `40`. The sensor numbers are `4`, `5`, and `6` respectively.

▼ To View the Sensor Base Unit of Measurement for All Sensors for a Resource

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiSensorBaseUnits.1.40

HPI-B0101-MIB::saHpiSensorBaseUnits.1.40.false.3 = INTEGER: unspecified(1)
HPI-B0101-MIB::saHpiSensorBaseUnits.1.40.false.4 = INTEGER: degreesC(2)
HPI-B0101-MIB::saHpiSensorBaseUnits.1.40.false.5 = INTEGER: degreesC(2)
HPI-B0101-MIB::saHpiSensorBaseUnits.1.40.false.6 = INTEGER: degreesC(2)
```

For sensors `4`, `5`, and `6`, the sensor type is temperature and base measuement unit is degrees in Celsius.

▼ To View a Column From the Sensor Table for a Sensor of a Resource

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiSensorType.1.40.false.4

HPI-B0101-MIB::saHpiSensorType.1.40.false.4 = INTEGER: temperature(2)
```

where `1` is the domain ID, `40` is the resource ID, `false` is the `isHistorical` value, and `4` is the sensor number.

▼ To View the Current State of All of the Sensors for All of the Resources of a Domain

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiCurrentSensorStateTable.1
```

where `1` is domain ID.

▼ To View a Column From the Current Sensor State Table for All of the Resources of a Domain

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiCurrentSensorStateValue.1

HPI-B0101-MIB::saHpiCurrentSensorStateValue.1.1.4096 = ""
HPI-B0101-MIB::saHpiCurrentSensorStateValue.1.1.4864 = ""
HPI-B0101-MIB::saHpiCurrentSensorStateValue.1.2.4112 = STRING: "27"
HPI-B0101-MIB::saHpiCurrentSensorStateValue.1.2.4113 = STRING: "5e0"
HPI-B0101-MIB::saHpiCurrentSensorStateValue.1.2.4114 = STRING: "350"
HPI-B0101-MIB::saHpiCurrentSensorStateValue.1.3.4112 = STRING: "39"
HPI-B0101-MIB::saHpiCurrentSensorStateValue.1.3.4113 = STRING: "0e0"
HPI-B0101-MIB::saHpiCurrentSensorStateValue.1.3.4114 = STRING: "200"
HPI-B0101-MIB::saHpiCurrentSensorStateValue.1.4.4112 = STRING: "45"
HPI-B0101-MIB::saHpiCurrentSensorStateValue.1.4.4113 = STRING: "1e2"
HPI-B0101-MIB::saHpiCurrentSensorStateValue.1.4.4114 = STRING: "200"
......
```

▼ To View a Column From the Current Sensor State Table for a Resource

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiCurrentSensorStateValue.1.40

HPI-B0101-MIB::saHpiCurrentSensorStateValue.1.40.3 = STRING: "0e0"
HPI-B0101-MIB::saHpiCurrentSensorStateValue.1.40.4 = STRING: "9.2e1"
HPI-B0101-MIB::saHpiCurrentSensorStateValue.1.40.5 = STRING: "9.4e1"
HPI-B0101-MIB::saHpiCurrentSensorStateValue.1.40.6 = STRING: "3.3e1"
```

where `1` is the domain ID and `40` is the resource ID.

For resource `40`, there are three temperature sensors. The current values are 92, 94, and 33 degrees Celsius, respectively.

▼ To View a Column From the Current Sensor State Table for a Sensor of a Resource

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-
MIB::saHpiCurrentSensorStateValue.1.40.4

HPI-B0101-MIB::saHpiCurrentSensorStateValue.1.40.4 = STRING: "9.2e1"
```

where `1` is the domain ID, `40` is the resource ID, and `4` is the sensor number.

Getting and Setting Sensor Thresholds

The information in this section relates only to the threshold sensors (that is, the sensors that are categorized as threshold). The threshold information of sensors is contained in six tables:

- `saHpiSensorThdUpCriticalTable`, which includes information on the upper critical threshold for all of the threshold sensors for all of the resources
- `saHpiSensorThdUpMajorTable`, which includes information on the upper major threshold for all of the threshold sensors for all of the resources
- `saHpiSensorThdUpMinorTable`, which includes information on the upper minor threshold for all of the threshold sensors for all of the resources
- `saHpiSensorThdLowCriticalTable`, which includes information on the lower critical threshold for all of the threshold sensors for all of the resources
- `saHpiSensorThdLowMajorTable`, which includes information on the lower major threshold for all of the threshold sensors for all of the resources
- `saHpiSensorThdLowMinorTable`, which includes information on the lower minor threshold for all of the threshold sensors for all of the resources

The information in the tables is on same line and includes:

- Current threshold value
- Whether or not the value is readable
- Whether or not the value is writable

The index to the threshold tables is `domainID.resourceID.isHistorical.sensorNum`.

▼ To View All of the Information From the Upper Critical Sensor Threshold Table for All of the Sensors on All of the Resources of a Domain

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiSensorThdUpCriticalTable.1
```

where 1 is the domain ID.

▼ To View a Column From the Upper Critical Sensor Threshold Table for All of the Sensors on All of the Resources

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiSensorThdUpCriticalValue.1

HPI-B0101-MIB::saHpiSensorThdUpCriticalValue.1.2.false.4113 = STRING: "4e2"
HPI-B0101-MIB::saHpiSensorThdUpCriticalValue.1.3.false.4113 = STRING: "4e2"
HPI-B0101-MIB::saHpiSensorThdUpCriticalValue.1.4.false.4113 = STRING: "4e2"
HPI-B0101-MIB::saHpiSensorThdUpCriticalValue.1.5.false.4113 = STRING: "4e2"
HPI-B0101-MIB::saHpiSensorThdUpCriticalValue.1.6.false.4113 = STRING: "4e2"
HPI-B0101-MIB::saHpiSensorThdUpCriticalValue.1.7.false.4113 = STRING: "4e2"
HPI-B0101-MIB::saHpiSensorThdUpCriticalValue.1.8.false.4113 = STRING: "4e2"
HPI-B0101-MIB::saHpiSensorThdUpCriticalValue.1.9.false.4113 = STRING: "4e2"
......
```

▼ To View a Column From the Upper Critical Sensor Threshold Table on All of the Sensors of a Resource

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-
MIB::saHpiSensorThdUpCriticalValue.1.40

HPI-B0101-MIB::saHpiSensorThdUpCriticalValue.1.40.false.4 = STRING: "1.2e2"
HPI-B0101-MIB::saHpiSensorThdUpCriticalValue.1.40.false.5 = STRING: "1.2e2"
HPI-B0101-MIB::saHpiSensorThdUpCriticalValue.1.40.false.6 = STRING: "1.2e2"
```

where `1` is domain ID and `40` is resource ID.

The 4, 5, and 6 values are temperature sensors with measurement units in degrees of Celsius. The sensors have an upper critical threshold value of 120 degrees Celsius.

▼ To View a Column From the Upper Critical Sensor Threshold Table for a Sensor of a Resource

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-
MIB::saHpiSensorThdUpCriticalValue.1.40.false.4
```

where `4` is the sensor number, `40` is the resource, and `1` is the domain.

▼ To Set the Sensor Threshold for a Sensor

1. Confirm that the sensor threshold is writable;

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-
MIB::saHpiSensorThdUpCriticalIsWritable.1.40.false.4

HPI-B0101-MIB::saHpiSensorThdUpMinorIsWritable.1.40.false.4 = INTEGER: true(1)
```

This command returns the sensor information for the sensor number `4`.

2. **Confirm that the value being used is within the acceptable range for the sensor:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-
MIB::saHpiSensorReadingMinValue.1.40.false.4

HPI-B0101-MIB::saHpiSensorReadingMinValue.1.40.false.4 = STRING: "-4e1"

snmpwalk -v 2c -c public ShMMIP HPI-B0101-
MIB::saHpiSensorReadingMaxValue.1.40.false.4

HPI-B0101-MIB::saHpiSensorReadingMaxValue.1.40.false.4 = STRING: "1.25e2"
```

   The acceptable range of values for sensor `4` of resource `40` is -40 to 125.

3. **Type:**

```
snmpset -v 2c -c public ShMMIP HPI-B0101-
MIB::saHpiSensorThdUpCriticalValue.1.40.false.4 s 1.23e2
```

   where `1` is domain ID, `40` is resource ID, `4` is the sensor number , `s` indicates the type of value (which is string), and `1.23e2` is the value being set.

Getting and Setting Information on Controls

Control information is contained in six tables based on the control type. The following list contains the name and description of the tables:

- `saHpiCtrlAnalogTable` (for analog controls)
- `saHpiCtrlDigitalTable` (for digital controls)
- `saHpiCtrlDiscreteTable` (for discrete controls)
- `saHpiCtrlTextTable` (for text controls)
- `saHpiCtrlStreamTable` (for stream controls)
- `saHpiCtrlOemTable` (for OEM controls)

The information in all the tables is on similar lines; however, based on the control type, extra fields might appear in some of the tables. The common information includes:

- Control number
- Control mode
- Control state
- Control default state
- Whether or not the mode is read-only

- Whether or not the control is write-only

The index to all of the control tables is `domainID.resourceID.isHistorical.EntryID`.

▼ To View Information for All of the Analog Controls for All of the Resources

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiCtrlAnalogTable.1
```

where `1` is domain ID.

▼ To View a Column of the Control Analog Table for All of the Resources

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiCtrlAnalogMode.1

HPI-B0101-MIB::saHpiCtrlAnalogMode.1.2.false.0 = INTEGER: auto(1)
HPI-B0101-MIB::saHpiCtrlAnalogMode.1.3.false.0 = INTEGER: auto(1)
HPI-B0101-MIB::saHpiCtrlAnalogMode.1.4.false.0 = INTEGER: auto(1)
HPI-B0101-MIB::saHpiCtrlAnalogMode.1.5.false.0 = INTEGER: auto(1)
HPI-B0101-MIB::saHpiCtrlAnalogMode.1.6.false.0 = INTEGER: auto(1)
HPI-B0101-MIB::saHpiCtrlAnalogMode.1.7.false.0 = INTEGER: auto(1)
HPI-B0101-MIB::saHpiCtrlAnalogMode.1.8.false.0 = INTEGER: auto(1)
HPI-B0101-MIB::saHpiCtrlAnalogMode.1.9.false.0 = INTEGER: auto(1)
HPI-B0101-MIB::saHpiCtrlAnalogMode.1.10.false.0 = INTEGER: auto(1)
......
```

This command returns the control mode for all of the analog controls.

▼ To View a Column of the Control Analog Table for a Resource

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiCtrlAnalogMode.1.31

HPI-B0101-MIB::saHpiCtrlAnalogMode.1.31.false.0 = INTEGER: auto(1)
HPI-B0101-MIB::saHpiCtrlAnalogMode.1.31.false.1 = INTEGER: auto(1)
HPI-B0101-MIB::saHpiCtrlAnalogMode.1.31.false.2 = INTEGER: manual(2)
```

This command returns the control mode for all of the analog controls for resource `31`.

▼ To View the Control State for All of the Analog Controls for a Specific Resource

1. **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiCtrlAnalogState.1.31

HPI-B0101-MIB::saHpiCtrlAnalogState.1.31.false.0 = INTEGER: 1
HPI-B0101-MIB::saHpiCtrlAnalogState.1.31.false.1 = INTEGER: 900
HPI-B0101-MIB::saHpiCtrlAnalogState.1.31.false.2 = INTEGER: 3
```

2. **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiRdrIdString.1.31

HPI-B0101-MIB::saHpiRdrIdString.1.31.false.65536 = STRING: "Blue LED"
HPI-B0101-MIB::saHpiRdrIdString.1.31.false.65537 = STRING: "LED 1"
HPI-B0101-MIB::saHpiRdrIdString.1.31.false.65538 = STRING: "LED 2"
HPI-B0101-MIB::saHpiRdrIdString.1.31.false.69680 = STRING: "FRU Desired Power"
HPI-B0101-MIB::saHpiRdrIdString.1.31.false.70144 = STRING: "FRU Reboot and
Diagnostic Control"
HPI-B0101-MIB::saHpiRdrIdString.1.31.false.70656 = STRING: "ATCA-Fan"
HPI-B0101-MIB::saHpiRdrIdString.1.31.false.131077 = STRING: "FRU 3 HOT_SWAP"
HPI-B0101-MIB::saHpiRdrIdString.1.31.false.131084 = STRING: "Fan Tray 0"
HPI-B0101-MIB::saHpiRdrIdString.1.31.false.131196 = STRING: "Temp_In Left"
HPI-B0101-MIB::saHpiRdrIdString.1.31.false.131280 = STRING: "24V FT 0"
HPI-B0101-MIB::saHpiRdrIdString.1.31.false.131281 = STRING: "-48A bus FT 0"
HPI-B0101-MIB::saHpiRdrIdString.1.31.false.131282 = STRING: "-48A FT 0"
HPI-B0101-MIB::saHpiRdrIdString.1.31.false.131283 = STRING: "-48B bus FT 0"
HPI-B0101-MIB::saHpiRdrIdString.1.31.false.131284 = STRING: "-48B FT 0"
HPI-B0101-MIB::saHpiRdrIdString.1.31.false.131285 = STRING: "-48A FT 0 Fuse"
HPI-B0101-MIB::saHpiRdrIdString.1.31.false.131286 = STRING: "-48B FT 0 Fuse"
HPI-B0101-MIB::saHpiRdrIdString.1.31.false.196608 = STRING: "Fan Tray 0"

snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiRdrRowPointer.1.31

HPI-B0101-MIB::saHpiRdrRowPointer.1.31.false.65536 = OID: HPI-B0101-
MIB::saHpiCtrlOemNum.1.31.false.0
HPI-B0101-MIB::saHpiRdrRowPointer.1.31.false.65537 = OID: HPI-B0101-
MIB::saHpiCtrlOemNum.1.31.false.1
HPI-B0101-MIB::saHpiRdrRowPointer.1.31.false.65538 = OID: HPI-B0101-
MIB::saHpiCtrlOemNum.1.31.false.2
HPI-B0101-MIB::saHpiRdrRowPointer.1.31.false.69680 = OID: HPI-B0101-
MIB::saHpiCtrlAnalogNum.1.31.false.1
HPI-B0101-MIB::saHpiRdrRowPointer.1.31.false.70144 = OID: HPI-B0101-
MIB::saHpiCtrlAnalogNum.1.31.false.0
HPI-B0101-MIB::saHpiRdrRowPointer.1.31.false.70656 = OID: HPI-B0101-
MIB::saHpiCtrlAnalogNum.1.31.false.2
HPI-B0101-MIB::saHpiRdrRowPointer.1.31.false.131077 = OID: HPI-B0101-
MIB::saHpiSensorNum.1.31.false.5
HPI-B0101-MIB::saHpiRdrRowPointer.1.31.false.131084 = OID: HPI-B0101-
MIB::saHpiSensorNum.1.31.false.12
HPI-B0101-MIB::saHpiRdrRowPointer.1.31.false.131196 = OID: HPI-B0101-
MIB::saHpiSensorNum.1.31.false.124
HPI-B0101-MIB::saHpiRdrRowPointer.1.31.false.131280 = OID: HPI-B0101-
MIB::saHpiSensorNum.1.31.false.208
HPI-B0101-MIB::saHpiRdrRowPointer.1.31.false.131281 = OID: HPI-B0101-
MIB::saHpiSensorNum.1.31.false.209
```

```
HPI-B0101-MIB::saHpiRdrRowPointer.1.31.false.131282 = OID: HPI-B0101-
MIB::saHpiSensorNum.1.31.false.210
HPI-B0101-MIB::saHpiRdrRowPointer.1.31.false.131283 = OID: HPI-B0101-
MIB::saHpiSensorNum.1.31.false.211
HPI-B0101-MIB::saHpiRdrRowPointer.1.31.false.131284 = OID: HPI-B0101-
MIB::saHpiSensorNum.1.31.false.212
HPI-B0101-MIB::saHpiRdrRowPointer.1.31.false.131285 = OID: HPI-B0101-
MIB::saHpiSensorNum.1.31.false.213
HPI-B0101-MIB::saHpiRdrRowPointer.1.31.false.131286 = OID: HPI-B0101-
MIB::saHpiSensorNum.1.31.false.214
HPI-B0101-MIB::saHpiRdrRowPointer.1.31.false.196608 = OID: HPI-B0101-
MIB::saHpiInventoryPersistent.1.31.false.0
```

Resource `31` has three analog controls. They represent FRU desired power, FRU reboot and diagnostic control, and ATCA-fan, respectively. The first two are managed automatically. The third is manual (that is, the control can be managed by SNMP manager).

▼ To View a Column of the Control Analog Table for a Control of a Resource

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-
MIB::saHpiCtrlAnalogState.1.31.false.2

HPI-B0101-MIB::saHpiCtrlAnalogState.1.31.false.2 = INTEGER: 3
```

This command returns the state of the analog control of resource `31` with entry ID `2`.

▼ To Set the State of an Analog Control

1. Confirm that the mode is manual and that the value is in the acceptable range:

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-
MIB::saHpiCtrlAnalogDefaultMinState.1.31.2.2

HPI-B0101-MIB::saHpiCtrlAnalogDefaultMinState.1.31.false.2 = INTEGER: 0

snmpwalk -v 2c -c public ShMMIP HPI-B0101-
MIB::saHpiCtrlAnalogDefaultMaxState.1.31.2.2

HPI-B0101-MIB::saHpiCtrlAnalogDefaultMaxState.1.31.false.2 = INTEGER: 15
```

The range of acceptable values for this analog control is 0–15.

2. Set the analog control:

```
snmpset -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiCtrlAnalogState.1.31.2.2 i
11
```

This command modifies the state of the analog control of resource `31` with entry ID `2`. The command sets the state of control to `11`.

Getting Information About the IDR

The inventory data repository (IDR) information is contained in three tables:

- `saHpiInventoryTable`
- `saHpiAreaTable`
- `saHpiFieldTable`

The `saHpiInventoryTable` is the high-level table that contains infomation such as:

- Updated count of inventory
- Number of areas
- Whether or not the table is read-only

This information is stored for all of the IDRs for all of the resources.

saHpiInventoryTable Information

The index to saHpiInventoryTable is domainID.resourceID.isHistorical.InventoryID.

▼ To View the High-Level Inventory Information for All of the Resources of a Domain

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiInventoryTable.1
```

where 1 is the domain ID.

▼ To View a Column of the Inventory Table for All of the Resources of a Domain

1. **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiInventoryNumAreas.1

HPI-B0101-MIB::saHpiInventoryNumAreas.1.1.false.0 = Gauge32: 47
HPI-B0101-MIB::saHpiInventoryNumAreas.1.27.false.0 = Gauge32: 2
HPI-B0101-MIB::saHpiInventoryNumAreas.1.28.false.0 = Gauge32: 47
HPI-B0101-MIB::saHpiInventoryNumAreas.1.29.false.0 = Gauge32: 47
HPI-B0101-MIB::saHpiInventoryNumAreas.1.30.false.0 = Gauge32: 2
HPI-B0101-MIB::saHpiInventoryNumAreas.1.31.false.0 = Gauge32: 3
HPI-B0101-MIB::saHpiInventoryNumAreas.1.32.false.0 = Gauge32: 2
HPI-B0101-MIB::saHpiInventoryNumAreas.1.33.false.0 = Gauge32: 2
HPI-B0101-MIB::saHpiInventoryNumAreas.1.34.false.0 = Gauge32: 3
HPI-B0101-MIB::saHpiInventoryNumAreas.1.35.false.0 = Gauge32: 3
HPI-B0101-MIB::saHpiInventoryNumAreas.1.36.false.0 = Gauge32: 2
HPI-B0101-MIB::saHpiInventoryNumAreas.1.37.false.0 = Gauge32: 3
HPI-B0101-MIB::saHpiInventoryNumAreas.1.38.false.0 = Gauge32: 2
HPI-B0101-MIB::saHpiInventoryNumAreas.1.45.false.0 = Gauge32: 8
HPI-B0101-MIB::saHpiInventoryNumAreas.1.51.false.0 = Gauge32: 4
......
```

This command returns the number of areas for all IDRs for all of the resources. The number of areas in IDR 0 for resource 1 in domain 1 is 47. The number of areas in IDR 0 for resource 27 in domain 1 is 2.

2. **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiResourceTag.1.1

HPI-B0101-MIB::saHpiResourceTag.1.1.false = STRING: "Shelf Resource"
```

This command returns the information for resource `1`. The number of areas for IDR `0` for shelf resource is 47.

▼ To View a Column of the Inventory Table for a Resource

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiInventoryNumAreas.1.1

HPI-B0101-MIB::saHpiInventoryNumAreas.1.1.false.0 = Gauge32: 47
......
```

This command returns the number of areas for all IDRs for resource `1`.

▼ To View a Column of the Inventory Table for a Resource and IDR Entry ID

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-
MIB::saHpiInventoryNumAreas.1.1.false.0

HPI-B0101-MIB::saHpiInventoryNumAreas.1.1.false.0 = Gauge32: 47
```

This command returns the number of areas for IDR `0` for resource `1`.

`saHpiAreaTable` Information

The `saHpiAreaTable` contains information about whether or not an area is read-only and the number of fields in the area. This information is stored for all areas for all IDRs for all of the resources. The index to the table is `domainID.resourceID.isHistorical.InventoryID.AreaID`.

▼ To View All of the Information on All Areas for All of the Resources of a Domain

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiAreaTable.1
```

where `1` is domain ID.

▼ To View a Column of the Area Table for All of the Resources

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiAreaNumDataFields.1
```

This command returns the number of data fields in all areas for all of the resources.

▼ To View a Column of the Area Table for a Resource

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiAreaNumDataFields.1.1

HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.0 = Gauge32: 8
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.1 = Gauge32: 2
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.2 = Gauge32: 33
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.3 = Gauge32: 10
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.4 = Gauge32: 8
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.5 = Gauge32: 19
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.6 = Gauge32: 19
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.7 = Gauge32: 21
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.8 = Gauge32: 19
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.9 = Gauge32: 9
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.10 = Gauge32: 7
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.11 = Gauge32: 7
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.12 = Gauge32: 7
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.13 = Gauge32: 7
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.14 = Gauge32: 7
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.15 = Gauge32: 7
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.16 = Gauge32: 2
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.17 = Gauge32: 12
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.18 = Gauge32: 12
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.19 = Gauge32: 12
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.20 = Gauge32: 10
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.21 = Gauge32: 10
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.22 = Gauge32: 12
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.23 = Gauge32: 12
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.24 = Gauge32: 10
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.25 = Gauge32: 22
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.26 = Gauge32: 2
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.27 = Gauge32: 2
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.28 = Gauge32: 2
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.29 = Gauge32: 2
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.30 = Gauge32: 2
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.31 = Gauge32: 2
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.32 = Gauge32: 2
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.33 = Gauge32: 2
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.34 = Gauge32: 2
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.35 = Gauge32: 2
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.36 = Gauge32: 2
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.37 = Gauge32: 2
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.38 = Gauge32: 2
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.39 = Gauge32: 2
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.40 = Gauge32: 2
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.41 = Gauge32: 2
```

```
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.42 = Gauge32: 2
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.43 = Gauge32: 2
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.44 = Gauge32: 7
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.45 = Gauge32: 6
HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.46 = Gauge32: 5
```

This command returns the number of data fields in all areas for resource `1` (that is, the shelf resource). The output shows the number of data fields in all areas of all IDRs for resource `1`.

▼ To View a Column for an IDR of a Resource

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-
MIB::saHpiAreaNumDataFields.1.1.false.0
```

This command returns the number of fields in all areas of IDR `0` for resource `1`, where `1.1.false.0` stands for `domainID.resourceID.isHistorical.InventoryD`.

▼ To View a Column for an Area of an IDR of a Resource

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-
MIB::saHpiAreaNumDataFields.1.1.false.0.46

HPI-B0101-MIB::saHpiAreaNumDataFields.1.1.false.0.46 = Gauge32: 5
```

This command returns the number of fields in area ID `46` of IDR `0` for resource `1`, where `1.1.false.0.46` stands for `domainID.resourceID.isHistorical.InventoryID.AreaID`.

`saHpiFieldTable` Information

The `saHpiFieldTable` contains information such as:

- Field type
- Field text
- Whether or not the field is read-only

This information is stored for all fields of all areas for all IDRs for all of the resources. The index to the table is `domainID.resourceID.isHistorical.InventoryI.AreaID.FieldID`.

▼ To View All of the Information on All of the Fields

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiFieldTable.1
```

where `1` is domain ID.

▼ To View a Column of the Field Table for All of the Fields

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiFieldType.1
```

This command returns the field type for all of the fields.

▼ To View a Column for All of the Fields of a Resource

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiFieldType.1.1
```

This command returns the field type for all of the fields for resource `1`.

▼ To View a Column for All of the Fields of an Area

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiFieldType.1.1.false.0.45

HPI-B0101-MIB::saHpiFieldType.1.1.false.0.45.0 = INTEGER: mfgDatetime(2)
HPI-B0101-MIB::saHpiFieldType.1.1.false.0.45.1 = INTEGER: manufacturer(3)
HPI-B0101-MIB::saHpiFieldType.1.1.false.0.45.2 = INTEGER: productName(4)
HPI-B0101-MIB::saHpiFieldType.1.1.false.0.45.3 = INTEGER: serialNumber(6)
HPI-B0101-MIB::saHpiFieldType.1.1.false.0.45.4 = INTEGER: partNumber(7)
HPI-B0101-MIB::saHpiFieldType.1.1.false.0.45.5 = INTEGER: fileId(8)
```

This command returns the field type for all of the fields of area 45, IDR 0, and resource 1, where 1.1.false.0.45 stands for domainID.resourceID.isHistorical.InventoryID.AreaID.

▼ To View a Column for a Field

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiFieldType.1.1.false.0.45.2

HPI-B0101-MIB::saHpiFieldText.1.1.false.0.45.2 = STRING: "14-slot Dual Star
Backplane, Radial IPMB"
```

This command returns the field text for field 1 of area 45, IDR 0, and resource 1, where 1.1.false.0.45.2 stands for domainID.resourceID.isHistorical.InventoryID.AreaID.FieldID.

Using the HPI Subagent to Manage the Custom Data Record

The snmpwalk command displays the data in the custom data record (CDR).

▼ To View the Area Type for All Areas of a Specific Resource

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiAreaType.1.31

HPI-B0101-MIB::saHpiAreaType.1.31.false.0.0 = INTEGER: productInfo(180)
HPI-B0101-MIB::saHpiAreaType.1.31.false.0.1 = INTEGER: boardInfo(179)
HPI-B0101-MIB::saHpiAreaType.1.31.false.0.2 = INTEGER: oem(193)
```

This command returns the area type information for resource 31 (fan tray 0). The area with ID 2 is an OEM area.

▼ To View the Field Text for All Fields for a Specific Area of a Specific Resource

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiFieldText.1.31.2.0.2

HPI-B0101-MIB::saHpiFieldText.1.31.false.0.2.0 = Hex-STRING: D0 02
HPI-B0101-MIB::saHpiFieldText.1.31.false.0.2.1 = Hex-STRING: 33 31 33 31 33 31
00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

There are two fields in area `2`. The first field with ID `0` has value of `D0 02`. The `D0` value indicates that the area is a CDR. The second field with ID `1` has the actual contents of the CDR and has a length of 255 bytes.

▼ To Modify the CDR Contents

- **Type:**

```
snmpset -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiFieldText.1.31.2.0.2.1 x "41
31 31 32 56 45 64"

HPI-B0101-MIB::saHpiFieldText.1.31.false.0.2.1 = STRING: "A112VEd"
```

Field `1` of the CDR is being set to a hexadecimal value of `41 31 31 32 56 45 64`. This action sets the field text to the corresponding ASCII characters.

▼ To Delete a Specific CDR Field for a Specific Area of a Specific Resource

- **Type:**

```
snmpset -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiFieldStatus.1.31.2.0.2.1 i 6

HPI-B0101-MIB::saHpiFieldStatus.1.31.false.0.2.1 = INTEGER: destroy(6)
```

The `saHpiFieldStatus` column can be used to delete a field. The command takes value `6` (that is, the destroy value) for deletion of a field. Thus, the `snmpset` command deletes field `1` for area `2` of resource `31`.

▼ To Check the Number of Fields in a Specific Area for a Specific Resource

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-
MIB::saHpiAreaNumDataFields.1.31.2.0.2

HPI-B0101-MIB::saHpiAreaNumDataFields.1.31.false.0.2 = Gauge32: 1
```

The output shows that there is only one field.

Using the Event Log and Event Tables

The Shelf Manager is notified about health and management status changes in the shelf through standard IPMI event messages that are logged in the IPMI system event log, as well as forwarded to the active Shelf Manager. IPMI controllers are configured to generate event messages when they detect a significant condition getting asserted or de-asserted in the system. This includes messages for events such as:

- Temperature threshold exceeded
- Voltage threshold exceeded
- Power fault
- Watchdog expired

IPMI event messages are typically associated with a sensor defined in the SDR. The type and event type of the sensor associated with an event helps the Shelf Manager and HPI user decide on actions to be taken on account of that event.

saHpiEventTable

The saHpiEventTable presents the list of all of the events present in the HPI system. The table contains:

- Event type
- Timestamp when event was generated
- Event severity
- Pointer to a subtable that has event details

The subtable depends on the event type. For instance, if the event type is a sensor, then the subtable is the saHpiSensorEventTable, or if the event type is a hot-swap, then the subtable is the saHpiHotSwapEventTable.

▼ To View All of the Information From the Event Table

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiEventTable.1
```

▼ To View a Column From the Event Table

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiEventType.1
```

saHpiSensorEventTable

Based on the event type, event details are stored in one of several tables, such as the saHpiSensorEventTable or the saHpiHotSwapEventTable. The examples in this section pertain to the saHpiSensorEventTable, but the method to access the other event tables is the same.

The saHpiSensorEventTable contains information on the event type, the event category, and the event timestamp. The index to the table is domainID.resourceID.sensorNum.eventSeverity.eventEntryID.

▼ To View All of the Information From the Sensor Event Table

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiSensorEventTable.1
```

where `1` is the domain ID.

▼ To View a Column of the Sensor Event Table

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiSensorEventType.1

HPI-B0101-MIB::saHpiSensorEventType.1.36.4352.major.0 = INTEGER: oemSensor(193)
HPI-B0101-MIB::saHpiSensorEventType.1.36.4352.ok.1 = INTEGER: oemSensor(193)
HPI-B0101-MIB::saHpiSensorEventType.1.38.4352.major.0 = INTEGER: oemSensor(193)
HPI-B0101-MIB::saHpiSensorEventType.1.38.4352.ok.1 = INTEGER: oemSensor(193)
HPI-B0101-MIB::saHpiSensorEventType.1.44.7.critical.1 = INTEGER: voltage(3)
HPI-B0101-MIB::saHpiSensorEventType.1.44.7.critical.20 = INTEGER: voltage(3)
HPI-B0101-MIB::saHpiSensorEventType.1.44.7.major.0 = INTEGER: voltage(3)
HPI-B0101-MIB::saHpiSensorEventType.1.44.7.major.21 = INTEGER: voltage(3)
HPI-B0101-MIB::saHpiSensorEventType.1.44.8.critical.3 = INTEGER: voltage(3)
HPI-B0101-MIB::saHpiSensorEventType.1.44.8.critical.22 = INTEGER: voltage(3)
HPI-B0101-MIB::saHpiSensorEventType.1.44.8.major.2 = INTEGER: voltage(3)
HPI-B0101-MIB::saHpiSensorEventType.1.44.8.major.23 = INTEGER: voltage(3)
HPI-B0101-MIB::saHpiSensorEventType.1.44.9.critical.5 = INTEGER: voltage(3)
HPI-B0101-MIB::saHpiSensorEventType.1.44.9.critical.24 = INTEGER: voltage(3)
......
```

This command returns the event type for all of the events.

▼ To View a Column of the Sensor Event Table for a Resource

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiSensorEventType.1.44
```

This command returns the event type for all of the sensor events originating from resource `44`.

▼ To View a Column of the Sensor Event Table for a Sensor of a Resource

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiSensorEventType.1.44.7

HPI-B0101-MIB::saHpiSensorEventType.1.44.7.critical.1 = INTEGER: voltage(3)
HPI-B0101-MIB::saHpiSensorEventType.1.44.7.critical.20 = INTEGER: voltage(3)
HPI-B0101-MIB::saHpiSensorEventType.1.44.7.major.0 = INTEGER: voltage(3)
HPI-B0101-MIB::saHpiSensorEventType.1.44.7.major.21 = INTEGER: voltage(3)
......
```

This command returns the event type for all sensor events originating from resource `44`, sensor `7`.

`saHpiEventLogInfoTable`

The `saHpiEventLogInfo` contains high-level information of event log for resources with `EVENT_LOG` capability. The information includes:

- Size of event log
- Number of the current entries in event log
- Whether or not the event log has overflown

The table also has a column that can be used to clear all of the events in the log. The index to the table is `domainID.resourceID`.

▼ To View the Event Log Information for All of the Resources of a Domain

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiEventLogInfoTable.1

HPI-B0101-MIB::saHpiEventLogInfoEntries.1.27 = Gauge32: 42
HPI-B0101-MIB::saHpiEventLogInfoEntries.1.4294967295 = Gauge32: 30
HPI-B0101-MIB::saHpiEventLogInfoSize.1.27 = Gauge32: 65535
HPI-B0101-MIB::saHpiEventLogInfoSize.1.4294967295 = Gauge32: 200
HPI-B0101-MIB::saHpiEventLogInfoUserEventMaxSize.1.27 = Gauge32: 0
HPI-B0101-MIB::saHpiEventLogInfoUserEventMaxSize.1.4294967295 = Gauge32: 255
HPI-B0101-MIB::saHpiEventLogInfoUpdateTimestamp.1.27 = Hex-STRING: 10 89 9F 92
0A 1E 34 00
HPI-B0101-MIB::saHpiEventLogInfoUpdateTimestamp.1.4294967295 = Hex-STRING: 10
89 9F 92 3A 2C 8A 70
HPI-B0101-MIB::saHpiEventLogInfoTime.1.27 = Hex-STRING: 10 89 9F 94 99 C4 E2 00
HPI-B0101-MIB::saHpiEventLogInfoTime.1.4294967295 = Hex-STRING: 10 89 9F 93 80
2E A7 C8
HPI-B0101-MIB::saHpiEventLogInfoIsEnabled.1.27 = INTEGER: true(1)
HPI-B0101-MIB::saHpiEventLogInfoIsEnabled.1.4294967295 = INTEGER: true(1)
HPI-B0101-MIB::saHpiEventLogInfoOverflowFlag.1.27 = INTEGER: false(2)
HPI-B0101-MIB::saHpiEventLogInfoOverflowFlag.1.4294967295 = INTEGER: false(2)
HPI-B0101-MIB::saHpiEventLogInfoOverflowResetable.1.27 = INTEGER: false(2)
HPI-B0101-MIB::saHpiEventLogInfoOverflowResetable.1.4294967295 = INTEGER:
true(1)
HPI-B0101-MIB::saHpiEventLogInfoOverflowAction.1.27 = INTEGER: drop(1)
HPI-B0101-MIB::saHpiEventLogInfoOverflowAction.1.4294967295 = INTEGER:
overwrite(2)
HPI-B0101-MIB::saHpiEventLogInfoOverflowReset.1.27 = INTEGER: undefined(0)
HPI-B0101-MIB::saHpiEventLogInfoOverflowReset.1.4294967295 = INTEGER:
undefined(0)
HPI-B0101-MIB::saHpiEventLogClear.1.27 = INTEGER: false(2)
HPI-B0101-MIB::saHpiEventLogClear.1.4294967295 = INTEGER: false(2)
HPI-B0101-MIB::saHpiEventLogState.1.27 = INTEGER: true(1)
HPI-B0101-MIB::saHpiEventLogState.1.4294967295 = INTEGER: true(1)
```

The output shows that there are two resources with ID `27` and `4294967295` that have an event log. The `4294967295` value is reserved to indicate the domain event log (DEL). The other resource indicates a system event log (SEL).

The SEL has currently 42 entries. It can hold a maximum of 65535 entries, and in case it overflows, new events will be dropped.

▼ To View a Column of the Event Log Information Table

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiEventLogInfoEntries.1

HPI-B0101-MIB::saHpiEventLogInfoEntries.1.27 = Gauge32: 42
HPI-B0101-MIB::saHpiEventLogInfoEntries.1.4294967295 = Gauge32: 30
```

This command returns the current number of entries in the event log for all of the resources.

▼ To View a Column of the Event Log Information Table for a Resource

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiEventLogInfoEntries.1 .27

HPI-B0101-MIB::saHpiEventLogInfoEntries.1.27 = Gauge32: 42
```

This command returns the current number of entries in event log for resource `27`.

`saHpiEventLogTable`

The `saHpiEventLogTable` contains the event log records for all of the resources. The table contains the following information:

- Event type
- Timestamp when event was added
- Pointer to other event table that has event details

The pointer depends on the event type. For instance, if event type is sensor, then the pointer points to the `saHpiSensorEventLogTable`, or if event type is a hot-swap, then the pointer points to the `saHpiHotSwapEventLogTable`. The index to the table is `domainID.resourceID.EventLogIndex`.

▼ To View the Information From the `saHpiEventLogTable` for All Resources

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiEventLogTable.1
```

where `1` is domain ID.

▼ To View a Column of the `saHpiEventLogTable` for All of the Resources

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiEventLogType.1

HPI-B0101-MIB::saHpiEventLogType.1.27.0 = INTEGER: sensor(3)
HPI-B0101-MIB::saHpiEventLogType.1.27.1 = INTEGER: sensor(3)
HPI-B0101-MIB::saHpiEventLogType.1.27.2 = INTEGER: hotswap(5)
....
```

This command returns the event type of all of the events for all of the resources. For resource `27`, the first event log entry is a sensor type. The second entry is also a sensor, and the third is a hot-swap type.

▼ To View an Event Log Row Pointer for All of the Events for All of the Resources

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiEventLogRowPointer.1

HPI-B0101-MIB::saHpiEventLogRowPointer.1.27.0 = OID: HPI-B0101-
MIB::saHpiSensorEventLogTimestamp.1.38.4352.ok.0
HPI-B0101-MIB::saHpiEventLogRowPointer.1.27.1 = OID: HPI-B0101-
MIB::saHpiSensorEventLogTimestamp.1.38.5.informational.1
HPI-B0101-MIB::saHpiEventLogRowPointer.1.27.2 = OID: HPI-B0101-
MIB::saHpiHotSwapEventLogTimestamp.1.36.informational.0
......
```

The row pointer is based on the event type. The first two events are sensor events and point to an entry in the `saHpiSensorEventLog`. The third event is a hot-swap event and points to an entry in the `saHPIHotSwapLog`. These pointers can be used to access the event details.

▼ To View a Column of the Event Log Table for a Resource

● **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiEventLogType.1.27
```

This command returns the event type of all of the events logged in resource `27`.

▼ To View a Column of the Event Log Table for a Resource and an Event

● **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiEventLogType.1.27.2

HPI-B0101-MIB::saHpiEventLogType.1.27.2 = INTEGER: hotswap(5)
```

This command returns the event type of event log ID `2`, logged in resource `27`.

saHpiSensorEventLogTable

The `saHpiSensorEventLogTable` is based on the event type. Event details are stored in one of several tables, such as the `saHpiSensorEventLogTable` and the `saHpiHotSwapEventLogTable`. The examples in this section pertain to the `saHPISensorEventLogTable`, but the method to access other event log tables is same. The `saHpiSensorEventLogTable` contains information such as:

- Event type
- Event category
- Event timestamp

The index to the table is `domainID.resourceID.sensorNum.eventSeverity.eventEntryID`.

In the examples in this section, the resource ID stands for the resource that is the event source, not the resource that logs the event.

▼ To View All of the Information From the Sensor Event Log Table

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiSensorEventLogTable.1
```

where 1 is domain ID.

▼ To View a Column of the Sensor Event Log Table

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiSensorEventLogType.1

HPI-B0101-MIB::saHpiSensorEventLogType.1.27.4097.major.0 = INTEGER: voltage(3)
HPI-B0101-MIB::saHpiSensorEventLogType.1.36.2.informational.1 = INTEGER:
voltage(3)
HPI-B0101-MIB::saHpiSensorEventLogType.1.36.4.informational.2 = INTEGER:
voltage(3)
......
```

This command returns the event type for all of the events.

▼ To View a Column of the Sensor Event Log Table for a Resource

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiSensorEventLogType.1.40
```

This command returns the event type for all of the sensor events originating from resource 40.

▼ To View a Column of the Sensor Event Log Table for a Sensor of a Resource

- **Type:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiSensorEventLogType.1.40.4
```

This command returns the event type for all of the sensor events originating from resource `40`, sensor `4`.

Clearing Event Log Entries

The `saHpiEventLogInfoTable` contains a column named `saHpiEventLogClear`. This column can be set to `1` to delete all of the event log entries.

▼ To Clear the Entries of a Specific Resource From the System Event Log

1. **Type:**

```
snmpset -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiEventLogClear.1.27 i 1

HPI-B0101-MIB::saHpiEventLogClear.1.27 = INTEGER: true(1)
```

This clears the system event log of resource `27`.

2. **Confirm the deletion by checking the number of entries in event log:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiEventLogInfoEntries.1.27

HPI-B0101-MIB::saHpiEventLogInfoEntries.1.27 = Gauge32: 0
```

The output shows that the number of entries is `0`.

▼ To Clear the Event Log of a Specific Resource From the Domain Event Log

1. **Type:**

```
snmpset -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiEventLogClear.1.4294967295
i 1

HPI-B0101-MIB::saHpiEventLogClear.1.4294967295 = INTEGER: true(1)
```

This command clears the event log of resource `4294967295`.

2. **Confirm the deletion by checking the number of entries in the domain event log:**

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-
MIB::saHpiEventLogInfoEntries.1.4294967295

HPI-B0101-MIB::saHpiEventLogInfoEntries.1.4294967295 = Gauge32: 0
```

Configuring Traps and Processing Notifications

Asynchronous events of interest in the system are communicated to the SNMP managers through the use of SNMP version 1 and 2 traps.

In addition to the set of notifications supported by the `hpiSubagent` (based on OpenHPI events and defined in the HPI-B0101-MIB, as of R3), the master agent (`snmpd`) also generates generic traps such as cold start during the daemon start-up.

This section provides information related to the configuration of trap generation for the SNMP daemon, as well as an overview of the SNMP notifications supported by the Sun Netra CT900 server ShMM.

Trap Configuration

You can configure SNMP traps on the ShMM by editing the `/etc/snmpd.conf` file.

For additional information about editing the `snmpd.conf` file, also refer to the *Sun Netra CT900 Server Administration and Reference Manual*.

▼ To Configure Traps for SNMP Version 1

- **Insert the following line in the** `snmpd.conf` **file:**

```
trapsink target-host community target-port
```

▼ To Configure Traps for SNMP Version 2

- **Insert the following line in the** `snmpd.conf` **file:**

```
trap2sink target-host community target-port
```

The following example shows the syntax for both versions:

```
trapsink 129.149.2.132 public 9162
trap2sink 129.149.2.132 public 9162
```

You can use multiple `trapsink` or `trap2sink` entries to specify multiple trap destinations.

In a dual ShMM configuration, you can configure the active and backup ShMMs to generate traps based on incoming events. As of ATCA R3, the SNMP manager is expected to handle the filtering of any duplicate traps originating from both the active and the backup ShMM for a event.

Notification Processing

The following information provides descriptions of `hpiSubagent` notifications and some examples of processing `hpiSubagent` notifications.

TABLE 2-1 contains the SNMP notifications that are defined by the HPI-B0101-MIB.

TABLE 2-1 SNMP Notifications

Notification	Description
`saHpiSensorNotification`	Sensor event notification. After receiving this notification, management applications should refresh any cached information regarding the sensor indicated in the notification.
`saHpiSensorEnableChangeNotification`	Sensor enable change event notification.
`saHpiResourceNotification`	Resource failure or restoration event notifications. After receiving this notification, management applications should refresh any cached resource information.

TABLE 2-1 SNMP Notifications *(Continued)*

Notification	Description
saHpiDomainNotification	Domain events are used to announce the addition of domain references and the removal of domain references to the DRT.
saHpiWatchdogNotification	Watchdog notification.
saHpiHotSwapNotification	Hot-swap notification. After receiving this notification, management applications should refresh any cached resource information, as well as any cached information regarding sensors associated with the indicated resource.
saHpiSoftwareNotification	Audit events report a discrepancy in the audit process. Audits are typically performed by high-availability software to detect problems. Audits might look for such things as corrupted data stores, inconsistent RPT information, or improperly managed queues. Startup events report a failure to start up properly, or inconsistencies in persisted data.
saHpiOemNotification	OEM event notifications. For reference, refer to the OEM event cause of state change defined in the HPI to ATCA Mapping Specification.
saHpiUserNotification	User events can be used for storing custom events created by an HPI user when injecting events into the event log using saHpiEventLogEntryAdd().

As of version ATCA R3, the following are not supported by the hpiSubagent:

- saHpiSensorEnableChangeNotification
- saHpiDomainNotification, saHpiWatchdogNotification
- saHpiSoftwareNotification, and saHpiUserNotification

Example: Cold Start Traps

The following is an example of an SNMP version 1 cold start trap:

```
2007-04-26 14:43:02 vsp77-193 [10.4.77.193] (via UDP: [10.4.77.193]:1024) TRAP,
SNMP v1,
community public
SNMPv2-SMI::enterprises.8072.3.2.10 Cold Start Trap (0) Uptime: 0:00:00.24
```

The following is an example of an SNMP version 2 cold start trap:

```
2007-04-26 14:42:26 vsp77-193 [UDP: [10.4.77.193]:1024]:
SNMPv2-MIB::sysUpTime.0 = Timeticks: (38) 0:00:00.38 SNMPv2-MIB::snmpTrapOID.0
= OID:
SNMPv2-MIB::coldStart SNMPv2-MIB::snmpTrapEnterprise.0 = OID: SNMPv2-
SMI::enterprises.
8072.3.2.10:
```

Example: Hotswap 1

The following example illustrates the extraction of a board from the Sun Netra CT900 server, after the latch is released and the board is in the inactive state (solid blue LED).

The output shown is from the Net-SNMP command line tool, snmptrapd:

```
Apr 19 12:56:37 sunmc16 snmptrapd[19852]: [ID 702911
daemon.warning] vsp77-67.SFBay.Sun.COM [10.4.77.67]: Trap,
SAF-TC-MIB::internet.2.1.1.3.0 = Timeticks: (217825) 0:36:18.25,
SAF-TC-MIB::internet.6.3.1.1.4.1.0 = OID:
HPI-B0101-MIB::saHpiHotSwapNotification, HPI-B0101-
MIB::saHpiDomainActiveAlarms.1 = Gauge32: 35,
HPI-B0101-MIB::saHpiResourceId.1.39.false = Gauge32: 39, HPI-
B0101-MIB::saHpiEventSeverity.1.3 = INTEGER:
informational(4), HPI-B0101-
MIB::saHpiHotSwapEventState.1.39.informational.5 = INTEGER:
inactive(1),
HPI-B0101-
MIB::saHpiHotSwapEventPreviousState.1.39.informational.5 =
INTEGER: extractionPending(4)
```

The trap is processed as follows.

