

Sun Netra CT900 服务器

软件开发指南



文件号码: E22016-01
2011 年 3 月

版权所有©2008, 2009, 2010, 2011, Oracle 和/或其附属公司。保留所有权利。

本软件和相关文档是根据许可证协议提供的, 该许可证协议中规定了关于使用和公开本软件和相关文档的各种限制, 并受知识产权法的保护。除非在许可证协议中明确许可或适用法律明确授权, 否则不得以任何形式、任何方式使用、拷贝、复制、翻译、广播、修改、授权、传播、分发、展示、执行、发布或显示本软件和相关文档的任何部分。除非法律要求实现互操作, 否则严禁对本软件进行逆向工程设计、反汇编或反编译。

此文档所含信息可能随时被修改, 恕不另行通知, 我们不保证该信息没有错误。如果贵方发现问题, 请书面通知我们。

如果将本软件或相关文档交付给美国政府, 或者交付给以美国政府名义获得许可证的任何机构, 必须符合以下规定:

U.S. GOVERNMENT RIGHTS. Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

本软件或硬件是为了在各种信息管理应用领域内的一般使用而开发的。它不应被应用于任何存在危险或潜在危险的应用领域, 也不是为此而开发的, 其中包括可能会产生人身伤害的应用领域。如果在危险应用领域内使用本软件或硬件, 贵方应负责采取所有适当的防范措施, 包括备份、冗余和其它确保安全使用本软件或硬件的措施。对于因在危险应用领域内使用本软件或硬件所造成的一切损失或损害, Oracle Corporation 及其附属公司概不负责。

Oracle 和 Java 是 Oracle 和/或其附属公司的注册商标。其他名称可能是各自所有者的商标。

AMD、Opteron、AMD 徽标以及 AMD Opteron 徽标是 Advanced Micro Devices 的商标或注册商标。Intel 和 Intel Xeon 是 Intel Corporation 的商标或注册商标。所有 SPARC 商标均是 SPARC International, Inc 的商标或注册商标, 并应按照许可证的规定使用。UNIX 是通过 X/Open Company, Ltd 授权的注册商标。

本软件或硬件以及文档可能提供了访问第三方内容、产品和服务的方式或有关这些内容、产品和服务的信息。对于第三方内容、产品和服务, Oracle Corporation 及其附属公司明确表示不承担任何种类的担保, 亦不对其承担任何责任。对于因访问或使用第三方内容、产品或服务所造成的任何损失、成本或损害, Oracle Corporation 及其附属公司概不负责。



请回收



Adobe PostScript

目录

前言 xvii

1. 编程环境 1

Sun Netra CT900 服务器 1

硬件说明 2

机框 2

机框报警面板 2

机框管理卡 2

交换机 3

软件说明 4

管理框架 6

ATCA 中的智能平台管理概述 6

机框管理器和机框管理卡 8

机框管理器特性 8

硬件资源分层结构 9

系统管理员 (System Administrator) 接口选项 9

OpenHPI 10

OpenHPI 概述 10

实体 10

资源 10

会话	11
域	11
管理应用程序框架	11
2. 简单网络管理协议	15
SNMP 概述	16
ShMM SNMP 体系结构	17
ShMM SNMP 代理配置	18
了解 MIB 变量描述	18
SAF-HPI MIB	18
SAF-HPI MIB 表分层结构	19
实体表	19
传感器表	21
事件表	22
事件日志表	23
配置 hpiSubagent	24
▼ 启用读写访问	24
▼ 启用子代理的 SNMP 版本 3 使用	24
SNMP 使用示例	26
获取资源相关信息	26
▼ 查看域的所有资源的所有信息	26
▼ 查看域的所有资源的数据列	27
▼ 查看域的特定资源	29
获取属性相关信息	29
▼ 查看域的所有资源的 RDR 条目	30
▼ 查看域的所有资源的 RDR 表中的列	30
▼ 查看域的资源 RDR 表中的列	31
▼ 查看每个 RDR 条目表示的内容	31
▼ 查看 RDR EntryID 的 RDR 表中的列	31

获取传感器相关信息 32

- ▼ 查看域中所有资源的所有传感器的相关信息 32
- ▼ 查看域的所有资源的传感器表中的列 33
- ▼ 查看某资源的传感器表中的列 33
- ▼ 查看某资源的所有传感器的基本度量单位 34
- ▼ 查看某资源的传感器的传感器表中的列 34
- ▼ 查看域的所有资源的所有传感器的当前状态 34
- ▼ 查看域的所有资源的当前传感器状态表中的列 34
- ▼ 查看某资源的当前传感器状态表中的列 35
- ▼ 查看某资源的传感器的当前传感器状态表中的列 35

获取和设置传感器阈值 36

- ▼ 查看域的所有资源的所有传感器的传感器临界阈值上限表中的所有信息 36
- ▼ 查看所有资源的所有传感器的传感器临界阈值上限表中的列 37
- ▼ 查看某资源的所有传感器的传感器临界阈值上限表中的列 37
- ▼ 查看某资源的传感器的传感器临界阈值上限表中的列 37
- ▼ 为传感器设置传感器阈值 38

获取和设置有关控制的信息 38

- ▼ 查看所有资源的所有模拟控制的相关信息 39
- ▼ 查看所有资源的模拟控制表的列 39
- ▼ 查看某资源的模拟控制表的列 40
- ▼ 查看特定资源的所有模拟控制的控制状态 40
- ▼ 查看某资源的控制的模拟控制表的列 42
- ▼ 设置模拟控制的状态 42

获取有关 IDR 的信息 43

saHpiInventoryTable 信息 43

- ▼ 查看域的所有资源的高级别库存信息 43
- ▼ 查看域的所有资源的库存表的列 44
- ▼ 查看某资源的库存表的列 44
- ▼ 查看与资源和 IDR 条目 ID 对应的库存表的列 45

saHpiAreaTable 信息 45

- ▼ 查看域的所有资源的所有区域的所有信息 45
- ▼ 查看所有资源的区域表的列 45
- ▼ 查看某资源的区域表的列 46
- ▼ 查看某资源的 IDR 的列 47
- ▼ 查看某资源的 IDR 的区域的列 47

saHpiFieldTable 信息 47

- ▼ 查看所有字段的所有信息 48
- ▼ 查看所有字段的字段表的列 48
- ▼ 查看某资源的所有字段的列 48
- ▼ 查看区域的所有字段的列 48
- ▼ 查看字段的列 49

使用 HPI 子代理管理自定义数据记录 49

- ▼ 查看特定资源的所有区域的区域类型 49
- ▼ 查看特定资源的特定区域的所有字段的字段文本 50
- ▼ 修改 CDR 内容 50
- ▼ 删除特定资源的特定区域的特定 CDR 字段 51
- ▼ 检查特定资源的特定区域中的字段数量 51

使用事件日志和事件表 51

saHpiEventTable 52

- ▼ 查看事件表中的所有信息 52
- ▼ 查看事件表中的列 52

saHpiSensorEventTable 52

- ▼ 查看传感器事件表中的所有信息 53
- ▼ 查看传感器事件表的列 53
- ▼ 查看某资源的传感器事件表的列 53
- ▼ 查看某资源的传感器的传感器事件表的列 54

saHpiEventLogInfoTable	54
▼ 查看域的所有资源的事件日志信息	55
▼ 查看事件日志信息表的列	56
▼ 查看某资源的事件日志信息表的列	56
saHpiEventLogTable	56
▼ 查看所有资源的 saHpiEventLogTable 中的信息	57
▼ 查看所有资源的 saHpiEventLogTable 的列	57
▼ 查看所有资源的所有事件的事件日志行指针	57
▼ 查看某资源的事件日志表的列	58
▼ 查看某资源和事件的事件日志表的列	58
saHpiSensorEventLogTable	58
▼ 查看传感器事件日志表中的所有信息	58
▼ 查看传感器事件日志表的列	59
▼ 查看某资源的传感器事件日志表的列	59
▼ 查看某资源的传感器的传感器事件日志表的列	59
清除事件日志条目	60
▼ 从系统事件日志中清除特定资源的条目	60
▼ 从域事件日志中清除特定资源的事件日志	60
配置陷阱和处理通知	61
陷阱配置	61
▼ 配置 SNMP 版本 1 的陷阱	61
▼ 配置 SNMP 版本 2 的陷阱	61
通知处理	62
示例：冷启动陷阱	62
示例：热交换 1	63
示例：热交换 2	64
示例：超过温度传感器阈值	65

3. 智能平台管理接口驱动程序 67

IPMI 概述 68

操作系统支持和 IPMI 安装 68

▼ 安装 IPMI 驱动程序 69

IPMI 用户界面 69

IPMI 编程示例 70

获取设备 ID 70

对 LED 指示灯进行编程 72

IPMI 命令 77

Sun ATCA 板上支持的 IPMI/ATCA 命令 77

Sun 和 OEM IPMI 命令 83

Set AMC timeout params (设置 AMC 超时参数), 操作码: 0xF1,
网络功能: 0x2E 86

Get AMC timeout parameters (获取 AMC 超时参数), 操作码 0xF0,
网络功能: 0x2E 87

Set boot page (设置引导页面), 操作码 0x82, 网络功能: 0x2E 87

Get boot page (获取引导页面), 操作码 0x81, 网络功能: 0x2E 88

Set front panel reset button state (设置前面板复位按钮状态),
操作码 0x83, 网络功能: 0x2e 88

Get front panel reset button state (获取前面板复位按钮状态),
操作码 0x84, 网络功能: 0x2E 89

Set IPMC control bits (设置 IPMC 控制位), 操作码 0xE9, 网络
功能: 0x2E 89

Get IPMC control bits (获取 IPMC 控制位), 操作码 0xE8, 网络
功能: 0x2E 90

Set management port (设置管理端口), 操作码 0x9B, 网络功能:
0x2E 91

Get management port (获取管理端口), 操作码 0x9C, 网络功能:
0x2E 91

Get NIC IPMI PT firmware version (获取 NIC IPMI PT 固件版本),
操作码 0x87, 网络功能: 0x2E 92

Get version (获取版本), 操作码 0x80, 网络功能: 0x2E 92

Get Status (获取状态), 操作码 0x00, 网络功能: 0x2E 93

Graceful Payload Reset (正常有效载荷复位), 操作码 0x11, 网络功能: 0x2E 94

Set Payload Shutdown Timeout (设置有效载荷关闭超时), 操作码 0x16, 网络功能: 0x2E 95

Get Payload Shutdown Timeout (获取有效载荷关闭超时), 操作码 0x15, 网络功能: 0x2E 95

Set SOL fail over link change timeouts (设置 SOL 故障转移链路更改超时), 操作码 0xE7, 网络功能: 0x2E 96