1. Filter important information from the trap:
   - Source IP of trap.

     In the example, it is 10.4.77.67.
   - Field 3 of trap (resourceid).

     In the example, it is 39.
   - Fields 5 and 6 of trap (current and previous hotswap states of the resource).

     In the example, the previous hotswap state is extractionPending(4) and current hotswap state is inactive(1).
2. Check that the source IP address of the trap is the IP address of the active ShMM that is being monitored.

   This ensures that the traps from the backup ShMM are not processed.
3. Check the current and previous hotswap states.
   - If either the current or the previous hotswap state is notPresent(5), management applications should refresh all sensor information from the `hpiSubagent`,because the FRU that was added or removed from the system might have associated sensors.

- If neither the current nor the previous hotswap state is notPresent(5), management applications need only refresh cached voltage sensor information.

4. Refresh cached resource information.

Example: Hotswap 2

The following example illustrates the extraction of a Sun Netra CP3020 board from the Sun Netra CT900 server, after the latch is released and the board is in the inactive state (solid blue LED).

The output shown is from the Net-SNMP command line tool, snmptrapd:

```
2008-03-06 15:37:48 shmm972-1 [UDP: [10.7.97.202]:1024]:
SAF-TC-MIB::internet.2.1.1.3.0 = Timeticks: (23293) 0:03:52.93
SAF-TC-MIB::internet.6.3.1.1.4.1.0 = OID: HPI-B0101-
MIB::saHpiHotSwapNotification HPI-B0101-
MIB::saHpiDomainActiveAlarms.0 = Gauge32: 2 HPI-B0101-
MIB::saHpiResourceId.0.37.false = Gauge32: 37 HPI-B0101-
MIB::saHpiEventSeverity.
1.3.6.1.4.1.18568.2.1.1.3.1.18.1.2.0.37.5.1 = INTEGER: ok(5) HPI-
B0101-MIB::saHpiHotSwapEventState.0.37.ok.1 =
INTEGER: notPresent(5) HPI-B0101-
MIB::saHpiHotSwapEventPreviousState.0.37.ok.1 = INTEGER:
inactive(1)
```

From this output, we see that the `saHpiHotSwapNotification` contains the following objects, with corresponding values:

- `saHpiDomainActiveAlarms.0` = `Gauge32: 2`
- `saHpiResourceId.0.37.false` = `Gauge32: 37`
- `saHpiEventSeverity.1.3.6.1.4.1.18568.2.1.1.3.1.18.1.2.0.37.5.1` = `INTEGER: ok(5)`
- `saHpiHotSwapEventState.0.37.ok.1` = `INTEGER: notPresent(5)`
- `saHpiHotSwapEventPreviousState.0.37.ok.1` = `INTEGER: inactive(1)`

These objects indicate that resource 37 has transitioned from `inactive` to `notPresent`.

Additional information can be retrieved based on the resource ID, from the `saHpiResourceTable`, however, this must be done before the board is extracted from the system.

Example: Temperature Sensor Threshold Exceeded

This example shows a notification that is generated as a result of an upper non-critical threshold being crossed on a temperature sensor.

The output shown is from the Net-SNMP command line tool, snmptrapd:

```
2008-03-06 16:23:37 shmm972-1 [UDP: [10.7.97.202]:1024]:
SAF-TC-MIB::internet.2.1.1.3.0 = Timeticks: (298337) 0:49:43.37
SAF-TC-MIB::internet.6.3.1.1.4.1.0 = OID: HPI-B0101-
MIB::saHpiSensorNotification HPI-B0101-
MIB::saHpiDomainActiveAlarms.0 = Gauge32: 2 HPI-B0101-
MIB::saHpiResourceId.0.44.false = Gauge32: 44 HPI-B0101-
MIB::saHpiEventSeverity.
1.3.6.1.4.1.18568.2.1.1.3.1.12.1.2.0.44.5.3.2 = INTEGER: minor(3)
HPI-B0101-MIB::saHpiSensorEventType.0.44.5.minor.2
= INTEGER: temperature(2) HPI-B0101-
MIB::saHpiSensorEventCategory.0.44.5.minor.2 = INTEGER:
threshold(2) HPIB0101-
MIB::saHpiSensorEventState.0.44.5.minor.2 = STRING: UPPER_MINOR
HPI-B0101-
MIB::saHpiSensorEventTriggerReadingType.0.44.5.minor.2 = INTEGER:
undefined(0) HPI-B0101-
MIB::saHpiSensorEventTriggerReading.0.44.5.minor.2 = "" HPI-
B0101-MIB::saHpiSensorEventTriggerThresholdType.
0.44.5.minor.2 = INTEGER: undefined(0) HPI-B0101-
MIB::saHpiSensorEventTriggerThreshold.0.44.5.minor.2 = ""
```

From this output, we can see that the `saHpiSensorNotification` contains the following objects:

- `saHpiDomainActiveAlarms.0 = Gauge32: 2`
- `saHpiResourceId.0.44.false = Gauge32: 44`
- `saHpiEventSeverity.1.3.6.1.4.1.18568.2.1.1.3.1.12.1.2.0.44.5.3.2 = INTEGER: minor(3)`
- `saHpiSensorEventType.0.44.5.minor.2 = INTEGER: temperature(2)`
- `saHpiSensorEventCategory.0.44.5.minor.2 = INTEGER: threshold(2)`
- `saHpiSensorEventState.0.44.5.minor.2 = STRING: UPPER_MINOR`
- `saHpiSensorEventTriggerReadingType.0.44.5.minor.2 = INTEGER: undefined(0)`
- `saHpiSensorEventTriggerReading.0.44.5.minor.2 = ""`
- `saHpiSensorEventTriggerThresholdType.0.44.5.minor.2 = INTEGER: undefined(0)`
- `saHpiSensorEventTriggerThreshold.0.44.5.minor.2 = ""`

These objects indicate that the temperature measured by sensor 5 on resource 44 has exceeded its upper-minor (upper non-critical) threshold. This event has a severity of minor.

The sensor number is embedded in the index value of the variable bindings. By definition in the MIB, the third value of the index refers to the `saHpiSensorNum`. And from the previous example, using `?saHpiSensorEventType.0.44.5.minor.2?`, we can see that the third value, the sensor number, is 5.

Additional information about the resource/sensor can be retrieved from `saHpiSensorTable` and `saHpiRdrTable` based on the resource and sensor IDs.

CHAPTER 3

Intelligent Platform Management Interface Driver

IPMI is a messaging protocol that defines how to monitor system hardware, control system components, and retrieve hardware event logs. IPMI describes how multiple embedded management controllers collaborate. The latest revision, IPMI v2.0, added standardized console access, called serial-over-LAN (SOL) re-direction, stronger security through AES encryption, and enhanced support for blade and modular systems.

You benefit by using an autonomous management subsystem in an ATCA shelf because the management subsystem is not affected by failures in the main CPU or OS. Thus, a higher level of system manageability is achieved.

In the ATCA architecture, IPMI is a key element for managing system resources. This chapter provides examples of applications that use the IPMI driver on the blade.

This chapter contains the following topics:

IPMI Overview

IPMB is the management bus in an ATCA system. Each blade has a IPMI controller to interface with the IPMB. The Sun Netra CP3*xxx* blades have an IPMI controller on board to meet the PICMG standard. The Solaris OS IPMI driver is the interface to the IPMI controller on the host or blade.

You need the IPMI driver to communicate to the local IPMI controller or other IPMI clients. For instance, with the IPMI driver, you can:

- Program the blade front panel LEDs.
- Program the watchdog timer in the IPMI controller.
- Receive a message, like a shutdown request, from other IPMI clients (typically, the Shelf Manager).

Operating System Support and IPMI Installation

The IPMI driver is supported on the following configurations:

- Solaris 10 and Solaris 10 1/06 OS on the CP3010 blade
- Solaris 10 and Solaris 10 6/06 OS on the CP3020 blade
- Solaris 10 and Solaris 10 6/06 OS on the CP3060 blade

Each platform requires two packages:

- `SUNWctipmi.u` and `SUNWctipmic` on the CP3010 blade
- `SUNWctipmi.v` and `SUNWctipmic` on the CP3020 and CP3060 blades

You can obtain these packages from the Oracle Support site:

`https://support.oracle.com`

▼ To Install the IPMI Driver

1. **Add the** SUNWctipmi.v **package:**

```
# pkgadd -d . SUNWctipmi.v
```

2. **Add the** SUNWctipmic **package:**

```
# pkgadd -d . SUNWctipmic
```

3. **Reboot the system:**

```
# reboot -- -rv
```

Note – Answer *yes* to any questions during the installation.

IPMI User Interface

For the supported features, the IPMI driver user interface is compatible with the Linux OpenIPMI driver user interface.

The IPMI device node has the following interfaces:

- /dev/ipmidev/0
- ioctl(2)
- IPMICTL_SEND_CMD
- IPMICTL_RECEIVE_MSG
- IPMICTL_RECEIVE_MSG_TRUNC
- IPMICTL_SET_GETS_EVENTS_CMD

The IPMI driver has the following poll(2) flags:

- POLLPRI
- POLLIN

The ipmi.h and ipmi_msgdef.h header files in the /usr/include/sys directory define the interfaces.

IPMI Programming Examples

This section contains two programming examples of how to use the IPMI driver. The first example shows how to get a device ID, and the second example shows how to program the LEDs.

Getting a Device ID

The following example shows how to use the IPMI driver to get a device ID.

EXAMPLE 3-1 IPMI Device ID Example

```
#include <stdio.h>
#include <strings.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <sys/ioccom.h>
#include <sys/ipmi.h>

char *devnode = "/dev/ipmidev/0";

int
main(int argc, char *argv[])

{
  int i, fd, ret = 0;
  uchar_t data[60];
  struct ipmi_reqreq;
  struct ipmi_recv recv;
  struct ipmi_system_interface_addr addr, addr1;

  /* open the ipmi device */
  if ((fd = open(devnode, O_RDWR)) < 0){
      fprintf(stderr, "Can't open ipmi device: %s\n", devnode);
      exit (1);
  };

  addr.addr_type = IPMI_SYSTEM_INTERFACE_ADDR_TYPE;
  addr.channel = 0;
  addr.lun = 0;
```

EXAMPLE 3-1 IPMI Device ID Example *(Continued)*

```
    /* send command */
    req.addr = (u_char *)&addr;
    req.addr_len = sizeof (addr);
    req.msgid = 123;
    req.msg.netfn = IPMI_NETFN_APP_REQUEST;
    req.msg.cmd = IPMI_GET_DEVICE_ID_CMD;
    req.msg.data_len = 0;
    req.msg.data = NULL;

    req.msgid++;
    ret = ioctl(fd, IPMICTL_SEND_COMMAND, (char *)&req);

    /* receive the command response */
    recv.msg.data = data;
    recv.msg.data_len = sizeof (data);
    recv.addr = (u_char *)&addr1;
    recv.addr_len = sizeof (addr1);
    ret = ioctl(fd, IPMICTL_RECEIVE_MSG_TRUNC, &recv);

    if (ret != 0) {
        perror("Error in ioctl IPMICTL_RECEIVE_MSG_TRUNC: ");
    } else {
        /*
         * Print the packet
         */
        printf("Packet:\t\trecv_type = %d; msgid = %d\n",
                recv.recv_type, recv.msgid);

        printf("Address:\t");
        printf("addr_type=0x%x", addr1.addr_type);
        printf("; channel=0x%x", (int)addr1.channel);
        printf("; lun=0x%x", (int)addr1.lun);
        printf("\n");
```

EXAMPLE 3-1 IPMI Device ID Example *(Continued)*

```
        printf("Msg:\t\t");
        printf("netfn=0x%x", recv.msg.netfn);
        printf("; cmd=0x%x", recv.msg.cmd);
        printf("; data_len=%d", recv.msg.data_len);
        printf("\n");

        printf("Data:\t\t");
        for (i = 0; i < recv.msg.data_len; i++)
            printf("%x, ", (int)recv.msg.data[i]);
        printf("\n");
    }

    close(fd);
    return(0);
}
```

Programming the LEDs

The following example shows how to use the IPMI driver to program the system's LEDs.

EXAMPLE 3-2 IPMI LED Programming Example

```
/*
 * Copyright 2007 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 *
 *ipmi LED programming examples
 *
 *
Reference:
Section 3.2.5 "Front Board Face Plate Indicators",
PICMG 3.0 R2.0 AdvancedTCA Base Specification ECN-002, Dated: May 5, 2006
set channel "0x0f"
set luno "0x00"
set msg_id "9"
set netfn "0x2c"
set cmd "0x07"
set data_cnt 6
```

EXAMPLE 3-2 IPMI LED Programming Example *(Continued)*

```
set group_id "0x00"
set byte1 "$led_id_arg"
set byte2 "$led_func_arg"
set byte3 "$on_duration_arg"
set byte4 "$lamp_color_arg"
set cmd_data "$fru_dev_id_arg $byte1 $byte2 $byte3 $byte4"
 *
 */
#pragma ident   "@(#)ipmi_led.c 1.1     07/05/09 SMI"

#include <stdio.h>
#include <strings.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <sys/ioccom.h>
#include <sys/ipmi.h>

char *devnode = "/dev/ipmidev/0";
#define_DEMO_TIME8/* 8 seconds */


void
demo1(intfd)
{
  int    ret = 0;
  uchar_t data[60];
  struct ipmi_reqreq;
  struct ipmi_system_interface_addraddr;

  printf("***LED demo1\n");
  addr.addr_type = IPMI_SYSTEM_INTERFACE_ADDR_TYPE;
  addr.channel = 0xf;
  addr.lun = 0;

  /* send command */
  req.addr = (u_char *)&addr;
  req.addr_len = sizeof (addr);
  req.msgid = 9;
  req.msg.netfn = 0x2c;
  req.msg.cmd = 7;
```

EXAMPLE 3-2 IPMI LED Programming Example *(Continued)*

```
  req.msg.data_len = 6;
  req.msg.data = data;
  data[0]= 0x0;/* group id */
  data[1]= 0x0;/* fru dev id  */
  data[2]= 0x1;/* led id */

  /* led off */
  printf("LED 1 (OOS): off\n");
  data[3]= 0x0;/* led func */
  data[4]= 0x0;/* led duration */
  data[5]= 0xf;/* led color */

  req.msgid++;
  ret = ioctl(fd, IPMICTL_SEND_COMMAND, (char *)&req);

  /* led blinks  */
  printf("LED 1 (OOS): blink every 0.5 second\n");
  data[3]= 0x32;/* led off duration */
  data[4]= 0x32;/* led on duration */
  data[5]= 0xf;/* led color */

  req.msgid++;
  ret = ioctl(fd, IPMICTL_SEND_COMMAND, (char *)&req);
  sleep(_DEMO_TIME);

  /* led back to local control  */
  printf("LED 1 (OOS): restore to local control\n");
  data[3]= 0xfc;/* led func */
  data[4]= 0x0;/* led duration */
  data[5]= 0xf;/* led color */

  req.msgid++;
  ret = ioctl(fd, IPMICTL_SEND_COMMAND, (char *)&req);
}


void
demo2_sub(intfd, int led_id, int led_func, int led_duration, int led_color)
{
  int ret = 0;
  uchar_t data[60];
  struct ipmi_reqreq;
  struct ipmi_system_interface_addr addr;
```

EXAMPLE 3-2 IPMI LED Programming Example *(Continued)*

```
  addr.addr_type = IPMI_SYSTEM_INTERFACE_ADDR_TYPE;
  addr.channel = 0xf;
  addr.lun = 0;

  req.addr = (u_char *)&addr;
  req.addr_len = sizeof (addr);
  req.msgid = 9;
  req.msg.netfn = 0x2c;
  req.msg.cmd = 7;
  req.msg.data_len = 6;
  req.msg.data = data;
  data[0]= 0x0;/* group id */
  data[1]= 0x0;/* fru dev id  */
  data[2]= led_id;/* led id */
  data[3]= led_func;/* led func */
  data[4]= led_duration;/* led duration */
  data[5]= led_color;/* led color */

  req.msgid++;

  /* send command */
  ret = ioctl(fd, IPMICTL_SEND_COMMAND, (char *)&req);
}

void
demo2(intfd)
{
  int led;

  printf("***LED demo2\n");

  for (led=0; led<3; led++){

      /* led off */
      printf("LED %d: off\n", led);
      demo2_sub(fd, led, 0, 0, 0xf);

      /* led blink with default color */
      printf("LED %d: slow blink (off=2.5s, on=1s)\n", led);
      demo2_sub(fd, led, 0xfa, 0x64, 0xf);
      sleep(_DEMO_TIME);

      /* led blink with default color */
      printf("LED %d: fast blink (off=on=0.2s)\n", led);
      demo2_sub(fd, led, 0x14, 0x14, 0xf);
      sleep(_DEMO_TIME);
```

EXAMPLE 3-2 IPMI LED Programming Example *(Continued)*

```
      /* led lamp test with default color */
      printf("LED %d: lamp test\n", led);
      demo2_sub(fd, led, 0xfb, 0xfa, 0xf);
      sleep(_DEMO_TIME);

      /* led back to local control  */
      printf("LED %d: restore to local control\n\n", led);
      demo2_sub(fd, led, 0xfc, 0x0, 0xf);
  }
}
int
main(int argc, char *argv[])
{
  int fd;

  /* open the ipmi device */
  if ((fd = open(devnode, O_RDWR)) < 0){
      fprintf(stderr, "Can't open ipmi device: %s\n", devnode);
      exit (1);
  };

  printf("Programming LED demo starting in 5 seconds\n");
  sleep(5);

  demo1(fd);
  demo2(fd);

  close(fd);
  return(0);
}
```

IPMI Commands

This section lists all IPMI/ATCA commands and Sun OEM commands supported on ATCA blades. References to applicable specifications are provided for more information.

IPMI/ATCA Commands Supported on Sun ATCA Boards

TABLE 3-1 IPMI Global Device Commands, Net Function: Application (0x06/0x07)

Command	Op Code	Platforms Supported	Interfaces Supported	Comments
`Get Device ID`	0x1	All	Payload, IPMB	
`Cold Reset`	0x2	All	Payload, IPMB.	The cold reset command resets the IPMC. The node state is retained after the reset; however, issuing this command can have adverse effects on the system. Ref: IPMI 1.5, section 17.3
`Warm Reset`	0x3	All	Payload, IPMB	The warm reset command resets the IPMC. The node state is retained after the reset; however, issuing this command can have adverse effects on the system. Ref: IPMI 1.5, section 17.3
`Get Self Test Results`	0x4	All	Payload, IPMB	For all boards, this command is supported from R3U1 onwards. In pre-R3U1 releases, this command is not supported.
`Broadcast 'Get Device ID'`	0x1	All	IPMB only	This command is used for board discovery purposes on IPMB bus only. It is not to be sent from the payload.

TABLE 3-2 BMC Watchdog Timer Commands, Net Function: Application (0x06/0x07)

Command	Op Code	Platforms Supported	Interfaces Supported	Comments
`Reset Watchdog Timer`	0x22	All	Payload, IPMB	This command starts and pats the watchdog once the watchdog parameters are set using the `Set Watchdog Timer` command. It must be used after correctly setting the watchdog parameters. Ref: IPMI 1.5, section 21.5
`Set Watchdog Timer`	0x24	All	Payload, IPMB	Timer actions 'pre-timeout interrupt' and 'power cycle' are not supported. Ref IPMI 1.5, Section 21.6
`Get Watchdog Timer`	0x25	All	Payload, IPMB	Ref IPMI 1.5, Section 21.7

TABLE 3-3 BMC Device and Messaging Commands, Net Function: Application, (0x06/0x07)

Command	Op Code	Platforms Supported	Interfaces Supported	Comments
`Send Message`	0x34	All	Payload, IPMB	Ref IPMI 1.5, section 18.7
`Master Write-Read`	0x52	All	Payload, IPMB	The user has to be aware of characteristics of the device being accessed. This command should not be issued addressing IPMI bus. Ref IPMI 1.5, section 18.10.

TABLE 3-4 Event Commands, Net Function: Sensor/Event, (0x04/0x05)

Command	Op Code	Platforms Supported	Interfaces Supported	Comments
`Set Event Receiver`	0x00	All	Payload, IPMB.	This command sets the event receiver's address and LUN. By default, the event receiver is address 0x20 (that is., ShMM). This address should not be changed, because the events will not get logged. Ref IPMI 1.5, section 23.1 On getting this command, IPMC is supposed to resend the asserted events, which it does except for the IPMC reset event, if supported, on the board. This action is performed to ensure smooth NetConsole operation.
`Get Event Receiver`	0x01	All	Payload, IPMB	Ref IPMI 1.5, section 23.2
`Platform Event`	0x02	All	Payload, IPMB	This command logs an event in SEL. If IPMC gets this command from payload, it sends it to ShMM for logging in the SEL; however, sending this command from ShMM does not make sense.

TABLE 3-5 Sensor Device Commands, Net Function: Sensor/Event, (0x04/0x05)

Command	Op Code	Platforms Supported	Interfaces Supported	Comments
`Get Device SDR Info`	0x20	All	Payload, IPMB	Ref: IPMI 1.5, section 29.2
`Get Device SDR`	0x21	All	Payload, IPMB	Ref: IPMI 1.5, section 29.3
`Reserve Device SDR Repository`	0x22	All	Payload, IPMB	Ref: IPMI 1.5, section 29.4
`Set Sensor Hysteresis`	0x24	All	Payload, IPMB	Ref: IPMI 1.5, section 29.6
`Get Sensor Hysteresis`	0x25	All	Payload, IPMB	Ref: IPMI 1.5, section 29.7

TABLE 3-5 Sensor Device Commands, Net Function: Sensor/Event, (0x04/0x05)

Command	Op Code	Platforms Supported	Interfaces Supported	Comments
`Set Sensor Threshold`	0x26	All	Payload, IPMB	Ref: IPMI 1.5, section 29.8
`Get Sensor Threshold`	0x27	All	Payload, IPMB	Ref: IPMI 1.5, section 29.9
`Set Sensor Event Enable`	0x28	All	Payload, IPMB	Ref: IPMI 1.5, section 29.10
`Get Sensor Event Enable`	0x29	All	Payload, IPMB	Ref: IPMI 1.5, section 29.11
`Get Sensor Event Status`	0x2B	All	Payload, IPMB	Ref: IPMI 1.5, section 29.13
`Get Sensor Reading`	0x2D	All	Payload, IPMB	Ref: IPMI 1.5, section 29.14

TABLE 3-6 FRU Device Commands, Net Function: Storage, (0xA/0xB)

Command	Op Code	Platforms Supported	Interfaces Supported	Comments
`Get FRU Inventory Area Info`	0x10	All	Payload, IPMB	Ref: IPMI 1.5, section 28.1
`Read FRU Data`	0x11	All	Payload, IPMB	Ref: IPMI 1.5, section 28.2
`Write FRU Data`	0x12	All	Payload, IPMB	Ref: IPMI 1.5, section 28.3

TABLE 3-7 ATCA Commands, Net Function: ATCA (0x2C/0x2D)

Command	Op Code	Platforms Supported	Interfaces Supported	Comments
`Get PICMG Properties`	0x00	All.	Payload, IPMB	Ref: PICMG 3.0R2.0ECN002, Section 3-10
`Get Address Info`	0x01	All.	Payload, IPMB	Ref: PICMG 3.0R2.0ECN002, Section 3-9
`FRU Control`	0x04	All.	payload, IPMB	Ref: PICMG 3.0R2.0ECN002, Section 3-25

TABLE 3-7 ATCA Commands, Net Function: ATCA (0x2C/0x2D) *(Continued)*

Command	Op Code	Platforms Supported	Interfaces Supported	Comments
`Get FRU LED Properties`	0x5	All.	Payload, IPMB	Ref: PICMG 3.0R2.0ECN002, Section 3-27
`Get LED Color Capabilities`	0x6	All.	Payload, IPMB	Ref: PICMG 3.0R2.0ECN002, Section 3-28
`Set FRU LED State`	0x7	All.	Payload, IPMB	Ref: PICMG 3.0R2.0ECN002, Section 3-29
`Get FRU LED State`	0x8	All.	Payload, IPMB	Ref: PICMG 3.0R2.0ECN002, Section 3-30
`Set IPMB State`	0x9	All.	IPMB	Ref: PICMG 3.0R2.0ECN002, Section 3-65
`Set FRU Activation Policy`	0xA	All.	Payload, IPMB	Ref: PICMG 3.0R2.0ECN002, Section 3-19
`Get FRU Activation Policy`	0xB	All.	Payload, IPMB	Ref: PICMG 3.0R2.0ECN002, Section 3-20
`Set FRU Activation`	0xC	All.	Payload, IPMB	Ref: PICMG 3.0R2.0ECN002, Section 3-18
`Get Device Locator Record ID`	0xD	All.	Payload, IPMB	Ref: PICMG 3.0R2.0ECN002, Section 3-35
`Set Port State`	0xE	All.	IPMB	Ref: PICMG 3.0R2.0ECN002, Section 3-54
`Get Port State`	0xF	All.	Payload, IPMB	Ref: PICMG 3.0R2.0ECN002, Section 3-55
`Compute Power Properties`	0x10	All.	IPMB	Ref: PICMG 3.0R2.0ECN002, Section 3-77
`Set Power Level`	0x11	All.	IPMB	Ref: PICMG 3.0R2.0ECN002, Section 3-79

TABLE 3-7 ATCA Commands, Net Function: ATCA (0x2C/0x2D) *(Continued)*

Command	Op Code	Platforms Supported	Interfaces Supported	Comments
`Get Power Level`	0x12	All.	Payload, IPMB	Ref: PICMG 3.0R2.0ECN002, Section 3-78
`Get IPMB Link info`	0x18	All.	Payload, IPMB	Ref: PICMG 3.0R2.0ECN002, Section 3-63
`FRU control capabilities`	0x1E	All.	Payload, IPMB	The graceful reboot option might be returned as supported in some versions of IPMC firmware, however, in the absence of support in the OS, this feature will not work. Ref: PICMG 3.0R2.0ECN002, Section 3-24

Sun and OEM IPMI Commands

TABLE 3-8 Sun OEM Commands, Net Function: OEM, (0x2E/0x2F)

Command	Op Code	Platforms Supported	Interfaces Supported	Comments
`Set AMC timeout params`	0xF1	CP3220 CP3260 CP3270 T3-1BA	Payload, IPMB.	This command can be sent from ShMM, Payload, or Debug interface to set the timeout value for AMCs to come up. IPMC does not release the payload reset until all AMCs get to M4 state or until this timeout times out. The timeout value is in seconds. IPMC stores this timeout value in persistent storage, and the value is retained across board resets.
`Get AMC timeout parameter`	0xF0	CP3220 CP3260 CP3270 T3-1BA	Payload, IPMB	This command can be sent from ShMM, Payload, or Debug port to read the default AMC timeout value.
`Set boot page`	0x81	CP3020 CP3060 CP3220 CP3250 CP3260 CP3270	Payload, IPMB	This command can be sent from ShMM, Payload, or Debug interface to set the BIOS boot page. The default value for the boot page is 0. The value set by the user is stored in SEEPROM. Upon next boot, the same value of the boot page will be used.
`Get boot page`	0x82	CP3020 CP3060 CP3220 CP3250 CP3260 CP3270	Payload, IPMB	This command can be sent from ShMM, Payload, or Debug interface to read the boot page settings for BIOS boot.
`Set front panel reset button state`	0x83	CP3010 CP3220 CP3020 CP3270	Payload, IPMB	This command can be used by software to change the way the front panel reset is handled by CPLD when this button is pressed. Default on CPLD power up is 10.
`Get front panel reset button state.`	0x84	CP3220 CP3010 CP3020 CP3270	Payload, IPMB	This command returns current settings of the front panel reset button handling. By default on CPLD power on, it comes up as 10, (i.e., pressing this button causes POR to CPU).

TABLE 3-8 Sun OEM Commands, Net Function: OEM, (0x2E/0x2F) *(Continued)*

Command	Op Code	Platforms Supported	Interfaces Supported	Comments
`Set IPMC control bits`	0xE9	CP3220 CP3260 CP3270 T3-1BA	Payload, IPMB	This command gives or takes control to or from IPMC to control various functions that might be controlled by IPMC or by external entities. Users must always perform a read, modify, and write sequence when changing any of the bits in the control byte.
`Get IPMC control bits`	0xE8	CP3220 CP3260 CP3270 T3-1BA	Payload, IPMB	This command returns current settings of IPMC control bits. Bit 0 controls the Green LED behavior.
`Set management port`	0x9B	T3-1BA	Payload, IPMB	This command routes management port access to the front or rear panel.
`Get management port`	0x9C	T3-1BA	Payload, IPMB	This command returns current settings of management port access.
`Get NIC IPMI PT firmware version`	0x87	CP3010 CP3020 CP3220	Payload, IPMB	This command returns the version string for IPMI-PT firmware running in the Broadcom NIC chip.
`Get version`	0x80	CP3270 T3-1BA	Payload, IPMB	This command returns IPMC firmware version and standby CPLD version. Although this command returns IPMC firmware version with CPLD version, the primary reason for this command is to provide CPLD version for IPMC version. In place of this command, use the IPMI get device ID command.
`Get Status`	0x00	CP3020 CP3060 CP3220 CP3250 CP3260 CP3270 T3-1BA	Payload, IPMB	This command returns the current IPMC alert status.
`Graceful Payload Reset`	0x11	CP3220 CP3250 CP3260 CP3270 T3-1BA	Payload, IPMB	This command is used to notify the carrier IPMC about completion of payload shutdown.

TABLE 3-8 Sun OEM Commands, Net Function: OEM, (0x2E/0x2F) *(Continued)*

Command	Op Code	Platforms Supported	Interfaces Supported	Comments
`Set SOL fail over link change timeouts`	0xE7	CP3270 T3-1BA	Payload, IPMB	This command sets the time for which IPMC waits to switch to second link when primary link fails, and the time it waits to switch back to primary channel if the primary channel link comes back up. Wait times are useful to filter out the link up/down bounces.
`Get SOL fail over link change timeouts`	0xE6	CP3270 T3-1BA	Payload, IPMB	This command returns current settings of IPMC control bits. Bit 0 controls the Green LED behavior, and bit 1 controls Fail LED behavior.
`Set Payload Shutdown Timeout`	0x16	CP3220 CP3250 CP3260 CP3270 T3-1BA	Payload, IPMB	This command sets the time out value for payload shutdown.
`Get Payload Shutdown Timeout`	0x15	CP3220 CP3250 CP3260 CP3270 T3-1BA	Payload, IPMB	This command returns the current value of payload shutdown timeout.
`Set Thermal Trip`	E5	T3-1BA	Payload, IPMB	This command enables or disables the thermal trip threshold which determines when to shut down a blade server.
`Get Thermal Trip`	0xE4	T3-1BA	Payload, IPMB	This command returns the value of the thermal trip.
`Set XAUI mux control`	0x95	CP3260 T3-1BA	Payload, IPMB	This command is used to route the XAUI1 and XAUI2 interfaces to either Zone 2 or Zone 3.
`Get XAUI mux control`	0x96	CP3260 T3-1BA	Payload, IPMB	This command returns the current setting for the XAUI1 and XAUI2 interface routing (either Zone 2 or Zone 3) for the board.

Tip – The following sections provide more detail about these commands.

`Set AMC timeout params`, Op Code: 0xF1, Net Function: 0x2E

This command can be sent from ShMM, Payload, or Debug interface to set the timeout value for AMCs to come up.

```
Data Bytes:
Request:
        Byte1: 00
        Byte2: 00
        Byte3: 6F or 2A (Sun legacy)
        Byte4: Delay LSB
        Byte5: Delay MSB
Response:
       Byte1: Completion Code
                00 = OK
                C1 = Command not supported
                CC = Invalid data in request
               (See IPMI spec for other completion codes)
       Byte2: 00
       Byte3: 00
       Byte4: 6F or 2A (Sun legacy)
```

`Get AMC timeout parameters`, Op Code 0xF0, Net Function: 0x2E

This command can be sent from ShMM, Payload, or Debug port to read the default AMC timeout value.

```
Data Bytes:
Request:
        Byte1: 00
        Byte2: 00
        Byte3: 6F or 2A (Sun legacy)
Response:
       Byte1: Completion Code
                00 = OK
                C1 = Command not supported
                CC = Invalid data in request
            CB = this is returned if parameter was not set earlier.
               (See IPMI spec for other completion codes)
       Byte2: 00
       Byte3: 00
       Byte4: 6F or 2A (Sun legacy)
       Byte5: Delay LSB
       Byte6: Delay MSB
```

`Set boot page`, Op Code 0x82, Net Function: 0x2E

This command can be sent from ShMM, Payload, or Debug interface to set the BIOS boot page. The default value for the boot page is 0. Bits 7 to 1 should be set to zeroes. The value set by the user is stored in SEEPROM. Upon next boot, the same value of the boot page will be used.

```
Data Bytes:
Request:
        Byte1: 00
        Byte2: 00
        Byte3: 6F or 2A (Sun legacy)
Response:
        Byte1: Completion Code
                00 = OK
                C1 = Command not supported
                CC = Invalid data in request
                CB = Parameter not set
       Byte2: 00
       Byte3: 00
       Byte4: 6F or 2A (Sun legacy)
       Byte5: Boot page value. 0 = page 0, 1 = page 1.
```

`Get boot page`, Op Code 0x81, Net Function: 0x2E

This command can be sent from ShMM, Payload, or Debug interface to read the BIOS boot page.

```
Data Bytes:
Request:
        Byte1: 00
        Byte2: 00
        Byte3: 6F or 2A (Sun legacy)
        Byte4: Boot page. 0 or 1.
Response:
        Byte1: Completion Code
                00 = OK
                C1 = Command not supported
                CC = Invalid data in request
       Byte2: 00
       Byte3: 00
       Byte4: 6F or 2A (Sun legacy)
```

`Set front panel reset button state`, Op Code 0x83, Net Function: 0x2e

This command can be used by software to change the way the front panel reset is handled by CPLD when this button is pressed. Default on CPLD power up is 10.

```
Data Bytes:
Request:
        Byte1: 00
        Byte2: 00
        Byte3: 6F or 2A (Sun legacy)
        Byte4: Front Panel Rest button settings.
               Bits 7 to 2  = 0
               Bits 1 and 0 = Front panel button state.
                      00 = Reset IPMC and hard reset to system.
                      01 = NMI to System.
                      10 = Hard reset to system.
                      11 = Front panel reset button disabled.
Response:
        Byte1: Completion Code
                00 = OK
                C1 = Command not supported
                CC = Invalid data in request
       Byte2: 00
       Byte3: 00
       Byte4: 6F or 2A (Sun legacy)
```

`Get front panel reset button`, Op Code 0x84, Net Function: 0x2E

This command returns current settings of the front panel reset button handling. By default on CPLD power on it comes up as 10, i.e., pressing this button causes Power on Reset to CPU.

```
Data Bytes:
Request:
        Byte1: 00
        Byte2: 00
        Byte3: 6F or 2A (Sun legacy)
Response:
        Byte1: Completion Code
                00 = OK
                C1 = Command not supported
                CC = Invalid data in request
       Byte2: 00
       Byte3: 00
```

```
        Byte4: 6F or 2A (Sun legacy)
        Byte5:  Front panel reset button setting.
                Bits 7 to 2   = Zeros.
                Bits 1 and 0 = Front panel button state.
                       00 = Reset IPMC and assert POR to CPU.
                       01 = XIR to CPU.
                       10 = POR to CPU.
                       11 = Front panel reset button disabled.
```

Set IPMC control bits, Op Code 0xE9, Net Function: 0x2E

This command can be used to set the configuration of the blade server's LED and the AMC shutdown behavior.

Note – Users must always perform a read, modify, and write sequence when changing any of the bits in the control byte.

```
Data Bytes:
Request:
        Byte1: 00
        Byte2: 00
        Byte3: 6F or 2A (Sun legacy)
        Byte4: Control byte.
                • Bit 0 = LED 2 (green) control bit:
                  - 1 = IPMC controls green LED.
                  - 0 = IPMC does not control green LED.
                • Bit 1 = LED 1 (amber or red OOS) ontrol bit:
                  - 1 = IPMC controls LED 1 for default behavior.
                  - 0 = IPMC does not control LED 1.
                • Bit 2 = AMC latch control bit:
                  - 1 = IPMC initiates shutdown of AMC upon latch
                       opening.
                  - 0 = IPMC does not initiate shutdown of AMC upon
                       latch opening.
                • Bits 3 to 7 = Reserved for future use. Write as is. (See Note)
Response:
        Byte1: Completion Code
                00 = OK
                C1 = Command not supported
                CC = Invalid data in request
       Byte2: 00
       Byte3: 00
       Byte4: 6F or 2A (Sun legacy)
```

Note – If an attempt is made to write 0 to any reserved bits (3 to 7), IPMC will reject the command with completion code 0xCC.

`Get IPMC control bits`, Op Code 0xE8, Net Function 0x2E.

This command returns current configuration of the blade server's LED and the AMC shutdown behavior.

```
Data Bytes:
Request:
        Byte1: 00
        Byte2: 00
        Byte3: 6F or 2A (Sun legacy)
Response:
        Byte1: Completion Code
                00 = OK
                C1 = Command not supported
                CC = Invalid data in request
       Byte2: 00
       Byte3: 00
       Byte4: 6F or 2A (Sun legacy)
       Byte5: IPMC control bits.
               •Bit 0: LED 2 (green) control bit.
               •Bit 1: LED 1 (amber or red OOS) control bit.
               •Bit 2: AMC latch control bit.
               •Bits 3 - 7: Reserved for future use.
```

`Set management port`, Op Code 0x9B, Net Function: 0x2E

This command can be used to route management port access to font or rear panel.

```
Data Bytes:
Request:
        Byte1: 00
        Byte2: 00
        Byte3: 6F
        Byte4: Control byte.
               Bits 7 to 1  = Reserved. Write zeros.
               Bits 0:
               • 1 => Route port to front (default.
               • 0 => Route port to rear (ARTM).
```

```
Response:
        Byte1: Completion Code
                00 = OK
                C1 = Command not supported
                CC = Invalid data in request
       Byte2: 00
       Byte3: 00
       Byte4: 6F
```

`Get management port`, Op Code 0x9C, Net Function 0x2E.

This command returns current settings of management port access.

```
Data Bytes:
Request:
        Byte1: 00
        Byte2: 00
        Byte3: 6F
Response:
        Byte1: Completion Code
                00 = OK
                C1 = Command not supported
                CC = Invalid data in request
       Byte2: 00
       Byte3: 00
       Byte4: 6F
       Byte5: IPMC control bit.
              Bits 7 - 1 : Reserved for future use.
              Bits 0:
                    1 => Route port to front (default.
                    0 => Route port to rear.
```

`Get NIC IPMI PT firmware version`, Op Code 0x87, Net Function: 0x2E

This command returns the IPMI PT firmware version string.

```
Data Bytes:
Request:
  Byte1: 00
  Byte2: 00
  Byte3: 6F or 2A (Sun legacy)
Response:
Byte1: Completion Code
```

```
  00 = OK
  C1 = Command not supported
  CC = Invalid data in request
  CB = Could not read NIC
Byte2: 00
Byte3: 00
Byte4: 6F or 2A (Sun legacy)
Byte5-20: The version number as ASCII string.
```

`Get version`, Op Code 0x80, Net Function: 0x2E

This command returns IPMC firmware version and standby CPLD version. Although this command returns IPMC firmware version with CPLD version, the primary reason for this command is to provide CPLD version for IPMC version. In place of this command, use the IPMI get device ID command.

```
Data Bytes:
Request:
        Byte1: 00
        Byte2: 00
        Byte3: 6F or 2A (Sun legacy)
Response:
        Byte1: Completion Code
                00 = OK
                CC = Invalid data in request
                (See IPMI spec for all completion codes.)
       Byte2: 00
       Byte3: 00
       Byte4: 6F or 2A (Sun legacy)
       Byte5: CPLD version
       Byte6: REV1 Byte of IPMC Firmware
       Byte7: REV2 Byte of IPMC Firmware
       Byte8:
                Bit 7 to Bit 1: Reserved
                Bit 8 to Bit 1: Reserved
                      1 => Test release.
                      0 => Regular release.
       Byte9: Reserved for future use.(ignore)
       ByteA: Reserved for future use.(ignore)
```

Note – IPMC version is read as low nibble of REV1, high nibble of REV2, and low nibble of REV2.

`Get Status`, Op Code 0x00, Net Function: 0x2E

This command returns the current IPMC alert status.

```
Op code: 0x00.
Net function: OEM (0x2E)
Request data:
     Byte 1: 00
     Byte 2: 40
     Byte 3: 0A
Response data:
     Byte 1 Completion code.
               OK = 0
               Command not supported = 0xC1
               Invalid data in request = 0xCC
     Byte 2: 00
     Byte 3: 40
     Byte 4: 0A
     Byte 5:
               Bit 0: 0 IPMC control over payload disabled.*
               Bits 1,2: IPMC mode.*
               Bit 3: Sensor Alert.*
               Bit 4: Reset Alert.
               Bit 5: Shutdown Alert.
               Bit 6: Diagnostic interrupt request.
               Bit 7: Graceful reboot request.
     Byte 6:
               Bits 0-3: Metallic bus 1 events.*
               Bits 4-7: Metallic bus 2 events.*
     Byte 7:
               Bits 0-3: Clock bus 1 events.*
               Bits 4-7: Clock bus 2 events.*
     Byte 8:
               Bits 0-3: Clock bus 3 events.*
               Bit 4:    Receive message queue alert.*
               Bits 5-7: Not applicable.
     Byte 9:
               Bit 0: Non-Intelligent RTM reset alert.*
               Bit 1: Non-Intelligent RTM shut down alert.*
               Bit 2: Non-Intelligent RTM diagnostic interrupt
                      alert. *
               Bit 3: Non-Intelligent RTM graceful reboot alert.*
               Bits 4-7: Not applicable.

* These options are not applicable to this specification.
```

`Graceful Payload Reset`, Op Code 0x11, Net Function: 0x2E

This command is used to notify the carrier IPMC about completion of payload shutdown. On getting this command from payload and before the shutdown timer has expired, it goes ahead with the follow up action.

```
Op code: 0x11
Net function: OEM(0x2E)
Request data:
     Byte 1: 00
     Byte 2: 40
     Byte 3: 0A
     Byte 4: FRU ID(Optional. Default is 0)
Response data:
     Byte 1: Completion code.
               00 = OK.
               C1 = Command not supported.
               CC = Invalid data in request.
     Byte 2: 00
     Byte 3: 40
     Byte 4: 0A
```

`Set Payload Shutdown Timeout`, Op Code 0x16, Net Function: 0x2E

This command sets the time out value for payload shutdown. On getting a shutdown request, IPMC sends alert to payload to get ready for power shutdown and after this time out, IPMC turns off the power. Value is retained across IPMC resets. Timeout value is in 100 ms tick, that is, a value of 0x32 (50 decimal) means 50 ticks of 100 ms which is 5 seconds.

```
Op code: 0x16
Net function: OEM(0x2E)
Request data:
     Byte 1: 00
     Byte 2: 40
     Byte 3: 0A
     Byte 4: Timeout value LS Byte.
     Byte 5: Timeout value MS Byte.
Response data:
     Byte 1: Completion code.
               00 = OK.
               0xC1 = Command not supported.
               0xCC = Invalid data in request.
```

```
      Byte 2: 00
      Byte 3: 40
      Byte 4: 0A
```

`Get Payload Shutdown Timeout`, Op Code 0x15, Net Function: 0x2E

This command shall return the current value of payload shutdown timeout. Timeout value is in 100 ms ticks, that is., a value of 0x32 (50 decimal) means 50 ticks of 100 ms which is 5 seconds.

```
Op code: 0x15.
Net function: OEM (0x2E)
Request data:
     Byte 1: 00
     Byte 2: 40
     Byte 3: 0A
Response data:
     Byte 1:Completion code.
               OK = 0
               Command not supported = 0xC1
               Invalid data in request = 0xCC
     Byte 2: 00
     Byte 3: 40
     Byte 4: 0A
     Byte 5: Payload shutdown timeout LSB.
     Byte 6: Payload shutdown timeout MSB.(
```

`Set SOL fail over link change timeouts`, Op Code 0xE7, Net Function 0x2E.

This command sets the time for which IPMC waits to switch to second serial over LAN (SOL) link when primary link fails, and the time it waits to switch back to primary channel if the primary channel link comes back up. Wait times are useful to filter out the link up/down bounces.

Wait times are in seconds. For example, a number 10 (0xA) in Byte 4 means IPMC will wait 10 seconds before switching the link to secondary channel. And a number 15(0xf) means IPMC will wait for 15 seconds before switching back to primary channel once it comes back up.

```
Data Bytes:
Request:
        Byte1: 00
        Byte2: 00
        Byte3: 6F or 2A (Sun legacy)
        Byte4: Primary Link down, fail-over wait time.
        Byte5: Primary Link up, wait time to switch to primary.
Response:
        Byte1: Completion Code
                00 = OK
                C1 = Command not supported
                CC = Invalid data in request
       Byte2: 00
       Byte3: 00
       Byte4: 6F or 2A (Sun legacy)
```

`Get SOL fail over link change timeouts`, Op Code 0xE6, Net Function 0x2E.

This command returns current settings of IPMC control bits for serial over LAN (SOL). Bit 0 controls the Green LED behavior, and bit 1 controls Fail LED behavior.

```
Data Bytes:
Request:
        Byte1: 00
        Byte2: 00
        Byte3: 6F or 2A (Sun legacy)
Response:
        Byte1: Completion Code
                00 = OK
                C1 = Command not supported
                CC = Invalid data in request
       Byte2: 00
```

```
        Byte3: 00
        Byte4: 6F or 2A (Sun legacy)
        Byte5: Primary Link down, fail-over wait time.
        Byte6: Primary Link up, wait time to switch to primary.
```

`Set Thermal Trip`, Op Code E5, Net Function: 0x2E

This command can be used to enable or disable the thermal trip. The thermal trip setting determines if a blade server shuts down because maximum temperature is reached. This feature is available only on the Netra SPARC T3-1BA blade server.

Caution – Damage to blades and systems can occur if temperature thresholds are reached and shut down does not occur. Unless the operating situation warrants overriding the default, use the default value.