Get SOL fail over link change timeouts (获取 SOL 故障转移链路更改超时), 操作码 0xE6, 网络功能: 0x2E 96

Set Thermal Trip (设置极限温度), 操作码 E5, 网络功能: 0x2E 97

Get Thermal Trip (获取极限温度), 操作码 0xE4, 网络功能: 0x2E 98

Set XAUI mux control (设置 XAUI mux 控制), 操作码 0x95, 网络功能: 0x2E 98

Get XAUI mux control (获取 XAUI mux 控制), 操作码 0x96, 网络功能: 0x2E 99

- A. 实体路径 101**
- B. 资源数据记录 105**
- C. Sun Netra CP3140 SNMP MIB 对象和陷阱 125**
- D. 传感器对应表和故障隔离 215**
 - 机箱传感器 216
 - PEM 传感器 226
 - PEM 传感器故障说明 227
 - 风扇托盘传感器 228
 - SAP 传感器 229
- E. ShMM 传感器对应表和故障隔离 231**
 - ShMM 传感器 232

F.	Sun Netra CP3020 刀片服务器传感器对应表和故障隔离	235
	Sun Netra CP3020 刀片服务器传感器列表	236
G.	Sun Netra CP3220 刀片服务器传感器对应表和故障隔离	241
	Sun Netra CP3220 刀片服务器传感器列表	242
H.	Sun Netra CP3060 刀片服务器传感器对应表和故障隔离	249
	Sun Netra CP3060 刀片服务器传感器列表	250
I.	Sun Netra CP3250 刀片服务器传感器对应表和故障隔离	257
	Sun Netra CP3250 刀片服务器传感器列表	258
J.	Sun Netra CP3260 刀片服务器传感器对应表和故障隔离	263
	Sun Netra CP3260 刀片服务器传感器列表	264
K.	Sun Netra CP3270 刀片服务器传感器对应表和故障隔离	271
	Sun Netra CP3270 刀片服务器传感器列表	272
L.	Netra SPARC T3-1BA 刀片服务器传感器对应表和故障隔离	277
	Netra SPARC T3-1BA 刀片服务器传感器列表	278
M.	Sun Netra CP32x0 ARTM 传感器对应表和故障隔离	283
	Sun Netra CP32x0 ARTM 传感器列表	284
	词汇表	289
	索引	295

图

-
- 图 1-1 Sun Netra CT900 服务器中软件与硬件接口的逻辑表示 5
- 图 1-2 ATCA 机框示例 7
- 图 1-3 OpenHPI 体系结构 11
- 图 1-4 HPI 应用程序、OpenHPI 守护进程和 RMCP 服务器的关系 12
- 图 1-5 HPI 应用程序和 OpenIPMI 驱动程序的关系 13
- 图 2-1 SNMP 管理关系概述 16
- 图 2-2 SNMP 体系结构 17
- 图 2-3 实体表关系 20
- 图 2-4 传感器表关系 21
- 图 2-5 事件表关系 22
- 图 2-6 事件日志表关系 23
- 图 D-1 机箱级传感器位置 - 前面 224
- 图 D-2 机箱级传感器位置 - 后面 225
- 图 D-3 PEM 传感器 226
- 图 D-4 风扇托盘传感器 228
- 图 D-5 SAP 传感器 229
- 图 F-1 Netra CP3020 电压分配和 H8 传感器对应关系 239
- 图 F-2 Sun Netra CP3020 刀片服务器和 RTM 温度监视及 H8 传感器对应关系 240
- 图 G-1 Sun Netra CP3220 电压分配和 H8 传感器对应关系 247
- 图 G-2 Sun Netra CP3220 刀片服务器和 RTM 温度监视及 H8 传感器对应关系 248

图 H-1	Sun Netra CP3060 电压分配和 H8 传感器对应关系	254
图 H-2	Sun Netra CP3060 温度监视和 H8 传感器对应关系	255
图 J-1	Sun Netra CP3260 电压分配和 IPMC 传感器对应关系	268
图 J-2	Sun Netra CP3260 温度监视和 H8 传感器对应关系	269
图 M-1	Sun Netra CP32x0 ARTM-HD 电压分配和 IPMC 传感器对应关系	285
图 M-2	Sun Netra CP32x0 ARTM-HD 温度监视和 H8 传感器对应关系	286

表

表 1-1	适用于系统管理员的 Sun Netra CT 服务器软件	4
表 2-1	SNMP 通知	62
表 3-1	IPMI 全局设备命令, 网络功能: 应用程序 (0x06/0x07)	77
表 3-2	BMC 监视程序计时器命令, 网络功能: 应用程序 (0x06/0x07)	78
表 3-3	BMC 设备和消息传送命令, 网络功能: 应用程序, (0x06/0x07)	78
表 3-4	事件命令, 网络功能: 传感器/事件, (0x04/0x05)	79
表 3-5	传感器设备命令, 网络功能: 传感器/事件, (0x04/0x05)	79
表 3-6	FRU 设备命令, 网络功能: 存储, (0xA/0xB)	81
表 3-7	ATCA 命令, 网络功能: ATCA (0x2C/0x2D)	81
表 3-8	Sun OEM 命令, 网络功能: OEM, (0x2E/0x2F)	83
表 A-1	资源表	101
表 B-1	Sun Netra CP3010 板资源数据记录	106
表 B-2	Sun Netra CP3020 板资源数据记录	108
表 B-3	Sun Netra CP3060 板资源数据记录	110
表 B-4	Sun Netra CP3140 交换机资源数据记录	112
表 B-5	Sun Netra CP3240 交换机资源数据记录	114
表 B-6	Sun Netra CP3220 板资源数据记录	118
表 B-7	Sun Netra CP3260 板资源数据记录	120
表 B-8	Sun Netra CP32x0 双 SAS 存储高级后部转换模块 (ARTM-HD) 资源数据记录	122
表 B-9	Sun Netra CP3250 板资源数据记录	123

表 C-1	802.3AD 链路聚合 MIB	125
表 C-2	RFC 2934 PIM-SM/DM MIB	128
表 C-3	RFC 2933 IGMP MIB	130
表 C-4	RFC 2932 IPv4 多播路由 MIB	131
表 C-5	RFC 2819 RMON MIB	132
表 C-6	RFC 2787 VRRP MIB	139
表 C-7	RFC 2737 实体 MIB (版本 2)	141
表 C-8	RFC 2674 VLAN MIB (P 网桥 MIB、Q 网桥 MIB)	142
表 C-9	RFC 2620 Radius 记帐客户机 MIB	147
表 C-10	RFC 2618 Radius 验证客户机 MIB	148
表 C-11	RFC 2233 接口 MIB	149
表 C-12	RFC 1850 OSPF MIB	151
表 C-13	RFC 1724 RIPv2 MIB	155
表 C-14	RFC 1657 BGP4 MIB	156
表 C-15	RFC 1643 以太网 MIB	158
表 C-16	RFC 1493 网桥 MIB	159
表 C-17	RFC 1213 Mib-2 MIB	161
表 C-18	POWER-ETHERNET-MIB	168
表 C-19	LVL7-POWER-ETHERNET-MIB	169
表 C-20	IEEE8021-PAE-MIB dot1x MIB	170
表 C-21	FASTPATH-SECURITY-MIB	173
表 C-22	FASTPATH-MULTICAST-MIB	173
表 C-23	FASTPATH-MGMT-SECURITY-MIB	175
表 C-24	FASTPATH-DHCPSEVER-PRIVATE-MIB	175
表 C-25	FASTPATH-BGP-MIB	178
表 C-26	FASTPATH 交换 MIB	179
表 C-27	FASTPATH 路由 MIB	191
表 C-28	FASTPATH Radius MIB	195
表 C-29	FASTPATH QOS DiffServ MIB	196
表 C-30	FASTPATH QOS DiffServ 扩展 MIB	201

表 C-31	FASTPATH QOS BW MIB	203
表 C-32	FASTPATH QOS ACL MIB	204
表 C-33	FASTPATH-INVENTORY-MIB	205
表 C-34	draft-ietf-idmr-dvmrp-mib-11 DVMRP MIB	206
表 C-35	RFC 3289 DiffServ MIB	208
表 D-1	传感器对应表	216
表 D-2	PEM 传感器故障说明	227
表 D-3	输入 1 的故障状况说明	227
表 E-1	ShMM 传感器对应表	232
表 F-1	Sun Netra CP3020 刀片服务器传感器	236
表 G-1	Sun Netra CP3220 刀片服务器传感器	242
表 H-1	Sun Netra CP3060 刀片服务器传感器列表	250
表 I-1	Sun Netra CP3250 刀片服务器传感器列表	258
表 J-1	Sun Netra CP3260 刀片服务器传感器列表	264
表 K-1	Sun Netra CP3270 刀片服务器传感器列表	272
表 L-1	Netra SPARC T3-1BA 刀片服务器传感器列表	278
表 M-1	Sun Netra CP32x0 ARTM-HD 传感器列表	284
表 M-2	板的传感器编号转换	286
表 M-3	板的传感器编号转换	287

前言

本文档概括介绍了在 Oracle Sun Netra CT900 服务器中实施的高级电信计算体系结构（AdvancedTCA 或 ATCA）硬件和软件。

本指南介绍了 SNMP 公共域软件等系统功能的配置和用法，以及用于编写如下应用程序的环境：

- 使用 HPI 框架的应用程序
- 在刀片上使用 IPMI 驱动程序的应用程序

本指南还提供了访问平台编译器以便编写如下应用程序所需的信息：

- 编写使用 OpenHPI API 的应用程序（如第 1 章所述）
 - 编写使用 IPMI 驱动程序的应用程序（如第 3 章所述）
-

使用 UNIX 命令

本文档中不包含有关基本 UNIX® 命令以及关闭系统、启动系统和配置设备等步骤的信息。欲获知此类信息，请参阅以下文档：

- 系统附带的软件文档
- Oracle Solaris 操作系统的有关文档，其 URL 如下：

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

Shell 提示符

Shell	提示符
C shell	<i>machine-name%</i>
C shell 超级用户	<i>machine-name#</i>
Bourne shell 和 Korn shell	\$
Minix shell 超级用户	#

相关文档

下表中列出了 Sun Netra CT900 服务器文档。除《Important Safety Information for Sun Hardware Systems》外，您可以从以下位置联机获取所列出的所有文档：

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

应用	书名	格式	所在位置
入门	《Sun Netra CT900 服务器入门指南》	印刷品 PDF	产品套件 联机提供
最新信息	《Sun Netra CT900 Server Product Notes》	PDF HTML	联机提供
概述	《Sun Netra CT900 服务器概述》	PDF HTML	联机提供
安装	《Sun Netra CT900 服务器安装指南》	PDF HTML	联机提供
升级	《Sun Netra CT900 Server Upgrade Guide》	PDF HTML	风扇托盘套件 联机提供
服务	《Sun Netra CT900 Server Service Manual》	PDF HTML	联机提供

应用	书名	格式	所在位置
参考	《Sun Netra CP3140 Switch Software Reference Manual》	PDF HTML	联机提供
安全	《Sun Netra CT900 Server Safety and Compliance Guide》	PDF HTML	联机提供
安全	《Important Safety Information for Sun Hardware Systems》（仅提供印刷版本）	印刷品	产品套件

您可能需要参考以下产品的有关文档来获取其他信息：Oracle Solaris OS、OpenBoot PROM 固件、Sun Netra CP3010 板、Sun Netra CP3020 板、Sun Netra ATCA CP3220 刀片服务器、Sun Netra ATCA CP3060 刀片服务器和 Sun Netra ATCA CP3260 刀片服务器。您可以联机获取这些文档。

文档、支持和培训

这些 Web 站点提供了一些其他资源：

- 文档 <http://www.oracle.com/technetwork/indexes/documentation/index.html>
- 支持 <https://support.oracle.com>
- 培训 <https://education.oracle.com>

第1章

编程环境

本章概述了软件环境，而软件环境是为 Sun Netra CT900 服务器开发应用程序的基础：

- 第 1 页的 “Sun Netra CT900 服务器”
 - 第 2 页的 “硬件说明”
 - 第 4 页的 “软件说明”
 - 第 6 页的 “管理框架”
-

Sun Netra CT900 服务器

Netra CT 900 服务器是基于底板的、可进行机架装配的高级电信计算体系结构（AdvancedTCA® 或 ATCA）包交换服务器。

Netra CT 900 服务器符合以下规范：

- PICMG® 3.0 修订版 2.0 AdvancedTCA 规范
- PICMG 3.1 修订版 1.0 AdvancedTCA 规范

Netra CT 900 服务器的硬件组件可细分为四部分：

- 机框
- 机框报警面板
- 机框管理卡
- 交换机

注 – 高级电信计算体系结构® (Advanced Telecom Computing Architecture®, ATCA) 采用了术语**机框**以与传统意义上电信的典型规范相一致，而术语**机箱**在本质上与其用意相同。

硬件说明

本节介绍了 Sun Netra CT900 服务器的主要组件。

机框

机框具有十二个节点板插槽和一个冗余基础结构（交换机、管理、电源和冷却），使其适用于电信级通信设备和 **Internet** 应用程序。除了高可用性外，Netra CT 900 服务器还具有高度模块化、可伸缩性和可维护性等特性。

可热交换的系统组件提供了内置冗余，从而可以简化替换过程并最大限度地减少服务时间。借助冗余的机框管理卡，客户可以远程管理多个处理器板并执行机框诊断，从而增强系统的可靠性。为 PICMG 3.0/3.1 交换机保留了两个 8U 插槽。Netra CT 900 服务器可通过中间背板而无需使用电缆来路由以太网信号，从而可节省设置、维护和维修时间，同时可避免碰到传统电缆连接方法在散热方面所遇到的难题。

机框报警面板

机框报警面板 (shelf alarm panel, SAP) 是安装在机框右上方的可移除模块，位于机框中的插槽 9 至插槽 14 上方。它提供用于连接机框管理卡的串行控制台接口的连接器、Telco 报警连接器、Telco 报警 LED 指示灯、用户可定义的 LED 指示灯和报警静音按钮。

机框报警面板上的 I²C 总线设备连接到两个机框管理卡的仅主控 I²C 总线。只有活动的机框管理卡才可以访问机框报警面板。

机框管理卡

Netra CT 900 服务器有两个用于机框管理卡的专用插槽。每个机框管理卡是一种外形规格为 78 x 280 毫米的板，并带有用于机框管理夹层 (shelf management mezzanine, ShMM) 设备的 SODIMM 插槽。Netra CT 900 服务器配有辐射型 IPMB，并设计为使用两个冗余机框管理卡。机框管理卡还包含用于三个可热交换风扇托盘的风扇控制器，并可为两个交换机提供单独的以太网连接。

ShMM 的双 IPMB 接口（通过 Netra CT 900 服务器中间背板的辐射状连接）连接到 ATCA 节点板上的双 IPMB。每个机框管理卡有一个不供用户使用的以太网端口；相反，机框管理卡的以太网流量会被路由至交换机上的以太网端口。机框管理卡的串行流量和 Telco 报警流量被路由至机框报警面板上的端口和 LED 指示灯。

机框管理卡包含几个板载设备，可基于 ShMM 实现对机框不同方面的管理。这些设备包括基于 I²C 的硬件监视和控制设备，以及通用输入/输出 (General Purpose Input/Output, GPIO) 扩展设备。

交换机

Netra CT 900 服务器的交换机是一个 AdvancedTCA 3.0 和 3.1 选项 1 交换机。这意味着该交换机在一个印刷电路板 (printed circuit board, PCB) 上实现两种不同的交换式网络。通过将基本 (3.0) 网络和扩展光纤 (3.1) 网络分开，该交换机可提供独立的控制平面和数据平面。它通过 3.0 基本交换接口提供 10/100/1000BASE-T 以太网交换功能，通过 3.1 扩展交换接口提供 1000BASE-X 以太网交换功能。这两种网络都是可完全管理的，并可与功能强健的 FASTPATH 管理套件协同工作。两种网络都支持第 2 层交换和第 3 层路由。该交换机还支持后部转换模块，以便扩展与上行链路端口的连通性。

软件说明

Sun Netra CT900 服务器软件包括：

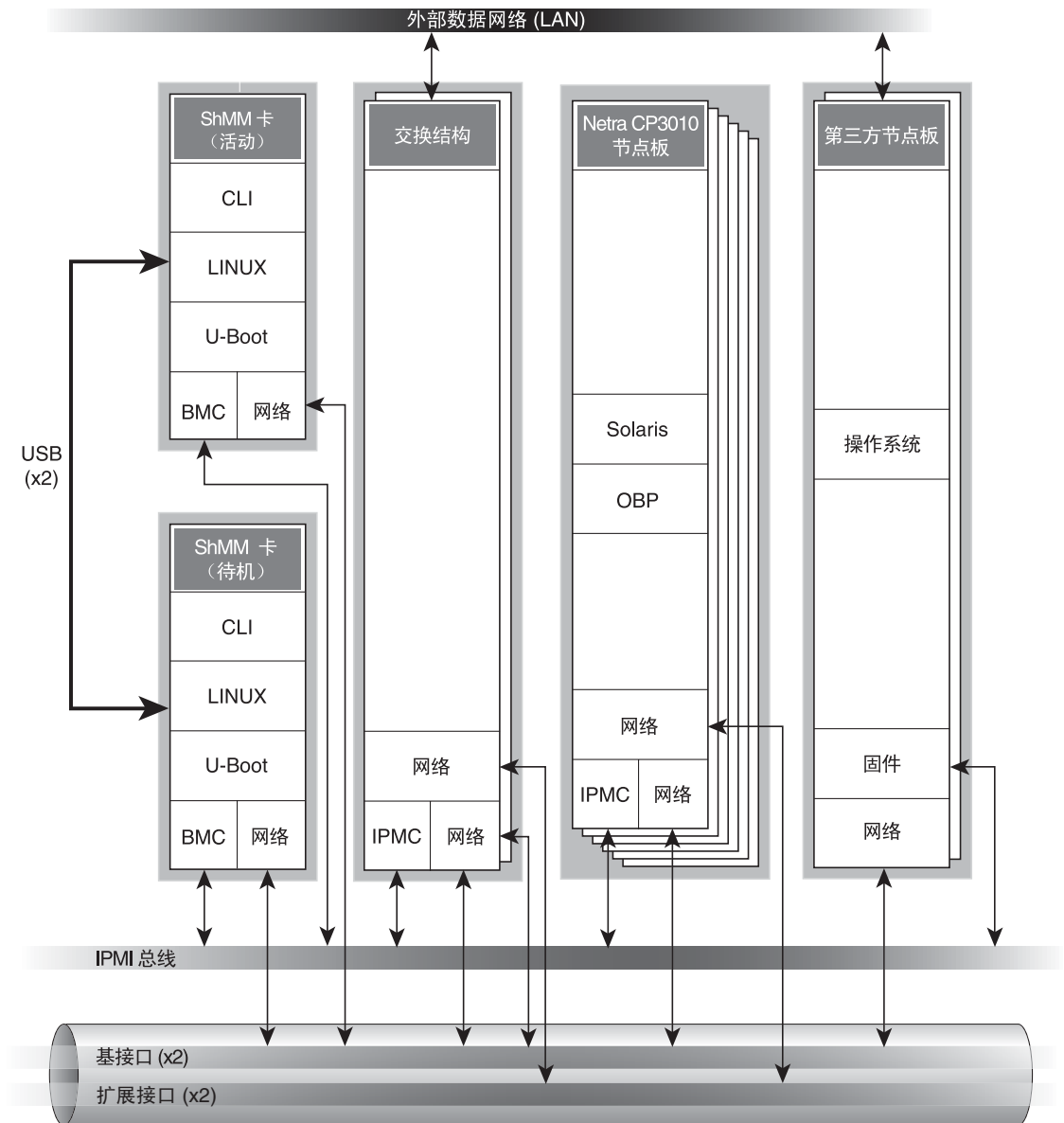
- 机框管理器
- 操作系统和应用程序
- 固件

表 1-1 对软件进行了说明，图 1-1 提供了软件与硬件的逻辑示意图。

表 1-1 适用于系统管理员的 Sun Netra CT 服务器软件

类别	名称	说明
机框管理	IPM Sentry 机框管理器	机框管理器软件在机框管理卡 (ShMM) 上运行，出厂时就已安装。它提供了远程管理控制协议 (Remote Management Control Protocol, RMCP) 以及对 IPMI 的 CLI 访问，以便管理服务器。
	命令行界面 (Command-line interface, CLI)	CLI 是机框管理器的板载用户界面。
操作系统和应用程序	Oracle Solaris 操作系统 (Solaris OS)	Solaris OS 在 Sun 支持的 ATCA 兼容节点板上运行，这些节点板类似于 Sun Netra CP3010、Sun Netra CP3020 和 CP3060 节点板。可以选择在 Sun Netra 节点板上预先安装 Solaris 10。用户可下载和安装 Solaris 10 和其他版本的 Solaris OS。
	Monta Vista Carrier Grade Linux OS	Sun Netra CP3020 也可以运行 Monta Vista Carrier Grade Linux OS。
固件	OpenBoot PROM 固件	Sun 支持的节点板（如 Sun Netra CP3010 板）上的固件，用于控制引导。它还有诊断功能。
	U-Boot	机框管理卡上的固件，用于执行开机自检 (power-on self-test, POST) 并控制机框管理卡软件的引导。
	智能平台管理控制器 (Intelligent Platform Management Controller, IPMC)	系统管理控制器固件，用于实现通过 Sun 支持的节点板（如 Sun Netra CP3010 板）上的 IPMI 控制器进行通信。