In extreme situations such as operating in a war zone, there may be a requirement by the user to override the maximum temperature thresholds to prevent shutdowns of blade servers. Referred to as "war-zone mode," users can override thermal trip to keep blades, and subsequently their systems, running, even if they reach maximum temperature thresholds. Sensors will still record the threshold violation event, even when the shut down is disabled.

```
Data Bytes:
Request:
        Byte1: 00
        Byte2: 00
        Byte3: 6F or 2A (Sun legacy)
        Byte4: Control byte.
               Bits 7 to 1  = Reserved. Write zeros.
               Bits 0:
               • 1 => Enable thermal trip (default.
               • 0 => Disable thermal trip.
Response:
        Byte1: Completion Code
                00 = OK
                C1 = Command not supported
                CC = Invalid data in request
       Byte2: 00
       Byte3: 00
       Byte4: 6F or 2A (Sun legacy)
```

Get Thermal Trip, Op Code 0xE4, Net Function: 0x2E

This command returns current settings of thermal trip.

```
Data Bytes:
Request:
        Byte1: 00
        Byte2: 00
        Byte3: 6F or 2A (Sun legacy)
Response:
        Byte1: Completion Code
                00 = OK
                C1 = Command not supported
                CC = Invalid data in request
       Byte2: 00
       Byte3: 00
       Byte4: 6F or 2A (Sun legacy)
       Byte5: Current state:
              • 1 => Thermal trip enabled (default).
              • 0 => Thermal trip disabled (war-zone mode).
```

Set XAUI mux control, Op Code 0x95, Net Function: 0x2E

This command can be used to route XAUI1 and XAUI2 interfaces to either Zone 2 or Zone 3. Applicable to Sun Netra CP3260 board only.

```
Data Bytes:
Request:
        Byte1: 00
        Byte2: 00
        Byte3: 6F or 2A (Sun legacy)
        Byte4: Control byte.
             Bits 7 to 2 = Reserved for future use.Write as zeros.
               Bit 1       =  1 => Route XAUI2 to Zone 2
                              0 => Route XAUI2 to Zone 3
               Bit 0       =  1 => Route XAUI1 to Zone 2
                              0 => Route XAUI1 to Zone 3
Response:
        Byte1: Completion Code
                00 = OK
                C1 = Command not supported
                CC = Invalid data in request
                (See IPMI spec for all completion codes.)
```

```
        Byte2: 00
        Byte3: 00
        Byte4: 6F or 2A (Sun legacy)
```

Get XAUI mux control, Op Code 0x96, Net Function: 0x2E

This command returns the current XAUI1 and XAUI2 interfaces route setting, either Zone 2 or Zone 3. Applicable to Sun Netra CP3260 board only.

```
Data Bytes:
Request:
        Byte1: 00
        Byte2: 00
        Byte3: 6F or 2A (Sun legacy)
Response:
        Byte1: Completion Code
                00 = OK
                C1 = Command not supported
                CC = Invalid data in request
                (See IPMI spec for all completion codes.)
       Byte2: 00
       Byte3: 00
       Byte4: 6F or 2A (Sun legacy)
       Byte5: Control byte.
           Bits 7 to 2 = Reserved for future use.Returned as zeros.
              Bits 1         1 => Route XAUI2 to Zone 2.
                             0 => Route XAUI2 to Zone 3.
              Bits 0         1 => Route XAUI1 to Zone 2.
                             0 => Route XAUI1 to Zone 3.
```

APPENDIX **A**

Entity Paths

Entities represent the physical components in the system. Each entity has a unique identifier called an entity path. The entity path is defined by the component's location in the physical containment hierarchy of the system. An entity path consists of a series of {entity type, entity location} pairs, starting from the entity and ending at the *root* of the system hierarchy.

For example, the entity path of a blade in slot 4 of an ATCA chassis at position 3 would be:

```
{SAHPI_ENT_SBC_BLADE,1},
{SAHPI_ENT_PHYSICAL_SLOT,4},
{SAHPI_ENT_ADVANCEDTCA_CHASSIS, 3},
{SAHPI_ENT_ROOT, 0}
```

where `SAHPI_ENT_ROOT` is the entity type and `0` is the entity location.

TABLE A-1 contains an abbreviated example of the resource table for a Sun Netra CT900 server. In this example, the system contains two ShMM 500 shelf managers, two CP3140 switch blades (slots 7 and 8), one CP3010 blade (slot 3), one CP3020 blade (slot 14), and one CP3060 blade (slot 12).

TABLE A-1 Resource Table

Resource Tag	Entity Path
Shelf Resource	`{SYSTEM_CHASSIS,1}`
OEM Slot 1	`{SYSTEM_CHASSIS,1}{OEM_SYSINT_SPECIFIC,1}`
ATCA Board Slot 1	`{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,1}`
ATCA Board Slot 2	`{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,2}`
ATCA Board Slot 3	`{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,3}`
ATCA Board Slot 4	`{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,4}`
ATCA Board Slot 5	`{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,5}`

TABLE A-1 Resource Table *(Continued)*

Resource Tag	Entity Path
ATCA Board Slot 6	{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,6}
ATCA Board Slot 7	{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,7}
ATCA Board Slot 8	{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,8}
ATCA Board Slot 9	{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,9}
ATCA Board Slot 10	{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,10}
ATCA Board Slot 11	{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,11}
ATCA Board Slot 12	{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,12}
ATCA Board Slot 13	{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,13}
ATCA Board Slot 14	{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,14}
Power Entry Module Slot 1	{SYSTEM_CHASSIS,1}{POWER_ENTRY_MODULE_SLOT,1}
Power Entry Module Slot 2	{SYSTEM_CHASSIS,1}{POWER_ENTRY_MODULE_SLOT,2}
Shelf FRU Information Slot 1	{SYSTEM_CHASSIS,1}{SHELF_FRU_DEVICE_SLOT,1}
Shelf FRU Information Slot 2	{SYSTEM_CHASSIS,1}{SHELF_FRU_DEVICE_SLOT,2}
Dedicated ShMc Slot 1	{SYSTEM_CHASSIS,1}{SHELF_MANAGER_SLOT,1}
Dedicated ShMc Slot 2	{SYSTEM_CHASSIS,1}{SHELF_MANAGER_SLOT,2}
Fan Tray Slot 1	{SYSTEM_CHASSIS,1}{FAN_TRAY_SLOT,1}
Fan Tray Slot 2	{SYSTEM_CHASSIS,1}{FAN_TRAY_SLOT,2}
Fan Tray Slot 3	{SYSTEM_CHASSIS,1}{FAN_TRAY_SLOT,3}
Alarm Slot 1	{SYSTEM_CHASSIS,1}{ALARM_SLOT,1}
PPS BMC	{SYSTEM_CHASSIS,1}{SHELF_MANAGER,0}
Shelf EEPROM 1	{SYSTEM_CHASSIS,1}{SHELF_FRU_DEVICE_SLOT,1}{SHELF_FRU_DEVICE,1}
Shelf EEPROM 2	{SYSTEM_CHASSIS,1}{SHELF_FRU_DEVICE_SLOT,2}{SHELF_FRU_DEVICE,2}
SAP Board	{SYSTEM_CHASSIS,1}{ALARM_SLOT,1}{ALARM_MANAGER,1}
Fan Tray 0	{SYSTEM_CHASSIS,1}{FAN_TRAY_SLOT,1}{COOLING_UNIT,1}
Fan Tray 1	{SYSTEM_CHASSIS,1}{FAN_TRAY_SLOT,2}{COOLING_UNIT,2}
Fan Tray 2	{SYSTEM_CHASSIS,1}{FAN_TRAY_SLOT,3}{COOLING_UNIT,3}
PEM A	{SYSTEM_CHASSIS,1}{POWER_ENTRY_MODULE_SLOT,1}{POWER_SUPPLY,1}
PEM B	{SYSTEM_CHASSIS,1}{POWER_ENTRY_MODULE_SLOT,2}{POWER_SUPPLY,2}

TABLE A-1 Resource Table *(Continued)*

Resource Tag	Entity Path
CP3140H-BEG	{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,7}{PICMG_FRONT_BLADE,7}
CP3140H-BEG	{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,8}{PICMG_FRONT_BLADE,8}
ShMM-500	{SYSTEM_CHASSIS,1}{SHELF_MANAGER_SLOT,1}{SHELF_MANAGER,1}
ShMM-500	{SYSTEM_CHASSIS,1}{SHELF_MANAGER_SLOT,2}{SHELF_MANAGER,2}
NetraCP-3020	{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,14}{PICMG_FRONT_BLADE,14}
	{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,14}{PICMG_FRONT_BLADE,14}{PROCESSOR,0}
	{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,14}{PICMG_FRONT_BLADE,14}{POWER_MODULE,0}
RTM Slot	{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,14}{PICMG_FRONT_BLADE,14}{RTM_SLOT,1}
NetraCP-3010	{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,3}{PICMG_FRONT_BLADE,3}
RTM Slot	{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,3}{PICMG_FRONT_BLADE,3}{RTM_SLOT,1}
	{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,3}{PICMG_FRONT_BLADE,3}{PROCESSOR,0}
	{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,3}{PICMG_FRONT_BLADE,3}{POWER_MODULE,0}
NetraCP-3060	{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,12}{PICMG_FRONT_BLADE,12}
	{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,12}{PICMG_FRONT_BLADE,12}{PROCESSOR,0}
	{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,12}{PICMG_FRONT_BLADE,12}{POWER_MODULE,0}
RTM Slot	{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,12}{PICMG_FRONT_BLADE,12}{RTM_SLOT,1}{BACK_PANEL_BOARD,1}

APPENDIX **B**

Resource Data Records

A resource data record (RDR) defines the management instruments (sensors, controls, watchdog timers, inventory data repositories, or annunciators) associated with a resource.

This appendix contains the following RDRs:

TABLE B-1 contains the resource data records for the Sun Netra CP3010 boards.

TABLE B-1 Sun Netra CP3010 Board Resource Data Records

ID String	Type
Blue LED	`ctrlRdr(2)`
LED 1	`ctrlRdr(2)`
LED 2	`ctrlRdr(2)`
FRU Desired Power	`ctrlRdr(2)`
IPMB-A State Control	`ctrlRdr(2)`
IPMB-B State Control	`ctrlRdr(2)`
FRU Reboot and Diagnostic Control	`ctrlRdr(2)`
FRU IPM Controller Reset Control	`ctrlRdr(2)`
FRU 0 Hot Swap	`sensorRdr(3)`
System Event	`sensorRdr(3)`
RTM Presence	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 1	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 2	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 1	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 2	`sensorRdr(3)`
IPMB Physical	`sensorRdr(3)`
NetraCP-3010	`inventoryRdr(4)`
{RTM_SLOT,1}	
FRU Activation Control	`ctrlRdr(2)`
Slot State Sensor	`sensorRdr(3)`
Assigned Power Sensor	`sensorRdr(3)`
Maximum Power Capability Sensor	`sensorRdr(3)`
{PROCESSOR,0}	
BMC Watchdog	`sensorRdr(3)`
CPU1 Temp	`sensorRdr(3)`
CPU2 Temp	`sensorRdr(3)`

TABLE B-1 Sun Netra CP3010 Board Resource Data Records *(Continued)*

ID String	Type
Inlet Temp	`sensorRdr(3)`
Version change	`sensorRdr(3)`
{POWER_MODULE,0}	
+12.0V	`sensorRdr(3)`
-12.0V	`sensorRdr(3)`
+5.0V VCC	`sensorRdr(3)`
+3.3V Main	`sensorRdr(3)`
+3.3V StandBy	`sensorRdr(3)`
VBAT	`sensorRdr(3)`
VDD Core0	`sensorRdr(3)`
VDD Core1	`sensorRdr(3)`
VTT 1.25V	`sensorRdr(3)`
VDD 1.2V	`sensorRdr(3)`
VCC TM 2.5V	`sensorRdr(3)`
VDD +2.5V	`sensorRdr(3)`
VDD +1.5V	`sensorRdr(3)`

TABLE B-2 contains the resource data records for the Sun Netra CP3020 boards.

TABLE B-2 Sun Netra CP3020 Board Resource Data Records

ID String	Type
Blue LED	`ctrlRdr(2)`
LED 1	`ctrlRdr(2)`
LED 2	`ctrlRdr(2)`
FRU Desired Power	`ctrlRdr(2)`
IPMB-A State Control	`ctrlRdr(2)`
IPMB-B State Control	`ctrlRdr(2)`
FRU Reboot and Diagnostic Control	`ctrlRdr(2)`
FRU IPM Controller Reset Control	`ctrlRdr(2)`
FRU 0 HOT_SWAP	`sensorRdr(3)`
System Event	`sensorRdr(3)`
RTM Presence	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 1	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 2	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 1	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 2	`sensorRdr(3)`
IPMB Physical	`sensorRdr(3)`
NetraCP-3020	`inventoryRdr(4)`
{PROCESSOR,0}	
BMC Watchdog	`sensorRdr(3)`
CPU Tcontrol	`sensorRdr(3)`
Board Temp	`sensorRdr(3)`
ADM Internal Tem	`sensorRdr(3)`
Version change	`sensorRdr(3)`
{POWER_MODULE,0}	
+12.0V Run	`sensorRdr(3)`
-12.0V Run	`sensorRdr(3)`

TABLE B-2 Sun Netra CP3020 Board Resource Data Records *(Continued)*

ID String	Type
VCC 5V Run	`sensorRdr(3)`
+3.3V Run	`sensorRdr(3)`
+3.3V ALW	`sensorRdr(3)`
VCC RTC	`sensorRdr(3)`
VDD Core Run	`sensorRdr(3)`
VCC 1.8V Dual	`sensorRdr(3)`
DDR VTT 1.3V Run	`sensorRdr(3)`
VCC 1.2V Run	`sensorRdr(3)`
VCC 5V ALW	`sensorRdr(3)r`
VDD PU 2.5V Run	`sensorRdr(3)`
DDR VDD 2.6V Run	`sensorRdr(3)`
VCC 1.8V Run	`sensorRdr(3)`
{RTM_SLOT,1}	
FRU Activation Control	`ctrlRdr(2)`
Slot State Sensor	`sensorRdr(3)`
Assigned Power Sensor	`sensorRdr(3)`
Maximum Power Capability Sensor	`sensorRdr(3)`

TABLE B-3 contains the resource data records for the Sun Netra CP3060 boards.

TABLE B-3 Sun Netra CP3060 Board Resource Data Records

ID String	Type
Blue LED	`ctrlRdr(2)`
LED 1	`ctrlRdr(2)`
LED 2	`ctrlRdr(2)`
FRU Desired Power	`ctrlRdr(2)`
IPMB-A State Control	`ctrlRdr(2)`
IPMB-B State Control	`ctrlRdr(2)`
FRU Reboot and Diagnostic Control	`ctrlRdr(2)`
FRU IPM Controller Reset Control	`ctrlRdr(2)`
AMC Power On Sequence Commit	`ctrlRdr(2)`
AMC Power On Sequence #0	`ctrlRdr(2)`
FRU 0 Hot Swap	`sensorRdr(3)`
RTM Presence	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 1	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 2	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 1	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 2	`sensorRdr(3)`
IPMB Physical	`sensorRdr(3)`
AMC Power On Sequence Commit Status	`sensorRdr(3)`
NetraCP-3060	`inventoryRdr(4)`
{PROCESSOR,0}	
BMC Watchdog	`sensorRdr(3)`
CPU Temp1	`sensorRdr(3)`
CPU Temp2	`sensorRdr(3)`
Board Temp	`sensorRdr(3)`
Version change	`sensorRdr(3)`
{POWER_MODULE,0}	

TABLE B-3 Sun Netra CP3060 Board Resource Data Records *(Continued)*

ID String	Type
12.0V	`sensorRdr(3)`
5.0V	`sensorRdr(3)`
3.3V	`sensorRdr(3)`
3.3V STBY	`sensorRdr(3)`
2.5V STBY	`sensorRdr(3)`
1.0V	`sensorRdr(3)`
1.2V CPU	`sensorRdr(3)`
1.2V	`sensorRdr(3)`
1.5V	`sensorRdr(3)f`
0.9V VTTL	`sensorRdr(3)`
0.9V VTTR	`sensorRdr(3)`
1.8V DDR2L	`sensorRdr(3)`
1.8V DDR2R	`sensorRdr(3)`
2.5V	`sensorRdr(3)`
1.2V STBY	`sensorRdr(3)`
{RTM_SLOT,1}{BACK_PANEL_BOARD,1}	
FRU Desired Power	`ctrlRdr(2)`
FRU Reboot and Diagnostic Control	`ctrlRdr(2)`
RTM Hot Swap	`sensorRdr(3)`

TABLE B-4 contains the resource data records for the Sun Netra CP3140 switch.

TABLE B-4 Sun Netra CP3140 Switch Resource Data Records

ID String	Type
Blue LED	`ctrlRdr(2)`
LED 1	`ctrlRdr(2)`
LED 2	`ctrlRdr(2)`
FRU Desired Power	`ctrlRdr(2)`
IPMB-A State Control	`ctrlRdr(2)`
IPMB-B State Control	`ctrlRdr(2)`
FRU Reboot and Diagnostic Control	`ctrlRdr(2)`
FRU IPM Controller Reset Control	`ctrlRdr(2)`
FRU 0 HOT_SWAP	`sensorRdr(3)`
-48V ALARM	`sensorRdr(3)`
RTM Present	`sensorRdr(3)`
OOS LED	`sensorRdr(3)`
ACTIVE LED	`sensorRdr(3)`
5V	`sensorRdr(3)`
3.3V	`sensorRdr(3)`
2.5V	`sensorRdr(3)`
1.5V	`sensorRdr(3)`
1.25V	`sensorRdr(3)`
Board Temp1	`sensorRdr(3)`
Board Temp2	`sensorRdr(3)`
IPMC Firmware	`sensorRdr(3)`
BMC Watchdog	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 1 Channel 1	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 1 Channel 2	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 1	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 2	`sensorRdr(3)`

TABLE B-4 Sun Netra CP3140 Switch Resource Data Records *(Continued)*

ID String	Type
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 3	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 4	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 5	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 6	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 7	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 8	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 9	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 10	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 11	`sensorRdr(3)t`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 12	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 13	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 14	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 15	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 16	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 1	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 2	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 3	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 4	`sensorRdr(3)`

TABLE B-4 Sun Netra CP3140 Switch Resource Data Records *(Continued)*

ID String	Type
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 5	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 6	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 7	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 8	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 9	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 10	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 11	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 12	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 13	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 14	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 15	`sensorRdr(3)`
IPMB LINK	`sensorRdr(3)`
CP3140H-BEG	`inventoryRdr(4)t`

TABLE B-5 contains the resource data records for the Sun Netra CP3240 switch.

TABLE B-5 Sun Netra CP3240 Switch Resource Data Records

ID String	Type
Blue LED	`ctrlRdr(2)`
LED 1	`ctrlRdr(2)`
LED 2	`ctrlRdr(2)`
FRU Desired Power	`ctrlRdr(2)`
IPMB-A State Control	`ctrlRdr(2)`
IPMB-B State Control	`ctrlRdr(2)`
FRU Reboot and Diagnostic Control	`ctrlRdr(2)`
FRU IPM Controller Reset Control	`ctrlRdr(2)`
Hot Swap	`sensorRdr(3)`
Hot Swap AMC 0	`sensorRdr(3)`
Hot Swap AMC 1	`sensorRdr(3)`
Hot Swap AMC 2	`sensorRdr(3)`
Site 1 PWR cur	`sensorRdr(3)`
Site 1 PWR	`sensorRdr(3)`
Site 1 MP	`sensorRdr(3)`
Site 2 PWR cur	`sensorRdr(3)`
Site 2 PWR	`sensorRdr(3)`
Site 2 MP	`sensorRdr(3)`
Site 3 PWR cur	`sensorRdr(3)`
Site 3 PWR	`sensorRdr(3)`
Site 3 MP	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 1 Channel 1	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 1 Channel 2	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 1	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 2	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 3	`sensorRdr(3)`

TABLE B-5 Sun Netra CP3240 Switch Resource Data Records *(Continued)*

ID String	Type
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 4	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 5	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 6	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 7	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 8	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 9	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 10	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 11	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 12	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 13	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 14	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 15	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 16	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 1 Channel 1	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 1	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 1 Channel 2	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 2	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 1 Channel 3	`sensorRdr(3)`

TABLE B-5 Sun Netra CP3240 Switch Resource Data Records *(Continued)*

ID String	Type
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 3	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 1 Channel 4	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 4	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 1 Channel 5	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 5	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 1 Channel 6	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 6	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 1 Channel 7	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 7	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 1 Channel 8	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 8	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 1 Channel 9	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 9	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 1 Channel 10	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 10	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 1 Channel 11	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 11	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 1 Channel 12	`sensorRdr(3)`

TABLE B-5 Sun Netra CP3240 Switch Resource Data Records *(Continued)*

ID String	Type
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 12	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 1 Channel 13	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 13	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 1 Channel 14	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 14	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 1 Channel 15	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 15	`sensorRdr(3)`
IPMB Physical	`sensorRdr(3)`
CP3240H-BEX-Z	`inventoryRdr(4)`
{PROCESSOR,0}	
BMC Watchdog	`sensorRdr(3)`
Base CPU Temp	`sensorRdr(3)n`
Fabric CPU Temp	`sensorRdr(3)`
{POWER_MODULE,0}	
+12.0V	`sensorRdr(3)`
+3.3V	`sensorRdr(3)`
+2.5V	`sensorRdr(3)`
+1.25V	`sensorRdr(3)`
+1.5V	`sensorRdr(3)`
+1.8V	`sensorRdr(3)`
+1.0V	`sensorRdr(3)`
+1.2V	`sensorRdr(3)`
{BACK_PANEL_BOARD,0}	
RTM Hot Swap	`sensorRdr(3)`
RTM Presence	`sensorRdr(3)`
RTM Temp	`sensorRdr(3)`

TABLE B-5 Sun Netra CP3240 Switch Resource Data Records *(Continued)*

ID String	Type
{OPERATING_SYSTEM,0}	
Base Early	`sensorRdr(3)`
Base Full	`sensorRdr(3)`
Base Good	`sensorRdr(3)`
Fabric Early	`sensorRdr(3)`
Fabric Full	`sensorRdr(3)`
Fabric Good	`sensorRdr(3)`
{RTM_SLOT,1}{BACK_PANEL_BOARD,1}	
Blue LED	`ctrlRdr(2)`
FRU Desired Power	`ctrlRdr(2)`
FRU Reboot and Diagnostic Control	`ctrlRdr(2)`
XCP3240H-RTM-CUZ	`inventoryRdr(4)E`

TABLE B-6 contains the resource data records for the Sun Netra CP3220 boards.

TABLE B-6 Sun Netra CP3220 Board Resource Data Records

ID String	Type
Blue LED	`ctrlRdr(2)`
LED 1	`ctrlRdr(2)`
LED 2	`ctrlRdr(2)`
FRU Desired Power	`ctrlRdr(2)`
IPMB-A State Control	`ctrlRdr(2)`
IPMB-B State Control	`ctrlRdr(2)`
FRU Reboot and Diagnostic Control	`ctrlRdr(2)`
FRU IPM Controller Reset Control	`ctrlRdr(2)`
FRU 0 Hot Swap	`sensorRdr(3)`
HotSwap AMC 5	`sensorRdr(3)`
HotSwap AMC 6	`sensorRdr(3)`
Board inlet Temp	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 1	`sensorRdr(3)`
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 2	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 1	`sensorRdr(3)`
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 2	`sensorRdr(3)`
IPMB Physical	`sensorRdr(3)`
NetraCP-3220	`inventoryRdr(4)`
{PROCESSOR,0}	
BMC Watchdog	`sensorRdr(3)`
CPU Case Temp	`sensorRdr(3)`
Zone-3 Temp	`sensorRdr(3)`
AMC Area Temp	`sensorRdr(3)`
Version change	`sensorRdr(3)`
{POWER_MODULE,0}	
12.0V	`sensorRdr(3)`

TABLE B-6 Sun Netra CP3220 Board Resource Data Records *(Continued)*

ID String	Type
5.0V	`sensorRdr(3)`
3.3V	`sensorRdr(3)`
3.3V STBY	`sensorRdr(3)`
Battery Voltage	`sensorRdr(3)`
VCC 1.15V M Dual	`sensorRdr(3)`
Proc0 0.9V DDR	`sensorRdr(3)`
VCC 1.2V HT	`sensorRdr(3)`
Proc0 Core NB	`sensorRdr(3)`
VCC 1.15V M Run	`sensorRdr(3)`
VCC 1.2V Run	`sensorRdr(3)`
Proc0 1.8V DDR	`sensorRdr(3)`
VCC 1.5V Run	`sensorRdr(3)`
Proc0 Core	`sensorRdr(3)`
PM Primary Temp	`sensorRdr(3)`
PM Sec Temp	`sensorRdr(3)`
-48V A Rail	`sensorRdr(3)`
-48V B Rail	`sensorRdr(3)`
-48V Voltage	`sensorRdr(3)`
-48V Current	`sensorRdr(3)`
12V Current	`sensorRdr(3)`

TABLE B-7 contains the resource data records for the Sun Netra CP3260 boards.

TABLE B-7 Sun Netra CP3260 Board Resource Data Records

ID String	Type
Blue LED	ctrlRdr(2)
LED 1	ctrlRdr(2)
LED 2	ctrlRdr(2)
FRU Desired Power	ctrlRdr(2)
IPMB-A State Control	ctrlRdr(2)
IPMB-B State Control	ctrlRdr(2)
FRU Reboot and Diagnostic Control	ctrlRdr(2)
FRU IPM Controller Reset Control	ctrlRdr(2)
AMC Power On Sequence Commit	ctrlRdr(2)
FRU 0 Hot Swap	sensorRdr(3)
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 1	sensorRdr(3)
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 2	sensorRdr(3)
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 1 Channel 1	sensorRdr(3)
E-Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 1 Channel 2	sensorRdr(3)
IPMB Physical	sensorRdr(3)
AMC Power On Sequence Commit Status	sensorRdr(3)
Netra CP3260	inventoryRdr(4)
{PROCESSOR,0}	
BMC Watchdog	sensorRdr(3)
CPU Temp1	sensorRdr(3)
CPU Temp2	sensorRdr(3)
Board Temp	sensorRdr(3)
{POWER_MODULE,0}	
12.0V	sensorRdr(3)
5.0V	sensorRdr(3)
3.3V	sensorRdr(3)

TABLE B-7 Sun Netra CP3260 Board Resource Data Records *(Continued)*

ID String	Type
3.3V STBY	sensorRdr(3)
3.0 VBAT/STBY	sensorRdr(3)
1.0V VDD	sensorRdr(3)
1.1V CPU	sensorRdr(3)
VDD 1.1V	sensorRdr(3)
1.5V	sensorRdr(3)
VDD 1.8V	sensorRdr(3)
VDD 2.5V	sensorRdr(3)
VDD_IO 1.2V	sensorRdr(3)

TABLE B-8 contains the resource data records for the Sun Netra CP32x0 ARTM-HD.

TABLE B-8 Sun Netra CP32x0 Dual SAS Storage Advanced Rear Transition Module (ARTM-HD) Resource Data Records

ID String	Type
Blue LED	ctrlRdr(2)
LED 1	ctrlRdr(2)
LED 2	ctrlRdr(2)
Application LED	1 ctrlRdr(2)
Application LED	2 ctrlRdr(2)
FRU Desired Power	ctrlRdr(2)
FRU Reboot and Diagnostic Control	ctrlRdr(2)
ARTM HotSwap	sensorRdr(3)
ARTM 3V3STBY	sensorRdr(3)
ARTM 3V3MAIN	sensorRdr(3)
ARTM 12V	sensorRdr(3)
ARTM 5V	sensorRdr(3)
ARTM 1V2	sensorRdr(3)
ARTM TEMP-AIR	sensorRdr(3)
ARTM TEMP-LSI	sensorRdr(3)
ARTM TEMP-ADM	sensorRdr(3)
CP32X0-RTM-HDD	inventoryRdr(4)A

TABLE B-9 contains the resource data records for the Sun Netra CP3250 board.

TABLE B-9 Sun Netra CP3250 Board Resource Data Records

ID String	Type
Blue LED	ctrlRdr(2)
LED 1	ctrlRdr(2)
LED 2	ctrlRdr(2)
FRU Desired Power	ctrlRdr(2)
IPMB-A State Control	ctrlRdr(2)
IPMB-B State Control	ctrlRdr(2)
FRU Reboot and Diagnostic Control	ctrlRdr(2)
FRU IPM Controller Reset Control	ctrlRdr(2)
AMC Power On Sequence Commit	ctrlRdr(2)
AMC Power On Sequence Commit Status	sensorRdr(3)
FRU 0 Hot Swap	sensorRdr(3)
ARTM HotSwap	sensorRdr(3)
Version change	sensorRdr(3)
P48V Alarm	sensorRdr(3)
IPMB Physical	sensorRdr(3)
E-Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 1	sensorRdr(3)
Keying Link State: 0 Intrface, Link Type 1, Link Type Ext 0 Channel 2	sensorRdr(3)
Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 1	sensorRdr(3)
Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 0 Channel 2	sensorRdr(3)
Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 1 Channel 1	sensorRdr(3)
Keying Link State: 1 Intrface, Link Type 2, Link Type Ext 1 Channel 2	sensorRdr(3)
Netra CP3250	inventoryRdr(4)
{PROCESSOR,0}	
BMC Watchdog	sensorRdr(3)
CPU Temp1	sensorRdr(3)

TABLE B-9 Sun Netra CP3250 Board Resource Data Records *(Continued)*

ID String	Type
CPU Temp2	sensorRdr(3)
Board Temp	sensorRdr(3)
Sys fw progress	sensorRdr(3)
Graceful reboot	sensorRdr(3)
{POWER_MODULE,0}	
12.0V	sensorRdr(3)
5.0V	sensorRdr(3)
3.3V	sensorRdr(3)
3.3V STBY	sensorRdr(3)
3.0 VBAT/STBY	sensorRdr(3)
1.0V VDD	sensorRdr(3)
1.1V CPU	sensorRdr(3)
VDD 1.1V	sensorRdr(3)
1.5V	sensorRdr(3)
VDD 1.8V FBDIMM	sensorRdr(3)
VDD 2.5V	sensorRdr(3)
VDD_IO 1.2V	sensorRdr(3)
VDD 1.8V M0	sensorRdr(3)

APPENDIX **C**

Sun Netra CP3140 SNMP MIB Objects and Traps

This appendix contains the SNMP MIB objects and traps that are supported or unsupported on the Sun Netra CP3140 switch blade. FASTPATH 4.2 is used on the Sun Netra CP3140 switch blade. FASTPATH 4.2 supports or does not support the objects and traps described in this appendix. Each table includes the name of the object, the support status of the object, and the access control.

For more information about SNMP on the Netra CP3140 switch blade, refer to the *Sun Netra CT900 Server Switch Software Reference Manual*. You can obtain this manual at:

`http://www.sun.com/documentation/`

TABLE C-1 802.3AD Link Aggregation MIB

Object	Support	Access
`lagMIBObjects` **Group**		
`dot3adTablesLastChanged`	Yes	RO
`dot3adAggTable`		
Index: `dot3adAggIndex`		
`dot3adAggMACAddress`	Yes	RO
`dot3adAggActorSystemPriority`	Yes	RW
`dot3adAggActorSystemID`	Yes	RO
`dot3adAggAggregateOrIndividual`	Yes	RO
`dot3adAggActorAdminKey`	Yes	RW
`dot3adAggActorOperKey`	Yes	RO

TABLE C-1 802.3AD Link Aggregation MIB *(Continued)*

Object	Support	Access
dot3adAggPartnerSystemID	Yes	RO
dot3adAggPartnerSystemPriority	Yes	RO
dot3adAggPartnerOperKey	Yes	RO
dot3adAggCollectorMaxDelay	Yes	RW
dot3adAggPortListTable		
Index: dot3adAggIndex		
dot3adAggPortListPorts	Yes	RO
dot3adAggPortTable		
Index: dot3adAggPortIndex		
dot3adAggPortActorSystemPriority	Yes	RW
dot3adAggPortActorSystemID	Yes	RO
dot3adAggPortActorAdminKey	Yes	RW
dot3adAggPortActorOperKey	Yes	RW
dot3adAggPortPartnerAdminSystemPriority	Yes	RW
dot3adAggPortPartnerOperSystemPriority	Yes	RO
dot3adAggPortPartnerAdminSystemID	Yes	RW
dot3adAggPortPartnerOperSystemID	Yes	RO
dot3adAggPortPartnerAdminKey	Yes	RW
dot3adAggPortPartnerOperKey	Yes	RO
dot3adAggPortSelectedAggID	Yes	RO
dot3adAggPortAttachedAggID	Yes	RO
dot3adAggPortActorPort	Yes	RO
dot3adAggPortActorPortPriority	Yes	RW
dot3adAggPortPartnerAdminPort	Yes	RW
dot3adAggPortPartnerOperPort	Yes	RO
dot3adAggPortPartnerAdminPortPriority	Yes	RW
dot3adAggPortPartnerOperPortPriority	Yes	RO
dot3adAggPortActorAdminState	Yes	RW

TABLE C-1 802.3AD Link Aggregation MIB *(Continued)*

Object	Support	Access
dot3adAggPortActorOperState	Yes	RO
dot3adAggPortPartnerAdminState	Yes	RW
dot3adAggPortPartnerOperState	Yes	RO
dot3adAggPortAggregateOrIndividual	Yes	RO
dot3adAggPortStatsTable		
Index: dot3adAggPortIndex		
dot3adAggPortStatsLACPDUsRx	Yes	RO
dot3adAggPortStatsMarkerPDUsRx	Yes	RO
dot3adAggPortStatsMarkerResponsePDUsRx	No	N/A
dot3adAggPortStatsUnknownRx	Yes	RO
dot3adAggPortStatsIllegalRx	Yes	RO
dot3adAggPortStatsLACPDUsTx	Yes	RO
dot3adAggPortStatsMarkerPDUsTx	No	N/A
dot3adAggPortStatsMarkerResponsePDUsTx	Yes	RO
dot3adAggPortDebugTable		
Index: dot3adAggPortIndex		
dot3adAggPortDebugRxState	No	N/A
dot3adAggPortDebugLastRxTime	No	N/A
dot3adAggPortDebugMuxState	No	N/A
dot3adAggPortDebugMuxReason	No	N/A
dot3adAggPortDebugActorChurnState	No	N/A
dot3adAggPortDebugPartnerChurnState	No	N/A
dot3adAggPortDebugActorChurnCount	No	N/A
dot3adAggPortDebugPartnerChurnCount	No	N/A
dot3adAggPortDebugActorSyncTransitionCount	No	N/A
dot3adAggPortDebugPartnerSyncTransitionCount	No	N/A
dot3adAggPortDebugActorChangeCount	No	N/A
dot3adAggPortDebugPartnerChangeCount	No	N/A

TABLE C-2 RFC 2934 PIM-SM/DM MIB

Object	Support	Access
`pim`		
`pimJoinPruneInterval`	Yes	RW
`pimInterfaceTable`		
Index: `pimInterfaceIfIndex`		
`pimInterfaceAddress`	Yes	RO
`pimInterfaceNetMask`	Yes	RO
`pimInterfaceMode`	Yes	RC
`pimInterfaceDR`	Yes	RO
`pimInterfaceHelloInterval`	Yes	RC
`pimInterfaceStatus`	Yes	RC
`pimInterfaceJoinPruneInterval`	No	N/A
`pimInterfaceCBSRPreference`	Yes	RC
`pimNeighborTable`		
Index: `pimNeighborAddress`		
`pimNeighborIfIndex`	Yes	RO
`pimNeighborUpTime`	Yes	RO
`pimNeighborExpiryTime`	Yes	RO
`pimNeighborMode`	Yes	RO
`pimIpMRouteTable`		
Indicies: `ipMRouteGroup`, `ipMRouteSource`, `ipMRouteSourceMask`		
`pimIpMRouteUpstreamAssertTimer`	Yes	RO
`pimIpMRouteAssertMetric`	Yes	RO
`pimIpMRouteAssertMetricPref`	Yes	RO
`pimIpMRouteAssertRPTBit`	Yes	RO
`pimIpMRouteFlags`	Yes	RO

TABLE C-2 RFC 2934 PIM-SM/DM MIB *(Continued)*

Object	Support	Access
pimIpMRouteNextHopTable		
Indicies: ipMRouteNextHopGroup, ipMRouteNextHopSource, ipMRouteNextHopSourceMask, ipMRouteNextHopIfIndex, ipMRouteNextHopAddress		
pimIpMRouteNextHopPruneReason	Yes	RO
pimRPTable		
Indicies: pimRPGroupAddress, pimRPAddress		
pimRPState	No	N/A
pimRPStateTimer	No	N/A
pimRPLastChange	No	N/A
pimRPRowStatus	No	N/A
pimRPSetTable		
Indicies: pimRPSetComponent, pimRPSetGroupAddress, pimRPSetGroupMask, pimRPSetAddress		
pimRPSetHoldTime	Yes	RO
pimRPSetExpiryTime	Yes	RO
pimCandidateRPTable		
Indicies: pimCandidateRPGroupAddress, pimCandidateRPGroupMask		
pimCandidateRPAddress	Yes	RO
pimCandidateRPRowStatus	Yes	RO
pimComponentTable		
Index: pimComponentIndex		
pimComponentBSRAddress	Yes	RO
pimComponentBSRExpiryTime	Yes	RO
pimComponentCRPHoldTime	Yes	RO
pimComponentStatus	Yes	RO

TABLE C-3 RFC 2933 IGMP MIB

Object	Support	Access
igmpInterfaceTable		
Index: igmpInterfaceIfIndex		
igmpInterfaceQueryInterval	Yes	RC
igmpInterfaceStatus	Yes	RC
igmpInterfaceVersion	Yes	RC
igmpInterfaceQuerier	Yes	RO
igmpInterfaceQueryMaxResponseTime	Yes	RC
igmpInterfaceQuerierUpTime	Yes	RO
igmpInterfaceQuerierExpiryTime	Yes	RO
igmpInterfaceVersion1QuerierTimer	No	N/A
igmpInterfaceWrongVersionQueries	Yes	RO
igmpInterfaceJoins	Yes	RO
igmpInterfaceProxyIfIndex	No	N/A
igmpInterfaceGroups	Yes	RO
igmpInterfaceRobustness	Yes	RC
igmpInterfaceLastMembQueryIntvl	Yes	RC
igmpCacheTable		
Indicies: igmpCacheAddress, igmpCacheIfIndex		
igmpCacheSelf	No	N/A
igmpCacheLastReporter	Yes	RO
igmpCacheUpTime	Yes	RO
igmpCacheExpiryTime	Yes	RO
igmpCacheStatus	Yes	RO
igmpCacheVersion1HostTimer	Yes	RO

TABLE C-4 RFC 2932 IPv4 Multicast Routing MIB

Object	Support	Access
ipMRoute		
ipMRouteEnable	Yes	RW
ipMRouteEntryCount	Yes	RO
ipMRouteTable		
Indicies: ipMRouteGroup, ipMRouteSource, ipMRouteSourceMask		
ipMRouteUpstreamNeighbor	Yes	RO
ipMRouteInIfIndex	Yes	RO
ipMRouteUpTime	Yes	RO
ipMRouteExpiryTime	Yes	RO
ipMRoutePkts	No	N/A
ipMRouteDifferentInIfPackets	No	N/A
ipMRouteOctets	No	N/A
ipMRouteProtocol	Yes	RO
ipMRouteRtProto	No	N/A
ipMRouteRtAddress	Yes	RO
ipMRouteRtMask	Yes	RO
ipMRouteRtType	Yes	RO
ipMRouteHCOctets	No	N/A
ipMRouteNextHopTable		
Indicies: ipMRouteNextHopGroup, ipMRouteNextHopSource, ipMRouteNextHopSourceMask, ipMRouteNextHopIfIndex, ipMRouteNextHopAddress		
ipMRouteNextHopState	No	N/A
ipMRouteNextHopUpTime	No	N/A
ipMRouteNextHopExpiryTime	No	N/A
ipMRouteNextHopClosestMemberHops	No	N/A
ipMRouteNextHopProtocol	No	N/A
ipMRouteNextHopPkts	No	N/A

TABLE C-4 RFC 2932 IPv4 Multicast Routing MIB *(Continued)*

Object	Support	Access
ipMRouteInterfaceTable		
Index: ipMRouteInterfaceIfIndex		
ipMRouteInterfaceTtl	Yes	RW
ipMRouteInterfaceProtocol	Yes	RO
ipMRouteInterfaceRateLimit	No	N/A
ipMRouteInterfaceInMcastOctets	No	N/A
ipMRouteInterfaceOutMcastOctets	No	N/A
ipMRouteInterfaceHCInMcastOctets	No	N/A
ipMRouteInterfaceHCOutMcastOctets	No	N/A
ipMRouteBoundaryTable		
Indicies: ipMRouteBoundaryIfIndex, ipMRouteBoundaryAddress, ipMRouteBoundaryAddressMask		
ipMRouteBoundaryStatus	Yes	RC
ipMRouteScopeNameTable		
Indicies: ipMRouteScopeNameAddress, ipMRouteScopeNameAddressMask, ipMRouteScopeNameLanguage		
ipMRouteScopeNameString	No	N/A
ipMRouteScopeNameDefault	No	N/A
ipMRouteScopeNameStatus	No	N/A

TABLE C-5 RFC 2819 RMON MIB

Object	Support	Access
etherStatsTable		
Index: etherStatsIndex		
etherStatsDataSource	Yes	RC
etherStatsDropEvents	Yes	RO
etherStatsOctets	Yes	RO
etherStatsPkts	Yes	RO

TABLE C-5 RFC 2819 RMON MIB *(Continued)*

Object	Support	Access
etherStatsBroadcastPkts	Yes	RO
etherStatsMulticastPkts	Yes	RO
etherStatsCRCAlignErrors	Yes	RO
etherStatsUndersizePkts	Yes	RO
etherStatsOversizePkts	Yes	RO
etherStatsFragments	Yes	RO
etherStatsJabbers	Yes	RO
etherStatsCollisions	Yes	RO
etherStatsPkts64Octets	Yes	RO
etherStatsPkts65to127Octets	Yes	RO
etherStatsPkts128to255Octets	Yes	RO
etherStatsPkts256to511Octets	Yes	RO
etherStatsPkts512to1023Octets	Yes	RO
etherStatsPkts1024to1518Octets	Yes	RO
etherStatsOwner	Yes	RC
etherStatsStatus	Yes	RC
historyControlTable		
Index: historyControlIndex		
historyControlDataSource	Yes	RC
historyControlBucketsRequested	Yes	RC
historyControlBucketsGranted	Yes	RO
historyControlInterval	Yes	RC
historyControlOwner	Yes	RC
historyControlStatus	Yes	RC
etherHistoryTable		
Indices: etherHistoryIndex, etherHistorySampleIndex		
etherHistoryIntervalStart	Yes	RO
etherHistoryDropEvents	Yes	RO

TABLE C-5 RFC 2819 RMON MIB *(Continued)*

Object	Support	Access
etherHistoryOctets	Yes	RO
etherHistoryPkts	Yes	RO
etherHistoryBroadcastPkts	Yes	RO
etherHistoryMulticastPkts	Yes	RO
etherHistoryCRCAlignErrors	Yes	RO
etherHistoryUndersizePkts	Yes	RO
etherHistoryOversizePkts	Yes	RO
etherHistoryFragments	Yes	RO
etherHistoryJabbers	Yes	RO
etherHistoryCollisions	Yes	RO
etherHistoryUtilization	Yes	RO
alarmTable		
Index: alarmIndex		
alarmInterval	Yes	RC
alarmVariable	Yes	RC
alarmSampleType	Yes	RC
alarmValue	Yes	RO
alarmStartupAlarm	Yes	RC
alarmRisingThreshold	Yes	RC
alarmFallingThreshold	Yes	RC
alarmRisingEventIndex	Yes	RC
alarmFallingEventIndex	Yes	RC
alarmOwner	Yes	RC
alarmStatus	Yes	RC
hostControlTable		
Index: hostControlIndex		
hostControlDataSource	No	N/A
hostControlTableSize	No	N/A

TABLE C-5 RFC 2819 RMON MIB *(Continued)*

Object	Support	Access
hostControlLastDeleteTime	No	N/A
hostControlOwner	No	N/A
hostControlStatus	No	N/A
hostTable		
Indicies: hostIndex, hostAddress		
hostCreationOrder	No	N/A
hostInPkts	No	N/A
hostOutPkts	No	N/A
hostInOctets	No	N/A
hostOutOctets	No	N/A
hostOutErrors	No	N/A
hostOutBroadcastPkts	No	N/A
hostOutMulticastPkts	No	N/A
hostTimeTable		
Indicies: hostTimeIndex, hostTimeCreationOrder		
hostTimeAddress	No	N/A
hostTimeInPkts	No	N/A
hostTimeOutPkts	No	N/A
hostTimeInOctets	No	N/A
hostTimeOutOctets	No	N/A
hostTimeOutErrors	No	N/A
hostTimeOutBroadcastPkts	No	N/A
hostTimeOutMulticastPkts	No	N/A
hostTopNControlTable		
Index: hostTopNControlIndex		
hostTopNHostIndex	No	N/A
hostTopNRateBase	No	N/A

TABLE C-5 RFC 2819 RMON MIB *(Continued)*

Object	Support	Access
hostTopNTimeRemaining	No	N/A
hostTopNDuration	No	N/A
hostTopNRequestedSize	No	N/A
hostTopNGrantedSize	No	N/A
hostTopNStartTime	No	N/A
hostTopNOwner	No	N/A
hostTopNStatus	No	N/A
hostTopNTable		
Indicies: hostTopNReport, hostTopNIndex		
hostTopNAddress	No	N/A
hostTopNRate	No	N/A
matrixControlTable		
Index: matrixControlIndex		
matrixControlDataSource	No	N/A
matrixControlTableSize	No	N/A
matrixControlLastDeleteTime	No	N/A
matrixControlOwner	No	N/A
matrixControlStatus	No	N/A
matrixSDTable		
Indicies: matrixSDIndex, matrixSDSourceAddress, matrixSDDestAddress		
matrixSDPkts	No	N/A
matrixSDOctets	No	N/A
matrixSDErrors	No	N/A
matrixDSTable		
Indicies: matrixDSIndex, matrixDSDestAddress, matrixDSSourceAddress		

TABLE C-5 RFC 2819 RMON MIB *(Continued)*

Object	Support	Access
matrixDSPkts	No	N/A
matrixDSOctets	No	N/A
matrixDSErrors	No	N/A
filterTable		
Index: filterIndex		
filterChannelIndex	No	N/A
filterPktDataOffset	No	N/A
filterPktData	No	N/A
filterPktDataMask	No	N/A
filterPktDataNotMask	No	N/A
filterPktStatus	No	N/A
filterPktStatusMask	No	N/A
filterPktStatusNotMask	No	N/A
filterOwner	No	N/A
filterStatus	No	N/A
channelTable		
Index: channelIndex		
channelIfIndex	No	N/A
channelAcceptType	No	N/A
channelDataControl	No	N/A
channelTurnOnEventIndex	No	N/A
channelTurnOffEventIndex	No	N/A
channelEventIndex	No	N/A
channelEventStatus	No	N/A
channelMatches	No	N/A
channelDescription	No	N/A
channelOwner	No	N/A
channelStatus	No	N/A