图 1-1 Sun Netra CT900 服务器中软件与硬件接口的逻辑表示



管理框架

机框管理器是面向 ATCA 产品的机框级管理解决方案。机框管理卡提供了在 ATCA 机框内运行机框管理器所必需的硬件。下面的概述重点介绍了对 ATCA 环境下使用的任一机框管理载体均通用的机框管理器以及机框管理卡的各个方面。

ATCA 中的智能平台管理概述

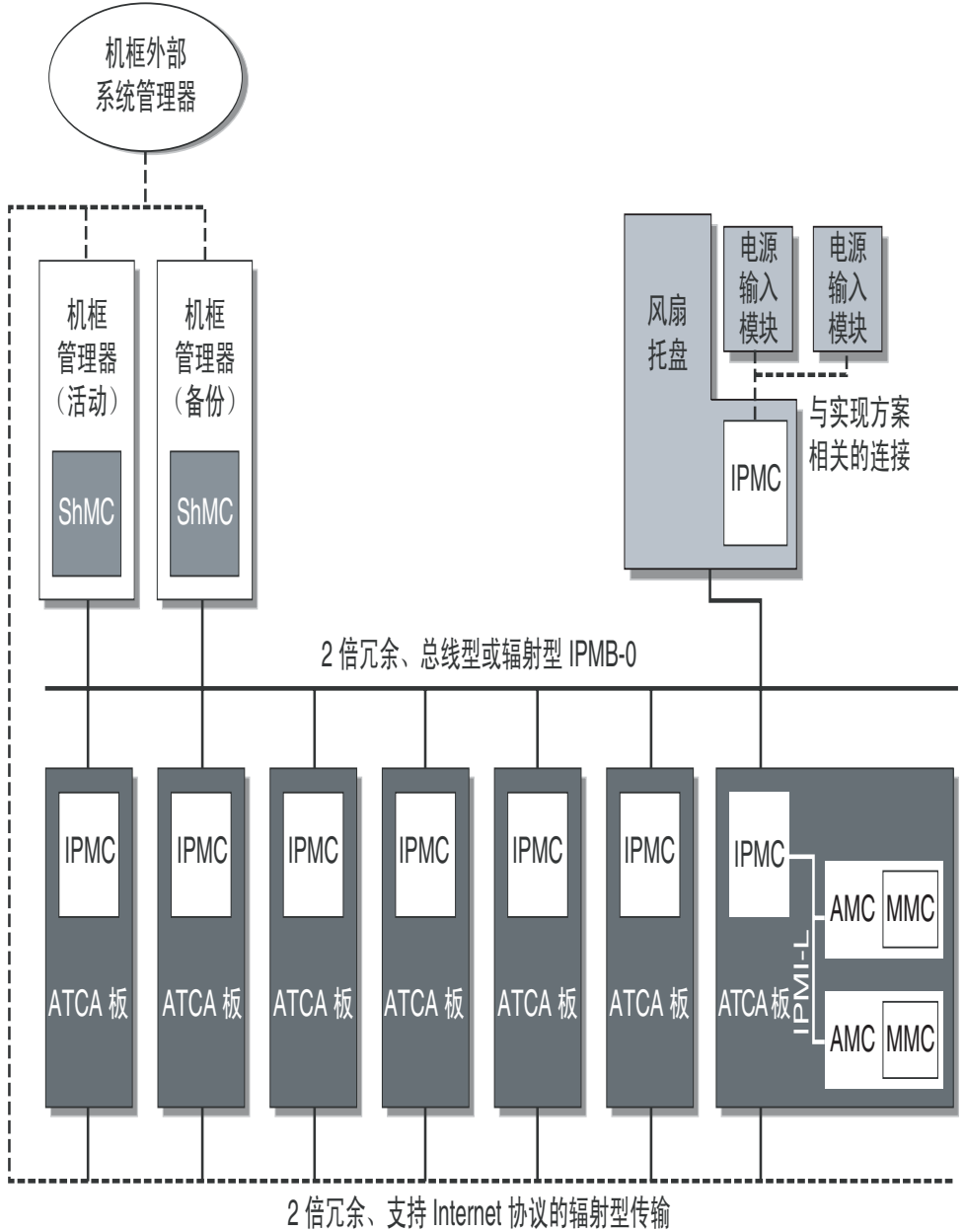
机框管理器和机框管理卡是专为 ATCA 等模块化平台设计的智能平台管理 (Intelligent Platform Management, IPM) 构建块, 其重点在于实现 FRU 动态填充和最大程度地提高服务可用性。IPMI 规范为管理此类平台奠定了坚实的基础, 但是为了提供更好的支持, 还需要重要的扩展功能。PICMG 3.0 (ATCA 规范) 定义了 IPMI 所必需的扩展功能。

AdvancedTCA 机框管理器在机框内与 IPM 控制器进行通信, 每个 IPM 控制器负责一个或多个现场可更换单元 (field replaceable unit, FRU) (如, 板、风扇托盘或电源输入模块) 的本地管理。机框内的管理通信主要通过智能平台管理总线 (Intelligent Platform Management Bus, IPMB) 进行, 而 IPMB 在 AdvancedTCA 中基于双冗余方式来实现。

PICMG 高级夹层卡 (AdvancedMC 或 AMC) 规范 AMC.0 定义了可热交换的夹层外形规格, 旨在顺畅地与 AdvancedTCA 的物理和管理体系结构相适合。

图 1-2 包括一个具有 IPMC 和两个已安装 AMC 模块的 AMC 载体, 每个 AMC 模块配有一个模块管理控制器 (Module Management Controller, MMC)。载体上的管理通信是通过 IPMB-L ("L" 代表本地) 实现的。

图 1-2 ATCA 机框示例



综合系统管理器（通常位于机框外部）可协调多个机框的活动。系统管理器通常通过以太网或串行接口与每个机框管理器进行通信。

图 1-2 显示了三个管理级别：板级、机框级和系统级。下一节介绍机框管理器软件和机框管理卡，它们实现了符合 ATCA 标准的机框管理器和机框管理控制器 (shelf management controller, ShMC)。

机框管理器和机框管理卡

机框管理器（符合 ATCA 机框管理器的要求）有两个主要职责：

- 管理和跟踪机框的 FRU 填充和通用基础结构，特别是电源、冷却和互连资源及其使用情况。在机框内部，这种管理和跟踪主要是通过智能平台管理总线 0 (Intelligent Platform Management bus 0, IPMB-0) 在机箱管理器与 IPM 控制器之间交互来实现的。
- 使综合系统管理器可通过系统管理器接口参与上述管理和跟踪，这通常是通过以太网实现的。

机框管理器软件主要用于日常任务，如打开或关闭机框电源、处理 FRU 的安装或拆卸，包括就电源和互连资源的分配进行协商，以及监视每个 FRU 的运行状况。此外，机框管理器还可以在机框出现异常时采取直接措施。例如，机框管理器为响应温度异常会提高风扇转速，如果这一步还不能满足需要，机框管理器甚至会关闭 FRU 电源来减少机框内的热负荷。

机框管理器特性

机框管理器软件的特性包括：

- 在机框管理卡上执行，该卡是一种紧凑型 SO-DIMM 模块，安装在机框的适当载板上。
- 符合 ATCA 规范。
- 通过 ATCA 指定的双冗余智能平台管理总线 (Intelligent Platform Management bus, IPMB) 监视机框内的活动。
- 接受并记录由机框内任一智能 FRU 发布的事件（反映温度和电压等异常）；根据可配置的 IPMI 平台事件过滤器发布机框外部的警报。
- 支持现场可更换单元 (field-replaceable unit, FRU) 的热交换，同时保持管理的全面可见。
- 通过机框管理实现的干接点中继，利用接口连接到标准的 Telco 报警基础结构。
- 支持机框管理器冗余实例以实现高可用性。
- 集成了监视程序计时器，如果没有定期进行选通处理，则使机框管理卡复位；此类复位会自动触发切换操作以转至备用机框管理卡（如果进行了此项配置的话）。
- 包含由电池供电的实时时钟，供时间戳事件使用。
- 实现了一组功能齐全的机框外部接口，包括 ATCA 所必需的 RMCP 以及 CLI，可通过以太网进行访问。

硬件资源分层结构

系统的每个可管理组件在系统中均标识为一个唯一实体。每个实体都由一个实体路径唯一地命名，实体路径会根据实体在系统中的物理包含关系来标识该组件。

实体路径由一组按顺序排列的 {实体类型, 实体位置} 对组成。该路径根据实体所在的实体以及其容器所在的实体来定义该实体在系统中的物理位置。

有关更多详细信息，请参阅 SAF-HPI-B.01.01 规范。可以从以下网址获得该规范：

<http://saforum.org/>

附录 A 显示了 Sun Netra CT900 服务器的简化资源表，其中包括两个 ShMM 500 机框管理器、两个 CT3140 交换机刀片、一个 CP3010 刀片、一个 CP3020 刀片和一个 CP3060 刀片。

附录 B 包含 3.2 PICMG 刀片的资源数据记录。该资源数据记录定义了与资源关联的管理工具（传感器、控制机制、监视程序计时器、库存数据系统信息库或报警器）。

系统管理员 (System Administrator) 接口选项

机框管理器的另一个主要子系统可实现系统管理员 (System Administrator) 接口。系统管理员 (System Administrator) 是一个逻辑概念，可涵盖软件以及操作中心中的操作人员。机框管理器提供了两个系统管理员 (System Administrator) 接口选项，它们提供了对与机框有关的同类信息和控制的不同访问机制：

- IPMI 局域网 (Local Area Network, LAN) 接口
- 命令行界面 (Command-line interface, CLI)

IPMI LAN 接口用于最大程度地提高独立实现的机框产品之间的互操作性。此接口是 ATCA 规范所要求的，并支持通过 RMCP 在 IPMI 与机框管理器之间传送消息。使用 RMCP 与机框通信的系统管理员应该能够与所有符合 ATCA 标准的机框管理器进行交互。此底层接口提供对机框 IPMI 内容的访问，包括系统管理员使用机框管理器作为代理向机框中的 IPMI 控制器发出 IPMI 命令的能力。

RMCP 是通过 LAN 连接到 IPMI 控制器的标准网络接口，并由 IPMI 1.5 规范进行定义。

CLI 提供了一组全面的文本命令，可以通过物理串行连接或 Telnet 连接向机框管理器发出这些命令。

OpenHPI

开放式硬件平台接口 (Open Hardware Platform Interface, OpenHPI) 定义了一个 C 应用程序编程接口, 以便访问如下平台管理功能:

- 配置 - 系统中的组件
- 组件详细信息 - 组件的供应商、型号、版本和序列号
- 状态 - 温度、电压、风扇速度和 LED 指示灯的状态
- 控制 - 打开、关闭和重置系统以及设置 WDT 的能力

有关 OpenHPI 以及支持的返回码的详细说明, 请参阅 OpenHPI 规范, 网址为:

<http://www.openhpi.org/>

OpenHPI 概述

服务可用性论坛 (Service Availability Forum, SAF) 硬件平台接口 (Hardware Platform Interface, HPI) 规定了用于监视和控制高可用性系统的通用机制。监视和控制这些系统的能力是通过一组一致的、独立于平台的编程接口提供的。HPI 规范提供了可用来与平台或系统的可管理子集进行交互的数据结构和功能定义。借助 HPI, 应用程序和中间件可通过标准化接口访问和管理硬件组件。

HPI 模型包括四个基本概念: 实体、资源、会话和域。本节对每个概念进行了简要介绍。

实体

实体表示系统的物理组件。每个实体都具有一个称为实体路径的唯一标识符, 实体路径由组件在系统的物理包含分层结构中的位置进行定义。

资源

资源提供对系统内实体的管理访问。通常, 资源表示由本地控制处理器执行的用于管理实体硬件的功能。每个资源负责向 HPI 用户提供一组管理工具和管理功能。在系统中, 可按添加和删除包含管理功能的可热交换系统组件的方式动态添加和删除资源。

会话

会话提供对由 HPI 用户实现的 HPI 的所有访问。在单个域中打开 HPI 会话；一个 HPI 用户可以同时打开多个会话，可以在给定的任一域中同时打开多个会话。通过会话，还可以访问由会话访问的域创建或转发的事件。HPI 用户通过会话访问系统，其中每个会话都是在域中打开的。通过会话，可访问域功能和可通过域访问的资源集。

域

可通过会话访问所有 HPI 用户功能，每个会话都与单个域相关联。域提供对零个或多个资源的访问，并提供一组关联的服务和功能。后者可逻辑归类到称为域控制器的抽象概念中。可通过域访问的资源列在域的资源存在表 (Resource Presence Table, RPT) 中。该表的内容可能会随时间而改变，域的会话管理功能会拒绝任何对当前未列在域资源存在表中的资源的访问尝试。

管理应用程序框架

如图 1-3 所示，管理应用程序通过 OpenHPI 库与 OpenHPI 守护进程进行对话。OpenHPI 守护进程通过插件与平台（本地或远程）进行对话。

图 1-3 OpenHPI 体系结构

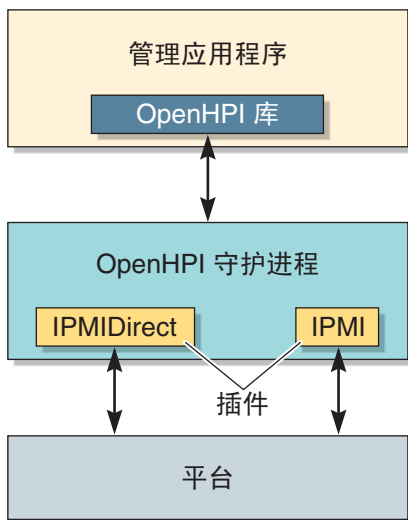
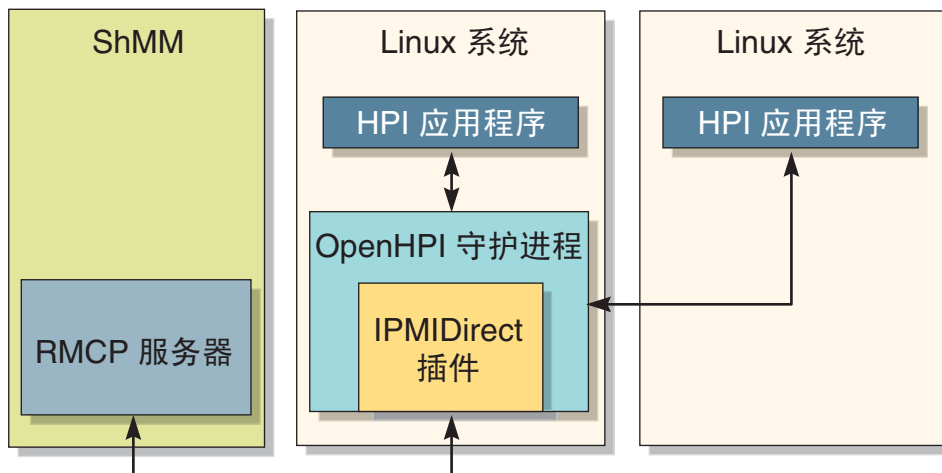


图 1-4 显示了运行 OpenHPI 守护进程（IPMI 直接插件）的 Linux OS 系统，该系统通过 RMCP 与 ShMM 进行通信，以进行机框管理。

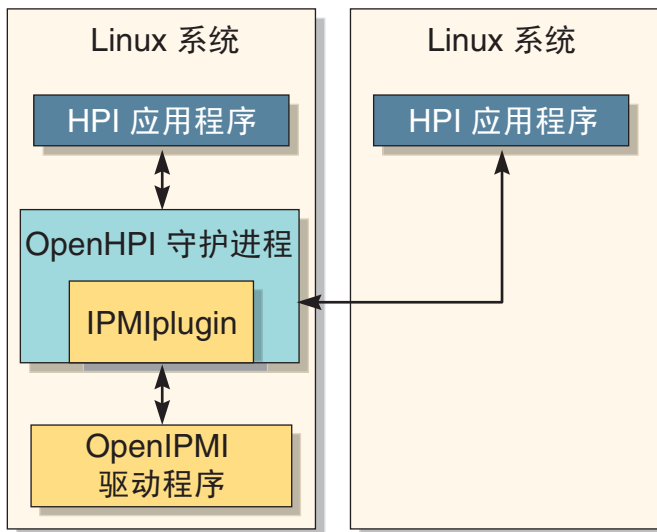
图 1-4 HPI 应用程序、OpenHPI 守护进程和 RMCP 服务器的关系



SAF HPI 在很大程度上凭借于智能平台管理接口 (Intelligent Platform Management Interface, IPMI) 规范规定的概念，以定义独立于平台的功能和数据格式。因此，在使用 IPMI 作为平台管理基础结构的平台上实现 HPI 接口会非常简单。但是，由于 HPI 是一个通用接口规范，所以它可在具备足够底层平台管理技术的任何平台上实现。

图 1-5 显示了在具有 OpenIPMI 驱动程序的系统上运行以进行本地管理的 OpenHPI 守护进程（IPMI 插件）。

图 1-5 HPI 应用程序和 OpenIPMI 驱动程序的关系



第2章

简单网络管理协议

如 Internet 工程任务组 (Internet Engineering Task Force, IETF) 所定义, 简单网络管理协议 (simple network management protocol, SNMP) 是 Internet 协议套件的一部分。网络管理系统使用 SNMP 监视网络连接设备, 以确定是否有触发管理性警示的条件。SNMP 由一组网络管理标准组成, 其中包括应用层协议、数据库结构以及一组数据对象。

本章描述 SNMP 网络协议, 并说明如何使用该协议。

本章包括以下主题:

- 第 16 页的 “SNMP 概述”
- 第 17 页的 “ShMM SNMP 体系结构”
- 第 18 页的 “ShMM SNMP 代理配置”
- 第 18 页的 “了解 MIB 变量描述”
- 第 24 页的 “配置 hpiSubagent”
- 第 26 页的 “SNMP 使用示例”
- 第 61 页的 “配置陷阱和处理通知”

有关 SNMP 的更多信息, 请访问:

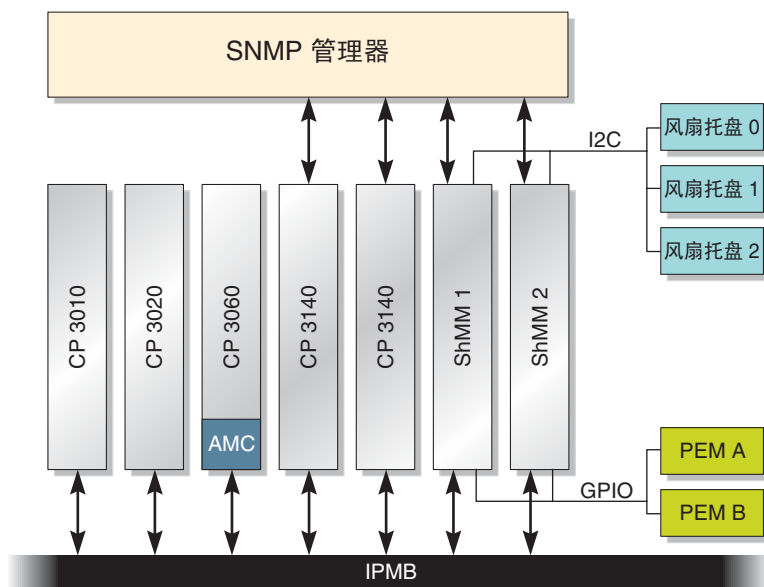
<http://net-snmp.sourceforge.net/>

SNMP 概述

设备必须具有与之关联的 SNMP 代理，才能管理该设备。代理可以接受表示设备状态的数据请求，并提供相应的响应。代理还可以控制设备状态。此外，代理可以生成 SNMP 陷阱，这些未经请求的消息发送给选定 NMS 以指示与设备相关的重要事件。