TABLE C-5 RFC 2819 RMON MIB *(Continued)*

Object	Support	Access
`bufferControlTable`		
Index: `bufferControlIndex`		
`bufferControlChannelIndex`	No	N/A
`bufferControlFullStatus`	No	N/A
`bufferControlFullAction`	No	N/A
`bufferControlCaptureSliceSize`	No	N/A
`bufferControlDownloadSliceSize`	No	N/A
`bufferControlDownloadOffset`	No	N/A
`bufferControlMaxOctetsRequested`	No	N/A
`bufferControlMaxOctetsGranted`	No	N/A
`bufferControlCapturedPackets`	No	N/A
`bufferControlTurnOnTime`	No	N/A
`bufferControlOwner`	No	N/A
`bufferControlStatus`	No	N/A
`captureBufferTable`		
Indicies: `captureBufferControlIndex`, `captureBufferIndex`		
`captureBufferPacketID`	No	N/A
`captureBufferPacketData`	No	N/A
`captureBufferPacketLength`	No	N/A
`captureBufferPacketTime`	No	N/A
`captureBufferPacketStatus`	No	N/A
`eventTable`		
Index: `eventIndex`		
`eventDescription`	Yes	RC
`eventType`	Yes	RC
`eventCommunity`	Yes	RC
`eventLastTimeSent`	Yes	RO

TABLE C-5 RFC 2819 RMON MIB *(Continued)*

Object	Support	Access
eventOwner	Yes	RC
eventStatus	Yes	RC
logTable		
Indicies: logEventIndex, logIndex		
logTime	Yes	RO
logDescription	Yes	RO

TABLE C-6 RFC 2787 VRRP MIB

Object	Support	Access
vrrpOperationsGroup		
vrrpNodeVersion	Yes	RO
vrrpNotificationCntl	Yes	RW
vrrpOperTable		
Indices: ifIndex, vrrpOperVrId		
vrrpOperVirtualMacAddr	Yes	RO
vrrpOperState	Yes	RO
vrrpOperAdminState	Yes	RC
vrrpOperPriority	Yes	RC
vrrpOperIpAddrCount	Yes	RO
vrrpOperMasterIpAddr	Yes	RO
vrrpOperPrimaryIpAddr	Yes	RC
vrrpOperAuthType	Yes	RC
vrrpOperAuthKey	Yes	RC
vrrpOperAdvertisementInterval	Yes	RC
vrrpOperPreemptMode	Yes	RC
vrrpOperVirtualRouterUpTime	Yes	RO
vrrpOperProtocol	Yes	RC

TABLE C-6 RFC 2787 VRRP MIB *(Continued)*

Object	Support	Access
vrrpOperRowStatus	Yes	RC
vrrpAssoIpTable		
Index: vrrpAssoIpAddr		
vrrpAssoIpAddrRowStatus	No	RC
vrrpStatisticsGroup		
vrrpRouterChecksumErrors	Yes	RO
vrrpRouterVersionErrors	Yes	RO
vrrpRouterVrIdErrors	Yes	RO
vrrpRouterStatsTable		
Augment: vrrpOperTable		
vrrpStatsBecomeMaster	Yes	RO
vrrpStatsAdvertiseRcvd	Yes	RO
vrrpStatsAdvertiseIntervalErrors	Yes	RO
vrrpStatsAuthFailures	Yes	RO
vrrpStatsIpTtlErrors	Yes	RO
vrrpStatsPriorityZeroPktsRcvd	Yes	RO
vrrpStatsPriorityZeroPktsSent	Yes	RO
vrrpStatsInvalidTypePktsRcvd	Yes	RO
vrrpStatsAddressListErrors	Yes	RO
vrrpStatsInvalidAuthType	Yes	RO
vrrpStatsAuthTypeMismatch	Yes	RO
vrrpStatsPacketLengthErrors	Yes	RO

TABLE C-7 RFC 2737 ENTITY MIB (version 2)

Object	Support	Access
entPhysicalTable		
Index: entPhysicalIndex		
entPhysicalDescr	Yes	RO
entPhysicalVendorType	Yes	RO
entPhysicalContainedIn	Yes	RO
entPhysicalClass	Yes	RO
entPhysicalParentRelPos	Yes	RO
entPhysicalName	Yes	RO
entPhysicalHardwareRev	Yes	RO
entPhysicalFirmwareRev	Yes	RO
entPhysicalSoftwareRev	Yes	RO
entPhysicalSerialNum	Yes	RO
entPhysicalMfgName	Yes	RO
entPhysicalModelName	Yes	RO
entPhysicalAlias	Yes	RO
entPhysicalAssetID	Yes	RO
entPhysicalIsFRU	Yes	RO
entLogicalTable		
Index: entLogicalIndex		
entLogicalDescr	No	N/A
entLogicalType	No	N/A
entLogicalCommunity	No	N/A
entLogicalTAddress	No	N/A
entLogicalTDomain	No	N/A
entLogicalContextEngineID	No	N/A
entLogicalContextName	No	N/A

TABLE C-7 RFC 2737 ENTITY MIB (version 2) *(Continued)*

Object	Support	Access
entLPMappingTable		
Indicies: entLogicalIndex, entLPPhysicalIndex		
entLPPhysicalIndex	No	N/A
entAliasMappingTable		
Index: entPhysicalIndex, entAliasLogicalIndexOrZero		
entAliasMappingIdentifier	No	N/A
entPhysicalContainsTable		
entPhysicalChildIndex	Yes	RO
entityGeneral		
entLastChangeTime	Yes	RO
Traps		
entConfigChange	Yes	

TABLE C-8 RFC 2674 VLAN MIB (P-Bridge, Q-Bridge MIBs)

Object	Support	Access
dot1dTpHCPortTable		
Index: dot1dTpPort		
dot1dTpHCPortInFrames	Yes	RO
dot1dTpHCPortOutFrames	Yes	RO
dot1dTpHCPortInDiscards	Yes	RO
dot1dTpPortOverflowTable		
Index: dot1dTpPort		
dot1dTpPortInOverflowFrames	Yes	RO
dot1dTpPortOutOverflowFrames	Yes	RO
dot1dTpPortInOverflowDiscards	Yes	RO

TABLE C-8 RFC 2674 VLAN MIB (P-Bridge, Q-Bridge MIBs) *(Continued)*

Object	Support	Access
dot1dExtBaseGroup		
dot1dDeviceCapabilities	Yes	RO
dot1dTrafficClassesEnabled	Yes	RW
dot1dGmrpStatus	Yes	RO
dot1dPortCapabilitiesTable		
Augment: dot1dBasePort Table		
dot1dPortCapabilities	Yes	RO
dot1dPortPriorityTable		
Augment: dot1dBasePort Table		
dot1dPortDefaultUserPriority	Yes	RW
dot1dPortNumTrafficClasses	Yes	RO
dot1dUserPriorityRegenTable		
Indicies: dot1dBasePort, dot1dUserPriority		
dot1dRegenUserPriority	No	N/A
dot1dTrafficClassTable		
Indicies: dot1dBasePort, dot1dTrafficClassPriority		
dot1dTrafficClass	Yes	RW
dot1dPortOutboundAccessPriorityTable		
Index: dot1dBasePort		
dot1dPortOutboundAccessPriority	No	N/A
dot1dPortGarpTable		
Augment: dot1dBasePort Table		
dot1dPortGarpJoinTime	Yes	RW
dot1dPortGarpLeaveTime	Yes	RW
dot1dPortGarpLeaveAllTime	Yes	RW

TABLE C-8 RFC 2674 VLAN MIB (P-Bridge, Q-Bridge MIBs) *(Continued)*

Object	Support	Access
dot1dPortGmrpTable		
Augment: dot1dBasePort Table		
dot1dPortGmrpStatus	Yes	RW
dot1dPortGmrpFailedRegistrations	Yes	RO
dot1dPortGmrpLastPduOrigin	Yes	RO
dot1qGroup		
dot1qVlanVersionNumber	Yes	RO
dot1qMaxVlanId	Yes	RO
dot1qMaxSupportedVlans	Yes	RO
dot1qNumVlans	Yes	RO
dot1qGvrpStatus	Yes	RW
dot1qFdbTable		
Index: dot1qFdbId		
dot1qFdbDynamicCount	Yes	RO
dot1qTpFdbTable		
Indicies: dot1qFdbId, dot1qTpFdbAddress		
dot1qTpFdbPort	Yes	RO
dot1qTpFdbStatus	Yes	RO
dot1qTpGroupTable		
Indicies: dot1qVlanIndex, dot1qTpGroupAddress		
dot1qTpGroupEgressPorts	No	N/A
dot1qTpGroupLearnt	No	N/A
dot1qForwardAllTable		
Index: dot1qVlanIndex		
dot1qForwardAllPorts	No	N/A

TABLE C-8 RFC 2674 VLAN MIB (P-Bridge, Q-Bridge MIBs) *(Continued)*

Object	Support	Access
dot1qForwardAllStaticPorts	No	N/A
dot1qForwardAllForbiddenPorts	No	N/A
dot1qForwardUnregisteredTable		
Index: dot1qVlanIndex		
dot1qForwardUnregisteredPorts	No	N/A
dot1qForwardUnregisteredStaticPorts	No	N/A
dot1qForwardUnregisteredForbiddenPorts	No	N/A
dot1qStaticUnicastTable		
Indicies: dot1qFdbId, dot1qStaticUnicastAddress, dot1qStaticUnicastReceivePort		
dot1qStaticUnicastAllowedToGoTo	No	N/A
dot1qStaticUnicastStatus	No	N/A
dot1qStaticMulticastTable		
Indicies: dot1qVlanIndex, dot1qStaticMulticastAddress, dot1qStaticMulticastReceivePort		
dot1qStaticMulticastStaticEgressPorts	No	N/A
dot1qStaticMulticastForbiddenEgressPorts	No	N/A
dot1qStaticMulticastStatus	No	N/A
dot1qVlanGroup		
dot1qVlanNumDeletes	Yes	RO
dot1qNextFreeLocalVlanIndex	Yes	RO
dot1qConstraintSetDefault	No	N/A
dot1qConstraintTypeDefault	No	N/A
dot1qVlanCurrentTable		
Indicies: dot1qVlanTimeMark, dot1qVlanIndex		
dot1qVlanFdbId	Yes	RO

TABLE C-8 RFC 2674 VLAN MIB (P-Bridge, Q-Bridge MIBs) *(Continued)*

Object	Support	Access
`dot1qVlanCurrentEgressPorts`	Yes	RO
`dot1qVlanCurrentUntaggedPorts`	Yes	RO
`dot1qVlanStatus`	Yes	RO
`dot1qVlanCreationTime`	Yes	RO
`dot1qVlanStaticTable`		
Index: `dot1qVlanIndex`		
`dot1qVlanStaticName`	Yes	RC
`dot1qVlanStaticEgressPorts`	Yes	RC
`dot1qVlanForbiddenEgressPorts`	Yes	RC
`dot1qVlanStaticUntaggedPorts`	Yes	RC
`dot1qVlanStaticRowStatus`	Yes	RC
`dot1qPortVlanTable`		
Augment: `dot1dBasePortEntry`		
`dot1qPvid`	Yes	RW
`dot1qPortAcceptableFrameTypes`	Yes	RW
`dot1qPortIngressFiltering`	Yes	RW
`dot1qPortGvrpStatus`	Yes	RW
`dot1qPortGvrpFailedRegistrations`	Yes	RO
`dot1qPortGvrpLastPduOrigin`	Yes	RO
`dot1qPortVlanStatisticsTable`		
Indicies: `dot1dBasePort`, `dot1qVlanIndex`		
`dot1qTpVlanPortInFrames`	No	N/A
`dot1qTpVlanPortOutFrames`	No	N/A
`dot1qTpVlanPortInDiscards`	No	N/A
`dot1qTpVlanPortInOverflowFrames`	No	N/A
`dot1qTpVlanPortOutOverflowFrames`	No	N/A
`dot1qTpVlanPortInOverflowDiscards`	No	N/A

TABLE C-8 RFC 2674 VLAN MIB (P-Bridge, Q-Bridge MIBs) *(Continued)*

Object	Support	Access
`dot1qPortVlanHCStatisticsTable`		
Indicies: `dot1dBasePort`, `dot1qVlanIndex`		
`dot1qTpVlanPortHCInFrames`	No	N/A
`dot1qTpVlanPortHCOutFrames`	No	N/A
`dot1qTpVlanPortHCInDiscards`	No	N/A
`dot1qLearningConstraintsTable`		
Indicies: `dot1qConstraintVlan`, `dot1qConstraintSet`		
`dot1qConstraintType`	No	N/A
`dot1qConstraintStatus`	No	N/A

TABLE C-9 RFC 2620 Radius Accounting Client MIB

Object	Support	Access
`radiusAccClient` **Group**		
`radiusAccClientInvalidServerAddresses`	Yes	RO
`radiusAccClientIdentifier`	Yes	RO
`radiusAccServerTable`		
Index: `radiusAccServerIndex`		
`radiusAccServerAddress`	Yes	RO
`radiusAccClientServerPortNumber`	Yes	RO
`radiusAccClientRoundTripTime`	Yes	RO
`radiusAccClientRequests`	Yes	RO
`radiusAccClientRetransmissions`	Yes	RO
`radiusAccClientResponses`	Yes	RO
`radiusAccClientMalformedResponses`	Yes	RO
`radiusAccClientBadAuthenticators`	Yes	RO
`radiusAccClientPendingRequests`	Yes	RO

TABLE C-9 RFC 2620 Radius Accounting Client MIB *(Continued)*

Object	Support	Access
radiusAccClientTimeouts	Yes	RO
radiusAccClientUnknownTypes	Yes	RO
radiusAccClientPacketsDropped	Yes	RO

TABLE C-10 RFC 2618 Radius Authentication Client MIB

Object	Support	Access
radiusAuthClient **Group**		
radiusAuthClientInvalidServerAddresses	Yes	RO
radiusAuthClientIdentifier	Yes	RO
radiusAuthServerTable		
Index: radiusAuthServerIndex		
radiusAuthServerAddress	Yes	RO
radiusAuthClientServerPortNumber	Yes	RO
radiusAuthClientRoundTripTime	Yes	RO
radiusAuthClientAccessRequests	Yes	RO
radiusAuthClientAccessRetransmissions	Yes	RO
radiusAuthClientAccessAccepts	Yes	RO
radiusAuthClientAccessRejects	Yes	RO
radiusAuthClientAccessChallenges	Yes	RO
radiusAuthClientMalformedAccessResponses	Yes	RO
radiusAuthClientBadAuthenticators	Yes	RO
radiusAuthClientPendingRequests	Yes	RO
radiusAuthClientTimeouts	Yes	RO
radiusAuthClientUnknownTypes	Yes	RO
radiusAuthClientPacketsDropped	Yes	RO

TABLE C-11 RFC 2233 Interfaces MIB

Object	Support	Access
interfaces		
ifNumber	No	N/A
ifMIBObjects		
ifTableLastChange	No	N/A
ifStackLastChange	No	N/A
ifTable		
Index: ifIndex		
ifDescr	Yes	RO
ifType	Yes	RO
ifMtu	Yes	RO
ifSpeed	Yes	RO
ifPhysAddress	Yes	RO
ifAdminStatus	Yes	RW
ifOperStatus	Yes	RO
ifLastChange	Yes	RO
ifInOctets	Yes	RO
ifInUcastPkts	Yes	RO
ifInNUcastPkts	Yes	RO
ifInDiscards	Yes	RO
ifInErrors	Yes	RO
ifInUnknownProtos	Yes	RO
ifOutOctets	Yes	RO
ifOutUcastPkts	Yes	RO
ifOutNUcastPkts	Yes	RO
ifOutDiscards	Yes	RO
ifOutErrors	Yes	RO
ifOutQLen	No	N/A
ifSpecific	No	N/A

TABLE C-11 RFC 2233 Interfaces MIB *(Continued)*

Object	Support	Access
ifXTable		
Index: ifIndex		
ifName	Yes	RO
ifInMulticastPkts	Yes	RO
ifInBroadcastPkts	Yes	RO
ifOutMulticastPkts	Yes	RO
ifOutBroadcastPkts	Yes	RO
ifHCInOctets	Yes	RO
ifHCInUcastPkts	Yes	RO
ifHCInMulticastPkts	Yes	RO
ifHCInBroadcastPkts	Yes	RO
ifHCOutOctets	Yes	RO
ifHCOutUcastPkts	Yes	RO
ifHCOutMulticastPkts	Yes	RO
ifHCOutBroadcastPkts	Yes	RO
ifLinkUpDownTrapEnable	Yes	RW
ifHighSpeed	Yes	RO
ifPromiscuousMode	Yes	RW
ifConnectorPresent	Yes	RO
ifAlias	No	N/A
ifCounterDiscontinuityTime	Yes	RO
ifStackTable		
Indicies: ifStackHigherLayer, ifStackLowerLayer		
ifStackStatus	No	N/A
ifRcvAddressTable		
Indicies: ifIndex, ifRcvAddressAddress		
ifRcvAddressStatus	No	N/A
ifRcvAddressType	No	N/A

TABLE C-11 RFC 2233 Interfaces MIB *(Continued)*

Object	Support	Access
ifTestTable		
Index: ifTestId		
ifTestStatus	No	N/A
ifTestType	No	N/A
ifTestResult	No	N/A
ifTestCode	No	N/A
ifTestOwner	No	N/A

TABLE C-12 RFC 1850 OSPF MIB

Object	Support	Access
ospfGeneralGroup		
ospfRouterId	Yes	RW
ospfAdminStat	Yes	RW
ospfVersionNumber	Yes	RO
ospfAreaBdrRtrStatus	Yes	RO
ospfASBdrRtrStatus	Yes	RW
ospfExternLsaCount	Yes	RO
ospfExternLsaCksumSum	Yes	RO
ospfTOSSupport	Yes	RW
ospfOriginateNewLsas	Yes	RO
ospfRxNewLsas	Yes	RO
ospfExtLsdbLimit	Yes	RW
ospfMulticastExtensions	Yes	RO
ospfExitOverflowInterval	Yes	RW
ospfDemandExtensions	Yes	RO
ospfAreaTable		
Index: ospfAreaId		
ospfAuthType	No	N/A

TABLE C-12 RFC 1850 OSPF MIB *(Continued)*

Object	Support	Access
ospfImportAsExtern	Yes	RC
ospfSpfRuns	Yes	RO
ospfAreaBdrRtrCount	Yes	RO
ospfAsBdrRtrCount	Yes	RO
ospfAreaLsaCount	Yes	RO
ospfAreaLsaCksumSum	Yes	RO
ospfAreaSummary	Yes	RC
ospfAreaStatus	Yes	RO
ospfStubAreaTable		
Indicies: ospfStubAreaId, ospfStubTOS		
ospfStubMetric	Yes	RC
ospfStubStatus	Yes	RC
ospfStubMetricType	Yes	RC
ospfLsdbTable		
Indicies: ospfLsdbAreaId, ospfLsdbType, ospfLsdbLsid, ospfLsdbRouterId		
ospfLsdbSequence	Yes	RO
ospfLsdbAge	Yes	RO
ospfLsdbChecksum	Yes	RO
ospfLsdbAdvertisement	Yes	RO
ospfAreaRangeTable		
Indicies: ospfAreaRangeAreaId, ospfAreaRangeNet		
ospfAreaRangeMask	Obsolete	
ospfAreaRangeStatus	Obsolete	
ospfAreaRangeEffect	Obsolete	

TABLE C-12 RFC 1850 OSPF MIB *(Continued)*

Object	Support	Access
ospfHostTable		
Indicies: ospfHostIpAddress, ospfHostTOS		
ospfHostMetric	No	N/A
ospfHostStatus	No	N/A
ospfHostAreaID	No	N/A
ospfIfTable		
Indicies: ospfIfIpAddress, ospfAddressLessIf		
ospfIfAreaId	Yes	RC
ospfIfType	Yes	RO
ospfIfAdminStat	Yes	RO
ospfIfRtrPriority	Yes	RC
ospfIfTransitDelay	Yes	RC
ospfIfRetransInterval	Yes	RC
ospfIfHelloInterval	Yes	RC
ospfIfRtrDeadInterval	Yes	RC
ospfIfPollInterval	No	N/A
ospfIfState	Yes	RO
ospfIfDesignatedRouter	Yes	RO
ospfIfBackupDesignatedRouter	Yes	RO
ospfIfEvents	Yes	RO
ospfIfAuthKey	Yes	RC
ospfIfStatus	Yes	RC
ospfIfMulticastForwarding	Yes	RO
ospfIfDemand	Yes	RO
ospfIfAuthType	Yes	RW

TABLE C-12 RFC 1850 OSPF MIB *(Continued)*

Object	Support	Access
`ospfIfMetricTable`		
Indicies: `ospfIfMetricIpAddress`, `ospfIfMetricAddressLessIf`, `ospfIfMetricTOS`		
`ospfIfMetricValue`	Yes	RW
`ospfIfMetricStatus`	Yes	RO
`ospfVirtIfTable`		
Indicies: `ospfVirtIfAreaId`, `ospfVirtIfNeighbor`		
`ospfVirtIfTransitDelay`	Yes	RW
`ospfVirtIfRetransInterval`	Yes	RW
`ospfVirtIfHelloInterval`	Yes	RW
`ospfVirtIfRtrDeadInterval`	Yes	RW
`ospfVirtIfState`	Yes	RO
`ospfVirtIfEvents`	Yes	RO
`ospfVirtIfAuthKey`	Yes	RO
`ospfVirtIfStatus`	Yes	RC
`ospfVirtIfAuthType`	Yes	RW
`ospfNbrTable`		
Indicies: `ospfNbrIpAddr`, `ospfNbrAddressLessIndex`		
`ospfNbrRtrId`	Yes	RO
`ospfNbrOptions`	Yes	RO
`ospfNbrPriority`	Yes	RO
`ospfNbrState`	Yes	RO
`ospfNbrEvents`	Yes	RO
`ospfNbrLsRetransQLen`	Yes	RO
`ospfNbmaNbrStatus`	Yes	RO
`ospfNbmaNbrPermanence`	Yes	RO
`ospfNbrHelloSuppressed`	Yes	RO

TABLE C-12 RFC 1850 OSPF MIB *(Continued)*

Object	Support	Access
ospfVirtNbrTable		
Indicies: ospfVirtNbrArea, ospfVirtNbrRtrId		
ospfVirtNbrIpAddr	Yes	RO
ospfVirtNbrOptions	Yes	RO
ospfVirtNbrState	Yes	RO
ospfVirtNbrEvents	Yes	RO
ospfVirtNbrLsRetransQLen	Yes	RO
ospfVirtNbrHelloSuppressed	Yes	RO
ospfExtLsdbTable		
Indicies: ospfExtLsdbType, ospfExtLsdbLsid, ospfExtLsdbRouterId		
ospfExtLsdbSequence	Yes	RO
ospfExtLsdbAge	Yes	RO
ospfExtLsdbChecksum	Yes	RO
ospfExtLsdbAdvertisement	Yes	RO
ospfAreaAggregateTable		
Indicies: ospfAreaAggregateAreaID, ospfAreaAggregateLsdbType, ospfAreaAggregateNet, ospfAreaAggregateMask		
ospfAreaAggregateStatus	Yes	RO
ospfAreaAggregateEffect	Yes	RW

TABLE C-13 RFC 1724 RIPv2 MIB

Object	Support	Access
rip2GlobalGroup		
rip2GlobalRouteChanges	Yes	RO
rip2GlobalQueries	Yes	RO
rip2IfStatTable		

TABLE C-13 RFC 1724 RIPv2 MIB *(Continued)*

Object	Support	Access
Index: rip2IfStatAddress		
rip2IfStatRcvBadPackets	Yes	RO
rip2IfStatRcvBadRoutes	Yes	RO
rip2IfStatSentUpdates	Yes	RO
rip2IfStatStatus	Yes	RC
rip2IfConfTable		
Index: rip2IfConfAddress		
rip2IfConfDomain	No	
rip2IfConfAuthType	Yes	RC
rip2IfConfAuthKey	Yes	RC
rip2IfConfSend	Yes	RC
rip2IfConfReceive	Yes	RC
rip2IfConfDefaultMetric	No	N/A
rip2IfConfStatus	Yes	RC
rip2IfConfSrcAddress	Yes	RO
rip2PeerTable		
Indicies: rip2PeerAddress, rip2PeerDomain		
rip2PeerLastUpdate	No	RO
rip2PeerVersion	No	RO
rip2PeerRcvBadPackets	No	RO
rip2PeerRcvBadRoutes	No	RO

TABLE C-14 RFC 1657 BGP4 MIB

Object	Support	Access
bgp		
bgpVersion	Yes	RO
bgpLocalAs	Yes	RO

TABLE C-14 RFC 1657 BGP4 MIB *(Continued)*

Object	Support	Access
bgpIdentifier	Yes	RO
bgpPeerTable		
Index: bgpPeerRemoteAddr		
bgpPeerIdentifier	Yes	RO
bgpPeerState	Yes	RO
bgpPeerAdminStatus	Yes	RW
bgpPeerNegotiatedVersion	Yes	RO
bgpPeerLocalAddr	Yes	RO
bgpPeerLocalPort	Yes	RO
bgpPeerRemotePort	Yes	RO
bgpPeerRemoteAs	Yes	RO
bgpPeerInUpdates	Yes	RO
bgpPeerOutUpdates	Yes	RO
bgpPeerInTotalMessages	Yes	RO
bgpPeerOutTotalMessages	Yes	RO
bgpPeerLastError	Yes	RO
bgpPeerFsmEstablishedTransitions	Yes	RO
bgpPeerFsmEstablishedTime	Yes	RO
bgpPeerConnectRetryInterval	Yes	RW
bgpPeerHoldTime	Yes	RO
bgpPeerKeepAlive	Yes	RO
bgpPeerHoldTimeConfigured	Yes	RW
bgpPeerKeepAliveConfigured	Yes	RW
bgpPeerMinASOriginationInterval	No	RW
bgpPeerMinRouteAdvertisementInterval	No	RW
bgpPeerInUpdateElapsedTime	Yes	RO
bgpRcvdPathAttrTable		
Indicies: bgpPathAttrDestNetwork, bgpPathAttrPeer		
bgpPathAttrOrigin	Obsolete	
bgpPathAttrASPath	Obsolete	

TABLE C-14 RFC 1657 BGP4 MIB *(Continued)*

Object	Support	Access
bgpPathAttrNextHop	Obsolete	
bgpPathAttrInterASMetric	Obsolete	
bgp4PathAttrTable		
Index: bgp4PathAttrIpAddrPrefix, bgp4PathAttrIpAddrPrefixLen, bgp4PathAttrPeer		
bgp4PathAttrOrigin	Yes	RO
bgp4PathAttrASPathSegment	Yes	RO
bgp4PathAttrNextHop	Yes	RO
bgp4PathAttrMultiExitDisc	Yes	RO
bgp4PathAttrLocalPref	Yes	RO
bgp4PathAttrAtomicAggregate	Yes	RO
bgp4PathAttrAggregatorAS	Yes	RO
bgp4PathAttrAggregatorAddr	Yes	RO
bgp4PathAttrCalcLocalPref	Yes	RO
bgp4PathAttrBest	Yes	RO
bgp4PathAttrUnknown	Yes	RO

TABLE C-15 RFC 1643 Ethernet MIB

Object	Support	Access
dot3StatsTable		
Index: dot3StatsIndex		
dot3StatsAlignmentErrors	Yes	RO
dot3StatsFCSErrors	Yes	RO
dot3StatsSingleCollisionFrames	Yes	RO
dot3StatsMultipleCollisionFrames	Yes	RO
dot3StatsSQETestErrors	Yes	RO
dot3StatsDeferredTransmissions	Yes	RO
dot3StatsLateCollisions	Yes	RO

TABLE C-15 RFC 1643 Ethernet MIB *(Continued)*

Object	Support	Access
dot3StatsExcessiveCollisions	Yes	RO
dot3StatsInternalMacTransmitErrors	Yes	RO
dot3StatsCarrierSenseErrors	Yes	RO
dot3StatsFrameTooLongs	Yes	RO
dot3StatsInternalMacReceiveErrors	Yes	RO
dot3StatsEtherChipSet	No	N/A
dot3CollTable		
Indicies: ifIndex, dot3CollCount		
dot3CollFrequencies	No	

TABLE C-16 RFC 1493 Bridge MIB

Object	Support	Access
dot1dBase		
dot1dBaseBridgeAddress	Yes	RO
dot1dBaseNumPorts	Yes	RO
dot1dBaseType	Yes	RO
dot1dBasePortTable		
Index: dot1dBasePort		
dot1dBasePortIfIndex	Yes	RO
dot1dBasePortCircuit	Yes	RO
dot1dBasePortDelayExceededDiscards	No	N/A
dot1dBasePortMtuExceededDiscards	No	N/A
dot1dStp		
dot1dStpProtocolSpecification	Yes	RO
dot1dStpPriority	Yes	RW

TABLE C-16 RFC 1493 Bridge MIB *(Continued)*

Object	Support	Access
`dot1dStpTimeSinceTopologyChange`	Yes	RO
`dot1dStpTopChanges`	Yes	RO
`dot1dStpDesignatedRoot`	Yes	RO
`dot1dStpRootCost`	Yes	RO
`dot1dStpRootPort`	Yes	RO
`dot1dStpMaxAge`	Yes	RO
`dot1dStpHelloTime`	Yes	RO
`dot1dStpHoldTime`	Yes	RO
`dot1dStpForwardDelay`	Yes	RO
`dot1dStpBridgeMaxAge`	Yes	RW
`dot1dStpBridgeHelloTime`	Yes	RW
`dot1dStpBridgeForwardDelay`	Yes	RW
`dot1dStpPortTable`		
Index: `dot1dStpPort`		
`dot1dStpPortPriority`	Yes	RW
`dot1dStpPortState`	Yes	RO
`dot1dStpPortEnable`	Yes	RW
`dot1dStpPortPathCost`	Yes	RW
`dot1dStpPortDesignatedRoot`	Yes	RO
`dot1dStpPortDesignatedCost`	Yes	RO
`dot1dStpPortDesignatedBridge`	Yes	RO
`dot1dStpPortDesignatedPort`	Yes	RO
`dot1dStpPortForwardTransitions`	Yes	RO
`dot1dTp`		
`dot1dTpLearnedEntryDiscards`	No	N/A
`dot1dTpAgingTime`	Yes	RW

TABLE C-16 RFC 1493 Bridge MIB *(Continued)*

Object	Support	Access
dot1dTpFdbTable		
Index: dot1dTpFdbAddress		
dot1dTpFdbPort	Yes	RO
dot1dTpFdbStatus	Yes	RO
dot1dTpPortTable		
Index: dot1dTpPort		
dot1dTpPortMaxInfo	Yes	RO
dot1dTpPortInFrames	Yes	RO
dot1dTpPortOutFrames	Yes	RO
dot1dTpPortInDiscards	Yes	RO
dot1dStaticTable		
Indicies: dot1dStaticAddress, dot1dStaticReceivePort		
dot1dStaticAllowedToGoTo	No	N/A
dot1dStaticStatus	No	N/A

TABLE C-17 RFC 1213 Mib-2 MIB

Object	Support	Access
system		
sysDescr	Yes	RO
sysObjectID	Yes	RO
sysUpTime	Yes	RO
sysContact	Yes	RW
sysName	Yes	RW
sysLocation	Yes	RW
sysServices	Yes	RO

TABLE C-17 RFC 1213 Mib-2 MIB *(Continued)*

Object	Support	Access
interfaces		
ifNumber	Yes	RO
ifTable		
Index: ifIndex		
ifDescr	Yes	RO
ifType	Yes	RO
ifMtu	Yes	RO
ifSpeed	Yes	RO
ifPhysAddress	Yes	RO
ifAdminStatus	Yes	RW
ifOperStatus	Yes	RO
ifLastChange	Yes	RO
ifInOctets	Yes	RO
ifInUcastPkts	Yes	RO
ifInNUcastPkts	Yes	RO
ifInDiscards	Yes	RO
ifInErrors	Yes	RO
ifInUnknownProtos	Yes	RO
ifOutOctets	Yes	RO
ifOutUcastPkts	Yes	RO
ifOutNUcastPkts	Yes	RO
ifOutDiscards	Yes	RO
ifOutErrors	Yes	RO
ifOutQLen	No	N/A
ifSpecific	No	N/A
atTable		
Index: atIfIndex, atNetAddress		
atPhysAddress	Depreciated	

TABLE C-17 RFC 1213 Mib-2 MIB *(Continued)*

Object	Support	Access
ip		
ipForwarding	Yes	RW
ipDefaultTTL	Yes	RO
ipInReceives	Yes	RO
ipInHdrErrors	Yes	RO
ipInAddrErrors	Yes	RO
ipForwDatagrams	Yes	RO
ipInUnknownProtos	Yes	RO
ipInDiscards	Yes	RO
ipInDelivers	Yes	RO
ipOutRequests	Yes	RO
ipOutDiscards	Yes	RO
ipOutNoRoutes	Yes	RO
ipReasmTimeout	Yes	RO
ipReasmReqds	Yes	RO
ipReasmOKs	Yes	RO
ipReasmFails	Yes	RO
ipFragOKs	Yes	RO
ipFragFails	Yes	RO
ipFragCreates	Yes	RO
ipRoutingDiscards	Yes	RO
ipAddrTable		
Index: ipAdEntAddr		
ipAdEntIfIndex	Yes	RO
ipAdEntNetMask	Yes	RO
ipAdEntBcastAddr	Yes	RO
ipAdEntReasmMaxSize	Yes	RO

TABLE C-17 RFC 1213 Mib-2 MIB *(Continued)*

Object	Support	Access
`ipRouteTable`		
Index: `ipRouteDest`		
`ipRouteIfIndex`	Yes	RO
`ipRouteMetric1`	Yes	RO
`ipRouteMetric2`	Yes	RO
`ipRouteMetric3`	Yes	RO
`ipRouteMetric4`	Yes	RO
`ipRouteNextHop`	Yes	RO
`ipRouteType`	Yes	RO
`ipRouteProto`	Yes	RO
`ipRouteAge`	No	N/A
`ipRouteMask`	Yes	RO
`ipRouteMetric5`	Yes	RO
`ipRouteInfo`	Yes	RO
`ipNetToMedia` **Table**		
Indices: `ipNetToMediaIfIndex`, `ipNetToMediaNetAddress`		
`ipNetToMediaPhysAddress`	Yes	RO
`ipNetToMediaType`	Yes	RO
`icmp` **Group**		
`icmpInMsgs`	Yes	RO
`icmpInErrors`	Yes	RO
`icmpInDestUnreachs`	Yes	RO
`icmpInTimeExcds`	Yes	RO
`icmpInParmProbs`	Yes	RO
`icmpInSrcQuenchs`	Yes	RO
`icmpInRedirects`	Yes	RO
`icmpInEchos`	Yes	RO
`icmpInEchoReps`	Yes	RO

TABLE C-17 RFC 1213 Mib-2 MIB *(Continued)*

Object	Support	Access
icmpInTimestamps	Yes	RO
icmpInTimestampReps	Yes	RO
icmpInAddrMasks	Yes	RO
icmpInAddrMaskReps	Yes	RO
icmpOutMsgs	Yes	RO
icmpOutErrors	Yes	RO
icmpOutDestUnreachs	Yes	RO
icmpOutTimeExcds	Yes	RO
icmpOutParmProbs	Yes	RO
icmpOutSrcQuenchs	Yes	RO
icmpOutRedirects	Yes	RO
icmpOutEchos	Yes	RO
icmpOutEchoReps	Yes	RO
icmpOutTimestamps	Yes	RO
icmpOutTimestampReps	Yes	RO
icmpOutAddrMasks	Yes	RO
icmpOutAddrMaskReps	Yes	RO
tcp **Group**		
tcpRtoAlgorithm	Yes	RO
tcpRtoMin	Yes	RO
tcpRtoMax	Yes	RO
tcpMaxConn	Yes	RO
tcpActiveOpens	Yes	RO
tcpPassiveOpens	Yes	RO
tcpAttemptFails	Yes	RO
tcpEstabResets	Yes	RO
tcpCurrEstab	Yes	RO
tcpInSegs	Yes	RO
tcpOutSegs	Yes	RO

TABLE C-17 RFC 1213 Mib-2 MIB *(Continued)*

Object	Support	Access
tcpRetransSegs	Yes	RO
tcpInErrs	Yes	RO
tcpOutRsts	Yes	RO
tcpConn **Table**		
Indicies: tcpConnLocalAddress, tcpConnLocalPort, tcpConnRemAddress, tcpConnRemPort		
tcpConnState	Yes	RO
udp **Group**		
udpInDatagrams	Yes	RO
udpNoPorts	Yes	RO
udpInErrors	Yes	RO
udpOutDatagrams	Yes	RO
udp **Table**		
Indicies: udpLocalAddress, udpLocalPort		
udpLocalAddress	Yes	RO
udpLocalPort	Yes	RO
egp **Group**		
egpInMsgs	No	N/A
egpInErrors	No	N/A
egpOutMsgs	No	N/A
egpOutErrors	No	N/A
egpAs	No	N/A
egpNeighTable		
Index: egpNeighAddr		
egpNeighState	No	N/A

TABLE C-17 RFC 1213 Mib-2 MIB *(Continued)*

Object	Support	Access
egpNeighAs	No	N/A
egpNeighInMsgs	No	N/A
egpNeighInErrs	No	N/A
egpNeighOutMsgs	No	N/A
egpNeighOutErrs	No	N/A
egpNeighInErrMsgs	No	N/A
egpNeighOutErrMsgs	No	N/A
egpNeighStateUps	No	N/A
egpNeighStateDowns	No	N/A
egpNeighIntervalHello	No	N/A
egpNeighIntervalPoll	No	N/A
egpNeighMode	No	N/A
egpNeighEventTrigger	No	N/A
snmp **Group**		
snmpInPkts	Yes	RO
snmpOutPkts	Obsolete	
snmpInBadVersions	Yes	RO
snmpInBadCommunityNames	Yes	RO
snmpInBadCommunityUses	Yes	RO
snmpInASNParseErrs	Yes	RO
snmpInTooBigs	Obsolete	
snmpInNoSuchNames	Obsolete	
snmpInBadValues	Obsolete	
snmpInReadOnlys	Obsolete	
snmpInGenErrs	Obsolete	
snmpInTotalReqVars	Obsolete	
snmpInTotalSetVars	Obsolete	
snmpInGetRequests	Obsolete	
snmpInGetNexts	Obsolete	

TABLE C-17 RFC 1213 Mib-2 MIB *(Continued)*

Object	Support	Access
`snmpInSetRequests`	Obsolete	
`snmpInGetResponses`	Obsolete	
`snmpInTraps`	Obsolete	
`snmpOutTooBigs`	Obsolete	
`snmpOutNoSuchNames`	Obsolete	
`snmpOutBadValues`	Obsolete	
`snmpOutGenErrs`	Obsolete	
`snmpOutGetRequests`	Obsolete	
`snmpOutGetNexts`	Obsolete	
`snmpOutSetRequests`	Obsolete	
`snmpOutGetResponses`	Obsolete	
`snmpOutTraps`	Obsolete	
`snmpEnableAuthenTraps`	Yes	RW
`snmpSilentDrops`	Yes	RO
`snmpProxyDrops`	Yes	RO

TABLE C-18 POWER-ETHERNET-MIB

Object	Support	Access
`pethPsePortTable`		
Indicies: `pethPsePortGroupIndex`, `pethPsePortIndex`		
`pethPsePortAdminEnable`	Yes	RW
`pethPsePortPowerPairsControlAbility`	Yes	RO
`pethPsePortPowerPairs`	Yes	RW
`pethPsePortDetectionStatus`	Yes	RO
`pethPsePortPowerPriority`	Yes	RW
`pethPsePortMPSAbsentCounter`	Yes	RO
`pethPsePortType`	Yes	RW
`pethPsePortPowerClassifications`	Yes	RO
`pethPsePortInvalidSignatureCounter`	Yes	RO

TABLE C-18 POWER-ETHERNET-MIB *(Continued)*

Object	Support	Access
`pethPsePortPowerDeniedCounter`	Yes	RO
`pethPsePortOverLoadCounter`	Yes	RO
`pethPsePortShortCounter`	Yes	RO
`pethMainPseTable`		
Index: `pethMainPseGroupIndex`		
`pethMainPsePower`	Yes	RO
`pethMainPseOperStatus`	Yes	RO
`pethMainPseConsumptionPower`	Yes	RO
`pethMainPseUsageThreshold`	Yes	RW
`pethNotificationControlTable`		
Index: `pethNotificationControlGroupIndex`		
`pethNotificationControlEnable`	Yes	RW

TABLE C-19 LVL7-POWER-ETHERNET-MIB

Object	Support	Access
`agentPethPsePortTable`		
Augment: `pethPsePortEntry`		
`agentPethPowerLimit`	Yes	RW
`agentPethOutputPower`	Yes	RO
`agentPethOutputCurrent`	Yes	RO
`agentPethOutputVolts`	Yes	RO

TABLE C-20 IEEE8021-PAE-MIB dot1x MIB

Object	Support	Access
dot1xPaeSystem **Group**		
dot1xPaeSystemAuthControl	Yes	RW
dot1xPaePortTable		
Index: dot1xPaePortNumber		
dot1xPaePortProtocolVersion	Yes	RO
dot1xPaePortCapabilities	Yes	RO
dot1xPaePortInitialize	Yes	RW
dot1xPaePortReauthenticate	Yes	RW
dot1xAuthConfigTable		
Index: dot1xPaePortNumber		
dot1xAuthPaeState	Yes	RO
dot1xAuthBackendAuthState	Yes	RO
dot1xAuthAdminControlledDirections	Yes	RO
dot1xAuthOperControlledDirections	Yes	RO
dot1xAuthAuthControlledPortStatus	Yes	RO
dot1xAuthAuthControlledPortControl	Yes	RW
dot1xAuthQuietPeriod	Yes	RW
dot1xAuthTxPeriod	Yes	RW
dot1xAuthSuppTimeout	Yes	RW
dot1xAuthServerTimeout	Yes	RW
dot1xAuthMaxReq	Yes	RW
dot1xAuthReAuthPeriod	Yes	RW
dot1xAuthReAuthEnabled	Yes	RW
dot1xAuthKeyTxEnabled	Yes	RO

TABLE C-20 IEEE8021-PAE-MIB dot1x MIB *(Continued)*

Object	Support	Access
dot1xAuthStatsTable		
Index: dot1xPaePortNumber		
dot1xAuthEapolFramesRx	Yes	RO
dot1xAuthEapolFramesTx	Yes	RO
dot1xAuthEapolStartFramesRx	Yes	RO
dot1xAuthEapolLogoffFramesRx	Yes	RO
dot1xAuthEapolRespIdFramesRx	Yes	RO
dot1xAuthEapolRespFramesRx	Yes	RO
dot1xAuthEapolReqIdFramesTx	Yes	RO
dot1xAuthEapolReqFramesTx	Yes	RO
dot1xAuthInvalidEapolFramesRx	Yes	RO
dot1xAuthEapLengthErrorFramesRx	Yes	RO
dot1xAuthLastEapolFrameVersion	Yes	RO
dot1xAuthLastEapolFrameSource	Yes	RO
dot1xAuthDiagTable		
Index: dot1xPaePortNumber		
dot1xAuthEntersConnecting	Yes	RO
dot1xAuthEapLogoffsWhileConnecting	Yes	RO
dot1xAuthEntersAuthenticating	Yes	RO
dot1xAuthAuthSuccessWhileAuthenticating	Yes	RO
dot1xAuthAuthTimeoutsWhileAuthenticating	Yes	RO
dot1xAuthAuthFailWhileAuthenticating	Yes	RO
dot1xAuthAuthReauthsWhileAuthenticating	Yes	RO
dot1xAuthAuthEapStartsWhileAuthenticating	Yes	RO
dot1xAuthAuthEapLogoffWhileAuthenticating	Yes	RO
dot1xAuthAuthReauthsWhileAuthenticated	Yes	RO
dot1xAuthAuthEapStartsWhileAuthenticated	Yes	RO
dot1xAuthAuthEapLogoffWhileAuthenticated	Yes	RO
dot1xAuthBackendResponses	Yes	RO

TABLE C-20 IEEE8021-PAE-MIB dot1x MIB *(Continued)*

Object	Support	Access
`dot1xAuthBackendAccessChallenges`	Yes	RO
`dot1xAuthBackendOtherRequestsToSupplicant`	Yes	RO
`dot1xAuthBackendNonNakResponsesFromSupplicant`	Yes	RO
`dot1xAuthBackendAuthSuccesses`	Yes	RO
`dot1xAuthBackendAuthFails`	Yes	RO
`dot1xAuthSessionStatsTable`		
Index: `dot1xPaePortNumber`		
`dot1xAuthSessionOctetsRx`	No	N/A
`dot1xAuthSessionOctetsTx`	No	N/A
`dot1xAuthSessionFramesRx`	No	N/A
`dot1xAuthSessionFramesTx`	No	N/A
`dot1xAuthSessionId`	No	N/A
`dot1xAuthSessionAuthenticMethod`	No	N/A
`dot1xAuthSessionTime`	No	N/A
`dot1xAuthSessionTerminateCause`	No	N/A
`dot1xAuthSessionUserName`	No	N/A
`dot1xSuppConfigTable`		
Index: `dot1xPaePortNumber`		
`dot1xSuppPaeState`	No	N/A
`dot1xSuppHeldPeriod`	No	N/A
`dot1xSuppAuthPeriod`	No	N/A
`dot1xSuppStartPeriod`	No	N/A
`dot1xSuppMaxStart`	No	N/A
`dot1xSuppStatsTable`		
Index: `dot1xPaePortNumber`		
`dot1xSuppEapolFramesRx`	No	N/A
`dot1xSuppEapolFramesTx`	No	N/A

TABLE C-20 IEEE8021-PAE-MIB dot1x MIB *(Continued)*

Object	Support	Access
dot1xSuppEapolStartFramesTx	No	N/A
dot1xSuppEapolLogoffFramesTx	No	N/A
dot1xSuppEapolRespIdFramesTx	No	N/A
dot1xSuppEapolRespFramesTx	No	N/A
dot1xSuppEapolReqIdFramesRx	No	N/A
dot1xSuppEapolReqFramesRx	No	N/A
dot1xSuppInvalidEapolFramesRx	No	N/A
dot1xSuppEapLengthErrorFramesRx	No	N/A
dot1xSuppLastEapolFrameVersion	No	N/A
dot1xSuppLastEapolFrameSource	No	N/A