图 2-1 从 SNMP 管理器角度出发高度概述了 Sun Netra CT900 服务器。风扇托盘和电源输入模块 (power entry module, PEM) 只是几个可通过 ShMM 管理的资源示例。

图 2-1 SNMP 管理关系概述



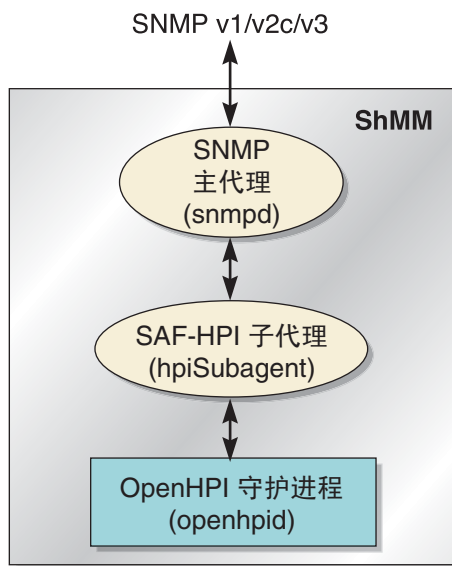
ShMM SNMP 体系结构

系统级 SNMP 支持是通过 ShMM 提供的，后者可以监视系统中每个 FRU 的状态。基于 ShMM 的 SNMP 支持是在主代理与子代理体系结构中实现的。hpiSubagent 是一个开源 OpenHPI SNMP 子代理，它基于服务可用性论坛硬件平台接口规范 (SAI-HPI-B.01.01)。

主代理的设计以及主代理与子代理之间的通信协议超出了本文档的讨论范围。

下图显示了基于 ShMM 的 SNMP 代理体系结构。

图 2-2 SNMP 体系结构



刀片（例如 CP3140 交换机刀片）可以通过刀片上的 SNMP 代理，以本地方式支持更多刀片特定的功能。

ShMM SNMP 代理配置

可以通过修改 `snmpd.conf` 和 `hpiSubagent.conf` 文件（均位于 `/etc` 目录），来配置基于 ShMM 的 SNMP 代理。

`hpiSubagent.conf` 文件包含用于配置 HPI 检查间隔、事件行以及事件溢出操作的参数。`snmpd.conf` 文件中的参数记录在 `snmpd.conf` 手册页中。

了解 MIB 变量描述

管理信息库 (management information base, MIB) 定义一个虚拟数据存储库，可通过 SNMP 软件（由代理维护的相应数据提供的内容）或通过从受管设备获取所需数据的代理来访问该虚拟数据存储库。对于由网络管理器写入虚拟数据存储库的数据，代理执行的操作将影响其自身状态或受管设备的状态。

在 Sun Netra CT900 服务器中，SNMP 支持是通过主代理与子代理体系结构提供的，其中主代理（从 R3.0 开始，由 PPS 提供）处理对非 Sun Netra CT900 服务器特定的 SNMP 对象（即 MIB2）的支持。Sun Netra CT900 服务器级 SNMP 支持主要由符合 SAF-HPI B-01-01 规范的 HPI MIB 定义，并由基于 ShMM 的 `hpiSubagent` 实现。每个 CP3140 交换机刀片还可以通过其 MIB 提供对相关本地对象的更多支持，通过运行在刀片上的 SNMP 代理可以直接访问这些 MIB。

SAF-HPI MIB

此 MIB 基于 SAI-HPI-B.01.01 规范定义 HPI 检测，该规范将硬件平台视为可分别管理的物理实体的集合。

实体的逻辑集合包括一个管理域。每个实体都有一组反映在实体表中的通用属性，并可能有存在于一个或多个表中的其他分类属性。每个实体都由一个实体路径唯一地命名，实体路径会根据实体在系统中的物理包含关系来标识该组件。有关 HPI 模型的更多详细信息，请参阅 HPI 规范。可以在以下位置获取 HPI 规范副本：

<http://www.saforum.org/>

SAF-HPI MIB 表分层结构

与每个受管实体相关的数据存储在由 SAF-HPI MIB 定义的几个表中。这些表之间的关系非常类似于在 HPI 规范中定义的概念（域->资源->实体）。要遍历这些表，必须首先根据 saHpiDomainId、saHpiResourceEntryId、saHpiResourceIsHistorical 和 saHpiRdrEntryId（如果适用）构建正确的索引。

以下各节中的图表说明了在 SAF-HPI MIB 中定义的一些常用表之间的关系。有关说明以及每个表中包含的对象列表，请参阅 MIB 文件。

可以在以下位置获取标准 MIB 的副本：

<http://www.faqs.org/>

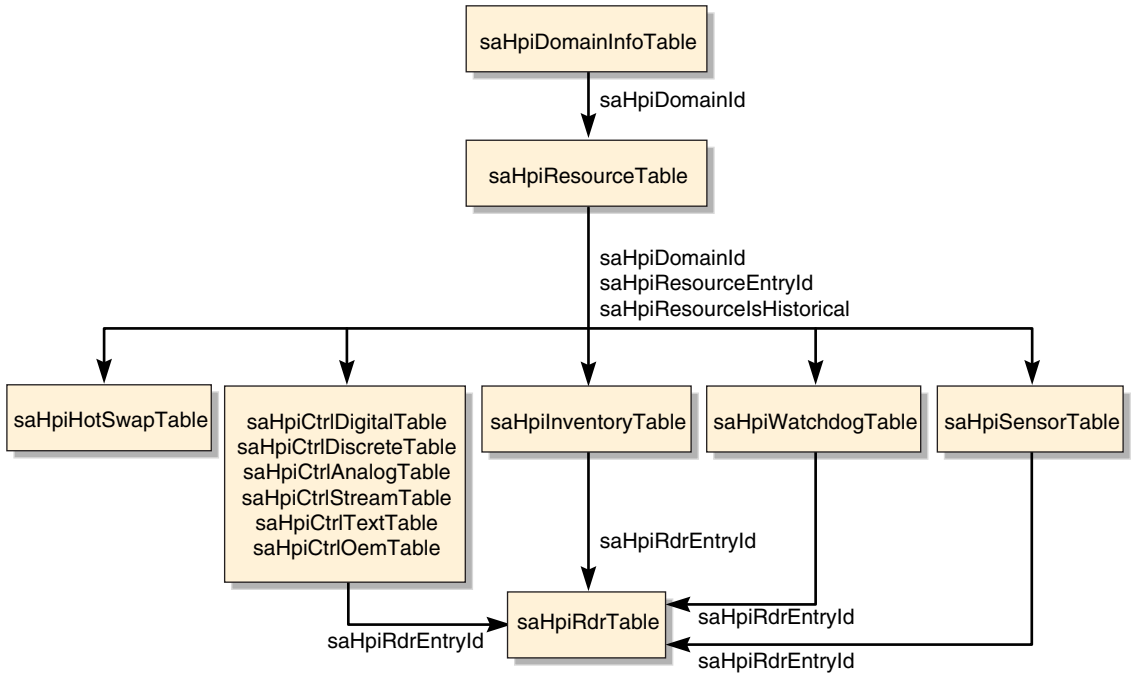
实体表

每个实体都有一组反映在实体表中的通用属性。实体可以具有存在于以下一个或多个表中的分类属性：

- 热交换表定义支持热交换的实体（通常称为 FRU）的管理属性。
- 控制表定义用于读取和设置与实体相关的控制的变量。
- 传感器表定义用于读取与实体相关的传感器的变量，以及用于控制该传感器的事件生成的变量。
- 监视程序表定义用于读取与实体相关的监视程序事件的变量。
- 库存控制表定义用于读取库存资源和更改设置的变量。

图 2-3 显示了实体表之间的关系。

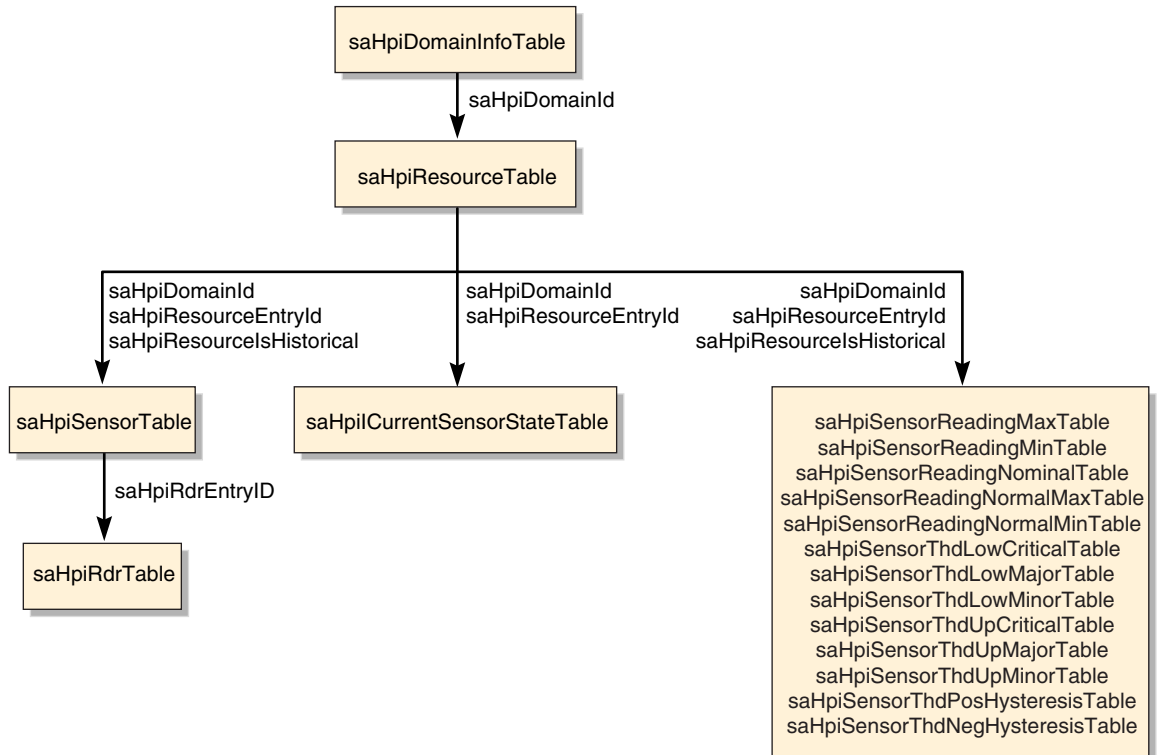
图 2-3 实体表关系



传感器表

图 2-4 显示了传感器表之间的关系。

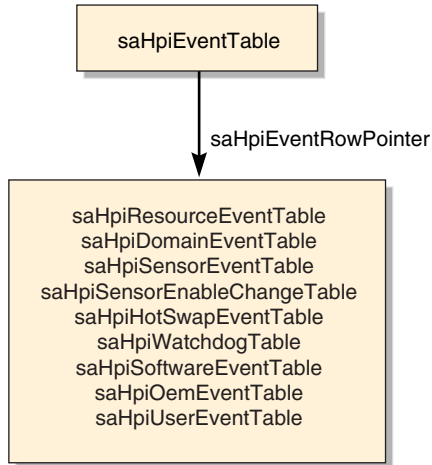
图 2-4 传感器表关系



事件表

saHpiEventTable 显示 HPI 系统中出现的所有事件的列表。此表用作主事件表，同时有个索引指向包含事件更多详细信息的特定子表。图 2-5 显示了 saHpiEventTable 和事件子表之间的关系。