TABLE C-21 FASTPATH-SECURITY-MIB

Object	Support	Access
agentSSLConfigGroup		
agentSSLAdminMode	Yes	RW
agentSSLSecurePort	Yes	RW
agentSSLProtocolLevel	Yes	RW
agentSSHConfigGroup		
agentSSHAdminMode	Yes	RW
agentSSHProtocolLevel	Yes	RW
agentSSHSessionsCount	Yes	RW

TABLE C-22 FASTPATH-MULTICAST-MIB

Object	Support	Access
agentMulticastIGMPConfigGroup		
agentMulticastIGMPAdminMode	Yes	RW

TABLE C-22 FASTPATH-MULTICAST-MIB *(Continued)*

Object	Support	Access
agentMulticastIGMPInterfaceTable		
Index: agentMulticastIGMPInterfaceIfIndex		
agentMulticastIGMPInterfaceAdminMode	Yes	RW
agentMulticastPIMConfigGroup		
agentMulticastPIMConfigMode	Yes	RW
agentMulticastPIMSMConfigGroup		
agentMulticastPIMSMAdminMode	Yes	RW
agentMulticastPIMSMDataThresholdRate	Yes	RW
agentMulticastPIMSMRegThresholdRate	Yes	RW
agentMulticastPIMSMStaticRPTable		
Indices: agentMulticastPIMSMStaticRPIpAddr, agentMulticastPIMSMStaticRPGroupIpAddr, agentMulticastPIMSMStaticRPGroupIpMask		
agentMulticastPIMSMStaticRPStatus	Yes	RW
agentMulticastPIMSMInterfaceTable		
Index: agentMulticastPIMSMInterfaceIndex		
agentMulticastPIMSMInterfaceCBSRHashMaskLength	Yes	RW
agentMulticastPIMSMInterfaceCRPPreference	Yes	RW
agentMulticastPIMDMConfigGroup		
agentMulticastPIMDMAdminMode	Yes	RW
agentMulticastRoutingConfigGroup		
agentMulticastRoutingAdminMode	Yes	RW
agentMulticastDVMRPConfigGroup		
agentMulticastDVMRPAdminMode	Yes	RW

TABLE C-23 FASTPATH-MGMT-SECURITY-MIB

Object	Support	Access
agentSSLConfigGroup		
agentSSLAdminMode	Yes	RW
agentSSLSecurePort	Yes	RW
agentSSLProtocolLevel	Yes	RW
agentSSHConfigGroup		
agentSSHAdminMode	Yes	RW
agentSSHProtocolLevel	Yes	RW
agentSSHSessionsCount	Yes	RW

TABLE C-24 FASTPATH-DHCPSERVER-PRIVATE-MIB

Object	Support	Access
agentDhcpServerGroup		
agentDhcpServerAdminMode	Yes	RW
agentDhcpServerPingPktNos	Yes	RW
agentDhcpServerAutomaticBindingsNos	Yes	RO
agentDhcpServerExpiredBindingsNos	Yes	RO
agentDhcpServerMalformedMessagesReceived	Yes	RO
agentDhcpServerDISCOVERMessagesReceived	Yes	RO
agentDhcpServerREQUESTMessagesReceived	Yes	RO
agentDhcpServerDECLINEMessagesReceived	Yes	RO
agentDhcpServerRELEASEMessagesReceived	Yes	RO
agentDhcpServerINFORMMessagesReceived	Yes	RO
agentDhcpServerOFFERMessagesSent	Yes	RO
agentDhcpServerACKMessagesSent	Yes	RO
agentDhcpServerNAKMessagesSent	Yes	RO
agentDhcpServerClearStatistics	Yes	RW

TABLE C-24 FASTPATH-DHCPSERVER-PRIVATE-MIB *(Continued)*

Object	Support	Access
agentDhcpServerBootpAutomatic	Yes	RW
agentDhcpServerPoolConfigGroup		
agentDhcpServerPoolNameCreate	Yes	RW
agentDhcpServerPoolConfigTable		
Index: agentDhcpServerPoolIndex		
agentDhcpServerPoolName	Yes	RO
agentDhcpServerPoolDefRouter	Yes	RW
agentDhcpServerPoolDNSServer	Yes	RW
agentDhcpServerPoolLeaseTime	Yes	RW
agentDhcpServerPoolType	Yes	RO
agentDhcpServerPoolNetbiosNameServer	Yes	RW
agentDhcpServerPoolNetbiosNodeType	Yes	RW
agentDhcpServerPoolNextServer	Yes	RW
agentDhcpServerPoolDomainName	Yes	RW
agentDhcpServerPoolBootfile	Yes	RW
agentDhcpServerPoolRowStatus	Yes	RW
agentDhcpServerPoolAllocationTable		
Augment: agentDhcpServerPoolConfigEntry		
agentDhcpServerPoolAllocationName	Yes	RO
agentDhcpServerDynamicPoolIpAddress	Yes	RW
agentDhcpServerDynamicPoolIpMask	Yes	RW
agentDhcpServerDynamicPoolIpPrefixLength	Yes	RW
agentDhcpServerPoolAllocationType	Yes	RO
agentDhcpServerManualPoolClientIdentifier	Yes	RW
agentDhcpServerManualPoolClientName	Yes	RW
agentDhcpServerManualPoolClientHWAddr	Yes	RW
agentDhcpServerManualPoolClientHWType	Yes	RW
agentDhcpServerManualPoolIpAddress	Yes	RW
agentDhcpServerManualPoolIpMask	Yes	RW

TABLE C-24 FASTPATH-DHCPSERVER-PRIVATE-MIB *(Continued)*

Object	Support	Access
agentDhcpServerManualPoolIpPrefixLength	Yes	RW
agentDhcpServerPoolConfigGroup		
agentDhcpServerExcludedAddressRangeCreate	Yes	RW
agentDhcpServerExcludedAddressRangeTable		
Index: agentDhcpServerExcludedRangeIndex		
agentDhcpServerExcludedStartIpAddress	Yes	RO
agentDhcpServerExcludedEndIpAddress	Yes	RO
agentDhcpServerExcludedAddressRangeStatus	Yes	RW
agentDhcpServerPoolConfigGroup		
agentDhcpServerPoolOptionCreate	Yes	RW
agentDhcpServerPoolOptionTable		
Indicies: agentDhcpServerPoolOptionIndex, agentDhcpServerPoolOptionCode		
agentDhcpServerOptionPoolName	Yes	RO
agentDhcpServerPoolOptionAsciiData	Yes	RW
agentDhcpServerPoolOptionHexData	Yes	RW
agentDhcpServerPoolOptionIpAddressData	Yes	RW
agentDhcpServerPoolOptionStatus	Yes	RW
agentDhcpServerLeaseGroup		
agentDhcpServerLeaseClearAllBindings	Yes	RW
agentDhcpServerLeaseTable		
Index: agentDhcpServerLeaseIPAddress		
agentDhcpServerLeaseIPMask	Yes	RO
agentDhcpServerLeaseHWAddress	Yes	RO
agentDhcpServerLeaseRemainingTime	Yes	RO
agentDhcpServerLeaseType	Yes	RO

TABLE C-24 FASTPATH-DHCPSERVER-PRIVATE-MIB *(Continued)*

Object	Support	Access
agentDhcpServerLeaseStatus	Yes	RW
agentDhcpServerAddressConflictGroup		
agentDhcpServerClearAllAddressConflicts	Yes	RW
agentDhcpServerAddressConflictLogging	Yes	RW
agentDhcpServerAddressConflictTable		
Index: agentDhcpServerAddressConflictIP		
agentDhcpServerAddressConflictDetectionType	Yes	RO
agentDhcpServerAddressConflictDetectionTime	Yes	RO
agentDhcpServerAddressConflictStatus	Yes	RW

TABLE C-25 FASTPATH-BGP-MIB

Object	Support	Access
agentBGPConfigGroup		
agentBGPAdminMode	Yes	RW
agentBGPDefaultMetric	Yes	RW
agentBGPDefaultMetricConfigured	Yes	RW
agentBGPDefaultInfoOriginate	Yes	RW
agentBgpPeerTable		
Augment: bgpPeerEntry		
agentBgpPeerAuthType	Yes	RC
agentBgpPeerAuthKey	Yes	RC
agentBGPRouteRedistTable		
Index: agentBGPRouteRedistSource		
agentBGPRouteRedistMode	Yes	RW
agentBGPRouteRedistMetric	Yes	RW
agentBGPRouteRedistMetricConfigured	Yes	RW

TABLE C-25 FASTPATH-BGP-MIB *(Continued)*

Object	Support	Access
agentBGPRouteRedistMatchInternal	Yes	RW
agentBGPRouteRedistMatchExternal1	Yes	RW
agentBGPRouteRedistMatchExternal2	Yes	RW
agentBGPRouteRedistMatchNSSAExternal1	Yes	RW
agentBGPRouteRedistMatchNSSAExternal2	Yes	RW
agentBGPRouteRedistDistList	Yes	RW
agentBGPRouteRedistDistListConfigured	Yes	RW

TABLE C-26 FASTPATH Switching MIB

Object	Support	Access
agentInventoryGroup		
agentInventorySysDescription	Yes	RO
agentInventoryMachineType	Yes	RO
agentInventoryMachineModel	Yes	RO
agentInventorySerialNumber	Yes	RO
agentInventoryFRUNumber	Yes	RO
agentInventoryMaintenanceLevel	Yes	RO
agentInventoryPartNumber	Yes	RO
agentInventoryManufacturer	Yes	RO
agentInventoryBurnedInMacAddress	Yes	RO
agentInventoryOperatingSystem	Yes	RO
agentInventoryNetworkProcessingDevice	Yes	RO
agentInventoryAdditionalPackages	Yes	RO
agentInventorySoftwareVersion	Yes	RO
agentTrapLogGroup		
agentTrapLogTotal	Yes	RO
agentTrapLogTotalSinceLastViewed	No	RO

TABLE C-26 FASTPATH Switching MIB *(Continued)*

Object	Support	Access
agentTrapLogTable		
Index: agentTrapLogIndex		
agentTrapLogSystemTime	Yes	RO
agentTrapLogTrap	Yes	RO
agentSupportedMibTable		
Index: agentSupportedMibIndex		
agentSupportedMibName	Yes	RO
agentSupportedMibDescription	Yes	RO
agentLoginSessionTable		
Index: agentLoginSessionIndex		
agentLoginSessionUserName	Yes	RO
agentLoginSessionIPAddress	Yes	RO
agentLoginSessionConnectionType	Yes	RO
agentLoginSessionIdleTime	Yes	RO
agentLoginSessionSessionTime	Yes	RO
agentLoginSessionStatus	Yes	RW
agentTelnetGroup		
agentTelnetLoginTimeout	Yes	RW
agentTelnetMaxSessions	Yes	RW
agentTelnetAllowNew	Yes	RW
agentUserConfigGroup		
agentUserConfigCreate	Yes	RW
agentUserConfig **Table**		
Index: agentUserIndex		

TABLE C-26 FASTPATH Switching MIB *(Continued)*

Object	Support	Access
agentUserName	Yes	RW
agentUserPassword	Yes	RW
agentUserAccessMode	Yes	RO
agentUserStatus	Yes	RW
agentUserAuthenticationType	Yes	RW
agentUserEncryptionType	Yes	RW
agentUserEncryptionPassword	Yes	RW
agentSerial **Group**		
agentSerialBaudrate	Yes	RW
agentSerialTimeout	Yes	RW
agentSerialCharacterSize	Yes	RO
agentSerialHWFlowControlMode	Yes	RO
agentSerialStopBits	Yes	RO
agentSerialParityType	Yes	RO
agentLagConfigGroup		
agentLagConfigCreate	Yes	RW
agentLagConfigStaticCapability	Yes	RW
agentLagSummaryConfig **Table**		
Index: agentLagSummaryLagIndex		
agentLagSummaryName	Yes	RW
agentLagSummaryFlushTimer	No	N/A
agentLagSummaryLinkTrap	Yes	RW
agentLagSummaryAdminMode	Yes	RW
agentLagSummaryStpMode	Yes	RW
agentLagSummaryAddPort	Yes	RW
agentLagSummaryDeletePort	Yes	RW
agentLagSummaryStatus	Yes	RW

TABLE C-26 FASTPATH Switching MIB *(Continued)*

Object	Support	Access
agentLagSummaryType	Yes	RO
agentLagDetailedConfig **Table**		
Indicies: agentLagDetailedLagIndex, agentLagDetailedIfIndex		
agentLagDetailedPortSpeed	Yes	RO
agentLagDetailedPortStatus	Yes	RO
agentNetworkConfig **Group**		
agentNetworkIPAddress	Yes	RW
agentNetworkSubnetMask	Yes	RW
agentNetworkDefaultGateway	Yes	RW
agentNetworkBurnedInMacAddress	Yes	RO
agentNetworkLocalAdminMacAddress	Yes	RW
agentNetworkMacAddressType	Yes	RW
agentNetworkConfigProtocol	Yes	RW
agentNetworkWebMode	Yes	RW
agentNetworkJavaMode	Yes	RW
agentNetworkMgmtVlan	Yes	RW
agentServicePortConfig **Group**		
agentServicePortIPAddress	Yes	RW
agentServicePortSubnetMask	Yes	RW
agentServicePortDefaultGateway	Yes	RW
agentServicePortBurnedInMacAddress	Yes	RO
agentServicePortConfigProtocol	Yes	RW
agentSnmpConfig **Group**		
agentSnmpCommunityCreate	Yes	RW
agentSnmpTrapReceiverCreate	Yes	RW
agentSnmpCommunityConfig **Table**		

TABLE C-26 FASTPATH Switching MIB *(Continued)*

Object	Support	Access
Index: agentSnmpCommunityIndex		
agentSnmpCommunityName	Yes	RW
agentSnmpCommunityIPAddress	Yes	RW
agentSnmpCommunityIPMask	Yes	RW
agentSnmpCommunityAccessMode	Yes	RW
agentSnmpCommunityStatus	Yes	RW
agentSnmpTrapReceiverConfig **Table**		
Index: agentSnmpTrapReceiverIndex		
agentSnmpTrapReceiverCommunityName	Yes	RW
agentSnmpTrapReceiverIPAddress	Yes	RW
agentSnmpTrapReceiverStatus	Yes	RW
agentSnmpTrapFlagsConfig **Group**		
agentSnmpAuthenticationTrapFlag	Yes	RW
agentSnmpLinkUpDownTrapFlag	Yes	RW
agentSnmpMultipleUsersTrapFlag	Yes	RW
agentSnmpSpanningTreeTrapFlag	Yes	RW
agentSnmpBroadcastStormTrapFlag	Yes	RW
agentSpanningTreeConfig **Group**		
agentSpanningTreeMode	Yes	RW
agentSwitchConfig **Group**		
agentSwitchBroadcastStormRecoveryMode	Yes	RW
agentSwitchDot3FlowControlMode	Yes	RW
agentSwitchAddressAgingTimeoutTable		
Index: dot1qFdbId		
agentSwitchAddressAgingTimeout	Yes	RW

TABLE C-26 FASTPATH Switching MIB *(Continued)*

Object	Support	Access
agentSwitchStaticMacFilteringTable		
Index: agentSwitchStaticMacFilteringVlanId		
agentSwitchStaticMacFilteringAddress	Yes	RW
agentSwitchStaticMacFilteringSourcePortMask	Yes	RW
agentSwitchStaticMacFilteringDestPortMask	Yes	RW
agentSwitchStaticMacFilteringStatus	Yes	RC
agentSwitchIGMPSnoopingGroup		
agentSwitchIGMPSnoopingAdminMode	Yes	RW
agentSwitchIGMPSnoopingGroupMembershipInterval	Yes	RW
agentSwitchIGMPSnoopingMaxResponseTime	Yes	RW
agentSwitchIGMPSnoopingExpirationTime	Yes	RW
agentSwitchIGMPSnoopingPortMask	Yes	RW
agentSwitchIGMPSnoopingMulticastControlFramesProcess	Yes	RO
agentSwitchMFDBTable		
Indicies: agentSwitchMFDBVlanId, agentSwitchMFDBMacAddress, agentSwitchMFDBProtocolType		
agentSwitchMFDBType	Yes	RO
agentSwitchMFDBDescription	Yes	RO
agentSwitchMFDBForwardingPortMask	Yes	RO
agentSwitchMFDBFilteringPortMask	Yes	RO
agentSwitchMFDBSummaryTable		
Indicies: agentSwitchMFDBSummaryVlanId, agentSwitchMFDBSummaryMacAddress		
agentSwitchMFDBSummaryForwardingPortMask	Yes	RO
agentSwitchMFDBGroup		
agentSwitchMFDBMaxTableEntries	Yes	RO
agentSwitchMFDBMostEntriesUsed	Yes	RO

TABLE C-26 FASTPATH Switching MIB *(Continued)*

Object	Support	Access
agentSwitchMFDBCurrentEntries	Yes	RO
agentTransferUploadConfig **Group**		
agentTransferUploadMode	Yes	RW
agentTransferUploadServerIP	Yes	RW
agentTransferUploadPath	Yes	RW
agentTransferUploadFilename	Yes	RW
agentTransferUploadDataType	Yes	RW
agentTransferUploadStart	Yes	RW
agentTransferUploadStatus	Yes	RO
agentTransferDownloadConfig **Group**		
agentTransferDownloadMode	Yes	RW
agentTransferDownloadServerIP	Yes	RW
agentTransferDownloadPath	Yes	RW
agentTransferDownloadFilename	Yes	RW
agentTransferDownloadDataType	Yes	RW
agentTransferDownloadStart	Yes	RW
agentTransferDownloadStatus	Yes	RO
agentPortMirroring **Group**		
agentMirroredPortIfIndex	Yes	RW
agentProbePortIfIndex	Yes	RW
agentPortMirroringMode	Yes	RW
agentDot3adAggPortTable		
Index: agentDot3adAggPort		
agentDot3adAggPortLACPMode	Yes	RW
agentPortConfig **Table**		
Index: agentPortDot1dBasePort		

TABLE C-26 FASTPATH Switching MIB *(Continued)*

Object	Support	Access
agentPortIfIndex	Yes	RO
agentPortIanaType	Yes	RO
agentPortSTPState	Yes	RO
agentPortSTPMode	Yes	RW
agentPortAdminMode	Yes	RW
agentPortPhysicalMode	No	N/A
agentPortPhysicalStatus	No	N/A
agentPortLinkTrapMode	Yes	RW
agentPortClearStats	Yes	RW
agentPortDefaultType	Yes	RW
agentPortType	Yes	RO
agentPortAutoNegAdminStatus	Yes	RW
agentPortDot3FlowControlMode	Yes	RW
agentPortDVlanTagMode	Yes	RW
agentPortDVlanTagEthertype	Yes	RW
agentPortDVlanTagCustomerId	Yes	RW
agentPortMaxFrameSizeLimit	Yes	RO
agentPortMaxFrameSize	Yes	RW
agentProtocolConfigGroup		
agentProtocolGroupCreate	Yes	RW
agentProtocolGroupTable		
Index: agentProtocolGroupId		
agentProtocolGroupName	Yes	RO
agentProtocolGroupVlanId	Yes	RW
agentProtocolGroupProtocolIP	Yes	RW
agentProtocolGroupProtocolARP	Yes	RW
agentProtocolGroupProtocolIPX	Yes	RW
agentProtocolGroupStatus	Yes	RW

TABLE C-26 FASTPATH Switching MIB *(Continued)*

Object	Support	Access
agentProtocolGroupPortTable		
Indicies: agentProtocolGroupId, agentProtocolGroupPortIfIndex		
agentProtocolGroupPortStatus	Yes	RC
agentStpSwitchConfigGroup		
agentStpConfigDigestKey	Yes	RO
agentStpConfigFormatSelector	Yes	RO
agentStpConfigName	Yes	RW
agentStpConfigRevision	Yes	RW
agentStpForceVersion	Yes	RW
agentStpAdminMode	Yes	RW
agentStpPortTable		
Index: ifIndex		
agentStpPortState	Yes	RW
agentStpPortStatsMstpBpduRx	Yes	RO
agentStpPortStatsMstpBpduTx	Yes	RO
agentStpPortStatsRstpBpduRx	Yes	RO
agentStpPortStatsRstpBpduTx	Yes	RO
agentStpPortStatsStpBpduRx	Yes	RO
agentStpPortStatsStpBpduTx	Yes	RO
agentStpPortUpTime	Yes	RO
agentStpPortMigrationCheck	Yes	RW
agentStpCstConfigGroup		
agentStpCstHelloTime	Yes	RO
agentStpCstMaxAge	Yes	RO
agentStpCstRegionalRootId	Yes	RO
agentStpCstRegionalRootPathCost	Yes	RO

TABLE C-26 FASTPATH Switching MIB *(Continued)*

Object	Support	Access
agentStpCstRootFwdDelay	Yes	RO
agentStpCstBridgeFwdDelay	Yes	RW
agentStpCstBridgeHelloTime	Yes	RW
agentStpCstBridgeHoldTime	Yes	RO
agentStpCstBridgeMaxAge	Yes	RW
agentStpCstPortTable		
Index: ifIndex		
agentStpCstPortOperEdge	Yes	RO
agentStpCstPortOperPointToPoint	Yes	RO
agentStpCstPortTopologyChangeAck	Yes	RO
agentStpCstPortEdge	Yes	RW
agentStpCstPortForwardingState	Yes	RO
agentStpCstPortId	Yes	RO
agentStpCstPortPathCost	Yes	RW
agentStpCstPortPriority	Yes	RW
agentStpCstDesignatedBridgeId	Yes	RO
agentStpCstDesignatedCost	Yes	RO
agentStpCstDesignatedPortId	Yes	RO
agentStpMstTable		
Index: agentStpMstId		
agentStpMstBridgePriority	Yes	RW
agentStpMstBridgeIdentifier	Yes	RO
agentStpMstDesignatedRootId	Yes	RO
agentStpMstRootPathCost	Yes	RO
agentStpMstRootPortId	Yes	RO
agentStpMstTimeSinceTopologyChange	Yes	RO
agentStpMstTopologyChangeCount	Yes	RO
agentStpMstTopologyChangeParm	Yes	RO

TABLE C-26 FASTPATH Switching MIB *(Continued)*

Object	Support	Access
agentStpMstRowStatus	Yes	RC
agentStpMstPortTable		
Indicies: agentStpMstId, ifIndex		
agentStpMstPortForwardingState	Yes	RO
agentStpMstPortId	Yes	RO
agentStpMstPortPathCost	Yes	RW
agentStpMstPortPriority	Yes	RW
agentStpMstDesignatedBridgeId	Yes	RO
agentStpMstDesignatedCost	Yes	RO
agentStpMstDesignatedPortId	Yes	RO
agentStpMstVlanTable		
Indicies: agentStpMstId, dot1qVlanIndex		
agentStpMstVlanRowStatus	Yes	RC
agentAuthenticationGroup		
agentAuthenticationListCreate	Yes	RW
agentUserConfigDefaultAuthenticationList	Yes	RW
agentAuthenticationListTable		
Index: agentAuthenticationListIndex		
agentAuthenticationListName	Yes	RO
agentAuthenticationListMethod1	Yes	RW
agentAuthenticationListMethod2	Yes	RW
agentAuthenticationListMethod3	Yes	RW
agentAuthenticationListStatus	Yes	RW
agentUserAuthenticationConfigTable		
Augment: agentUserConfigEntry		
agentUserAuthenticationList	Yes	RW

TABLE C-26 FASTPATH Switching MIB *(Continued)*

Object	Support	Access
agentUserPortConfigTable		
Augment: agentUserConfigEntry		
agentUserPortSecurity	Yes	RW
agentClassOfServicePortTable		
Indecies: ifIndex, agentClassOfServicePortPriority		
agentClassOfServicePortClass	Yes	RW
agentSystemConfig **Group**		
agentSaveConfig	Yes	RW
agentSaveConfigStatus	Yes	RW
agentClearConfig	Yes	RW
agentClearLags	Yes	RW
agentClearLoginSessions	Yes	RW
agentClearPasswords	Yes	RW
agentClearPortStats	Yes	RW
agentClearSwitchStats	Yes	RW
agentClearTrapLog	Yes	RW
agentClearVlan	Yes	RW
agentResetSystem	Yes	RO
agentCableTesterGroup		
agentCableTesterStatus	Yes	RW
agentCableTesterIfIndex	Yes	RW
agentCableTesterCableStatus	Yes	RO
agentCableTesterMinimumCableLength	Yes	RO
agentCableTesterMaximumCableLength	Yes	RO
agentCableTesterCableFailureLocation	Yes	RO

TABLE C-27 FASTPATH Routing MIB

Object	Support	Access
agentSwitchArpGroup		
agentSwitchArpAgeoutTime	Yes	RW
agentSwitchArpResponseTime	Yes	RW
agentSwitchArpMaxRetries	Yes	RW
agentSwitchArpCacheSize	Yes	RW
agentSwitchArpDynamicRenew	Yes	RW
agentSwitchArpTotalEntryCountCurrent	Yes	RO
agentSwitchArpTotalEntryCountPeak	Yes	RO
agentSwitchArpStaticEntryCountCurrent	Yes	RO
agentSwitchArpStaticEntryCountMax	Yes	RO
agentSwitchArpTable		
Index: agentSwitchArpIpAddress		
agentSwitchArpAge	Yes	RO
agentSwitchArpMacAddress	Yes	RC
agentSwitchArpInterface	Yes	RO
agentSwitchArpType	Yes	RO
agentSwitchArpStatus	Yes	RW
agentSwitchIpGroup		
agentSwitchIpRoutingMode	Yes	RW
agentSwitchIpInterfaceTable		
Index: agentSwitchIpInterfaceIfIndex		
agentSwitchIpInterfaceIpAddress	Yes	RW
agentSwitchIpInterfaceNetMask	Yes	RW
agentSwitchIpInterfaceClearIp	Yes	RW
agentSwitchIpInterfaceRoutingMode	Yes	RW

TABLE C-27 FASTPATH Routing MIB *(Continued)*

Object	Support	Access
agentSwitchIpRouterDiscoveryTable		
Index: agentSwitchIpRouterDiscoveryIfIndex		
agentSwitchIpRouterDiscoveryAdvertiseMode	Yes	RW
agentSwitchIpRouterDiscoveryIpAddress	Yes	RO
agentSwitchIpRouterDiscoveryMaxAdvertisementInterval	Yes	RW
agentSwitchIpRouterDiscoveryMinAdvertisementInterval	Yes	RW
agentSwitchIpRouterDiscoveryAdvertisementLifetime	Yes	RW
agentSwitchIpRouterDiscoveryPreferenceLevel	Yes	RW
agentSwitchIpRouterDiscoveryAdvertisementAddress	Yes	RW
agentSwitchIpVlanTable		
Index: agentSwitchIpVlanId		
agentSwitchIpVlanIfIndex	Yes	RO
agentSwitchIpVlanRoutingStatus	Yes	RC
agentRouterRipConfigGroup		
agentRouterRipAdminState	Yes	RW
agentRouterRipSplitHorizonMode	Yes	RW
agentRouterRipAutoSummaryMode	Yes	RW
agentRouterRipHostRoutesAcceptMode	Yes	RW
agentRouterRipDefaultMetric	Yes	RW
agentRouterRipDefaultMetricConfigured	Yes	RW
agentRouterRipDefaultInfoOriginate	Yes	RW
agentRipRouteRedistTable		
Index: agentRipRouteRedistSource		
agentRipRouteRedistMode	Yes	RW
agentRipRouteRedistMetric	Yes	RW
agentRipRouteRedistMetricConfigured	Yes	RW
agentRipRouteRedistMatchInternal	Yes	RW

TABLE C-27 FASTPATH Routing MIB *(Continued)*

Object	Support	Access
agentRipRouteRedistMatchExternal1	Yes	RW
agentRipRouteRedistMatchExternal2	Yes	RW
agentRipRouteRedistMatchNSSAExternal1	Yes	RW
agentRipRouteRedistMatchNSSAExternal2	Yes	RW
agentRipRouteRedistDistList	Yes	RW
agentRipRouteRedistDistListConfigured	Yes	RW
agentRouterOspfConfigGroup		
agentOspfDefaultMetric	Yes	RW
agentOspfDefaultMetricConfigured	Yes	RW
agentOspfDefaultInfoOriginate	Yes	RW
agentOspfDefaultInfoOriginateAlways	Yes	RW
agentOspfDefaultInfoOriginateMetric	Yes	RW
agentOspfDefaultInfoOriginateMetricConfigured	Yes	RW
agentOspfDefaultInfoOriginateMetricType	Yes	RW
agentRouterOspfRFC1583CompatibilityMode	Yes	RW
agentOspfRouteRedistTable		
Index: agentOspfRouteRedistSource		
agentOspfRouteRedistMode	Yes	RW
agentOspfRouteRedistMetric	Yes	RW
agentOspfRouteRedistMetricConfigured	Yes	RW
agentOspfRouteRedistMetricType	Yes	RW
agentOspfRouteRedistTag	Yes	RW
agentOspfRouteRedistSubnets	Yes	RW
agentOspfRouteRedistDistList	Yes	RW
agentOspfRouteRedistDistListConfigured	Yes	RW

TABLE C-27 FASTPATH Routing MIB *(Continued)*

Object	Support	Access
agentOspfIfTable		
Augment: ospfIfEntry		
agentOspfIfAuthKeyId	Yes	RC
agentOspfVirtIfTable		
Augment: ospfVirtIfEntry		
agentOspfVirtIfAuthKeyId	Yes	RW
agentOspfAreaTable		
Augment: ospfAreaEntry		
agentOspfAuthType	Yes	RW
agentSnmpTrapFlagsConfigGroupLayer3		
agentSnmpVRRPNewMasterTrapFlag	Yes	RW
agentSnmpVRRPAuthFailureTrapFlag	Yes	RW
agentBootpDhcpRelayGroup		
agentBootpDhcpRelayMaxHopCount	Yes	RW
agentBootpDhcpRelayForwardingIp	Yes	RW
agentBootpDhcpRelayForwardMode	Yes	RW
agentBootpDhcpRelayMinWaitTime	Yes	RW
agentBootpDhcpRelayCircuitIdOptionMode	Yes	RW
agentBootpDhcpRelayNumOfRequestsReceived	Yes	RO
agentBootpDhcpRelayNumOfRequestsForwarded	Yes	RO
agentBootpDhcpRelayNumOfDiscards	Yes	RO

TABLE C-28 FASTPATH Radius MIB

Object	Support	Access
agentRadiusConfigGroup		
agentRadiusMaxTransmit	Yes	RW
agentRadiusTimeout	Yes	RW
agentRadiusAccountingMode	Yes	RW
agentRadiusStatsClear	Yes	RW
agentRadiusAccountingIndexNextValid	Yes	RO
agentRadiusServerIndexNextValid	Yes	RO
agentRadiusAccountingConfig **Table**		
Index: agentRadi		
agentRadiusAccountingServerAddress	Yes	RW
agentRadiusAccountingPort	Yes	RW
agentRadiusAccountingSecret	Yes	RW
agentRadiusAccountingStatus	Yes	RW
agentRadiusServerConfig **Table**		
Index: agentRadi		
agentRadiusServerAddress	Yes	RW
agentRadiusServerPort	Yes	RW
agentRadiusServerSecret	Yes	RW
agentRadiusServerPrimaryMode	Yes	RW
agentRadiusServerCurrentMode	Yes	RO
agentRadiusServerMsgAuth	Yes	RW
agentRadiusServerStatus	Yes	RW

TABLE C-29 FASTPATH QOS DiffServ MIB

Object	Support	Access
agentDiffServGenStatusGroup		
Index:		
agentDiffServGenStatusAdminMode	Yes	RW
agentDiffServGenStatusClassTableSize	Yes	RO
agentDiffServGenStatusClassTableMax	Yes	RO
agentDiffServGenStatusClassRuleTableSize	Yes	RO
agentDiffServGenStatusClassRuleTableMax	Yes	RO
agentDiffServGenStatusPolicyTableSize	Yes	RO
agentDiffServGenStatusPolicyTableMax	Yes	RO
agentDiffServGenStatusPolicyInstTableSize	Yes	RO
agentDiffServGenStatusPolicyInstTableMax	Yes	RO
agentDiffServGenStatusPolicyAttrTableSize	Yes	RO
agentDiffServGenStatusPolicyAttrTableMax	Yes	RO
agentDiffServGenStatusServiceTableSize	Yes	RO
agentDiffServGenStatusServiceTableMax	Yes	RO
agentDiffServClassGroup		
Index:		
agentDiffServClassIndexNextFree	Yes	RO
agentDiffServClassTable		
Index: agentDiffServClassIndex		
agentDiffServClassName	Yes	RC
agentDiffServClassType	Yes	RC
agentDiffServClassAclNum	Yes	RC
agentDiffServClassRuleIndexNextFree	Yes	RO
agentDiffServClassStorageType	Yes	RC
agentDiffServClassRowStatus	Yes	RC

TABLE C-29 FASTPATH QOS DiffServ MIB *(Continued)*

Object	Support	Access
agentDiffServClassRuleTable		
Indicies: agentDiffServClassIndex, agentDiffServClassRuleIndex		
agentDiffServClassRuleMatchEntryType	Yes	RC
agentDiffServClassRuleMatchCos	Yes	RC
agentDiffServClassRuleMatchDstIpAddr	Yes	RC
agentDiffServClassRuleMatchDstIpMask	Yes	RC
agentDiffServClassRuleMatchDstL4PortStart	Yes	RC
agentDiffServClassRuleMatchDstL4PortEnd	Yes	RC
agentDiffServClassRuleMatchDstMacAddr	Yes	RC
agentDiffServClassRuleMatchDstMacMask	Yes	RC
agentDiffServClassRuleMatchEvery	Yes	RO
agentDiffServClassRuleMatchIpDscp	Yes	RC
agentDiffServClassRuleMatchIpPrecedence	Yes	RC
agentDiffServClassRuleMatchIpTosBits	Yes	RC
agentDiffServClassRuleMatchIpTosMask	Yes	RC
agentDiffServClassRuleMatchProtocolNum	Yes	RC
agentDiffServClassRuleMatchRefClassIndex	Yes	RC
agentDiffServClassRuleMatchSrcIpAddr	Yes	RC
agentDiffServClassRuleMatchSrcIpMask	Yes	RC
agentDiffServClassRuleMatchSrcL4PortStart	Yes	RC
agentDiffServClassRuleMatchSrcL4PortEnd	Yes	RC
agentDiffServClassRuleMatchSrcMacAddr	Yes	RC
agentDiffServClassRuleMatchSrcMacMask	Yes	RC
agentDiffServClassRuleMatchVlanId	Yes	RC
agentDiffServClassRuleMatchExcludeFlag	Yes	RC
agentDiffServClassRuleStorageType	Yes	RC
agentDiffServClassRuleRowStatus	Yes	RC

TABLE C-29 FASTPATH QOS DiffServ MIB *(Continued)*

Object	Support	Access
agentDiffServPolicyGroup		
Index: agentDiffServPolicyIndexNextFree	Yes	RO
agentDiffServPolicyTable		
Index: agentDiffServPolicyIndex		
agentDiffServPolicyName	Yes	RC
agentDiffServPolicyType	Yes	RC
agentDiffServPolicyInstIndexNextFree	Yes	RO
agentDiffServPolicyStorageType	Yes	RC
agentDiffServPolicyRowStatus	Yes	RC
agentDiffServPolicyInstTable		
Indicies: agentDiffServPolicyIndex, agentDiffServPolicyInstIndex		
agentDiffServPolicyInstClassIndex	Yes	RC
agentDiffServPolicyInstAttrIndexNextFree	Yes	RO
agentDiffServPolicyInstStorageType	Yes	RC
agentDiffServPolicyInstRowStatus	Yes	RC
agentDiffServPolicyAttrTable		
Indicies: agentDiffServPolicyIndex, agentDiffServPolicyInstIndex, agentDiffServPolicyAttrIndex		
agentDiffServPolicyAttrStmtEntryType	Yes	RC
agentDiffServPolicyAttrStmtBandwidthCrate	Yes	RC
agentDiffServPolicyAttrStmtBandwidthCrateUnits	Yes	RC
agentDiffServPolicyAttrStmtExpediteCrate	Yes	RC
agentDiffServPolicyAttrStmtExpediteCrateUnits	Yes	RC
agentDiffServPolicyAttrStmtExpediteCburst	Yes	RC
agentDiffServPolicyAttrStmtMarkCosVal	Yes	RC
agentDiffServPolicyAttrStmtMarkIpDscpVal	Yes	RC

TABLE C-29 FASTPATH QOS DiffServ MIB *(Continued)*

Object	Support	Access
agentDiffServPolicyAttrStmtMarkIpPrecedenceVal	Yes	RC
agentDiffServPolicyAttrStmtPoliceConformAct	Yes	RC
agentDiffServPolicyAttrStmtPoliceConformVal	Yes	RC
agentDiffServPolicyAttrStmtPoliceExceedAct	Yes	RC
agentDiffServPolicyAttrStmtPoliceExceedVal	Yes	RC
agentDiffServPolicyAttrStmtPoliceNonconformAct	Yes	RC
agentDiffServPolicyAttrStmtPoliceNonconformVal	Yes	RC
agentDiffServPolicyAttrStmtPoliceSimpleCrate	Yes	RC
agentDiffServPolicyAttrStmtPoliceSimpleCburst	Yes	RC
agentDiffServPolicyAttrStmtPoliceSinglerateCrate	Yes	RC
agentDiffServPolicyAttrStmtPoliceSinglerateCburst	Yes	RC
agentDiffServPolicyAttrStmtPoliceSinglerateEburst	Yes	RC
agentDiffServPolicyAttrStmtPoliceTworateCrate	Yes	RC
agentDiffServPolicyAttrStmtPoliceTworateCburst	Yes	RC
agentDiffServPolicyAttrStmtPoliceTworatePrate	Yes	RC
agentDiffServPolicyAttrStmtPoliceTworatePburst	Yes	RC
agentDiffServPolicyAttrStmtRandomdropMinThresh	Yes	RC
agentDiffServPolicyAttrStmtRandomdropMaxThresh	Yes	RC
agentDiffServPolicyAttrStmtRandomdropMaxDropProb	Yes	RC
agentDiffServPolicyAttrStmtRandomdropSamplingRate	Yes	RC
agentDiffServPolicyAttrStmtRandomdropDecayExponent	Yes	RC
agentDiffServPolicyAttrStmtShapeAverageCrate	Yes	RC
agentDiffServPolicyAttrStmtShapePeakCrate	Yes	RC
agentDiffServPolicyAttrStmtShapePeakPrate	Yes	RC
agentDiffServPolicyAttrStorageType	Yes	RC
agentDiffServPolicyAttrRowStatus	Yes	RC
agentDiffServPolicyPerfInTable		
Indicies: agentDiffServPolicyIndex, agentDiffServPolicyInstIndex, ifIndex		
agentDiffServPolicyPerfInOfferedOctets	Yes	RO

TABLE C-29 FASTPATH QOS DiffServ MIB *(Continued)*

Object	Support	Access
agentDiffServPolicyPerfInOfferedPackets	Yes	RO
agentDiffServPolicyPerfInDiscardedOctets	Yes	RO
agentDiffServPolicyPerfInDiscardedPackets	Yes	RO
agentDiffServPolicyPerfInHCOfferedOctets	Yes	RO
agentDiffServPolicyPerfInHCOfferedPackets	Yes	RO
agentDiffServPolicyPerfInHCDiscardedOctets	Yes	RO
agentDiffServPolicyPerfInHCDiscardedPackets	Yes	RO
agentDiffServPolicyPerfInStorageType	Yes	RO
agentDiffServPolicyPerfInRowStatus	Yes	RO
agentDiffServPolicyPerfOutTable		
Indicies: agentDiffServPolicyIndex, agentDiffServPolicyInstIndex, ifIndex		
agentDiffServPolicyPerfOutTailDroppedOctets	Yes	RO
agentDiffServPolicyPerfOutTailDroppedPackets	Yes	RO
agentDiffServPolicyPerfOutRandomDroppedOctets	Yes	RO
agentDiffServPolicyPerfOutRandomDroppedPackets	Yes	RO
agentDiffServPolicyPerfOutShapeDelayedOctets	Yes	RO
agentDiffServPolicyPerfOutShapeDelayedPackets	Yes	RO
agentDiffServPolicyPerfOutSentOctets	Yes	RO
agentDiffServPolicyPerfOutSentPackets	Yes	RO
agentDiffServPolicyPerfOutHCTailDroppedOctets	Yes	RO
agentDiffServPolicyPerfOutHCTailDroppedPackets	Yes	RO
agentDiffServPolicyPerfOutHCRandomDroppedOctets	Yes	RO
agentDiffServPolicyPerfOutHCRandomDroppedPackets	Yes	RO
agentDiffServPolicyPerfOutHCShapeDelayedOctets	Yes	RO
agentDiffServPolicyPerfOutHCShapeDelayedPackets	Yes	RO
agentDiffServPolicyPerfOutHCSentOctets	Yes	RO
agentDiffServPolicyPerfOutHCSentPackets	Yes	RO
agentDiffServPolicyPerfOutStorageType	Yes	RO
agentDiffServPolicyPerfOutRowStatus	Yes	RO

TABLE C-29 FASTPATH QOS DiffServ MIB *(Continued)*

Object	Support	Access
agentDiffServServiceTable		
Indicies: agentDiffServServiceIfIndex, agentDiffServServiceIfDirection		
agentDiffServServicePolicyIndex	Yes	RC
agentDiffServServiceIfOperStatus	Yes	RO
agentDiffServServiceStorageType	Yes	RC
agentDiffServServiceRowStatus	Yes	RC
agentDiffServServicePerfTable		
Indicies: agentDiffServServiceIfIndex, agentDiffServServiceIfDirection		
agentDiffServServicePerfOfferedOctets	Yes	RO
agentDiffServServicePerfOfferedPackets	Yes	RO
agentDiffServServicePerfDiscardedOctets	Yes	RO
agentDiffServServicePerfDiscardedPackets	Yes	RO
agentDiffServServicePerfSentOctets	Yes	RO
agentDiffServServicePerfSentPackets	Yes	RO
agentDiffServServicePerfHCOfferedOctets	Yes	RO
agentDiffServServicePerfHCOfferedPackets	Yes	RO
agentDiffServServicePerfHCDiscardedOctets	Yes	RO
agentDiffServServicePerfHCDiscardedPackets	Yes	RO
agentDiffServServicePerfHCSentOctets	Yes	RO
agentDiffServServicePerfHCSentPackets	Yes	RO

TABLE C-30 FASTPATH QOS DiffServ Extensions MIB

Object	Support	Access
agentDiffServClassifier		
agentDiffServAuxMfClfrNextFree	Yes	RO

TABLE C-30 FASTPATH QOS DiffServ Extensions MIB *(Continued)*

Object	Support	Access
agentDiffServAuxMfClfrTable		
Index: agentDiffServAuxMfClfrId		
agentDiffServAuxMfClfrDstAddr	Yes	RO
agentDiffServAuxMfClfrDstMask	Yes	RO
agentDiffServAuxMfClfrSrcAddr	Yes	RO
agentDiffServAuxMfClfrSrcMask	Yes	RO
agentDiffServAuxMfClfrProtocol	Yes	RO
agentDiffServAuxMfClfrDstL4PortMin	Yes	RO
agentDiffServAuxMfClfrDstL4PortMax	Yes	RO
agentDiffServAuxMfClfrSrcL4PortMin	Yes	RO
agentDiffServAuxMfClfrSrcL4PortMax	Yes	RO
agentDiffServAuxMfClfrCos	Yes	RO
agentDiffServAuxMfClfrTos	Yes	RO
agentDiffServAuxMfClfrTosMask	Yes	RO
agentDiffServAuxMfClfrDstMac	Yes	RO
agentDiffServAuxMfClfrDstMacMask	Yes	RO
agentDiffServAuxMfClfrSrcMac	Yes	RO
agentDiffServAuxMfClfrSrcMacMask	Yes	RO
agentDiffServAuxMfClfrVlanId	Yes	RO
agentDiffServAuxMfClfrStorage	Yes	RO
agentDiffServAuxMfClfrStatus	Yes	RO
agentDiffServIpPrecMarkActTable		
Index: agentDiffServIpPrecMarkActPrecedence		
agentDiffServIpPrecMarkActPrecedence	Yes	RO
agentDiffServCosMarkActTable		
Index: agentDiffServCosMarkActCos		
agentDiffServCosMarkActCos	Yes	RO

TABLE C-31 FASTPATH QOS BW MIB

Object	Support	Access
trafficClassGroup		
trafficClassCreate	Yes	RW
trafficClassTable		
Index: trafficClassIndex		
trafficClassName	Yes	RO
trafficClassIfIndex	Yes	RW
trafficClassVlanId	Yes	RW
trafficClassWeight	Yes	RW
trafficClassBandwidthAllocation	Yes	RW
trafficClassAcceptByteCount	Yes	RO
trafficClassStatus	Yes	RW
bandwidthAllocationGroup		
bandwidthAllocationCreate	Yes	RW
bandwidthAllocationTable		
Index: bandwidthAllocationIndex		
bandwidthAllocationName	Yes	RO
bandwidthAllocationMinBandwidth	Yes	RW
bandwidthAllocationMaxBandwidth	Yes	RW
bandwidthAllocationStatus	Yes	RW

TABLE C-32 FASTPATH QOS ACL MIB

Object	Support	Access
aclTable		
Index: aclIndex		
aclStatus	Yes	RC
aclIfTable		
Indicies: aclIndex, aclIfIndex, aclIfDirection		
aclIfStatus	Yes	RC
aclRuleTable		
Indicies: aclIndex, aclRuleIndex		
aclRuleAction	Yes	RC
aclRuleProtocol	Yes	RC
aclRuleSrcIpAddress	Yes	RC
aclRuleSrcIpMask	Yes	RC
aclRuleSrcL4Port	Yes	RC
aclRuleSrcL4PortRangeStart	Yes	RC
aclRuleSrcL4PortRangeEnd	Yes	RC
aclRuleDestIpAddress	Yes	RC
aclRuleDestIpMask	Yes	RC
aclRuleDestL4Port	Yes	RC
aclRuleDestL4PortRangeStart	Yes	RC
aclRuleDestL4PortRangeEnd	Yes	RC
aclRuleIPDSCP	Yes	RC
aclRuleIpPrecedence	Yes	RC
aclRuleIpTosBits	Yes	RC
aclRuleIpTosMask	Yes	RC
aclRuleStatus	Yes	RC

TABLE C-33 FASTPATH-INVENTORY-MIB

Object	Support	Access
agentInventoryStackGroup		
agentInventoryStackReplicateConfig	Yes	RW
agentInventoryStackReplicateCode	Yes	RW
agentInventoryStackReplicateCodeStatus	Yes	RO
agentInventoryStackReplicateSTK	Yes	RW
agentInventorySupportedUnitTable		
Index: agentInventorySupportedUnitIndex		
agentInventorySupportedUnitModelIdentifier	Yes	RO
agentInventorySupportedUnitDescription	Yes	RO
agentInventorySupportedUnitExpectedCodeVer	Yes	RO
agentInventoryUnitTable		
Index: agentInventoryUnitNumber		
agentInventoryUnitAssignNumber	Yes	RC
agentInventoryUnitType	Yes	RO
agentInventoryUnitSupportedUnitIndex	Yes	RC
agentInventoryUnitMgmtAdmin	Yes	RC
agentInventoryUnitHWMgmtPref	Yes	RO
agentInventoryUnitHWMgmtPrefValue	Yes	RO
agentInventoryUnitAdminMgmtPref	Yes	RC
agentInventoryUnitAdminMgmtPrefValue	Yes	RC
agentInventoryUnitStatus	Yes	RO
agentInventoryUnitDetectedCodeVer	Yes	RO
agentInventoryUnitDetectedCodeInFlashVer	Yes	RO
agentInventoryUnitUpTime	Yes	RO
agentInventoryUnitDescription	Yes	RW
agentInventoryUnitReplicateSTK	Yes	RW
agentInventoryUnitRowStatus	Yes	RC

TABLE C-33 FASTPATH-INVENTORY-MIB *(Continued)*

Object	Support	Access
agentInventorySlotTable		
Indicies: agentInventoryUnitNumber, agentInventorySlotNumber		
agentInventorySlotStatus	Yes	RO
agentInventorySlotPowerMode	Yes	RW
agentInventorySlotAdminMode	Yes	RW
agentInventorySlotInsertedCardType	Yes	RO
agentInventorySlotConfiguredCardType	Yes	RW
agentInventorySlotCapabilities	Yes	RO
agentInventoryCardTypeTable		
Index: agentInventoryCardIndex		
agentInventoryCardType	Yes	RO
agentInventoryCardModelIdentifier	Yes	RO
agentInventoryCardDescription	Yes	RO

TABLE C-34 draft-ietf-idmr-dvmrp-mib-11 DVMRP MIB

Object	Support	Access
dvmrpscalar		
dvmrpVersionString	Yes	RO
dvmrpGenerationId	No	N/A
dvmrpNumRoutes	Yes	RO
dvmrpReachableRoutes	Yes	RO
dvmrpInterfaceTable		
Index: dvmrpInterfaceIfIndex		
dvmrpInterfaceLocalAddress	Yes	RO
dvmrpInterfaceMetric	Yes	RC
dvmrpInterfaceStatus	Yes	RC
dvmrpInterfaceRcvBadPkts	Yes	RO

TABLE C-34 `draft-ietf-idmr-dvmrp-mib-11` DVMRP MIB *(Continued)*

Object	Support	Access
`dvmrpInterfaceRcvBadRoutes`	Yes	RO
`dvmrpInterfaceSentRoutes`	Yes	RO
`dvmrpInterfaceInterfaceKey`	No	N/A
`dvmrpInterfaceInterfaceKeyVersion`	No	N/A
`dvmrpNeighborTable`		
Indicies: `dvmrpNeighborIfIndex`, `dvmrpNeighborAddress`		
`dvmrpNeighborUpTime`	Yes	RO
`dvmrpNeighborExpiryTime`	Yes	RO
`dvmrpNeighborGenerationId`	Yes	RO
`dvmrpNeighborMajorVersion`	Yes	RO
`dvmrpNeighborMinorVersion`	Yes	RO
`dvmrpNeighborCapabilities`	Yes	RO
`dvmrpNeighborRcvRoutes`	Yes	RO
`dvmrpNeighborRcvBadPkts`	Yes	RO
`dvmrpNeighborRcvBadRoutes`	Yes	RO
`dvmrpNeighborState`	Yes	RO
`dvmrpRouteTable`		
Indicies: `dvmrpRouteSource`, `dvmrpRouteSourceMask`		
`dvmrpRouteUpstreamNeighbor`	Yes	RO
`dvmrpRouteIfIndex`	Yes	RO
`dvmrpRouteMetric`	Yes	RO
`dvmrpRouteExpiryTime`	Yes	RO
`dvmrpRouteUpTime`	Yes	RO
`dvmrpRouteNextHopTable`		
Indicies: `dvmrpRouteNextHopSource`, `dvmrpRouteNextHopSourceMask`, `dvmrpRouteNextHopIfIndex`		
`dvmrpRouteNextHopType`	Yes	RO
`dvmrpPruneTable`		

TABLE C-34 `draft-ietf-idmr-dvmrp-mib-11` DVMRP MIB *(Continued)*

Object	Support	Access
Indicies: `dvmrpPruneGroup`, `dvmrpPruneSource`, `dvmrpPruneSourceMask`		
`dvmrpPruneExpiryTime`	Yes	RO
Traps		
`dvmrpNeighborLoss`	Yes	
`dvmrpNeighborNotPruning`	Yes	

TABLE C-35 RFC 3289 DiffServ MIB

Object	Support	Access
`diffServDataPathTable`		
Indicies: `ifIndex`, `diffServDataPathIfDirection`		
`diffServDataPathStart`	Yes	RO
`diffServDataPathStorage`	Yes	RO
`diffServDataPathStatus`	Yes	RO
`diffServClassifier`		
`diffServClfrNextFree`	Yes	RO
`diffServClfrElementNextFree`	Yes	RO
`diffServMultiFieldClfrNextFree`	Yes	RO
`diffServMeter`		
`diffServMeterNextFree`	Yes	RO
`diffServTBParam`		
`diffServTBParamNextFree`	Yes	RO
`diffServAction`		
`diffServActionNextFree`	Yes	RO

TABLE C-35 RFC 3289 DiffServ MIB *(Continued)*

Object	Support	Access
`diffServCountActNextFree`	Yes	RO
`diffServAlgDrop`		
`diffServAlgDropNextFree`	Yes	RO
`diffServRandomDropNextFree`	Yes	RO
`diffServQueue`		
`diffServQNextFree`	Yes	RO
`diffServScheduler`		
`diffServSchedulerNextFree`	Yes	RO
`diffServMinRateNextFree`	Yes	RO
`diffServMaxRateNextFree`	Yes	RO
`diffServClfrTable`		
Index: `diffServClfrId`		
`diffServClfrStorage`	Yes	RO
`diffServClfrStatus`	Yes	RO
`diffServClfrElementTable`		
Indicies: `diffServClfrId`, `diffServClfrElementId`		
`diffServClfrElementPrecedence`	Yes	RO
`diffServClfrElementNext`	Yes	RO
`diffServClfrElementSpecific`	Yes	RO
`diffServClfrElementStorage`	Yes	RO
`diffServClfrElementStatus`	Yes	RO
`diffServMultiFieldClfrTable`		

TABLE C-35 RFC 3289 DiffServ MIB *(Continued)*

Object	Support	Access
Index: diffServMultiFieldClfrId		
diffServMultiFieldClfrAddrType	Yes	RO
diffServMultiFieldClfrDstAddr	Yes	RO
diffServMultiFieldClfrDstPrefixLength	Yes	RO
diffServMultiFieldClfrSrcAddr	Yes	RO
diffServMultiFieldClfrSrcPrefixLength	Yes	RO
diffServMultiFieldClfrDscp	Yes	RO
diffServMultiFieldClfrFlowId	Yes	RO
diffServMultiFieldClfrProtocol	Yes	RO
diffServMultiFieldClfrDstL4PortMin	Yes	RO
diffServMultiFieldClfrDstL4PortMax	Yes	RO
diffServMultiFieldClfrSrcL4PortMin	Yes	RO
diffServMultiFieldClfrSrcL4PortMax	Yes	RO
diffServMultiFieldClfrStorage	Yes	RO
diffServMultiFieldClfrStatus	Yes	RO
diffServMeterTable		
Index: diffServMeterId		
diffServMeterSucceedNext	Yes	RO
diffServMeterFailNext	Yes	RO
diffServMeterSpecific	Yes	RO
diffServMeterStorage	Yes	RO
diffServMeterStatus	Yes	RO
diffServTBParamTable		
Index: diffServTBParamId		
diffServTBParamType	Yes	RO
diffServTBParamRate	Yes	RO
diffServTBParamBurstSize	Yes	RO
diffServTBParamInterval	Yes	RO

TABLE C-35 RFC 3289 DiffServ MIB *(Continued)*

Object	Support	Access
diffServTBParamStorage	Yes	RO
diffServTBParamStatus	Yes	RO
diffServActionTable		
Index: diffServActionId		
diffServActionInterface	Yes	RO
diffServActionNext	Yes	RO
diffServActionSpecific	Yes	RO
diffServActionStorage	Yes	RO
diffServActionStatus	Yes	RO
diffServDscpMarkActTable		
Index: diffServDscpMarkActDscp		
diffServDscpMarkActDscp	Yes	RO
diffServCountActTable		
Index: diffServCountActId		
diffServCountActOctets	Yes	RO
diffServCountActPkts	Yes	RO
diffServCountActStorage	Yes	RO
diffServCountActStatus	Yes	RO
diffServAlgDropTable		
Index: diffServAlgDropId		
diffServAlgDropType	Yes	RO
diffServAlgDropNext	Yes	RO
diffServAlgDropQMeasure	Yes	RO
diffServAlgDropQThreshold	Yes	RO
diffServAlgDropSpecific	Yes	RO

TABLE C-35 RFC 3289 DiffServ MIB *(Continued)*

Object	Support	Access
diffServAlgDropOctets	Yes	RO
diffServAlgDropPkts	Yes	RO
diffServAlgRandomDropOctets	Yes	RO
diffServAlgRandomDropPkts	Yes	RO
diffServAlgDropStorage	Yes	RO
diffServAlgDropStatus	Yes	RO
diffServRandomDropTable		
Index: diffServRandomDropId		
diffServRandomDropMinThreshBytes	Yes	RO
diffServRandomDropMinThreshPkts	Yes	RO
diffServRandomDropMaxThreshBytes	Yes	RO
diffServRandomDropMaxThreshPkts	Yes	RO
diffServRandomDropProbMax	Yes	RO
diffServRandomDropWeight	Yes	RO
diffServRandomDropSamplingRate	Yes	RO
diffServRandomDropStorage	Yes	RO
diffServRandomDropStatus	Yes	RO
diffServQTable		
Index: diffServQId		
diffServQNext	Yes	RO
diffServQMinRate	Yes	RO
diffServQMaxRate	Yes	RO
diffServQStorage	Yes	RO
diffServQStatus	Yes	RO
diffServSchedulerTable		
Index: diffServSchedulerId		
diffServSchedulerNext	Yes	RO
diffServSchedulerMethod	Yes	RO

TABLE C-35 RFC 3289 DiffServ MIB *(Continued)*

Object	Support	Access
diffServSchedulerMinRate	Yes	RO
diffServSchedulerMaxRate	Yes	RO
diffServSchedulerStorage	Yes	RO
diffServSchedulerStatus	Yes	RO
diffServMinRateTable		
Index: diffServMinRateId		
diffServMinRatePriority	Yes	RO
diffServMinRateAbsolute	Yes	RO
diffServMinRateRelative	Yes	RO
diffServMinRateStorage	Yes	RO
diffServMinRateStatus	Yes	RO
diffServMaxRateTable		
Index: diffServMaxRateId		
diffServMaxRateLevel	Yes	RO
diffServMaxRateAbsolute	Yes	RO
diffServMaxRateRelative	Yes	RO
diffServMaxRateThreshold	Yes	RO
diffServMaxRateStorage	Yes	RO
diffServMaxRateStatus	Yes	RO

APPENDIX **D**

Sensor Map and Fault Isolation

This appendix defines the sensors within the CT900 chassis and, where applicable, defines what system function(s) is/are affected when a sensor is triggered.

Chassis Sensors

TABLE D-1 Sensor Map

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition (impacted function)
0	FRU 0 HOT_SWAP	Discrete (0x6f), "Hot Swap" (0xf0)	Hot swap for Active ShMM	
2	FRU 1 HOT_SWAP	Discrete (0x6f), "Hot Swap" (0xf0)	Hotswap RTM.	N/A
3	FRU 2 HOT_SWAP	Discrete (0x6f), "Hot Swap" (0xf0)	Hotswap for Shelf EEPROM (redundant PROM).	
4	FRU 8 HOT_SWAP	Discrete (0x6f), "Hot Swap" (0xf0)	Hot swap for SAP	
5	FRU 3 HOT_SWAP	Discrete (0x6f), "Hot Swap" (0xf0)	Hot swap for Fan Tray 0	
6	FRU 4 HOT_SWAP	Discrete (0x6f), "Hot Swap" (0xf0)	Hot swap for Fan Tray 1	
7	FRU 5 HOT_SWAP	Discrete (0x6f), "Hot Swap" (0xf0)	Hot swap for Fan Tray 2	
8	FRU 6 HOT_SWAP	Discrete (0x6f), "Hot Swap" (0xf0)	Hot swap for PEM A	
9	FRU 7 HOT_SWAP	Discrete (0x6f), "Hot Swap" (0xf0)	Hot swap for PEM B	
10	IPMB LINK 1	Discrete (0x6f), "IPMB Link" (0xf1)	IPMI bus to slot 7 (address 41h). Redundant pair (IPMB_A and IPMB_B)	If both the IPMB-A and IPMB links are disabled the shelf manager will not communicate with the blade in slot 7.
11	IPMB LINK 2	Discrete (0x6f), "IPMB Link" (0xf1)	IPMI bus to slot 8 (address 42h). Redundant pair (IPMB_A and IPMB_B)	If both the IPMB-A and IPMB links are disabled the shelf manager will not communicate with the blade in slot 8.
12	Fan Tray 0	Discrete (0x6f), "Entity Presence" (0x25)	Fan tray 0 present.	Lack of fan tray compromises thermal integrity. All Fan trays must be installed.

TABLE D-1 Sensor Map *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition (impacted function)
13	Fan Tray 1	Discrete (0x6f), "Entity Presence" (0x25)	Fan tray 1 present.	Lack of fan tray compromises thermal integrity. All Fan trays must be installed.
14	Fan Tray 2	Discrete (0x6f), "Entity Presence" (0x25)	Fan tray 2 present.	Lack of fan tray compromises thermal integrity. All Fan trays must be installed.
15	IPMB LINK 3	Discrete (0x6f), "IPMB Link" (0xf1)	IPMI bus to slot 6 (address 43h). Redundant pair (IPMB_A and IPMB_B)	If both the IPMB-A and IPMB links are disabled the shelf manager will not communicate with the blade in slot 6.
16	IPMB LINK 4	Discrete (0x6f), "IPMB Link" (0xf1)	IPMI bus to slot 9 (address 44h). Redundant pair (IPMB_A and IPMB_B)	If both the IPMB-A and IPMB links are disabled the shelf manager will not communicate with the blade in slot 9.
17	IPMB LINK 5	Discrete (0x6f), "IPMB Link" (0xf1)	IPMI bus to slot 5 (address 45h). Redundant pair (IPMB_A and IPMB_B)	If both the IPMB-A and IPMB links are disabled the shelf manager will not communicate with the blade in slot 5.
18	IPMB LINK 6	Discrete (0x6f), "IPMB Link" (0xf1)	IPMI bus to slot 10 (address 46h). Redundant pair (IPMB_A and IPMB_B)	If both the IPMB-A and IPMB links are disabled the shelf manager will not communicate with the blade in slot 10.
19	IPMB LINK 7	Discrete (0x6f), "IPMB Link" (0xf1)	IPMI bus to slot 4 (address 47h). Redundant pair (IPMB_A and IPMB_B)	If both the IPMB-A and IPMB links are disabled the shelf manager will not communicate with the blade in slot 4.
20	IPMB LINK 8	Discrete (0x6f), "IPMB Link" (0xf1)	IPMI bus to slot 11 (address 48h). Redundant pair (IPMB_A and IPMB_B)	If both the IPMB-A and IPMB links are disabled the shelf manager will not communicate with the blade in slot 11.

TABLE D-1 Sensor Map *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition (impacted function)
21	IPMB LINK 9	Discrete (0x6f), "IPMB Link" (0xf1)	IPMI bus to slot 3 (address 49h). Redundant pair (IPMB_A and IPMB_B)	If both the IPMB-A and IPMB links are disabled the shelf manager will not communicate with the blade in slot 3.
22	IPMB LINK 10	Discrete (0x6f), "IPMB Link" (0xf1)	IPMI bus to slot 12 (address 4Ah). Redundant pair (IPMB_A and IPMB_B)	If both the IPMB-A and IPMB links are disabled the shelf manager will not communicate with the blade in slot 12.
23	IPMB LINK 11	Discrete (0x6f), "IPMB Link" (0xf1)	IPMI bus to slot 2 (address 4Bh). Redundant pair (IPMB_A and IPMB_B)	If both the IPMB-A and IPMB links are disabled the shelf manager will not communicate with the blade in slot 2.
24	IPMB LINK 12	Discrete (0x6f), "IPMB Link" (0xf1)	IPMI bus to slot 13 (address 4Ch). Redundant pair (IPMB_A and IPMB_B)	If both the IPMB-A and IPMB links are disabled the shelf manager will not communicate with the blade in slot 13.
25	IPMB LINK 13	Discrete (0x6f), "IPMB Link" (0xf1)	IPMI bus to slot 1 (address 4Dh). Redundant pair (IPMB_A and IPMB_B)	If both the IPMB-A and IPMB links are disabled the shelf manager will not communicate with the blade in slot 1.
26	IPMB LINK 14	Discrete (0x6f), "IPMB Link" (0xf1)	IPMI bus to slot 14 (address 4Eh). Redundant pair (IPMB_A and IPMB_B)	If both the IPMB-A and IPMB links are disabled the shelf manager will not communicate with the blade in slot 14.
27	IPMB LINK 15	Discrete (0x6f), "IPMB Link" (0xf1)	IPMI Backplane	
120	Center Exhaust	Threshold (0x01), "Temperature" (0x01)	Exhaust air temperature, Center	If the exhaust air temperature goes above the UNR threshold the blades may overheat.
121	Left Exhaust	Threshold (0x01), "Temperature" (0x01)	Exhaust air temperature, Left	If the exhaust air temperature goes above the UNR threshold the blades may overheat.

TABLE D-1 Sensor Map *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition (impacted function)
122	Right Exhaust	Threshold (0x01), "Temperature" (0x01)	Exhaust air temperature, Right	If the exhaust air temperature goes above the UNR threshold the blades may overheat.
123	SAP Temp	Threshold (0x01), "Temperature" (0x01)	Temperature sensor on SAP board	If the SAP air temperature goes above the UNR threshold the blades may overheat.
124	Temp_In Left	Threshold (0x01), "Temperature" (0x01)	Temperature of inlet air Left, Located in fan tray	If the intake air temperature goes above the UNR threshold the computer room airconditioning has failed.
125	Temp_In Center	Threshold (0x01), "Temperature" (0x01)	Temperature of inlet air Center, Located in fan tray	If the intake air temperature goes above the UNR threshold the computer room airconditioning has failed.
126	Temp_In Right	Threshold (0x01), "Temperature" (0x01)	Temperature of inlet air Right, Located in fan tray	If the intake air temperature goes above the UNR threshold the computer room airconditioning has failed.
131	TELCO Alarms	Discrete (0x6f), "OEM reserved" (0xdf)	Telco event occurred.	
132	BMC Watchdog	Discrete (0x6f), "Watchdog 2" (0x23)	ATCA IPMI watchdog.	
133	SYSTEM EVENT	Discrete (0x6f), "System Event" (0x12)	System reconfiguration event.	

TABLE D-1 Sensor Map *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition (impacted function)
135	FT Oper. Status	Management Subsystem Health (28h)	Current cooling state of the shelf: • 00h = Full Redundancy = all fan trays defined in the Address Table are operational • 01h = Redundancy Lost = some of the fan trays defined in the Address Table are missing or nonoperational. With the HPDL default cooling management strategy, this causes the fan level for all remaining fan trays to be set to their maximum.	

TABLE D-1 Sensor Map *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition (impacted function)
136	Cooling State	Management Subsystem Health (28h)	00h = transition to OK. The cooling state is Normal • 01h transition to Non-Critical from OK. The cooling state is now Minor Alert, the previous cooling state was Normal. • 02h transition to Critical from less severe. The cooling state is now Major Alert, the previous cooling state was either Normal or Minor Alert. • 04h transition to Non-Critical from more severe. The cooling state is now Minor Alert, the previous cooling state was either Major or Critical Alert. • 05h transition to Critical from Non-recoverable. The current cooling state is Major Alert, the previous cooling state was Critical Alert. • 06h transition to Nonrecoverable. The current cooling state is now Critical Alert.	

TABLE D-1 Sensor Map *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition (impacted function)
137	Fans State	Management Subsystem Health (28h)	• 00h = transition to OK. The fans state is Normal (no thresholds are crossed on fan tachometer sensors). • 01h = transition to Non-Critical from OK. The fans state is now Minor Alert (non-critical thresholds are crossed for some tachometer.	
150	Air Filter	Discrete (0x6f), "Entity Presence" (0x25)	Air filter presence sensor.	If the air filter is not present the blades will get dirty and may overheat.
152	SAP	Discrete (0x6f), "Entity Presence" (0x25)	SAP presence.	With no SAP, there will be no Telco alarms. SAP temperature and exhaust temperatures not available with SAP missing.
162	PEM A In 2	Discrete (0x6f), "Entity Presence" (0x25)	PEM A input 2, before fuse.	If failed, FRUs powered by input 2 will not have power redundancy.
163	PEM A In 2 Fused	Discrete (0x6f), "Entity Presence" (0x25)	PEM A input 2, after fuse.	Sensor #162 and 163 can be used to determine if fuse is failed or if input is not connected. See Table 2.
164	PEM A In 1	Discrete (0x6f), "Entity Presence" (0x25)	PEM A input 1, before fuse.	If failed, FRUs powered by input 1 will not have power redundancy.
165	PEM A In 1 Fused	Discrete (0x6f), "Entity Presence" (0x25)	PEM A input 1, after fuse.	Sensor #164 and 165 can be used to determine if fuse is failed or if input is not connected. See Table 2.
166	PEM A In 4	Discrete (0x6f), "Entity Presence" (0x25)	PEM A input 4, before fuse.	If failed, FRUs powered by input 4 will not have power redundancy.

TABLE D-1 Sensor Map *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition (impacted function)
167	PEM A In 4 Fused	Discrete (0x6f), "Entity Presence" (0x25)	PEM A input 4, after fuse.	Sensor #166 and 167 can be used to determine if fuse is failed or if input is not connected. See Table 2.
168	PEM A In 3	Discrete (0x6f), "Entity Presence" (0x25)	PEM A input 3, before fuse.	If failed, FRUs powered by input 3 will not have power redundancy.
169	PEM A In 3 Fused	Discrete (0x6f), "Entity Presence" (0x25)	PEM A input 3, after fuse.	Sensor #168 and 169 can be used to determine if fuse is failed or if input is not connected. See Table 2.
174	PEM B In 2	Discrete (0x6f), "Entity Presence" (0x25)	PEM B input 2, before fuse.	If failed, FRUs powered by input 2 will not have power redundancy.
175	PEM B In 2 Fused	Discrete (0x6f), "Entity Presence" (0x25)	PEM B input 2, after fuse.	Sensor #174 and 175 can be used to determine if fuse is failed or if input is not connected. See Table 2.
176	PEM B In 1	Discrete (0x6f), "Entity Presence" (0x25)	PEM B input 1, before fuse.	If failed, FRUs powered by input 1 will not have power redundancy.
177	PEM B In 1 Fused	Discrete (0x6f), "Entity Presence" (0x25)	PEM B input 1, after fuse.	Sensor #176 and 177 can be used to determine if fuse is failed or if input is not connected. See Table 2.
178	PEM B In 4	Discrete (0x6f), "Entity Presence" (0x25)	PEM B input 4, before fuse.	If failed, FRUs powered by input 4 will not have power redundancy.
179	PEM B In 4 Fused	Discrete (0x6f), "Entity Presence" (0x25)	PEM B input 4, after fuse.	Sensor #178 and 179 can be used to determine if fuse is failed or if input is not connected. See Table 2.
180	PEM B In 3	Discrete (0x6f), "Entity Presence" (0x25)	PEM B input 3, before fuse.	If failed, FRUs powered by input 3 will not have power redundancy.

TABLE D-1 Sensor Map *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition (impacted function)
181	PEM B In 3 Fused	Discrete (0x6f), "Entity Presence" (0x25)	PEM B input 3, after fuse.	Sensor #180 and 181 can be used to determine if fuse is failed or if input is not connected. See Table 2.
192	PEM A	Discrete (0x6f), "Entity Presence" (0x25)	PEM A present.	If no PEM A, then PEM B will power up system.
193	PEM B	Discrete (0x6f), "Entity Presence" (0x25)	PEM B present.	If no PEM B, then PEM A will power up system.
194	Shelf EEPROM 1	Discrete (0x6f), "Entity Presence" (0x25)		
195	Shelf EEPROM 2	Discrete (0x6f), "Entity Presence" (0x25)		
200	PEM A Temp	Threshold (0x01), "Temperature" (0x01)	Temperature in PEM A	If the PEM temperature goes above the UNR threshold there is a cooling problem.
201	PEM B Temp	Threshold (0x01), "Temperature" (0x01)	Temperature in PEM B	If the PEM temperature goes above the UNR threshold there is a cooling problem.
208	24V FT 0	Discrete (0x6f), "Entity Presence" (0x25)	Output of 24V DC-DC converter OK	If the 24V DC-DC fails, fans in FT0 will fail
209	-48A bus FT 0	Discrete (0x6f), "Entity Presence" (0x25)	FT 0 A input, before fuse	If both A and B feeds are missing then the 24V DC-DC converter will not operate.
210	-48A FT 0	Discrete (0x6f), "Entity Presence" (0x25)	FT 0 A input, after fuse	If both A and B feeds are missing then the 24V DC-DC converter will not operate.
211	-48B bus FT 0	Discrete (0x6f), "Entity Presence" (0x25)	FT 0 B input, before fuse	If both A and B feeds are missing then the 24V DC-DC converter will not operate.
212	-48B FT 0	Discrete (0x6f), "Entity Presence" (0x25)	FT 0 B input, after fuse	If both A and B feeds are missing then the 24V DC-DC converter will not operate.

TABLE D-1 Sensor Map *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition (impacted function)
213	-48A FT 0 Fuse	Discrete (0x6f), "Entity Presence" (0x25)	FT 0 A input fuse	Sensor #209 and 210 are used to determine if fuse has failed or if input is not present.
214	-48B FT 0 Fuse	Discrete (0x6f), "Entity Presence" (0x25)	FT 0 B input fuse	Sensor #211 and 212 are used to determine if fuse has failed or if input is not present.
215	24V FT 1	Discrete (0x6f), "Entity Presence" (0x25)	Output of 24V DC-DC converter OK	If the 24V DC-DC fails, fans in FT1 will fail
216	-48A bus FT 1	Discrete (0x6f), "Entity Presence" (0x25)	FT 1 A input, before fuse	If both A and B feeds are missing then the 24V DC-DC converter will not operate.
217	-48A FT 1	Discrete (0x6f), "Entity Presence" (0x25)	FT 1 A input, after fuse	If both A and B feeds are missing then the 24V DC-DC converter will not operate.
218	-48B bus FT 1	Discrete (0x6f), "Entity Presence" (0x25)	FT 1 B input, before fuse	If both A and B feeds are missing then the 24V DC-DC converter will not operate.
219	-48B FT 1	Discrete (0x6f), "Entity Presence" (0x25)	FT 1 B input, after fuse	If both A and B feeds are missing then the 24V DC-DC converter will not operate.
220	-48A FT 1 Fuse	Discrete (0x6f), "Entity Presence" (0x25)	FT 1 A input fuse	Sensor #209 and 210 are used to determine if fuse has failed or if input is not present.
221	-48B FT 1 Fuse	Discrete (0x6f), "Entity Presence" (0x25)	FT 1 B input fuse	Sensor #211 and 212 are used to determine if fuse has failed or if input is not present.
222	24V FT 2	Discrete (0x6f), "Entity Presence" (0x25)	Output of 24V DC-DC converter OK	If the 24V DC-DC fails, fans in FT2 will fail
223	-48A bus FT 2	Discrete (0x6f), "Entity Presence" (0x25)	FT 2 A input, before fuse	If both A and B feeds are missing then the 24V DC-DC converter will not operate.

TABLE D-1 Sensor Map *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition (impacted function)
224	-48A FT 2	Discrete (0x6f), "Entity Presence" (0x25)	FT 2 A input, after fuse	If both A and B feeds are missing then the 24V DC-DC converter will not operate.
225	-48B bus FT 2	Discrete (0x6f), "Entity Presence" (0x25)	FT 2 B input, before fuse	If both A and B feeds are missing then the 24V DC-DC converter will not operate.
226	-48B FT 2	Discrete (0x6f), "Entity Presence" (0x25)	FT 2 B input, after fuse	If both A and B feeds are missing then the 24V DC-DC converter will not operate.
227	-48A FT 2 Fuse	Discrete (0x6f), "Entity Presence" (0x25)	FT 2 A input fuse	Sensor #209 and 210 are used to determine if fuse has failed or if input is not present.
228	-48B FT 2 Fuse	Discrete (0x6f), "Entity Presence" (0x25)	FT 2 B input fuse	Sensor #211 and 212 are used to determine if fuse has failed or if input is not present.
244	3V3_RAD	Discrete (0x6f), "Entity Presence" (0x25)	Power to the Radial IPMB circuitry.	Indicates a failure of both I2C-A and I2C-B power supplies. The Radial IPMB circuitry on the shelf manager carrier board will not work.

FIGURE D-1 Chassis Level Sensor Locations - Front

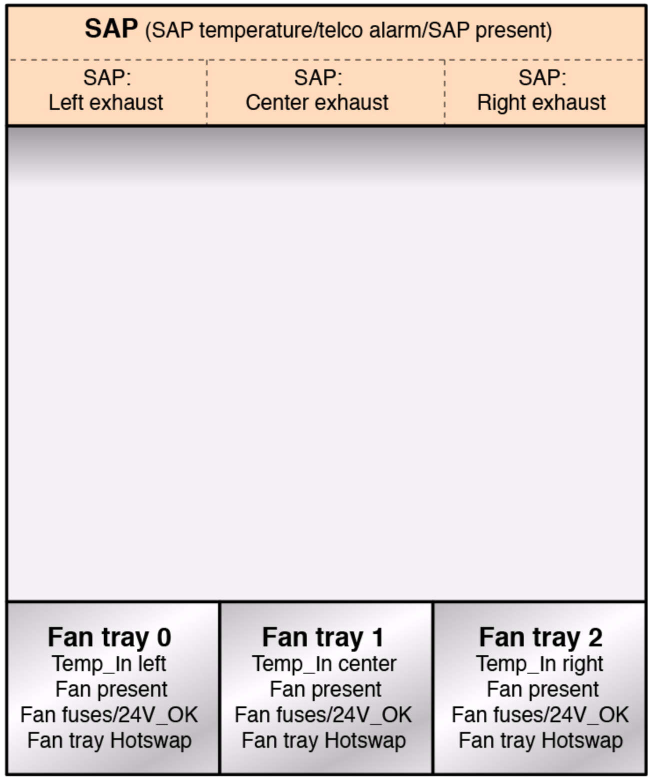
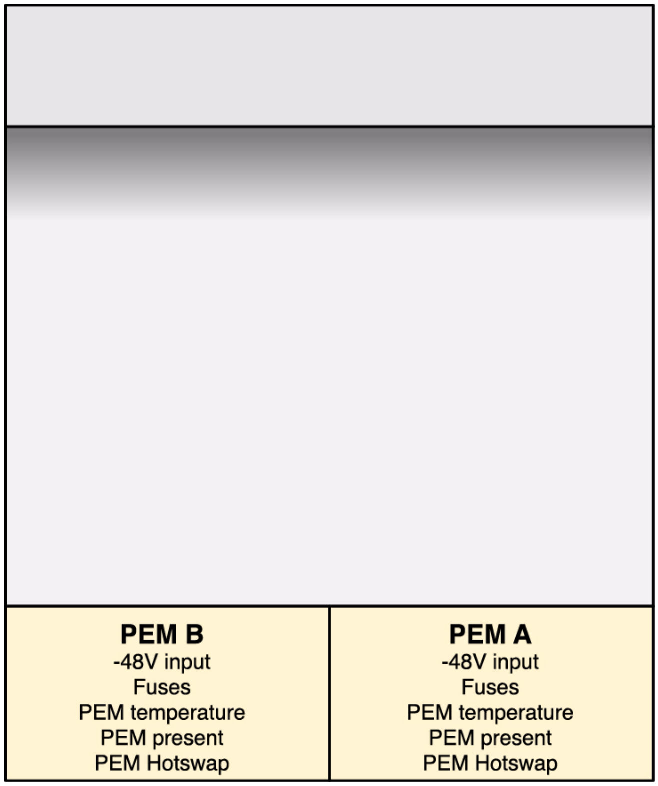
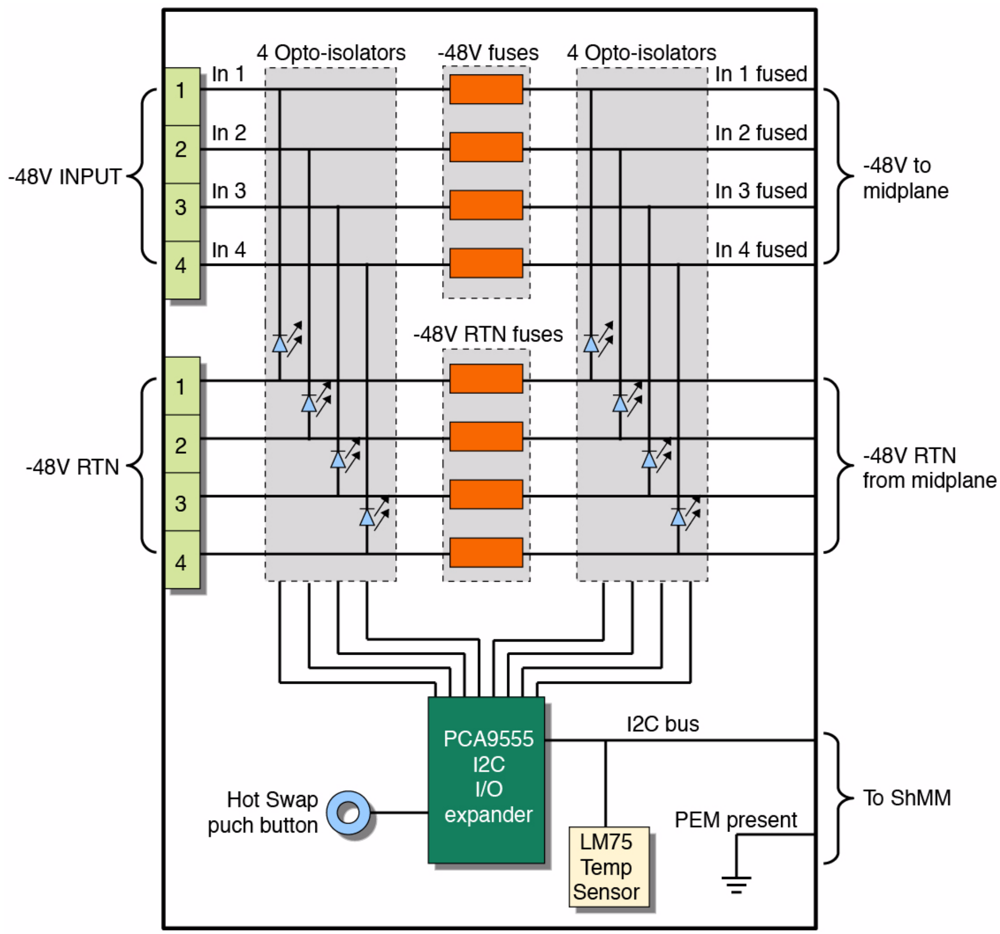


FIGURE D-2 Chassis Level Sensor Locations - Rear



PEM Sensors

FIGURE D-3 PEM Sensors



PEM Sensor Fault Interpretation

TABLE D-2 PEM Sensor Fault Interpretation

-48V INPUT	-48V Fuse	-48V RTN Fuse	-48V RTN	Input 1	Input 1 Fused
Present	OK	OK	Present	1	1
Present	OK	OK	Missing	0	0
Present	OK	Blown	Present	1	0
Present	OK	Blown	Missing	0	0
Present	Blown	OK	Present	1	0
Present	Blown	OK	Missing	0	0
Present	Blown	Blown	Present	1	0
Present	Blown	Blown	Missing	0	0
Missing	Don't Care	Don't Care	Don't Care	0	0

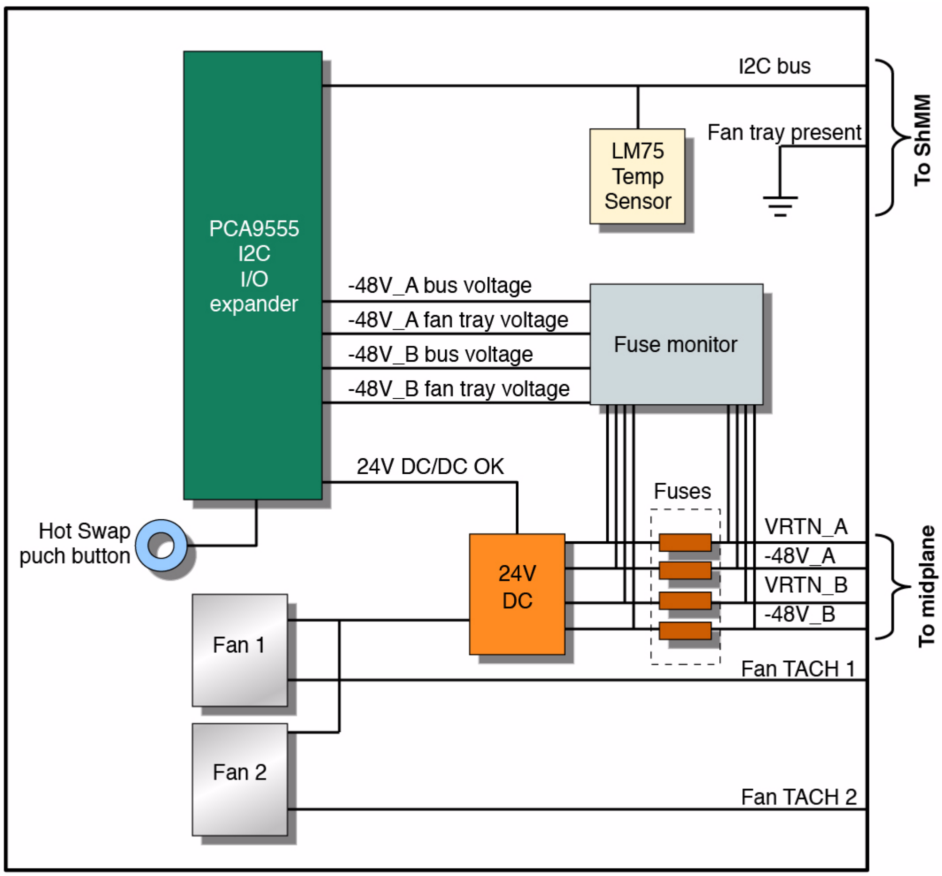
TABLE D-3 Fault Condition Interpretation for Input 1

In 1	In 1 Fused	Fault conditions
0	0	One or more of the following faults: -48V RTN missing;-48V RTN Fuse blown;48V Fuse blown; -48V INPUT missing
0	1	Not valid
1	0	One or more of the following faults: -48V RTN Fuse blown; -48V Fuse blown; -48V RTN Fuse blown
1	1	No fault

Note – If the same input on both PEMs fails, then selected slots, fan trays, and/or the Shelf Manager will be affected. Refer to the *CT900 Hardware System Specification* for definitions of which input feed supplies power to system components. This failure will only occur upon a double fault.

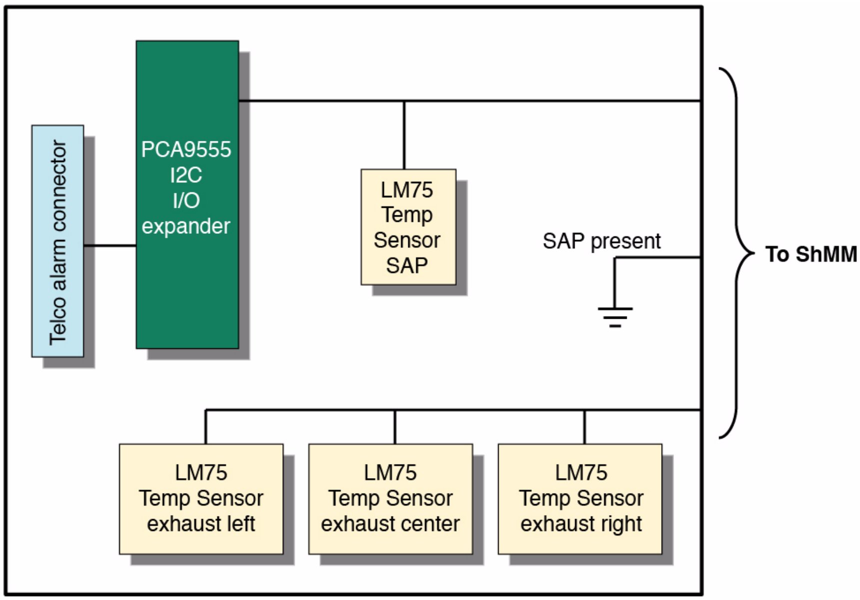
Fan Tray Sensors

FIGURE D-4 Fan Tray Sensors



SAP Sensors

FIGURE D-5 SAP Sensors



APPENDIX **E**

ShMM Sensor Map and Fault Isolation

This appendix defines the sensors within the CT900 ShMM card and, where applicable, defines what system function(s) is/are affected when a sensor is triggered.

ShMM Sensors

TABLE E-1 ShMM Sensor Map

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition (impacted function)
0	FRU 0 HOT_SWAP	Discrete (0x6f), "Hot Swap" (0xf0)	Hot swap for Active ShMM	
1	IPMB LINK	Discrete (0x6f), "IPMB Link" (0xf1)	IPMB link for ShMM carrier	Communication will fail
2	Local Temp	Threshold (0x01), "Temperature" (0x01)	Local temperature sensor	If the local temperature goes above the UNR threshold the ShMM carrier may overheat.
3	3V3_local	Threshold (0x01), "Voltage" (0x02)	Standby 3.3V signal on the ShMM carrier	Event will be logged
4	I2C_PWR_A	Threshold (0x01), Voltage" (0x02)	12V signal on the ShM carrier (I2C A power)	Event will be logged
5	I2C_PWR_B	Threshold (0x01), "Voltage" (0x02)	5V signal on the ShMM carrier (I2C B power)	Event will be logged
6	VBAT	Threshold (0x01), "Voltage" (0x02)	VBAT signal on the ShMMcarrier	Event will be logged
7	Fan Tach. 0	Threshold (0x01), "Fan" (0x04)	FT0 tach sensor for fan 1	Fan Fault will occur and FT red/SAP LED will be lit if threshold crossed.
8	Fan Tach. 1	Threshold (0x01), "Fan" (0x04)	FT0 tach sensor for fan 2	Fan Fault will occur and FT red/SAP LED will be lit if threshold crossed.
10	Fan Tach. 2	Threshold (0x01), "Fan" (0x04)	FT1 tach sensor for fan 1	Fan Fault will occur and FT red/SAP LED will be lit if threshold crossed.
11	Fan Tach. 3	Threshold (0x01), "Fan" (0x04)	FT1 tach sensor for fan 2	Fan Fault will occur and FT red/SAP LED will be lit if threshold crossed.
13	Fan Tach. 4	Threshold (0x01), "Fan" (0x04)	FT2 tach sensor for fan 1	Fan Fault will occur and FT red/SAP LED will be lit if threshold crossed.
14	Fan Tach. 5	Threshold (0x01), "Fan" (0x04)	FT2 tach sensor for fan 2	Fan Fault will occur and FT red/SAP LED will be lit if threshold crossed.

TABLE E-1 ShMM Sensor Map *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition (impacted function)
15	-48A Bus voltage	Discrete (0x6f), "Entity Presence" (0x25)	GPIO 12 presence on the carrier	Used to find if input is present or not.
16	-48B Bus voltage	Discrete (0x6f), "Entity Presence" (0x25)	GPIO 13 presence on the carrier	Used to find if input is present or not.
17	-48A ACB voltage	Discrete (0x6f), "Entity Presence" (0x25)	GPIO 14 presence on the carrier	Used to find if input is present or not.
18	-48B ACB voltage	Discrete (0x6f), "Entity Presence" (0x25)	GPIO 15 presence on the carrier	Used to find if input is present or not.
19	-48A ACB Fuse	Discrete (0x6f), "Entity Presence" (0x25)	GPIO12 || GPIO 14 This not mapped to a specific signal but describes the health on 48V A line	Used to find if input is present or not.

TABLE E-1 ShmM Sensor Map *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition (impacted function)
20	-48B ACB Fuse	Discrete (0x6f), "Entity Presence" (0x25)	GPIO13 || GPIO 15 This not mapped to a specific signal but describes the health on 48V	Used to find if input is present or not.
128	CPLD State	Discrete (0x6f), "OEM reserved" (0xde)	CPLD state sensor: • 0002h – Local Healthy • 0004h – Switchover Request Local • 0010h – Switchover Status LED 1 • 0200h – Remote Healthy (Status of other ShMM; 1 = healthy, 0 = not healthy) • 1000h – Local Presence (Status of ShMM; 1 = connected, 0 = not connected) • 2000h - Active	Events generated when ShMM CPLD state changes (including redundancy state changes).

TABLE E-1 ShMM Sensor Map *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition (impacted function)
129	Reboot Reason	OEM Reserved (0xdd)	The state mask for the sensor indicates the cause of the last reboot. (The sensor reading is always 0 and does not have any meaning.) • [1] The reboot was caused by a switchover operation. • [2] The reboot was caused by a forced switchover operation. • [3] The reboot was caused by the CLI command terminate. • [4] The reboot was caused by loss of the HEALTHY bit. • [5] The reboot was caused by loss of the ACTIVE bit. • [6] The reboot of the Backup ShMM happened because the redundancy connection was broken but the Active ShMM was still alive. • [7] The reboot happened due to an error during the Shelf Manager startup. • [8] The reboot was caused by the ShMM hardware watchdog. • [9] The reboot was initiated by software (a reboot() system call). • [10] The ShMM has been power cycled.	

APPENDIX **F**

Sun Netra CP3020 Blade Server Sensor Map and Fault Isolation

This appendix defines the sensors for the Sun Netra CP3020 blade server.

Sun Netra CP3020 Blade Server Sensor List

The Sun Netra CP3020 sensor numbers and names are reported by the on-blade server H8 chip.

TABLE F-1 Sun Netra CP3020 Blade Server Sensors

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
0	FRU 0 Hot Swap	Discrete (0x6f), "Hot Swap" (0xf0)	Hot swap for blade server FRU	N/A
1	RTM Hot Swap	Discrete(0x6F), Hotswap(0xf0)	Hotswap sensor for RTM	N/A
2	IPMB Physical	Discrete (0x6f), "IPMB Link" (0xf1)	Link Status of IPMB	No reply from IPMB (A or B). State of IPMB A or B bus is reported by monitoring the READY signal on the IPMB isolator.
3	BMC Watchdog	Discrete (0x6f), "Watchdog 2" (0x23)	Watchdog state of BMC	N/A
4	CPU Tcontrol (Max Normal Value = 70)	Threshold (0x01), "Temperature" (0x01)	Blade server temperature: Case temperature of Opteron CPU. Device =ADM 1026, U153 pin 25/26	IPMB isolator not ready. If this temperature goes above 75C, the H8 will shut down all power supplies and turn on the front panel OOS LED.
5	Blade Server Inlet Temp	Threshold (0x01), "Temperature" (0x01)	Bblade server temperature: Ambient @ blade server inlet. Sensor located @ bottom edge of blade server near power brick. Device =ADM 1026, U153 pin 27/28	If this temperature goes above 60C, the H8 will shut down all power supplies and turn on the front panel OOS LED.
6	ADM Internal Temp	Threshold (0x01), "Temperature" (0x01)	Blade server temperature: Ambient temp @ blade server exit. Device =ADM 1026, U153 Internal	If this temperature goes above 68C, the H8 will shut down all power supplies and all front panel LEDs are turned off.

TABLE F-1 Sun Netra CP3020 Blade Server Sensors *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
7	+12.0V Run	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of +12.0V power rail Device =ADM 1026, U153 pin 32	If this voltage is out of spec all other power rails will fail (Except STBY). Blade server and the RTM will not function. The H8 will be alive if 3.3V STBY is present and in spec. This rail is the power source for all DC/DC converters
8	-12.0V Run	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of -12.0V power rail Device =ADM 1026, U153 pin 31	This is the -12V power rail to the PMC slots. If this voltage is out of spec any installed PMC may not function.
9	VCC 5.0V Run	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 5.0V power rail. Device = ADM 1026, U153 pin 30	This voltage is one of the power sources for the memory VRM, 1.2V converter, PMC cards, BIOS chip, SAS HDD's, and ethernet. If this rail is out of spec the blade server will not function. The H8 will be alive if 3.3V STBY is present and in spec
10	+3.3V Run	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 3.3V power rail. Device = ADM 1026, U153 pin 7	This voltage is one of the power sources for the processor, 8132 I/O, pull-up resistors, and reset logic. If this rail is out of spec the Blade server will not function. The H8 will be alive if 3.3V STBY is present and in spec
11	+3.3V ALW	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 3.3V STBY power rail. Device =ADM 1026, U153 pin 22	If this voltage is out of spec Blade server and the H8 will not function. This rail is the power source for most of the components on the blade server including all I2C devices and the H8.

TABLE F-1 Sun Netra CP3020 Blade Server Sensors *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
12	VCC RTC	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 3.0 VBAT power rail. Device =ADM 1026, U153 pin 29	If this voltage is out of spec or goes to zero, the on blade server battery is bad or missing. The battery is not required for normal operation of the blade server while installed in the chassis and the -48V power source is applied. The function of the battery is for back up power to the CMOS and RTC when input power is removed or blade server is removed from the chassis..
13	VDD Core Run	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VCC 1.15V M Dual power rail (3.3V run + 3.3V STBY). Device = ADM 1026, U153 pin 33	This voltage is one of the power sources for the processor. If this rail is out of spec the blade server will not function. The H8 will be alive if 3.3V STBY is present and in spec.
14	VCC 1.8V Dual	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 1.8V CPU power rail. Device =ADM 1026, U153 pin 34	This voltage is one of the power sources for the AMD 8111 I/O Hub. If this rail is out of spec the blade server will not function. The H8 will be alive if 3.3V STBY is present and in spec
15	DDR VTT 1.3V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VCC 1.3V power rail. Device =ADM 1026, U153 pin35	This power rail provides termination voltage for main memory. If this rail is out of spec the blade server will not function. The H8 will be alive if 3.3V STBY is present and in spec
16	VCC 1.2V Run	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VCC 1.2V power rail. Device =ADM 1026, U153 pin 36	This voltage is one of the power sources for the processor, the the 1064 SAS controller and provides power to various pull-ups. If this rail is out of spec the blade server will not function. The H8 will be alive if 3.3V STBY is present and in spec

TABLE F-1 Sun Netra CP3020 Blade Server Sensors *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
17	VCC_5V_ALW	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VCC 5V Always power rail (sensor reads 1/2 value but F/W reports 2X sensor read value). Device =ADM 1026, U153 pin 38	This power rail enables the 5V & 3.3 run rails as well as ref V to several POKs. If this voltage is out of spec or goes to zero, blade server will not function. The H8 will be alive if 3.3V STBY is present and in spec.
18	VDD 2.5V PU Run	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VDD 2.5V Run power rail. Device =ADM 1026, U153 pin39	This power rail provides termination voltage for several critical processor signals. If this rail is out of spec the blade server might not function. The H8 will be alive if 3.3V STBY is present and in spec
19	DDR VDD 2.6V Run	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 2.6 V power rail. Device =ADM 1026, U153 pin 40	This voltage is one of the power sources for the CPU memory controller and memory. If this rail is out of spec the memory will not function. The H8 will be alive if 3.3V STBY is present and in spec
20	VCC 1V8 Run	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VCC 1.8V Run power rail. Device =ADM 1026, U153 pin 41	This voltage is one of the power sources for the AMD8111 I/O hub. If this rail is out of spec the blade server will not function. The H8 will be alive if 3.3V STBY is present and in spec

TABLE F-1 Sun Netra CP3020 Blade Server Sensors *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
21	System Event	Discrete (0x6f), "System Event" (0x12)		This sensor reports IPMC reset event to ShMM. This sensor lets NetConsole application know that IPMC has taken a reset and the NetConsole session has to be restarted.
22	RTM Presence	Discrete (0x6f), "Entity Presence" (0x25)		This sensor indicates the presence of an RTM.
23	Version Change	Discrete (0x6f), "Reserved" (0x2b) Belongs to entity: (0x3, 96) [FRU # 0]		IPMC reports event after the FW update/cold reset.

FIGURE F-1 Netra CP3020 Voltage Distribution and H8 Sensor Mapping

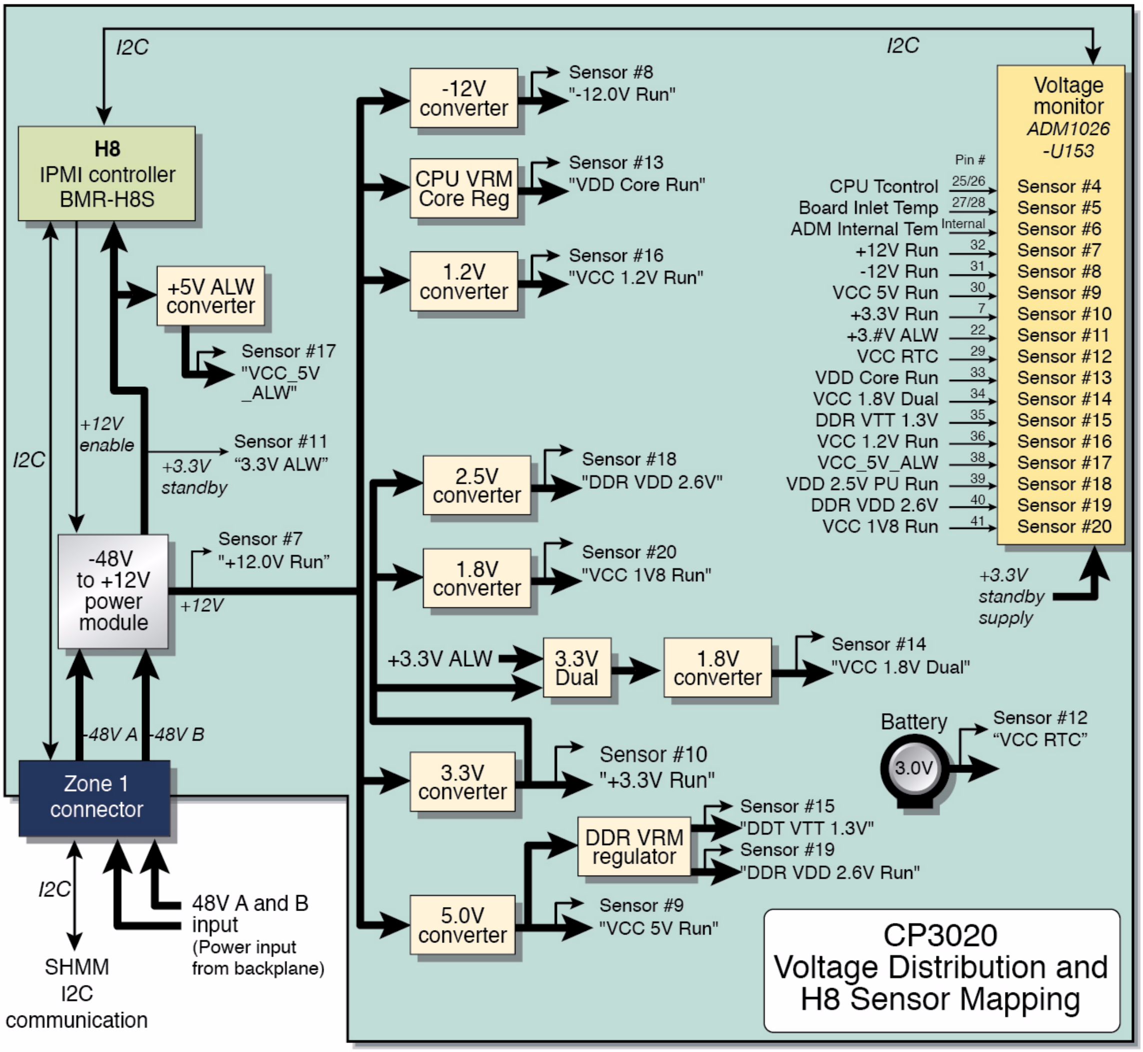
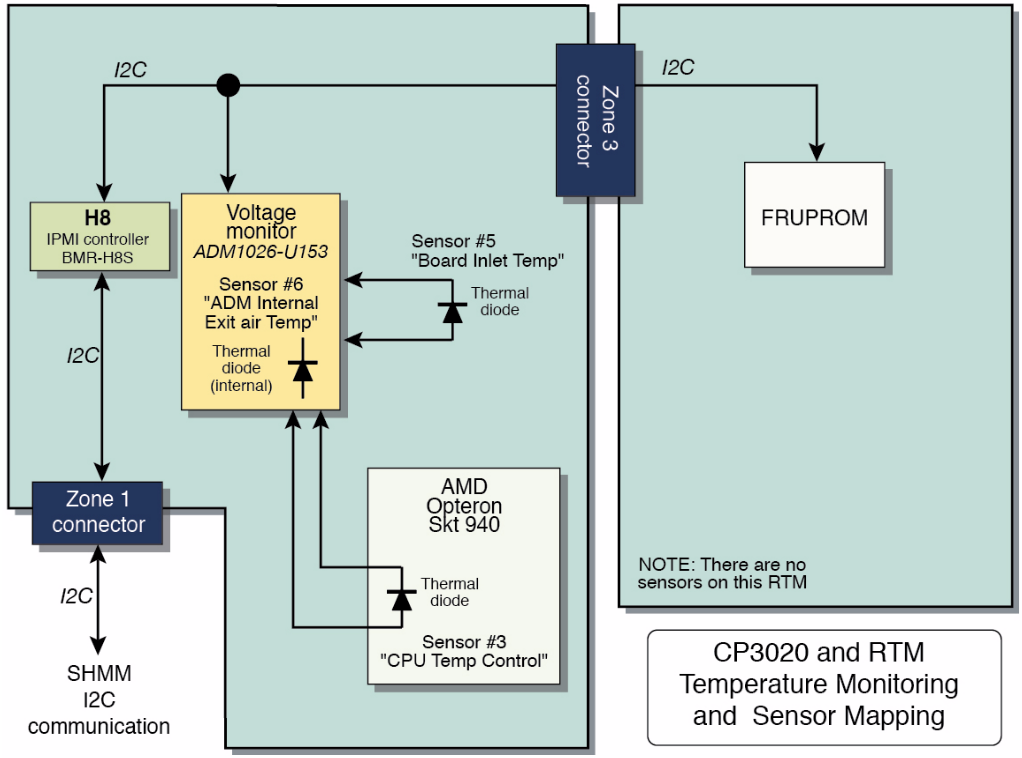


FIGURE F-2 Sun Netra CP3020 Blade Server and RTM Temperature Monitoring and H8 Sensor Mapping



APPENDIX **G**

Sun Netra CP3220 Blade Server Sensor Map and Fault Isolation

This appendix defines the sensors for the Sun Netra CP3220 blade server.

Sun Netra CP3220 Blade Server Sensor List

The Sun Netra CP3220 sensor numbers and names are reported by the on-board H8 processor via the ShMM within the ATCA chassis.

TABLE G-1 Sun Netra CP3220 Blade Server Sensors

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
0	FRU 0 Hot Swap	Discrete (0x6f), "Hot Swap" (0xf0)	Hot swap for CP3220 FRU	N/A
1	AMC 0 Hotswap	Discrete (0x6f), "Hot Swap" (0xf0)	Hot swap for AMC 0 (Bay B1)	N/A
2	AMC 1 Hotswap	Discrete (0x6f), "Hot Swap" (0xf0)	Hot swap for AMC 1 (Bay B2)	N/A
3	ARTM HotSwap	Discrete (0x6f), "Hot Swap" (0xf0)	Hot swap for ARTM FRU	N/A
4	IPMB Physical	Discrete (0x6f), "IPMB Link" (0xf1)	Link Status of IPMB	IPMB isolator not ready. State of IPMB A or B bus is reported by monitoring the READY signal on the IPMB isolator.
5	BMC Watchdog	Discrete (0x6f), "Watchdog 2" (0x23)	Watchdog state of BMC	N/A
6	CPU Case Temp	Threshold (0x01), "Temperature" (0x01)	CP3220 component temperature: Case temperature of Opteron CPU. Device = ADM 1026, U60 pin 25/26	If this temperature goes above 86C, the H8 will shut down all power supplies and turn on the front panel OOS LED.
7	Zone-3 Temp	“Temperature" (0x01)	Blade server temperature: Ambient @ top of the blade server near the Zone 3 Connector. Device = ADM 1026, U60 pin 27/28	There is no fault condition for this sensor; it is for information purposes only.

TABLE G-1 Sun Netra CP3220 Blade Server Sensors *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
8	AMC Area Temp	"Temperature" (0x01)	Blade server temperature: Ambient @ top of the blade server sensing the temp of AMC 0. Device = ADM 1026, U60 Internal	There is no fault condition for this sensor; it is for information purposes only.
9	12.0V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 12.0V power rail. Device = ADM 1026, U60 pin 32	If this voltage is out of spec all other power rails will fail (Except STBY). CP3220 and the ARTM will not function. The H8 will be alive if 3.3V STBY is present and in spec. This rail is the power source for all DC/DC converters
10	5.0V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 5.0V power rail. Device = ADM 1026, U60 pin 30	This voltage is one of the power sources for the processor, Nvidia I/O, USB, and CPLD. If this rail is out of spec the CP3220 will not function. The H8 will be alive if 3.3V STBY is present and in spec
11	3.3V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 3.3V power rail. Device = ADM 1026, U60 pin 7	This voltage is one of the power sources for the processor, Nvidia I/O, pull-up resistors, BIOS, and reset logic. If this rail is out of spec the CP3220 will not function. The H8 will be alive if 3.3V STBY is present and in spec
12	3.3V STBY	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 3.3V STBY power rail. Device = ADM 1026, U60 pin 22	If this voltage is out of spec CP3220 and the H8 will not function. This rail is the power source for most of the components on the CP3220 including all I2C devices and the H8.

TABLE G-1 Sun Netra CP3220 Blade Server Sensors *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
13	Battery Voltage	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 3.0 VBAT/STBY power rail. Device = ADM 1026, U60 pin 29	If this voltage is out of spec or goes to zero, the on blade server battery is bad or missing. The battery is not required for normal operation of the CP3220 while installed in the chassis and the -48V power source is applied. The function of the battery is for back up power to the CMOS and RTC when input power is removed or CP3220 is removed from the chassis.
14	VCC 1.15V M Dual	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VCC 1.15V M Dual power rail (3.3V run + 3.3V STBY). Device = ADM 1026, U60 pin 34	If this voltage is out of spec the I/O section on the NVIDIA MCP55 will not function. The H8 will be alive if 3.3V STBY is up. This rail is the power source for the I/O section of the NVIDIA MCP55 PRO
15	Proc0 0.9V DDR	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 0.9V CPU power rail. Device = ADM 1026, U60 pin 35	This voltage is one of the power sources for the processor memory controller as well as mem term power. If this rail is out of spec the CP3220 will not function. The H8 will be alive if 3.3V STBY is present and in spec
16	VCC 1.2V HT	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VCC 1.2V power rail. Device = ADM 1026, U60 pin36	This voltage is one of the power sources for the processor &, Nvidia Hyper-transport Bus. If this rail is out of spec the CP3220 will not function. The H8 will be alive if 3.3V STBY is present and in spec.
17	Proc0 Core NB	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of processor core power rail (voltage varies 1.1V - 1.4V). Device = ADM 1026, U60 pin 37	This voltage is one of the power sources for the processor core. If this rail is out of spec the CP3220 will not function. The H8 will be alive if 3.3V STBY is present and in spec.

TABLE G-1 Sun Netra CP3220 Blade Server Sensors *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
18	VCC 1.15V M Run	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VCC 1.15V Run power rail. Device = ADM 1026, U60 pin 38	If this voltage is out of spec or goes to zero, the CPU or HOST will not function. If all other power rails are up, the Service processor and H8 will still function. Supplies: FBDIMMs.
19	VCC 1.2V Run	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VCC 1.2V Run power rail. Device = ADM 1026, U60 pin39	This voltage is one of the power sources for the Nvidia MCP55 PRO. If this rail is out of spec most of the I/O of CP3220 will not function. The H8 will be alive if 3.3V STBY is present and in spec
20	Proc0 1.8V DDR	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of processor 1.8 V power rail. Device = ADM 1026, U60 pin 40	This voltage is one of the power sources for memory. If this rail is out of spec the memory bus (processor) will not function. The H8 will be alive if 3.3V STBY is present and in spec.
21	VCC 1.5V Run	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VCC 1.5V Run power rail. Device = ADM 1026, U60 pin 41	This voltage is one of the power sources for the Nvidia MCP55 PRO. It is also the power source for the 1.2V rail. If this rail is out of spec the CP3220 will not function. The H8 will be alive if 3.3V STBY is present and in spec
22	Proc0 Core	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of processor core rail (voltage varies 1.05V-1.4V). Device = ADM 1026, U60 pin 33	This voltage is one of the power sources for the CPU core. If this rail is out of spec the CP3220 will not function. The H8 will be alive if 3.3V STBY is present and in spec
23	Board Inlet Temp	"Temperature" (0x01)	Blade server temperature: Ambient @ board inlet. Sensor located @ bottom edge of blade server near power brick. Device = ADM 1032, U9 pin 2/3	There is no fault condition for this sensor; it is for information purposes only.

TABLE G-1 Sun Netra CP3220 Blade Server Sensors *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
24	PM Primary Temp	"Temperature" (0x01)	CP3220 component temperature: Temperature of the FETs on the primary side of the power brick. Sensor located internal to the Brick	There is no fault condition for this sensor; it is for information purposes only.
25	PM Sec Temp	"Temperature" (0x01)	CP3220 component temperature: Temperature of the FETs on the secondary side of the power brick. Sensor located internal to the Brick U2	There is no fault condition for this sensor; it is for information purposes only.
26	PM -48V A-Rail	Threshold (0x01), "Voltage" (0x02)	Voltage measurement between -48V A-side and RTN-A input. Sensor located internal to the Brick U2.	This voltage sensor is internal to the Power Brick. If the A-input power source drops below -36V or above -72V and the B input is in spec the power brick will report the low power but continue to operate normally.
27	PM -48V B-Rail	Threshold (0x01), "Voltage" (0x02)	Voltage measurement between -48V B-side and RTN-B input. Sensor located internal to the Brick U2.	This voltage sensor is internal to the Power Brick. If the B-input power source drops below -36V or above -72V and the A input is in spec the power brick will report the low power but continue to operate normally.
28	PM -48V Voltage	Threshold (0x01), "Voltage" (0x02)	Voltage measurement between HU- and HU+ IN. Sensor located internal to the Brick.	This voltage sensor is internal to the Power Brick. If both the A & B- input power sources drops below -36V or above -72V the power brick will shut down forcing all power (except battery back up) to 0V. No LED's will be illuminated.
29	-48V Current	Threshold (0x01), "Current" (0x03)	Current measurement of the -48V input after the input OR-ing. Sensor located internal to the Brick U2.	This sensor is for reporting purposes only.

TABLE G-1 Sun Netra CP3220 Blade Server Sensors *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
30	12V Current	Threshold (0x01), "Current" (0x03)	Current measurement of the 12V output of the power brick. Sensor located internal to the Brick U2.	This current sensor is internal to the Power Brick. If the output current of the 12VDC exceeds 6.48A the 12V output will be shut down and CP3220 will not function. However, The 3.3V stby will still be operational so the H8 will still function.
31	Version Change	Discrete (reserved), "Version" (0x2B)	Firmware update event	Reports event after the FW update/cold reset
32	System Event	Discrete (0x6f), "System Event" (0x12)		This sensor reports IPMC reset event to ShMM. This sensor lets NetConsole application know that IPMC has taken a reset and the NetConsole session has to be restarted.
33	Sys FW Progress	Discrete (0x6f), "System Firmware Progress" (0x0f),	Monitors the system firmware progress states. Not used.	Not Used. Reserved for future use.
34	Graceful Reboot	Discrete (0x6f) "OEM reserved" (0xc0)	Monitors the state of aa graceful reboot.	This sensor logs events into the system event log when a graceful reboot timer starts, stops, or expires.

FIGURE G-1 Sun Netra CP3220 Voltage Distribution and H8 Sensor Mapping

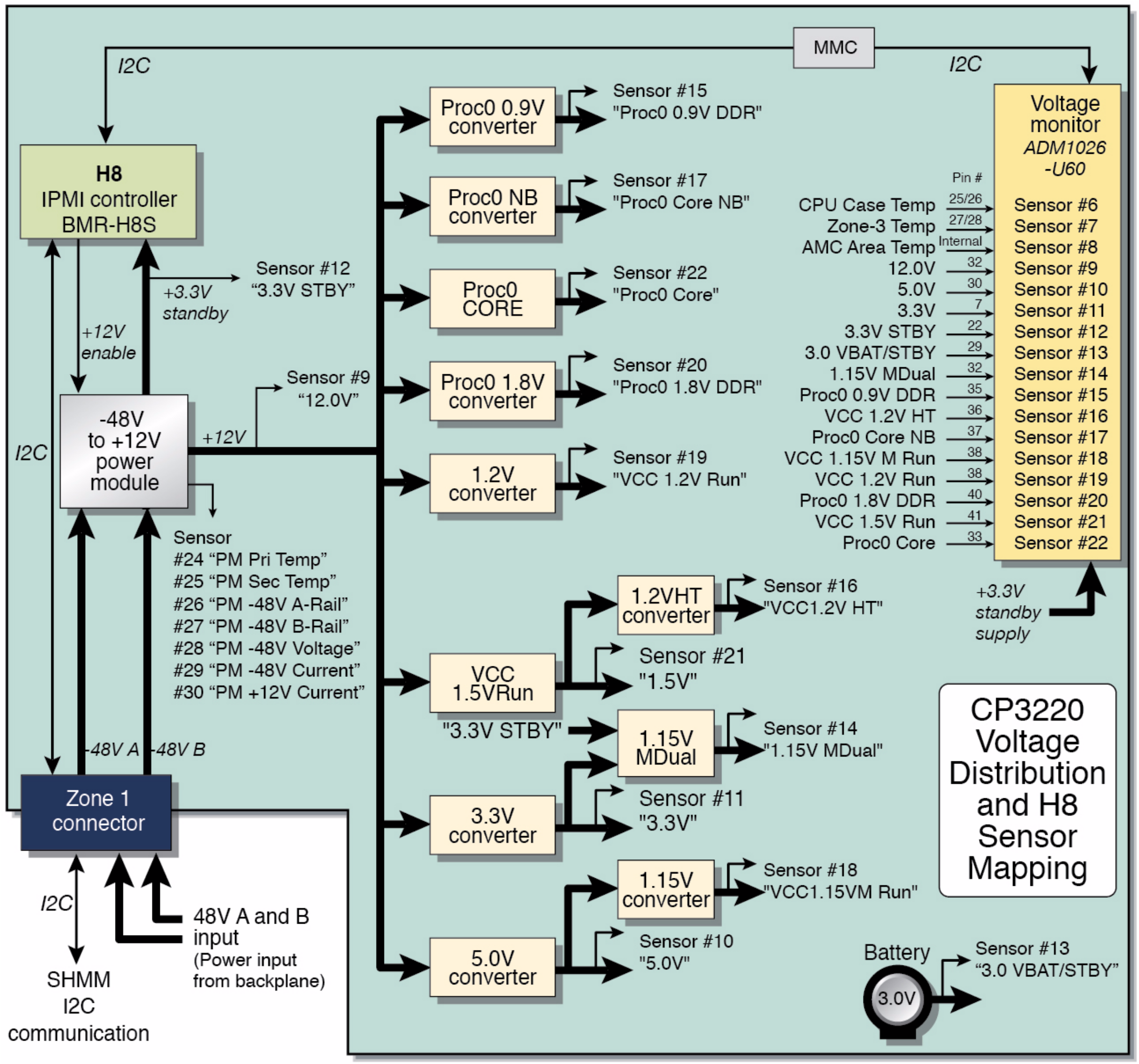
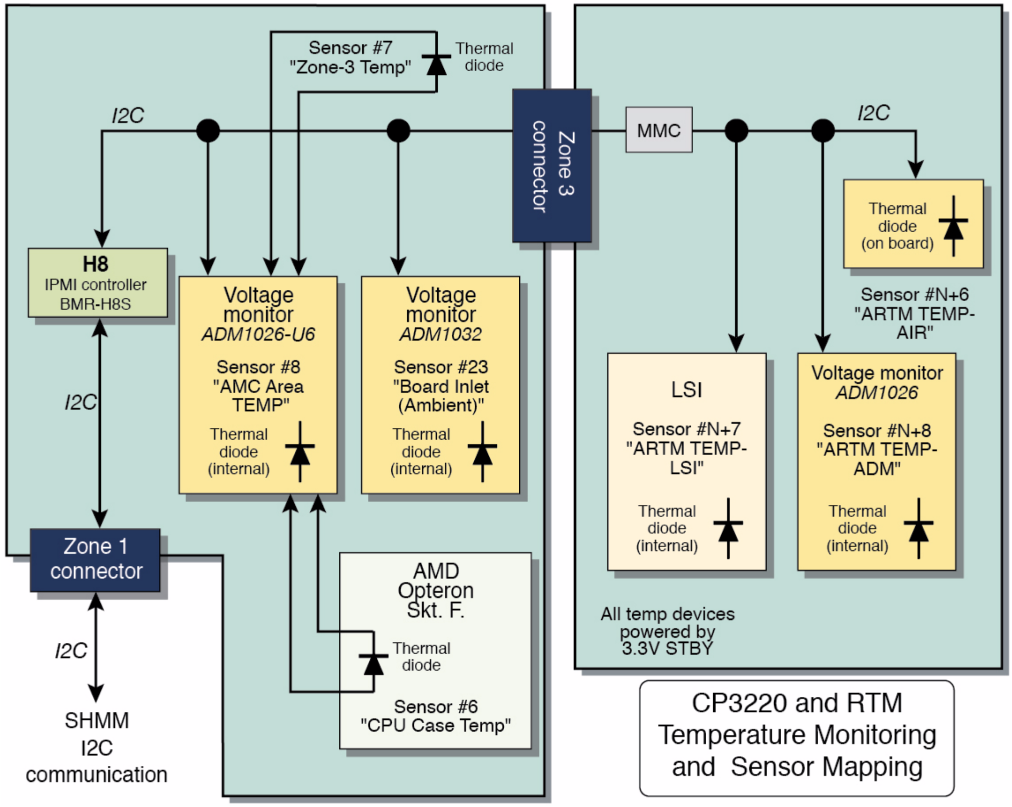


FIGURE G-2 Sun Netra CP3220 Blade Server and RTM Temperature Monitoring and H8 Sensor Mapping



APPENDIX H

Sun Netra CP3060 Blade Server Sensor Map and Fault Isolation

This appendix defines the sensors for the Sun Netra CP3060 blade server.

Sun Netra CP3060 Blade Server Sensor List

The Sun Netra CP3060 sensor names are reported by the on-board H8 processor via the Shelf Manager within the ATCA chassis.

TABLE H-1 Sun Netra CP3060 Blade Server Sensor List

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
0	FRU 0 Hot Swap	Discrete (0x6f), "Hot Swap" (0xf0)	Hot Swap Event for CP3060 FRU	Monitors FRU states as described in the ATCA spec
1	RTM HotSwap	Discrete (0x6f), "Hot Swap" (0xf0)	Hot Swap Event for RTM FRU	Monitors FRU states as described in the ATCA spec
2	Hotswap AMC 0	Discrete (0x6f), "Hot Swap" (0xf0)	Hot Swap Event for AMC FRU	Monitors FRU states as described in the ATCA spec
3	IPMB Physical	Discrete (0x6f), "IPMB Link" (0xf1)	Link Status of IPMB	State of IPMB A or B bus is reported by monitoring the READY signal on the IPMB isolator.
4	BMC Watchdog	Discrete (0x6f), "Watchdog 2" (0x23)	Watchdog state of BMC	N/A
5	CPU Temp1	Threshold (0x01), "Temperature" (0x01)	Internal die temperature of CPU	Power supplies/blade server shall be shut down
6	CPU Temp2	Threshold (0x01), "Temperature" (0x01)	Internal die temperature of CPU	Power supplies/blade server shall be shut down
7	Board Temp	Threshold (0x01), "Temperature" (0x01)	Blade server temperature: Ambient @ ADM1026	Power supplies/blade server shall be shut down
8	12.0V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 12.0V power rail	If this voltage is out of spec or goes to zero, all other power rails will fail (Except STBY). CP3060 and the RTM will not function. The IPMC will be alive if 3.3V STBY is up. Supplies: All DC/DC converters

TABLE H-1 Sun Netra CP3060 Blade Server Sensor List *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
9	5.0V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 5.0V power rail	If this voltage is out of spec or goes to zero, most other power rails will fail. CP3060 will not function. The IPMC will be alive if 3.3V STBY is up. Supplies: Most DC/DC converters
10	3.3V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 3.3V power rail	If this voltage is out of spec or goes to zero, many components on CP3060 will not function. Therefore, CP3060 will not function. The IPMC will be alive if 3.3V STBY is up. Supplies: Multiple support devices
11	3.3V STBY	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 3.3V STBY power rail	If this voltage is out of spec or goes to zero, CP3060 and the RTM will not function. The IPMC will also not function . Supplies: All I2C devices, IPMC domain
12	2.5V STBY	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 2.5V STBY power rail	If this voltage is out of spec or goes to zero, CP3060 and the RTM will not function. The IPMC will also not function . Supplies: IPMC domain
13	1.0V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 1.0V power rail	If this voltage is out of spec or goes to zero, many components on CP3060 will not function. Therefore, CP3060 will not function. The IPMC will be alive if 3.3V STBY is up. Supplies: Multiple support devices, Service Processor
14	1.2V CPU	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 1.2V CPU power rail	If this voltage is out of spec or goes to zero, the CPU or HOST will not function. If all other power rails are up, the Service processor and IPMC will still function. Supplies: CPU core

TABLE H-1 Sun Netra CP3060 Blade Server Sensor List *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
15	1.2V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 1.2V power rail	If this voltage is out of spec or goes to zero, many components on CP3060 will not function. Therefore, CP3060 will not function. The IPMC will be alive if 3.3V STBY is up. Supplies: Multiple support devices, Service Processor
16	1.5V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 1.5V power rail	If this voltage is out of spec or goes to zero, many components on CP3060 will not function. Therefore, CP3060 will not function. The IPMC will be alive if 3.3V STBY is up. Supplies: Multiple support devices
17	0.9V VTTL	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 0.9V VTTL power rail	If this voltage is out of spec or goes to zero, the CPU or HOST will not function. If all other power rails are up, the Service processor and IPMC will still function. Supplies: DDR DIMMs.
18	0.9V VTTR	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 0.9V VTTR power rail	If this voltage is out of spec or goes to zero, the CPU or HOST will not function. If all other power rails are up, the Service processor and IPMC will still function. Supplies: DDR DIMMs.
19	1.8V DDR2L	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 1.8V DDR2L power rail	If this voltage is out of spec or goes to zero, the CPU or HOST will not function. If all other power rails are up, the Service processor and IPMC will still function. Supplies: DDR DIMMs.

TABLE H-1 Sun Netra CP3060 Blade Server Sensor List *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
20	1.8V DDR2R	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 1.8V DDR2R power rail	If this voltage is out of spec or goes to zero, the CPU or HOST will not function. If all other power rails are up, the Service processor and IPMC will still function. Supplies: DDR DIMMs., multiple support devices
21	2.5V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VDD 2.5V power rail	If this voltage is out of spec or goes to zero, many components on CP3060 will not function. Therefore, CP3060 will not function. The IPMC will be alive if 3.3V STBY is up. Supplies: Multiple support devices, Service Processor
22	1.2V STBY	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 1.2V STBY power rail	If this voltage is out of spec or goes to zero, CP3060 and the RTM will not function. The IPMC will also not function . Supplies: All I2C devices, IPMC domain
23	AMC 12V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of AMC 12V power rail	If this voltage is out of spec or goes to zero, the AMC slot will not function. If this voltage is zero and CP3060 is functioning correctly, this implies the IPMC has not enabled the AMC slot.
24	AMC 3.3V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of AMC 3.3V power rail	If this voltage is out of spec or goes to zero, the AMC slot will not function. If this voltage is zero and CP3060 is functioning correctly, this implies the IPMC has not enabled the AMC slot.
25	RTM Presence	Discrete (0x6f), "Entity Presence" (0x25)	Presence of RTM	Indicates if an RTM is connected to CP3060
26	Version change	Discrete (0x6f), "reserved" (0x2b)	Firmware update event	Reports event after the FW update/cold reset

FIGURE H-1 Sun Netra CP3060 Voltage Distribution and H8 Sensor Mapping

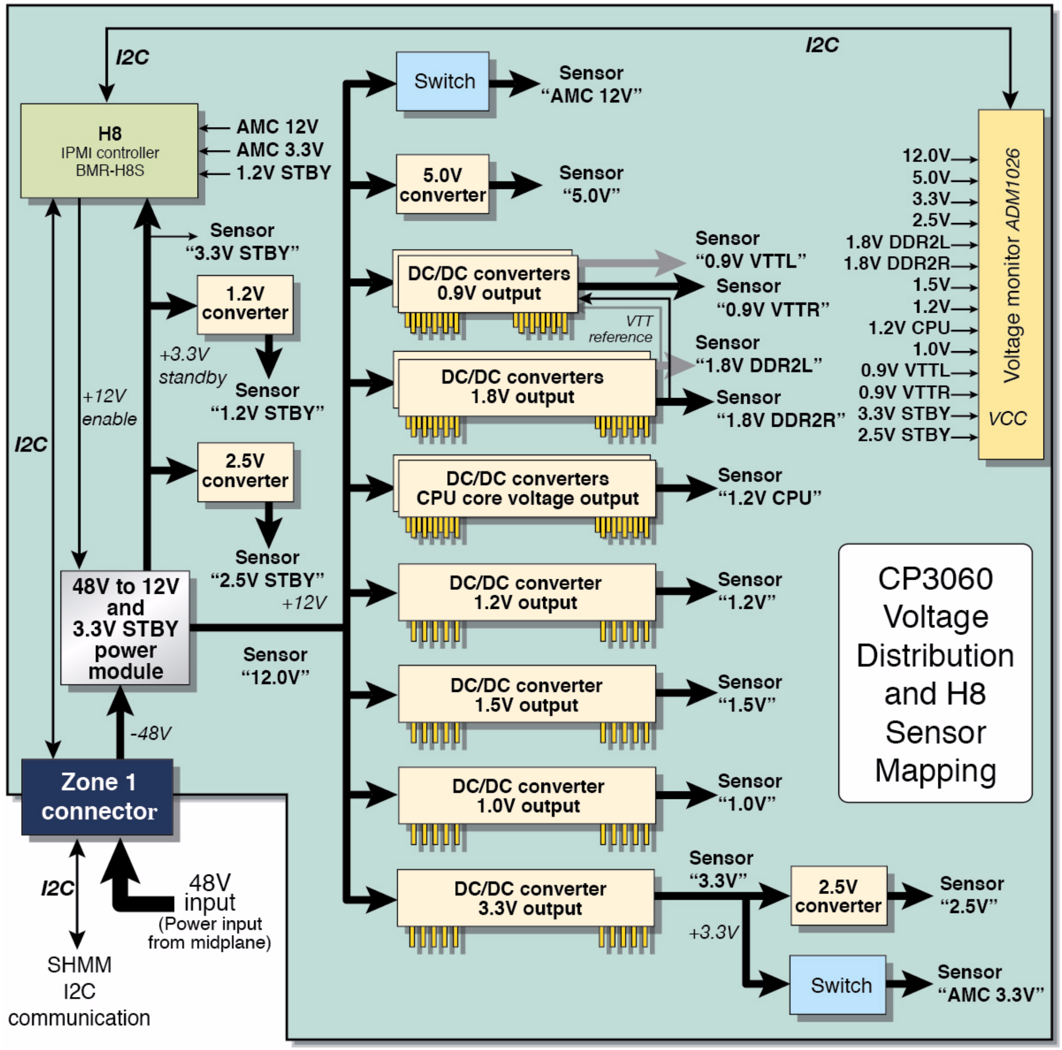
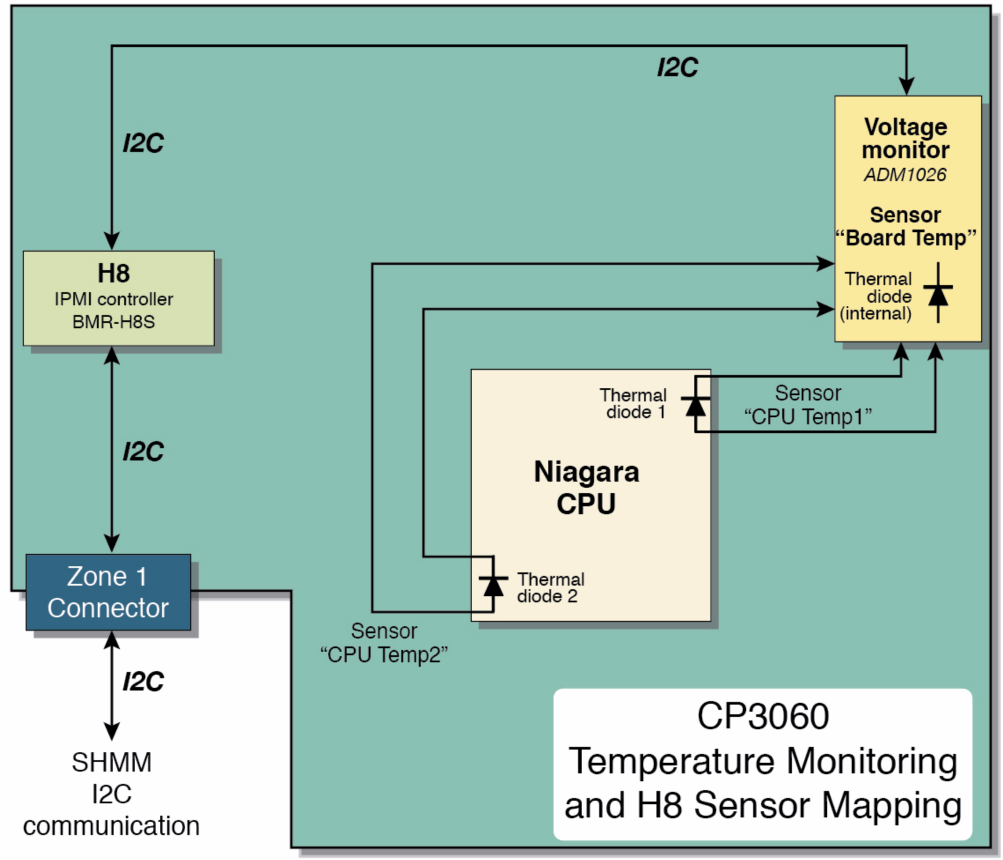


FIGURE H-2 Sun Netra CP3060 Temperature Monitoring and H8 Sensor Mapping



APPENDIX **I**

Sun Netra CP3250 Blade Server Sensor Map and Fault Isolation

This appendix defines the sensors for the Sun Netra CP3250 blade server.

Sun Netra CP3250 Blade Server Sensor List

The sensor numbers and names are reported by the on-blade server Sun Netra CP3250 IPMC processor via the ShMM, within the ATCA chassis.

TABLE I-1 Sun Netra CP3250 Blade Server Sensor List

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
0	FRU 0 Hot Swap	Discrete (0x6f), "Hot Swap" (0xf0)	Hot swap for CP3250 FRU	N/A
1	HotSwap AMC 5	Discrete (0x6f), "Hot Swap" (0xf0)	Hot swap for AMC FRU	N/A
2	HotSwap ARTM 15	Discrete (0x6f), "Hot Swap" (0xf0)	Hot swap for ARTM FRU	N/A
3	IPMB Physical	Discrete (0x6f), "Hot Swap" (0xf0)	Link Status of IPMB	No reply from IPMB (A or B). State of IPMB A or B bus is reported by monitoring the READY signal on the IPMB isolator.
4	BMC Watchdog	Discrete (0x6f), "Watchdog 2" (0x23)	Watchdog state of BMC	N/A
5	12.0V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 12.0V power rail	If this voltage is out of spec or goes to zero, all other power rails will fail (Except STBY). Blade server and the RTM will not function. The IPMC will be alive if 3.3V STBY is up. Supplies: All DC/DC converters
6	5.0V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 5.0V power rail	If this voltage is out of spec or goes to zero, most other power rails will fail. Blade server will not function. The IPMC will be alive if 3.3V STBY is up. Supplies: Most DC/DC converters

TABLE I-1 Sun Netra CP3250 Blade Server Sensor List *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
7	3.3V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 3.3V power rail	If this voltage is out of spec or goes to zero, many components on blade server will not function. Therefore, blade server will not function. The IPMC will be alive if 3.3V STBY is up. Supplies: Multiple support devices
8	3.3V STBY	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 3.3V STBY power rail.	If this voltage is out of spec blade server and the H8 will not function. This rail is the power source for the management components on the blade server including all I2C devices and the H8.
9	SuperCAP voltage	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 3.0V battery (ADM 1026, pin 29)	If this voltageis out of limit, the battery has failed or is not installed. Ignore message if not using the battery.
10	1.2V NTune	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 1.2V power rail.	If this voltage is out of spec or goes to zero, many components on blade server will not function. Therefore, blade server will not function.
11	CPU VTT	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of the VTT power rail.	If this voltage is out of spec or goes to zero, many components on blade server will not function. Therefore, blade server will not function.
12	1.5V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 1.5V power rail.	If this voltage is out of spec or goes to zero, many components on blade server will not function. Therefore, blade server will not function. The IPMC will be alive if 3.3V STBY is up. Supplies: CPU I/O, DIMMs

TABLE I-1 Sun Netra CP3250 Blade Server Sensor List *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
13	1.8 V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 1.8V power rail.	If this voltage is out of spec or goes to zero, many components on blade server will not function. Therefore, blade server will not function.
14	DDR2 VTT	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of DDR VTT power rail.	If this voltage is out of spec or goes to zero, many components on blade server will not function. Therefore, blade server will not function. Devices: DIMMs
15	1.05 V Core	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 1.05V power rail.	If this voltage is out of spec or goes to zero, many components on blade server will not function. Therefore, blade server will not function. Devices: CPU
16	1.5 V NTune	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 1.5V power rail.	If this voltage is out of spec or goes to zero, many components on blade server will not function. Therefore, blade server will not function. Devices: Neptune
17	VCC CPU1	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VCC CPU.	If this voltage is out of spec or goes to zero, many components on blade server will not function. Therefore, blade server will not function. Devices: CPU
18	VCC CPU0	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VCC CPU.	If this voltage is out of spec or goes to zero, many components on blade server will not function. Therefore, blade server will not function. Devices: CPU
19	Inlet 1 Temp Sen	Threshold (0x01), "Temperature" (0x01)	Ambient Temp, unused	N/A. To be deleted in future release.
20	Inlet 3 Temp Sen	Threshold (0x01), "Temperature" (0x01)	Ambient Temp, unused	N/A. To be deleted in future release.

TABLE I-1 Sun Netra CP3250 Blade Server Sensor List *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
21	Inlet 2 Temp Sen	Threshold (0x01), "Temperature" (0x01)	Ambient Temp, unused	N/A. To be deleted in future release.
22	MCH Temp Sensor	Threshold (0x01), "Temperature" (0x01)	Junction Temperature, Memory Controller Hub, Northbridge	This sensor indicates that the MCH is out of temperature range.
23	CPU_TEMP_SK0D0	Threshold (0x01), "Temperature" (0x01)	CPU Junction Temp, socket 0, domain 0	This sensor indicates that the CPU is out of temperature range. Above 94C will cause blade server to shutdown.
24	CPU_TEMP_SK0D1	Threshold (0x01), "Temperature" (0x01)	CPU Junction Temp, socket 0, domain 1	This sensor indicates that the CPU is out of temperature range. Above 94C will cause blade server to shutdown.
25	CPU_TEMP_SK1DO	Threshold (0x01), "Temperature" (0x01)	CPU Junction Temp, socket 1, domain 0	This sensor indicates that the CPU is out of temperature range. Above 94C will cause blade server to shutdown.
26	CPU_TEMP_SK1D1	Threshold (0x01), "Temperature" (0x01)	CPU Junction Temp, socket 1, domain 1	This sensor indicates that the CPU is out of temperature range. Above 94C will cause blade server to shutdown.
27	Version change	Discrete (0x6f), "reserved" (0x2b)	Firmware update event	Reports event after the FW update/cold reset
28	System Event	Discrete (0x6f), "System Event" (0x12)	System reset event	This sensor reports IPMC reset event to ShMM. This sensor lets NetConsole application know that IPMC has taken a reset and the NetConsole session has to be restarted.
29	CPU 0 presence	Discrete (0x6f), "Entity Presence" (0x25)		Indicates that CPU0 is installed.
30	CPU 1 presence	Discrete (0x6f), "Entity Presence" (0x25)		Indicates that CPU1 is installed.

TABLE I-1 Sun Netra CP3250 Blade Server Sensor List *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
31	P48V Alarm	Discrete (0x70), "OEM reserved" (0xc0)	Voltage measurement of 48V power	Detects either the 48V power input A or 48V power input B to power module that converts it to 12V.
32	Sys FW Progress	Discrete (0x6f), "System Firmware Progress" (0x0f),	Monitors the system firmware progress states.	This sensor monitors the firmware progress. The system firmware sends the firmware progress events to the system event log by way of the IPMC.
33	Graceful Reboot	Discrete (0x6f) "OEM reserved" (0xc0)	Monitors the state of aa graceful reboot.	This sensor logs events into the system event log when a graceful reboot timer starts, stops, or expires.

APPENDIX J

Sun Netra CP3260 Blade Server Sensor Map and Fault Isolation

This appendix defines the sensors for the Sun Netra CP3260 blade server.

Sun Netra CP3260 Blade Server Sensor List

The sensor numbers and names are reported by the on-board Sun Netra CP3260 IPMC processor via the ShMM, within the ATCA chassis.

TABLE J-1 Sun Netra CP3260 Blade Server Sensor List

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
0	FRU 0 Hot Swap	Discrete (0x6f), "Hot Swap" (0xf0)	Hot swap for CP3260 FRU	N/A
1	ARTM HotSwap	Discrete (0x6f), "Hot Swap" (0xf0)	Hotswap for RTM FRU	N/A
2	IPMB Physical	Discrete (0x6f), "IPMB Link" (0xf1)	Link Status of IPMB	No reply from IPMB (A or B). State of IPMB A or B bus is reported by monitoring the READY signal on the IPMB isolator.
3	BMC Watchdog	Discrete (0x6f), "Watchdog 2" (0x23)	Watchdog state of BMC	N/A
4	CPU Temp1	Threshold (0x01), "Temperature" (0x01)	Internal die temperature of CPU (Niagra 2)	If this temperature goes beyond 112C, all power supplies will be shut down and all front panel LEDs are turned off. The blue LED on the RTM remains ON.
5	CPU Temp2	Threshold (0x01), "Temperature" (0x01)	Internal die temperature of CPU (Niagra 2)	If this temperature goes beyond 112C, all power supplies will be shut down and all front panel LEDs are turned off. The blue LED on the RTM remains ON.

TABLE J-1 Sun Netra CP3260 Blade Server Sensor List *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
6	Board Temp	Threshold (0x01), "Temperature" (0x01)	Blade server temperature: Ambient @ ADM1026	If this temperature goes beyond 88C, all power supplies will be shut down and all front panel LEDs are turned off. The blue LED on the RTM remains ON.
7	12.0V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 12.0V power rail	If this voltage is out of spec or goes to zero, all other power rails will fail (Except STBY). CP3260 and the RTM will not function. The IPMC will be alive if 3.3V STBY is up. Supplies: All DC/DC converters
8	5.0V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 5.0V power rail	If this voltage is out of spec or goes to zero, most other power rails will fail. CP3260 will not function. The IPMC will be alive if 3.3V STBY is up. Supplies: Most DC/DC converters
9	3.3V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 3.3V power rail	If this voltage is out of spec or goes to zero, many components on CP3260 will not function. Therefore, CP3260 will not function. The IPMC will be alive if 3.3V STBY is up. Supplies: Multiple support devices
10	3.3V STBY	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 3.3V STBY power rail	If this voltage is out of spec or goes to zero, CP3260 and the RTM will not function. Supplies: All I2C devices, IPMC

TABLE J-1 Sun Netra CP3260 Blade Server Sensor List *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
11	3.0 VBAT/STBY	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 3.0 VBAT/STBY power rail	If this voltage is out of spec or goes to zero, the on blade server battery is bad or missing. The battery is not required for CP3260 or the RTM to function correctly. Supplies: 3.3V STBY
12	1.0V VDD	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 1.0V VDD power rail	If this voltage is out of spec or goes to zero, many components on CP3260 will not function. Therefore, CP3260 will not function. The IPMC will be alive if 3.3V STBY is up. Supplies: Multiple support devices, service processor
13	1.1V CPU	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 1.1V CPU power rail	If this voltage is out of spec or goes to zero, the CPU or HOST will not function. If all other power rails are up, the Service processor and IPMC will still function. Supplies: CPU core
14	VDD 1.1V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VDD 1.1V power rail	If this voltage is out of spec or goes to zero, many components on CP3260 will not function. Therefore, CP3260 will not function. The IPMC will be alive if 3.3V STBY is up. Supplies: Multiple support devices, service processor.

TABLE J-1 Sun Netra CP3260 Blade Server Sensor List *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
15	1.5V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 1.5V power rail	If this voltage is out of spec or goes to zero, many components on CP3260 will not function. Therefore, CP3260 will not function. The IPMC will be alive if 3.3V STBY is up. Supplies: CPU I/O, FBDIMMs
16	VDD 1.8V FBDIMM	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VDD 1.8V power rail	If this voltage is out of spec or goes to zero, the CPU or HOST will not function. If all other power rails are up, the Service processor and IPMC will still function. Supplies: FBDIMMs.
17	VDD 2.5V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VDD 2.5V power rail	If this voltage is out of spec or goes to zero, many components on CP3260 will not function. Therefore, CP3260 will not function. The IPMC will be alive if 3.3V STBY is up. Supplies: Multiple support devices, service processor
18	VDD_IO 1.2V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VDD_IO 1.2V power rail	If this voltage is out of spec or goes to zero, the CPU or HOST will not function. If all other power rails are up, the Service processor and IPMC will still function. Supplies: CPU I/O
19	Version Change	Discrete (0x6f), "reserved" (0x2b)	Firmware update event	IPMC reports event after the FW update/cold reset.
20	P48V Alarm	Discrete (0x70), "OEM reserved" (0xc0)	Voltage measurement of 48V power	Detecs either the 48V power input A or 48V power input B to power module that converts it to 12V.

TABLE J-1 Sun Netra CP3260 Blade Server Sensor List *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
21	VDD 1.8V M0	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VDD 1.8V power rail.	If this voltage is out of spec or goes to zero, the CPU or HOST will not function. If all other power rails are up, the Service processor and IPMC will still function. Supplies: FBDIMMs..
22	Sys FW Progress	Discrete (0x6f), "System Firmware Progress" (0x0f),	Monitors the system firmware progress states.	This sensor monitors the firmware progress. The system firmware sends the firmware progress events to the system event log by way of the IPMC.
23	Graceful Reboot	Discrete (0x6f) "OEM reserved" (0xc0)	Monitors the state of aa graceful reboot.	This sensor logs events into the system event log when a graceful reboot timer starts, stops, or expires.

FIGURE J-1 Sun Netra CP3260 Voltage Distribution and IPMC Sensor Mapping

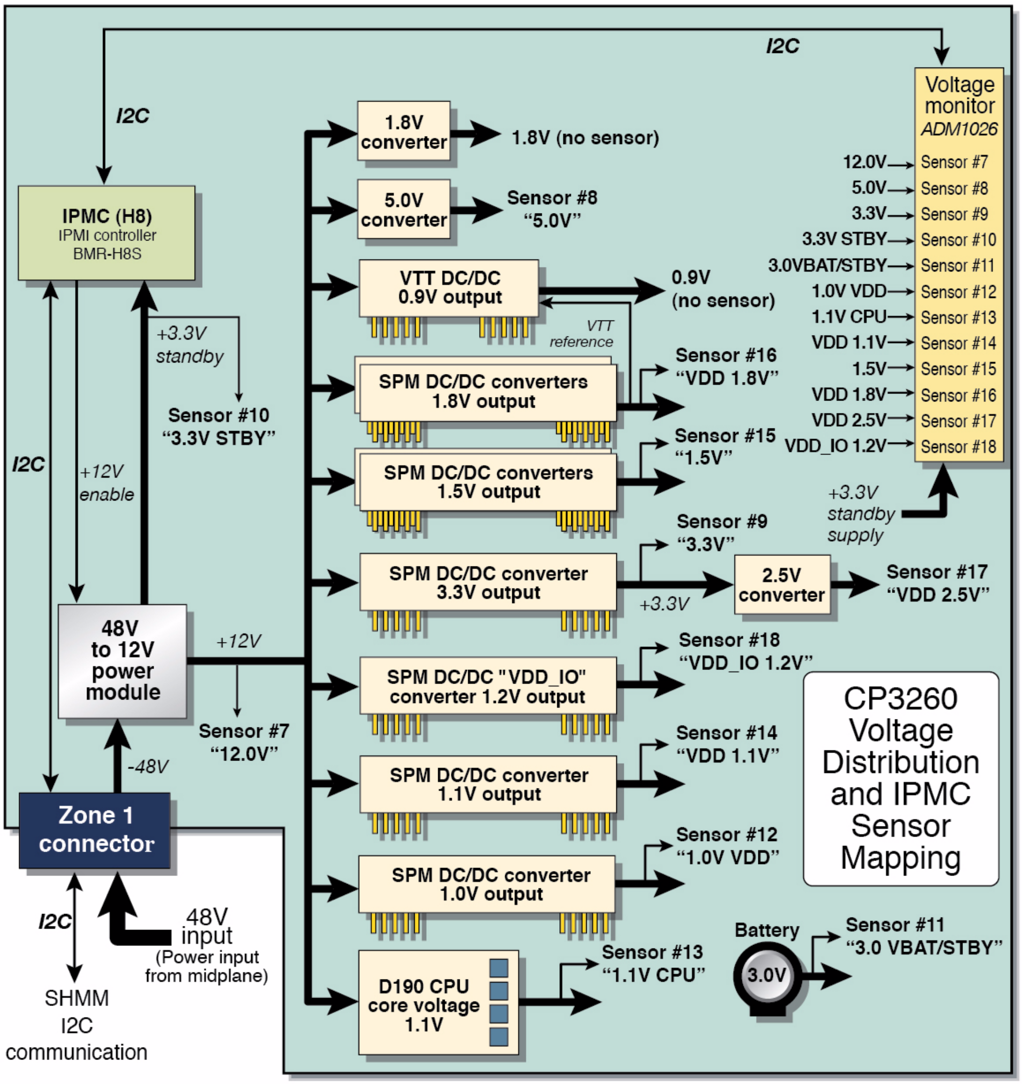
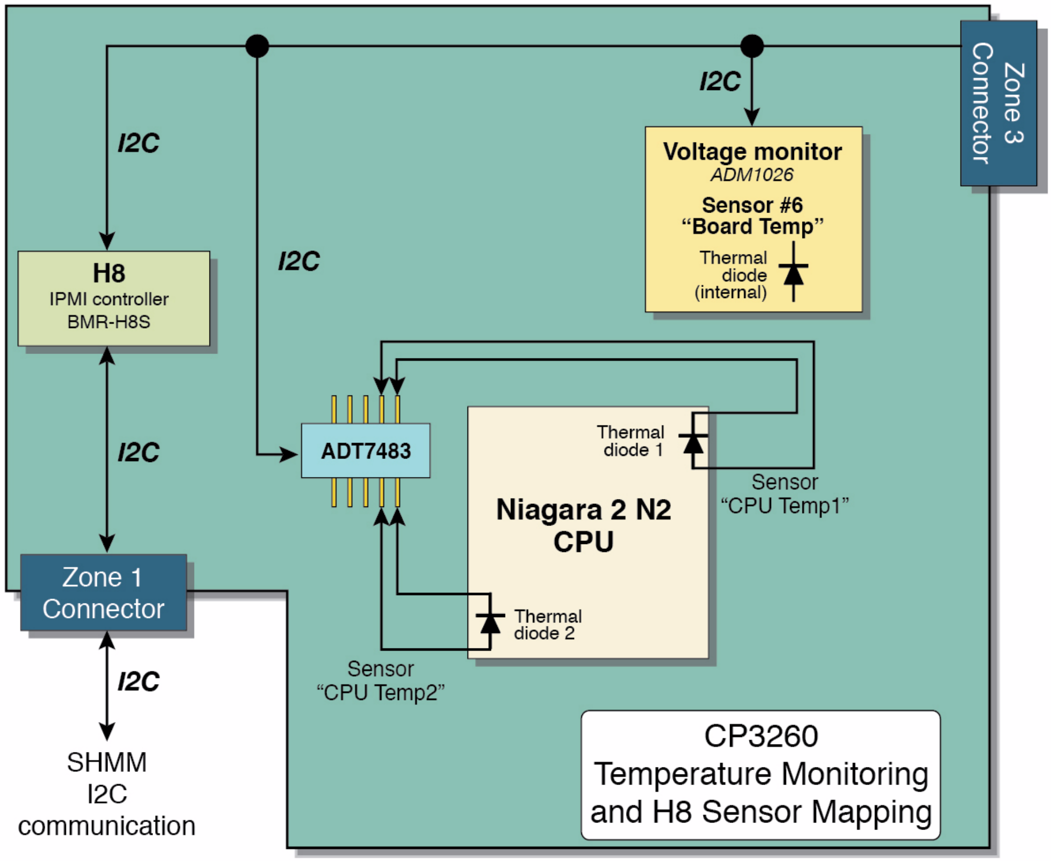


FIGURE J-2 Sun Netra CP3260 Temperature Monitoring and H8 Sensor Mapping



APPENDIX **K**

Sun Netra CP3270 Blade Server Sensor Map and Fault Isolation

This appendix defines the sensors for the Sun Netra CP3270 ATCA blade server.

Sun Netra CP3270 Blade Server Sensor List

The sensor numbers and names are reported by the on-blade server Sun Netra CP3270 IPMC processor via the ShMM, within the ATCA chassis.

TABLE K-1 Sun Netra CP3270 Blade Server Sensor List

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
0	FRU 0 Hot Swap	Discrete (0x6f), "Hot Swap" (0xf0)	Hot swap for CP3270 FRU	N/A
1	HotSwap AMC 5	Discrete (0x6f), "Hot Swap" (0xf0)	Hot swap for AMC FRU	N/A
2	HotSwap ARTM	Discrete (0x6f), "Hot Swap" (0xf0)	Hot swap for ARTM FRU	N/A
3	IPMB Physical	Discrete (0x6f), "Hot Swap" (0xf0)	Link Status of IPMB	No reply from IPMB (A or B). State of IPMB A or B bus is reported by monitoring the READY signal on the IPMB isolator.
4	BMC Watchdog	Discrete (0x6f), "Watchdog 2" (0x23)	Watchdog state of BMC	N/A
5	CPU 0 Temperature	Threshold (0x01), "Temperature" (0x01)	CPU Junction Temp, socket 0, domain 0	This sensor indicates that the CPU is out of temperature range. Above 94ºC will cause blade server to shutdown.
6	CPU 1 Temperature	Threshold (0x01), "Temperature" (0x01)	CPU Junction Temp, socket 0, domain 1	This sensor indicates that the CPU is out of temperature range. Above 94ºC will cause blade server to shutdown.
7	Vbat	Threshold, voltage	CMOS battery voltage	This sensor indicates a low battery condition and that the battery needs to be replaced.

TABLE K-1 Sun Netra CP3270 Blade Server Sensor List *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
8	P3V3_STBY	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 3.3V STBY power rail.	If this voltage is out of spec blade server and the H8 will not function. This rail is the power source for the management components on the blade server including all I2C devices and the H8.
9	P12V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 12.0V power rail	If this voltage is out of spec or goes to zero, all other power rails will fail (Except STBY). Blade server and the RTM will not function. The IPMC will be alive if 3.3V STBY is up. Supplies: All DC/DC converters
10	P5V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 5.0V power rail	If this voltage is out of spec or goes to zero, most other power rails will fail. Blade server will not function. The IPMC will be alive if 3.3V STBY is up. Supplies: Most DC/DC converters
11	P3V3	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 3.3V power rail	If this voltage is out of spec or goes to zero, many components on blade server will not function. Therefore, blade server will not function. The IPMC will be alive if 3.3V STBY is up. Supplies: Multiple support devices
12	P1V05_PCH	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VCC PCM.	If this voltage is out of spec or goes to zero, many components on blade server will not function. Therefore, blade server will not function. Devices: PCM

TABLE K-1 Sun Netra CP3270 Blade Server Sensor List *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
13	P1V5_DDR3_CPU0	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VCC MEMORY.	If this voltage is out of spec or goes to zero, many components on blade server will not function. Therefore, blade server will not function. Devices: MEMORY
14	P1V5_DDR3_CPU1	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VCC MEMORY.	If this voltage is out of spec or goes to zero, many components on blade server will not function. Therefore, blade server will not function. Devices: MEMORY
15	P0V75_DDR3_CPU0	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VCC MEMORY.	If this voltage is out of spec or goes to zero, many components on blade server will not function. Therefore, blade server will not function. Devices: MEMORY
16	P0V75_DDR3_CPU1	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VCC MEMORY.	If this voltage is out of spec or goes to zero, many components on blade server will not function. Therefore, blade server will not function. Devices: MEMORY
17	VTT CPU0	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of the VTT power rail.	If the voltage is 1.260 (out of spec) or goes to zero, many components on blade server will not function. Therefore, blade server will not function.
18	VTT CPU1	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of the VTT power rail.	If the voltage is 1.260 (out of spec) or goes to zero, many components on blade server will not function. Therefore, blade server will not function.

TABLE K-1 Sun Netra CP3270 Blade Server Sensor List *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
19	VCCP_CPU0	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VCC CPU.	If this voltage is out of spec or goes to zero, many components on blade server will not function. Therefore, blade server will not function. Devices: CPU
20	VCCP_CPU1	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VCC CPU.	If this voltage is out of spec or goes to zero, many components on blade server will not function. Therefore, blade server will not function. Devices: CPU
21	Version change	Discrete (0x6f), "reserved" (0x2b)	Firmware update event	Reports event after the FW update/cold reset
22	System Event	Discrete (0x6f), "System Event" (0x12)	System reset event	This sensor reports IPMC reset event to ShMM. This sensor lets NetConsole application know that IPMC has taken a reset and the NetConsole session has to be restarted.
23	CPU 0 presence	Discrete (0x6f), "Entity Presence" (0x25)		Indicates that CPU0 is installed.
24	CPU 1 presence	Discrete (0x6f), "Entity Presence" (0x25)		Indicates that CPU1 is installed.
25	P48V Alarm	Discrete (0x70), "OEM reserved" (0xc0)	Voltage measurement of 48V power	Detects either the 48V power input A or 48V power input B to power module that converts it to 12V.

TABLE K-1 Sun Netra CP3270 Blade Server Sensor List *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
26	Sys FW Progress	Discrete (0x6f), "System Firmware Progress" (0x0f),	Monitors the system firmware progress states.	This sensor monitors the firmware progress. The system firmware sends the firmware progress events to the system event log by way of the IPMC.
27	Graceful Reboot	Discrete (0x6f) "OEM reserved" (0xc0)	Monitors the state of graceful reboot.	This sensor logs events into the system event log when a graceful reboot timer starts, stops, or expires.
28	Therm Trip	Discrete (0x6f) "OEM reserved" (0xc0)	Monitors thermal trip state of CPUs.	This sensor tracks the Therm Trip bit in CPLD. If CPU generates thermal trip, CPLD shuts down the power and sets therm tip bit in CPLD (offset 2, bit 0). IPMC has to know about thermal trip power shutdown, put the blade server in M1 state, and generate appropriate event.

APPENDIX **L**

Netra SPARC T3-1BA Blade Server Sensor Map and Fault Isolation

This appendix defines the sensors for the Netra SPARC T3-1BA blade server, which is the next generation blade after the Sun Netra CP3260 ATCA blade server.

Netra SPARC T3-1BA Blade Server Sensor List

The sensor numbers and names are reported by the on-board IPMC processor via the ShMM, within the ATCA chassis.

TABLE L-1 Netra SPARC T3-1BA Blade Server Sensor List

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
0	FRU 0 Hot Swap	Discrete (0x6f), "Hot Swap" (0xf0)	Hot swap for blade server FRU	N/A
1	ARTM Hot Swap	Discrete (0x6f), "Hot Swap" (0xf0)	Hotswap for RTM FRU	N/A
2	IPMB Physical	Discrete (0x6f), "IPMB Link" (0xf1)	Link Status of IPMB	No reply from IPMB (A or B). State of IPMB A or B bus is reported by monitoring the READY signal on the IPMB isolator.
3	BMC Watchdog	Discrete (0x6f), "Watchdog 2" (0x23)	Watchdog state of BMC	N/A
4	CPU 0 Temp	Threshold (0x01), "Temperature" (0x01)	Internal die temperature of CPU	If this temperature goes beyond 110C, all power supplies will be shut down and all front panel LEDs are turned off. The blue LED on the RTM remains ON.
5	CPU 1 Temp	Threshold (0x01), "Temperature" (0x01)	Internal die temperature of CPU	If this temperature goes beyond 110C, all power supplies will be shut down and all front panel LEDs are turned off. The blue LED on the RTM remains ON.

TABLE L-1 Netra SPARC T3-1BA Blade Server Sensor List *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
6	Vbat	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 3.0 VBAT/STBY power rail	If this voltage is out of spec or goes to zero, the on blade server battery is bad or missing. The battery is not required for blade server or the ARTM to function correctly. Supplies: 3.3V STBY
7	3.3 V STBY	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 3.3V STBY power rail	If this voltage is out of spec or goes to zero, blade server and the RTM will not function. Supplies: All I2C devices, IPMC
8	12.0V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 12.0V power rail	If this voltage is out of spec or goes to zero, all other power rails will fail (Except STBY). Blade server and the RTM will not function. The IPMC will be alive if 3.3V STBY is up. Supplies: All DC/DC converters
9	5.0V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 5.0V power rail	If this voltage is out of spec or goes to zero, most other power rails will fail. Blade server will not function. The IPMC will be alive if 3.3V STBY is up. Supplies: Most DC/DC converters
10	3.3V MAIN	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 3.3V power rail	If this voltage is out of spec or goes to zero, many components on blade server will not function. Therefore, blade server will not function. The IPMC will be alive if 3.3V STBY is up. Supplies: Multiple support devices

TABLE L-1 Netra SPARC T3-1BA Blade Server Sensor List *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
11	B0 VDD 1.1V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VDD 1.1V power rail	If this voltage is out of spec or goes to zero, many components on blade server will not function. Therefore, blade server will not function. The IPMC will be alive if 3.3V STBY is up. Supplies: multiple support devices, service processor.
12	B1 VDD 1.1V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VDD 1.1V power rail	If this voltage is out of spec or goes to zero, many components on blade server will not function. Therefore, blade server will not function. The IPMC will be alive if 3.3V STBY is up. Supplies: multiple support devices, service processor.
13	B2 VDD 1.1V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VDD 1.1V power rail	If this voltage is out of spec or goes to zero, many components on blade server will not function. Therefore, blade server will not function. The IPMC will be alive if 3.3V STBY is up. Supplies: multiple support devices, service processor.
14	B3 VDD 1.1V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VDD 1.1V power rail	If this voltage is out of spec or goes to zero, many components on blade server will not function. Therefore, blade server will not function. The IPMC will be alive if 3.3V STBY is up. Supplies: multiple support devices, service processor.

TABLE L-1 Netra SPARC T3-1BA Blade Server Sensor List *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
15	RF CPU VDD 1.5V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VDD 1.5V power rail	If this voltage is out of spec or goes to zero, many components on blade server will not function. Therefore, blade server will not function. The IPMC will be alive if 3.3V STBY is up. Supplies: Multiple support devices, service processor
16	VTT MO 0.75V	Threshold (0x01), "Voltage" (0x02)	Voltage	
17	VDD MO 1.5V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VDD 1.5V power rail	If this voltage is out of spec or goes to zero, many components on blade server will not function. Therefore, blade server will not function. The IPMC will be alive if 3.3V STBY is up. Supplies: Multiple support devices, service processor
18	VDD 1.5V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VDD 1.5V power rail	If this voltage is out of spec or goes to zero, many components on blade server will not function. Therefore, blade server will not function. The IPMC will be alive if 3.3V STBY is up. Supplies: Multiple support devices, service processor
19	VDD 2.5V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of VDD 2.5V power rail	If this voltage is out of spec or goes to zero, many components on blade server will not function. Therefore, blade server will not function. The IPMC will be alive if 3.3V STBY is up. Supplies: Multiple support devices, service processor
20	Version change	Discrete (0x6f), "reserved" (0x2b)	Firmware update event	IPMC reports event after the FW update/cold reset.

TABLE L-1 Netra SPARC T3-1BA Blade Server Sensor List *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
21	System event	Discrete (0x6f, “System Event” (0x12)	Tracks IPMC resets.	Every time IPMC takes a reset, an event is sent to Shelf manager. This sensor supports SOL based NetConsole use.
22	P48V alarm	Discrete (0x6f, “System Event” (0x12)	Monitors the 48V rails.	Monitors the 48V rails and reports the state via state bits. IPMC shall directly read Power Module’s voltage registers to determine presence of power rails. If a voltage less than 38 volts, rail is considered to be absent. • 0x1 (Bit 0 set) Both rails absent. • 0x2 (Bit 1 set) Only Rail A is present. • 0x4 (Bit 2 set) Only Rail B is present. • 0x8 (Bit 3 set) Both Rail A and B are present.
23	Sys fw progress	Discrete (0x6f), "System Firmware Progress" (0x0f),	Monitors the system firmware progress states.	This sensor monitors the firmware progress. The system firmware sends the firmware progress events to the system event log by way of the IPMC.

TABLE L-1 Netra SPARC T3-1BA Blade Server Sensor List *(Continued)*

Sensor Number	Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
24	Graceful reboot	Discrete (0x6f) "OEM reserved" (0xc0)	Monitors the state of aa graceful reboot.	This sensor logs events into the system event log when a graceful reboot timer starts, stops, or expires.
25	Slot power	Threshold (0x01), "Voltage" (0x02)	Tracks the power being consumed by the slot.	This sensor is for information purposes and does not generate any events. The nominal value range for this sensor is from 5 Watts to ~300 Watts. The nominal value is set at 150 Watts. This is not indicative of a typical usage. Power usage shall depend upon the state of the blade and state of the OS running.
26	Thermal trip	Discrete (0x6f, “System Event” (0x12)	Reports upper non-recoverable (UNC) threshold violation events.	This sensor sends event notice if UNC threshold is reached.

APPENDIX **M**

Sun Netra CP32x0 ARTM Sensor Map and Fault Isolation

This appendix defines the sensors for the Sun Netra CP32x0 ARTM.

Note – The ARTM sensor numbers change depending on which node board is inserted and the configuration of the board (how many and what type of AMC cards are installed).

Related documentation is available at:

`http://docs.sun.com/app/docs/prod/cp32x0.sas?l=en#hic`

Sun Netra CP32x0 ARTM Sensor List

The sensor numbers and names are reported by the on-board Sun Netra CP32x0 IPMC processor via the ShMM, within the ATCA chassis.

TABLE M-1 Sun Netra CP32x0 ARTM-HD Sensor List

Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
ARTM 3V3STBY	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 3.3V STBY power rail on the RTM	If this voltage is out of spec or goes to zero, the RTM will not function and no LEDs on the RTM will be ON. If this voltage is zero and CP3260 is functioning correctly, this implies the IPMC has not enabled the RTM.
ARTM 3V3MAIN	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 3.3V power rail on the RTM	If this voltage is out of spec or goes to zero, the RTM will not function. If this voltage is zero and CP3260 is functioning correctly, this may imply the IPMC has not enabled the RTM.
ARTM 12V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 12.0V power rail on the RTM	If this voltage is out of spec or goes to zero, the RTM will not function. If this voltage is zero and CP3260 is functioning correctly, this may imply the IPMC has not enabled the RTM.
ARTM 5V	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 5.0V power rail on the RTM	If this voltage is out of spec or goes to zero, the RTM will not function. If this voltage is zero and CP3260 is functioning correctly, this may imply the IPMC has not enabled the RTM.
ARTM 1V2	Threshold (0x01), "Voltage" (0x02)	Voltage measurement of 1.2V power rail on the RTM	If this voltage is out of spec or goes to zero, the RTM will not function. If this voltage is zero and CP3260 is functioning correctly, this may imply the IPMC has not enabled the RTM.

TABLE M-1 Sun Netra CP32x0 ARTM-HD Sensor List *(Continued)*

Sensor Name	Sensor Type	Sensor Description	Fault Condition if Sensor Out of Limit
ARTM TEMP-AIR	Threshold (0x01), "Temperature" (0x01)	RTM Ambient temperature	No set threshold and thus no action taken. Only reported temperature shown.
ARTM TEMP-LSI	Threshold (0x01), "Temperature" (0x01)	Internal die temperature of LSI chip on RTM	No set threshold and thus no action taken. Only reported temperature shown.
ARTM TEMP-ADM	Threshold (0x01), "Temperature" (0x01)	RTM board temperature: Ambient @ ADM1026	No set threshold and thus no action taken. Only reported temperature shown.

Note – The ARTM sensor numbers in the following illustrations change depending on which node board is inserted and the configuration of the board (how many and what type of AMC cards are installed).

FIGURE M-1 Sun Netra CP32x0 ARTM-HD Voltage Distribution and IPMC Sensor Mapping

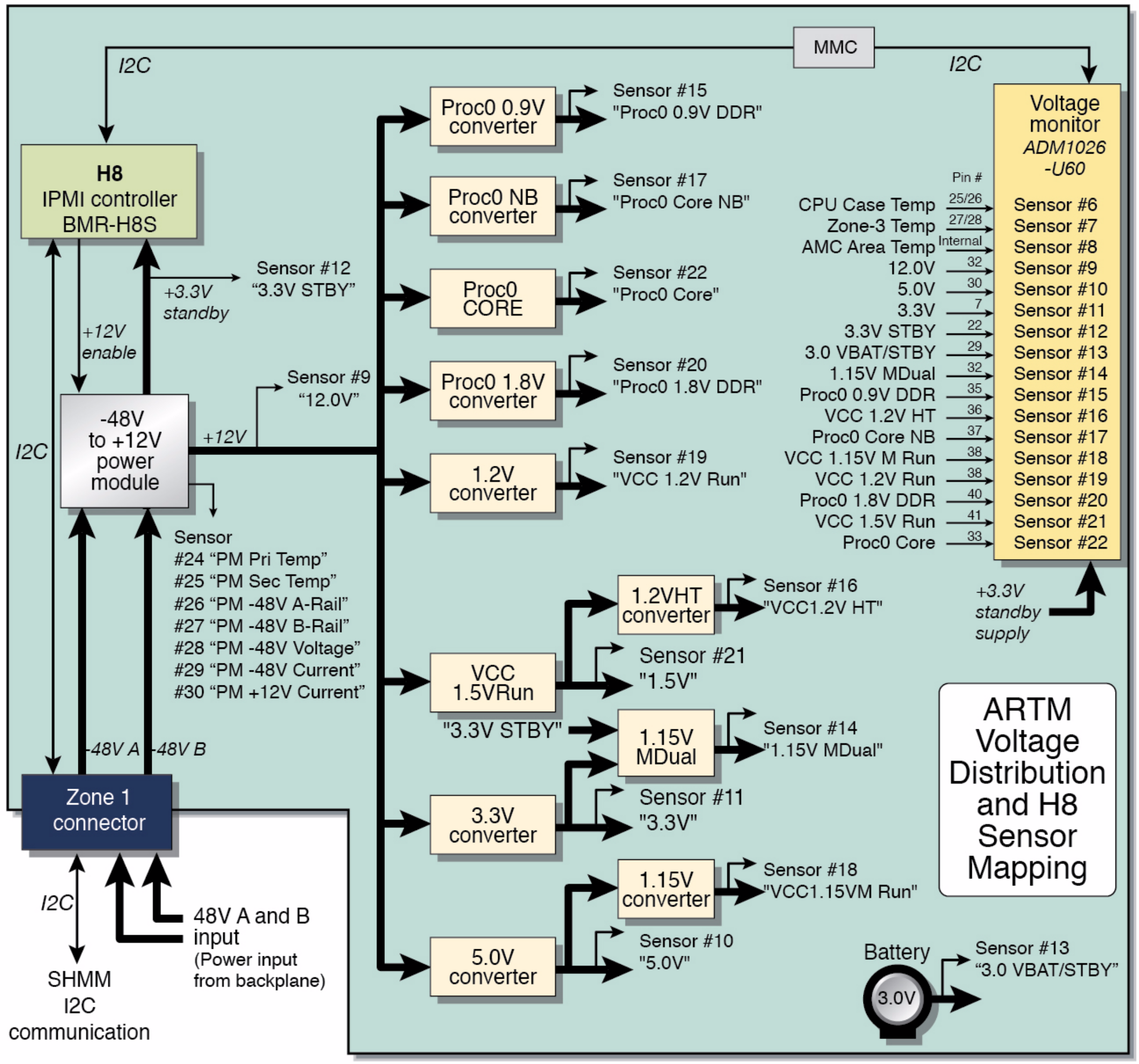


TABLE M-2 Sensor Number Conversion for Boards

Node Board	Sensor Numbering
Sun Netra CP3220 blade server w/o AMC	N = 32
Sun Netra CP3220 blade server w AMC	N = 32 plus AMC sensors (total number varies by vendor)
Sun Netra CP3260 blade server	N = 18

FIGURE M-2 Sun Netra CP32x0 ARTM-HD Temperature Monitoring and H8 Sensor Mapping

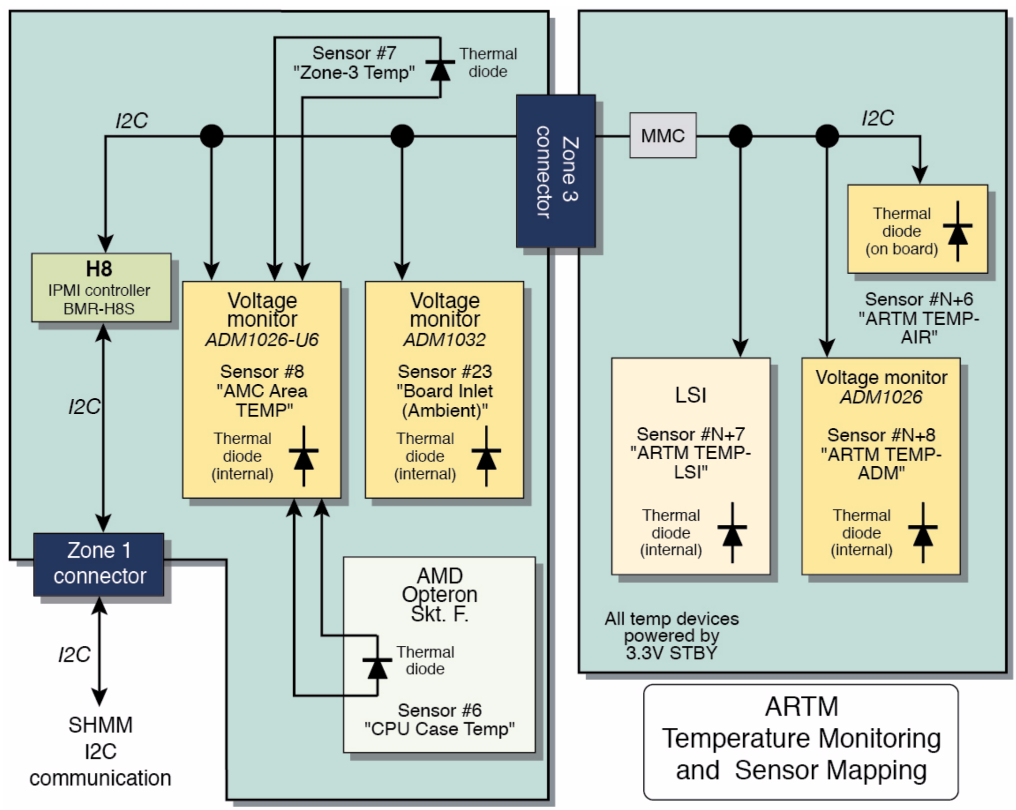


TABLE M-3 Sensor Number Conversion for Boards

Node Board	Sensor Numbering
Sun Netra CP3220 blade server w/o AMC	N = 32
Sun Netra CP3220 blade server w AMC	N = 32 plus AMC sensors (total number varies by vendor)
Sun Netra CP3260 blade server	N = 18

Glossary

Knowledge of the following terms and acronyms is useful in the administration of Oracle's Sun Netra CT900 server.

A

ATCA (Advanced Telecom Computing Architecture) Also referred to as AdvancedTCA. A series of industry standard specifications for the next generation of carrier grade communications equipment. AdvancedTCA incorporates the latest trends in high-speed interconnect technologies, next generation processors, and improved reliability, manageability and serviceability, resulting in a new blade (board) and chassis (shelf) form factor optimized for communications at the lowest cost due to standardization.

B

backup shelf management card Any shelf management card capable of assuming support for the shelf manager function.

Base channel A physical connection within the Base interface composed of up to four differential signal pairs. Each Base channel is the endpoint of a slot-to-slot connection within the base interface.

Base switch A switch that supports the Base interface. A Base switch provides 10/100/1000BASE-T packet switching services to all node boards installed in the shelf. In the Sun Netra CT900 server, the Base switches reside in physical slots 7 and 8 (logical slots 1 and 2) in the shelf and support connections to all node slots and boards. Boards that support the Fabric interface and Base interface are also referred to as “switches.”

Base interface An interface that is used to support 10/100 or 1000BASE-T connections between node boards and switches in a shelf. Midplanes are required to support the Base interface by routing four different signal pairs between all node board slots and each switch slot (in the Sun Netra CT900 server, the Base switch slots are physical slots 7 and 8, logical slots 1 and 2).

D

data transport interface A collection of point-to-point interfaces and bused signals intended to provide interconnect among the payloads on switches and node boards.

Dual Star topology An interconnect fabric topology in which two switch resources provide redundant connections to all end points within the network. A pair of switches provide redundant interconnects between node boards.

E

Electronic Keying or E-Keying A protocol used to describe the compatibility between the Base interface, Fabric interface, update channel interface, and synchronization clocks connections of front boards.

ETSI European Telecommunications Standards Institute.

F

Fabric channel A Fabric channel is comprised of two rows of signal pairs for a total of eight signal pairs per channel. Thus, each connector supports up to five channels available for board-to-board connectivity. A channel may also be viewed as being comprised of four 2-pair ports.

Fabric interface A Zone 2 interface that provides 15 connections per board or slot, each comprising up to 8 differential signal pairs (channels) supporting connections with up to 15 other slots or boards. Midplanes can support the Fabric interface in a variety of configurations including Full Mesh and Dual Star topologies. Boards that support the Fabric interface can be configured as fabric node boards, fabric switches, or mesh-enabled boards. Board implementations of the Fabric interface are defined by the PICMG 3.x subsidiary specifications.

field-replaceable unit (FRU) From a service point of view, the smallest irreducible elements of a server. Examples of FRUs are disk drives, I/O cards, and power entry modules. Note that a server, with all of its cards and other components, is not a FRU. However, an empty server is a FRU.

frame A physical or logical entity that can contain one or more shelves. Also called a rack, or, if enclosed, a cabinet.

front board A board that conforms to PICMG 3.0 mechanicals (8U x280mm), including a PCB and a panel. A front board connects with the Zone 1 and Zone 2 midplane connectors. It can optionally connect with a Zone 3 midplane connector or directly to a rear transition module connector and is installed into the front position in the shelf.

Full channel A Fabric channel connection that uses all eight differential signal pairs between end-points.

Full Mesh topology A Full Mesh configuration that can be supported within the Fabric interface to provide one dedicated channel of connectivity between each pair of slots within a shelf. Full Mesh-configured midplanes are capable of supporting mesh-enabled boards or switches and node boards installed in a Dual Star arrangement.

H

hot-swap The connection and disconnection of peripherals or other components without interrupting system operation. This facility may have design implications for both hardware and software.

I

I^2C Inter-integrated circuit bus. A multi-master, 2-wire serial bus used as the basis for current IPMBs.

IPMB (Intelligent Platform Management Bus) The lowest level hardware management bus as described in the Intelligent Platform Management Bus Communications Protocol specification.

IPMB-0 hub A hub device that provides multiple radial IPMB-0 links to various FRUs in the system. For example, an IPMB-0 hub is present in an ShMC that has radial IPMB-0 links.

IPMB-0 link With radial topology, the physical IPMB-0 segment between an IPMB-0 segment between an IPMB-0 hub and a single FRU. Each IPMB-0 link on an IPMB-0 hub is usually associated with a separate IPMB-0 sensor. An IPMB-0 link can also connect in a bused topology to multiple FRUs.

IPM controller (IPMC) The portion of a FRU that interfaces to the ATCA IPMB-0 and represents that FRU and any device subsidiary to it.

IPMI (Intelligent Platform Management Interface) A specification and mechanism for providing inventory management, monitoring, logging, and control for elements of a computer system. As defined in Intelligent Platform Management Interface specification.

L

logic ground A shelf-wide electrical net used on boards and midplanes as a reference and return path for logic-level signals that are carried between boards.

M

Mesh Enabled board A board that provides connectivity to all other boards within the midplane. Mesh Enabled boards support the Fabric interface and can also support the Base interface. Mesh Enabled boards can use 2 to 15 Fabric interface channels (typically all 15 channels) to support direct connections to all other boards in the shelf. The number of channels supported dictate the maximum number of boards that can be connected to within a shelf. Mesh Enabled boards that do not use the Base interface can be installed in the lowest available logical slot. Mesh Enabled boards supporting the Base interface can be Base switches, in which case they can support Base channels 1 and 2 and can be installed into logical slots 3 to 16. Boards supporting the Base interface use Base channels 1 and 2 only to support 10/100/1000BASE-T Ethernet.

midplane The functional equivalent of a backplane. The midplane is secured to the back of the server. The CPU card, I/O cards, and storage devices connect to the midplane from the front, and the rear transition modules connect to the midplane from the rear.

N

NEBS (Network Equipment/Building System) A set of requirements for equipment installed in telecommunications control offices in the United States. These requirements cover personnel safety, protection of property, and operational continuity. NEBS testing involves subjecting equipment to various vibration stresses, fire, and other environmental and quality metrics. There are three levels of NEBS compliance, each a superset of the preceding. NEBS level 3, the highest level, certifies that a piece of equipment can be safely deployed in an "extreme environment." A telecommunications central office is considered an extreme environment.

The NEBS standards are maintained by Telcordia Technologies, Inc., formerly Bellcore.

node board A board intended for use in a star topology midplane that has connectivity to a switch within the midplane. Node boards can support either or both the Base interface and Fabric interface. Boards supporting the Fabric interface use Fabric channels 1 and 2. Boards supporting the Base interface use Base channels 1 and 2 only to support 10/100/1000BASE-T Ethernet.

node slot A slot in the midplane that supports only node boards. A node slot is not capable of supporting a switch, thus a node board can never occupy logical slots 1 and 2. Node slots apply only to midplanes designed to support star topologies. Node slots support both the Base interface and Fabric interface. Typically, a node slot supports two or four Fabric channels and Base channels 1 and 2. Each two channel node slots establish connections to logical slots 1 and 2, respectively. Four channel node slots establish connections to logical slots 1, 2, 3, and 4, respectively.

P

PCI (Peripheral Component Interconnect) A standard for connecting peripherals to a computer. It runs at 0–33 MHz and carries 32 bits at a time over a 124-pin connector or 0-66MHz and carries 64 bits over a 188-pin connector. An address is sent in one cycle followed by one word of data (or several in burst mode).

Technically, PCI is a synchronous bus. It includes buffers to decouple the CPU from relatively slow peripherals and allow them to operate asynchronously. You can have a local PCI bus on a board or plug in PCI cards that adhere to the PCI specification. It is not asynchronous, because all devices operate on one common clock.

physical address An address that defines the physical slot location of a FRU. A physical address consists of a site type and site number.

PICMG (PCI Industrial Computer Manufacturers Group) A consortium of companies who develop open specifications for telecommunications and industrial computing applications, including the CompactPCI standard.

R

rear-access A configuration option for the Sun Netra CT900 server in which all of the cables come out from the back of the shelf.

rear transition module A card used only on the rear-access models of the Sun Netra CT900 server to extend the connectors to the back of the shelf.

Reliability, Availability, Serviceability (RAS) A hardware and software feature that implements or improves the reliability, availability and serviceability of a server.

S

shelf A collection of components that consists of the midplane, front boards, cooling devices, rear transition modules, and power entry modules. The shelf was historically known as a chassis.

shelf address A variable length, variable format descriptor of up to 20 bytes in length that provides a unique identifier for each shelf within a management domain.

shelf ground A safety ground and earth return that is connected to the frame and is available to all boards.

Shelf Manager The entity in the system that is responsible for managing the power, cooling, and interconnects (with Electronic Keying) in an AdvancedTCA shelf. The Shelf Manager also routes messages between the System Manager Interface and IPMB-0, provides interfaces to system repositories, and responds to event messages. The Shelf Manager can be partially or wholly deployed on the ShMC or System Manager Hardware.

ShMC (Shelf Management Controller) An IPMC that is also capable of supporting the functions required of the shelf manager.

SNMP Simple Network Management Protocol.

star topology A midplane topology having one or more hub slots providing connectivity among the supported node slots.

switch A board intended for use in a star topology midplane that provides connectivity to a number of node boards within the midplane. Switches can support either or both the Base interface and Fabric interface. Boards utilizing the Fabric interface typically provide switching resources to all 15 available Fabric channels. Switches supporting the Base interface are installed into logical slots 1 and 2 and use all 16 Base channels to provide 10/100/1000BASE-T Ethernet switching resources to up to 14 node boards and the other switch. One Base channel is assigned to support a connection to the shelf management card.

switch slot In a star topology midplane, switch slots must reside in logical slots 1 and 2. Switch slots support both the Base interface and Fabric interface. Switch slots located in logical sots 1 and 2 are capable of supporting both Base interface and Fabric interface switches. Logical slots 1 and 2 are always switch slots regardless of the fabric topology. These slots support up to 16 Base channels and up to 15 Fabric channels each.

system A managed entity that can include one or more of the following components: node and switches, shelves, and frames.

READ AND DELETE

1. To display the "Beta Draft" footer , show the BetaDraft conditional setting.
2. After Beta, hide the BetaDraft conditional setting.

Index