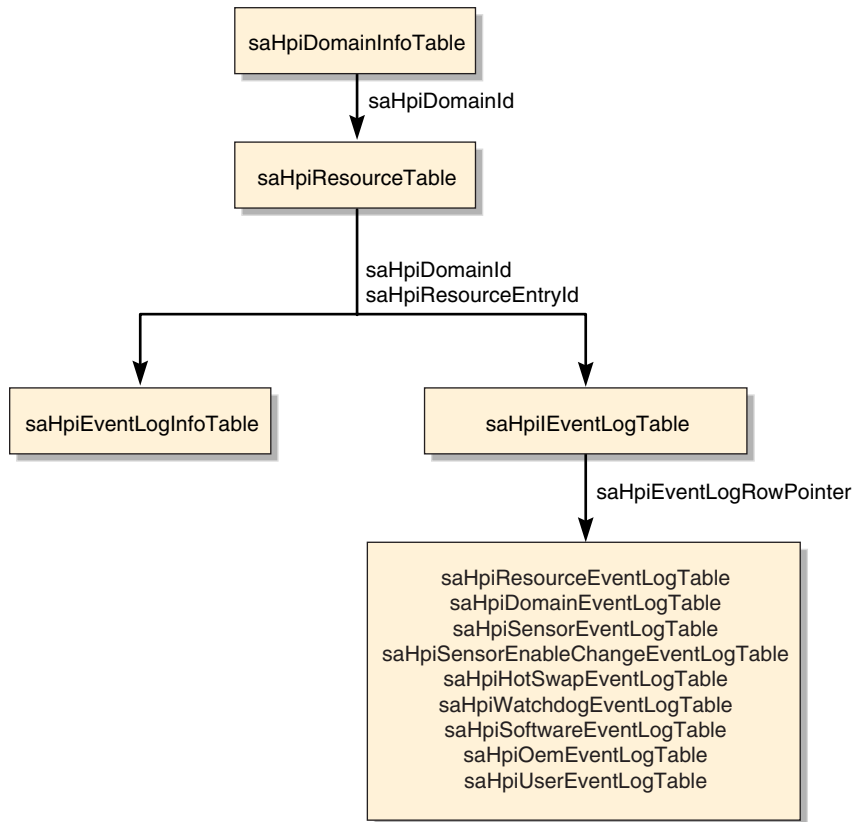
图 2-5 事件表关系



事件日志表

图 2-6 显示了事件日志表之间的关系。

图 2-6 事件日志表关系



配置 hpiSubagent

SNMP 子代理配置文件 `/etc/snmpd.conf` 定义 SNMP 子代理运行方式，并包括访问控制指令。下面的过程介绍如何设置访问控制以及如何启用 SNMP 版本 3 的使用。

▼ 启用读写访问

默认情况下，在 `snmpd.conf` 文件的访问控制部分中，`hpiSubagent` 配置为只读访问：

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

1. 将 `rocommunity` 值替换为 `rwcommunity`：

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

2. 重新启动 `hpiSubagent`：

```
# reboot
```

▼ 启用子代理的 SNMP 版本 3 使用

注 – 必须在两个 ShMM 卡上的 `snmpd.conf` 文件中进行如下更改。

1. 配置 `snmpd.conf` 文件中的 `engineID`：

```
engineID string
```

必须配置子代理，使 `engineID` 能够响应 SNMP 版本 3 消息。`engineID` 的默认值是为计算机主机名找到的第一个 IP 地址。

2. 为用户配置 username、验证类型和 authpassphrase。

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

MD5 和 SHA 都是验证类型。要使用 SHA，必须已经构建 OpenSSL 软件包并将其安装在 ShMM 上。DES 是保密性协议。如果未指定 *privpassphrase*，则假定为与 *authpassphrase* 是一样的。从发行版 3 开始，不支持 SHA 和 DES。它们包括在命令语法中，仅供参考。

下面是支持的命令语法示例：

```
createUser admin MD5 admin123
```

此命令行将创建名为 admin、验证类型为 MD5 且 *authpassphrase* 为 admin123 的用户。

注 – 从发行版 3.0 开始，不支持 SHA 和 DES 验证。

3. 配置用户的访问控制。

```
rouser admin
```

此配置行可提供对 admin 用户的只读访问。

```
rwuser admin
```

此配置行可提供对 admin 用户的读写访问。

4. 重新启动 hpiSubagent:

```
# reset
```

5. 使用 snmpwalk 命令检查 SNMP 的使用:

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

此示例是对使用 SNMP 版本 3 的 saHpiResourceTable 执行 snmpwalk。用户是 admin，验证类型是 MD5，authpassphrase 是 admin123。ShMMIP 是机框管理器的 IP 地址。

注 – 从发行版 3.0 开始，不支持 authPriv。

SNMP 使用示例

以下各节举例说明了如何使用 `snmpwalk` 命令查看 HPI 子代理 MIB 表的内容。所有示例都基于具有以下配置的机框：

- 一个活动 ShMM-500
- 三个风扇托盘
- 两个 PEM
- 两个交换机
- 一个 CP3010 刀片
- 一个 CP3020 刀片
- 一个安装了 AMC 的 CP3060 刀片

所有示例都包括 `ShMMIP`，它是机框管理器的 IP 地址。

获取资源相关信息

`saHpiResourceTable` 包含 ATCA 机框中所有资源的相关信息。资源包括插槽、ATCA 刀片、交换机和 ShMM 卡。信息包括 `ResourceId`、`ResourceTag`、`ResourceEntityPath` 和 `ResourceCapabilities`。该表的索引是 `domainID.resourceID.isHistorical`。

▼ 查看域的所有资源的所有信息

- 键入：

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

其中 `ShMMIP` 是活动机框管理器的 IP 地址，`1` 是 `domainID`。

▼ 查看域的所有资源的数据列

以下示例说明如何查看具有一个 ShMM 的机框上的所有资源的 ResourceTag。

- 键入：

```
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"
.....

```

以下示例说明如何查看具有两个 ShMM 的机框上的所有资源的 ResourceTag。

- 键入：

```

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 = ""
```

注 - 这两个示例中的资源 ID 都不是固定的或静态的。相同的 `snmpwalk` 命令可能会导致不同机框上的资源 ID 不同。即便在使用新 HPI 子代理实例的同一个机框上，所分配的资源 ID 也可能不同。

▼ 查看域的特定资源

- 键入:

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

在此命令示例中，`domainID` 是 1，资源 ID 是 40。

获取属性相关信息

`saHpiRdrTable` 包含所有资源的资源数据记录。信息包括 `RdrType`（其中 `Rdr` 是传感器、控制机制或监视程序）、`RdrEntityPath` 和 `RdrRowPointer`（是根据 `RdrType` 指向另一个表的指针）。如果 `RdrType` 是传感器，则条目是指向传感器表中条目的指针。如果 `RdrType` 是控制机制，则条目是指向控制表中条目的指针。该表的索引是 `domainID.resourceID.isHistorical.RDRID`。

▼ 查看域的所有资源的 RDR 条目

- 键入:

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

其中 1 是域 ID。

▼ 查看域的所有资源的 RDR 表中的列

- 键入:

```
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)
.....
```

此命令在 RdrType 中搜索所有数据记录。输出结果中显示了字符串 ctrlRdr 和一个整数。2 表示控制 RDR。3 表示传感器 RDR，4 表示库存 RDR。

▼ 查看域的资源 RDR 表中的列

- 键入:

```
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)
```

其中 1 是域 ID，40 是资源 ID。

▼ 查看每个 RDR 条目表示的内容

- 键入:

```
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"
```

▼ 查看 RDR EntryID 的 RDR 表中的列

- 键入:

```
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)
```

其中 1 是域 ID，40 是资源 ID，false 是 isHistorical 值，131076 是 RDR 条目 ID。

获取传感器相关信息

`saHpiSensorTable` 包含所有资源的所有传感器的相关信息。信息包括 `SensorType`（例如，温度或电压）、`SensorCategory`（例如，阈值、存在或启用）和 `SensorBaseUnits`（例如，伏特或摄氏度）。

`saHpiCurrentSensorStateTable` 包含所有资源的所有传感器的当前状态的相关信息，其中包括：

- 当前值
- 传感器的事件状态
- 是否启用传感器
- 是否从启用的传感器生成事件

`saHpiSensorTable` 的索引是 `domainID.resourceID.isHistorical.sensorNum`。

`saHpiCurrentSensorStateTable` 的索引是 `domainID.resourceID.sensorNum`。

▼ 查看域中所有资源的所有传感器的相关信息

- 键入：

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

其中 1 是 `domainID`。

▼ 查看域的所有资源的传感器表中的列

- 键入:

```
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)
.....
```

在此示例中，命令返回所有资源的所有传感器的传感器类型信息。

▼ 查看某资源的传感器表中的列

- 键入:

```
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)
```

此命令返回特定资源的传感器类型信息，其中 1 是域 ID，40 是资源 ID。
资源 40 有三个温度传感器。传感器编号分别是 4、5 和 6。

▼ 查看某资源的所有传感器的基本度量单位

- 键入:

```
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)
```

对于传感器 4、5 和 6，传感器类型是温度，基本度量单位是摄氏度。

▼ 查看某资源的传感器的传感器表中的列

- 键入:

```
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)
```

其中 1 是域 ID，40 是资源 ID，false 是 isHistorical 值，4 是传感器编号。

▼ 查看域的所有资源的所有传感器的当前状态

- 键入:

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

其中 1 是域 ID。

▼ 查看域的所有资源的当前传感器状态表中的列

- 键入:

```
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"  
.....
```

▼ 查看某资源的当前传感器状态表中的列

- 键入:

```
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"
```

其中 1 是域 ID，40 是资源 ID。

对于资源 40，有三个温度传感器。当前值分别是 92、94 和 33 摄氏度。

▼ 查看某资源的传感器的当前传感器状态表中的列

- 键入:

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiCurrentSensorStateValue.1.40.4  
  
HPI-B0101-MIB::saHpiCurrentSensorStateValue.1.40.4 = STRING: "9.2e1"
```

其中 1 是域 ID，40 是资源 ID，4 是传感器编号。

获取和设置传感器阈值

此节中的信息仅与阈值传感器（即归类为阈值的传感器）有关。传感器的阈值信息包含在以下六个表中：

- `saHpiSensorThdUpCriticalTable`，它包括所有资源的所有阈值传感器的临界阈值上限的相关信息
- `saHpiSensorThdUpMajorTable`，它包括所有资源的所有阈值传感器的重要阈值上限的相关信息
- `saHpiSensorThdUpMinorTable`，它包括所有资源的所有阈值传感器的次要阈值上限的相关信息
- `saHpiSensorThdLowCriticalTable`，它包括所有资源的所有阈值传感器的临界阈值下限的相关信息
- `saHpiSensorThdLowMajorTable`，它包括所有资源的所有阈值传感器的重要阈值下限的相关信息
- `saHpiSensorThdLowMinorTable`，它包括所有资源的所有阈值传感器的次要阈值下限的相关信息

这些表中的信息位于相同的行上，并包括：

- 当前阈值
- 值是否可读
- 值是否可写

阈值表的索引是 `domainID.resourceID.isHistorical.sensorNum`。

▼ 查看域的所有资源的所有传感器的传感器临界阈值上限表中的所有信息

- 键入：

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

其中 1 是域 ID。

▼ 查看所有资源的所有传感器的传感器临界阈值上限表中的列

- 键入:

```
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"
.....
```

▼ 查看某资源的所有传感器的传感器临界阈值上限表中的列

- 键入:

```
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"
```

其中 1 是域 ID，40 是资源 ID。

值 4、5 和 6 是温度传感器，度量单位为摄氏度。传感器的临界阈值上限为 120 摄氏度。

▼ 查看某资源的传感器的传感器临界阈值上限表中的列

- 键入:

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

其中 4 是传感器编号，40 是资源，1 是域。

▼ 为传感器设置传感器阈值

1. 确认传感器阈值可写:

```
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)
```

此命令返回与传感器编号 4 对应的传感器信息。

2. 确认所用的值处于传感器的可接受范围:

```
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"
```

资源 40 的传感器 4 的可接受值范围是 -40 至 125。

3. 键入:

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

其中 1 是域 ID, 40 是资源 ID, 4 是传感器编号, s 指示值类型 (是字符串), 1.23e2 是所设置的值。

获取和设置有关控制的信息

根据控制类型, 控制信息包含在以下六个表中。下面列出了这些表的名称及说明:

- saHpiCtrlAnalogTable (用于模拟控制)
- saHpiCtrlDigitalTable (用于数字控制)
- saHpiCtrlDiscreteTable (用于离散控制)
- saHpiCtrlTextTable (用于文本控制)
- saHpiCtrlStreamTable (用于流控制)
- saHpiCtrlOemTable (用于 OEM 控制)

上述所有表中的信息都位于类似的行上；但是，根据控制类型，某些表中可能会出现额外的字段。通用信息包括：

- 控制编号
- 控制模式
- 控制状态
- 控制默认状态
- 模式是否为只读
- 控制是否为只写

所有控制表的索引是 `domainID.resourceID.isHistorical.EntryID`。

▼ 查看所有资源的所有模拟控制的相关信息

- 键入：

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

其中 1 是域 ID。

▼ 查看所有资源的模拟控制表的列

- 键入：

```
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)  
.....
```

此命令返回所有模拟控制的控制模式。

▼ 查看某资源的模拟控制表的列

- 键入:

```
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)
```

此命令返回资源 31 的所有模拟控制的控制模式。

▼ 查看特定资源的所有模拟控制的控制状态

1. 键入:

```
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. 键入:

```
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
```

资源 31 具有三种模拟控制。它们分别表示 FRU 所需电源、FRU 重新引导和诊断控制，以及 ATCA 风扇。前两项控制是自动管理的。第三项控制是手动的（即，该项控制可以由 SNMP 管理器进行管理）。

▼ 查看某资源的控制的模拟控制表的列

- 键入：

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-  
MIB::saHpiCtrlAnalogState.1.31.false.2  
  
HPI-B0101-MIB::saHpiCtrlAnalogState.1.31.false.2 = INTEGER: 3
```

此命令返回资源 31（条目 ID 为 2）的模拟控制的状态。

▼ 设置模拟控制的状态

1. 确认模式是手动并且值处于可接受范围内：

```
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
```

此模拟控制的可接受值范围是 0 至 15。

2. 设置模拟控制:

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

此命令修改资源 31（条目 ID 为 2）的模拟控制的状态。此命令将该控制的状态设置为 11。

获取有关 IDR 的信息

库存数据系统信息库 (inventory data repository, IDR) 信息包含在以下三个表中:

- saHpiInventoryTable
- saHpiAreaTable
- saHpiFieldTable

saHpiInventoryTable 是包含如下信息的高级别表:

- 库存的更新计数
- 区域数量
- 表是否为只读

会为所有资源的所有 IDR 存储此信息。

saHpiInventoryTable 信息

saHpiInventoryTable 的索引是
domainID.resourceID.isHistorical.InventoryID。

▼ 查看域的所有资源的高级别库存信息

- 键入:

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

其中 1 是域 ID。

▼ 查看域的所有资源的库存表的列

1. 键入:

```
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
.....
```

此命令返回所有资源的所有 IDR 的区域数量。域 1 中资源 1 的 IDR 0 的区域数量是 47。域 1 中资源 27 的 IDR 0 的区域数量是 2。

2. 键入:

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

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

此命令返回资源 1 的信息。机架资源的 IDR 0 的区域数量是 47。

▼ 查看某资源的库存表的列

● 键入:

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

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

此命令返回资源 1 的所有 IDR 的区域数量。

▼ 查看与资源和 IDR 条目 ID 对应的库存表的列

- 键入:

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

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

此命令返回资源 1 的 IDR 0 的区域数量。

saHpiAreaTable 信息

saHpiAreaTable 包含有关区域是否为只读以及区域中字段数量的信息。会为所有资源的所有 IDR 的所有区域存储此信息。该表的索引是 domainID.resourceID.isHistorical.InventoryID.AreaID。

▼ 查看域的所有资源的所有区域的所有信息

- 键入:

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

其中 1 是域 ID。

▼ 查看所有资源的区域表的列

- 键入:

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

此命令返回所有资源的所有区域中的数据字段数量。

▼ 查看某资源的区域表的列

- 键入:

```
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
```

此命令返回资源 1（即机框资源）的所有区域中的数据字段数量。输出结果中显示资源 1 的所有 IDR 的所有区域中的数据字段数量。

▼ 查看某资源的 IDR 的列

- 键入:

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

此命令返回资源 1 的 IDR 0 的所有区域中的字段数量，其中 1.1.false.0 代表 domainID.resourceID.isHistorical.InventoryD。

▼ 查看某资源的 IDR 的区域的列

- 键入:

```
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
```

此命令返回资源 1 的 IDR 0 的区域 ID 46 中的字段数量，其中 1.1.false.0.46 代表 domainID.resourceID.isHistorical.InventoryID.AreaID。

saHpiFieldTable 信息

saHpiFieldTable 包含如下信息:

- 字段类型
- 字段文本
- 字段是否为只读

会为所有资源的所有 IDR 的所有区域中的所有字段存储此信息。该表的索引是 domainID.resourceID.isHistorical.InventoryI.AreaID.FieldID。

▼ 查看所有字段的所有信息

- 键入:

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

其中 1 是域 ID。

▼ 查看所有字段的字段表的列

- 键入:

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

此命令返回所有字段的字段类型。

▼ 查看某资源的所有字段的列

- 键入:

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

此命令返回资源 1 的所有字段的字段类型。

▼ 查看区域的所有字段的列

- 键入:

```
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)
```

此命令返回资源 1 的 IDR 0 的区域 45 的所有字段的字段类型，其中 1.1.false.0.45 代表 domainID.resourceID.isHistorical.InventoryID.AreaID。

▼ 查看字段的列

- 键入:

```
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"
```

此命令返回资源 1 的 IDR 0 的区域 45 的字段 1 的字段文本，其中 1.1.false.0.45.2 代表 domainID.resourceID.isHistorical.InventoryID.AreaID.FieldID。

使用 HPI 子代理管理自定义数据记录

snmpwalk 命令可显示自定义数据记录 (custom data record, CDR) 中的数据。

▼ 查看特定资源的所有区域的区域类型

- 键入:

```
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)
```

此命令返回资源 31 (风扇托盘 0) 的区域类型信息。ID 为 2 的区域是 OEM 区域。

▼ 查看特定资源的特定区域的所有字段的字段文本

- 键入:

```
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 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

区域 2 中有两个字段。ID 为 0 的第一个字段具有值 D0 02。值 D0 指示区域是 CDR。ID 为 1 的第二个字段具有 CDR 实际内容，长度为 255 字节。

▼ 修改 CDR 内容

- 键入:

```
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"
```

CDR 的字段 1 设置为十六进制值 41 31 31 32 56 45 64。此操作将字段文本设置为相应的 ASCII 字符。

▼ 删除特定资源的特定区域的特定 CDR 字段

- 键入:

```
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)
```

可以使用 `saHpiFieldStatus` 列来删除字段。该命令采用值 6（即，`destroy` 值）来删除字段。这样，`snmpset` 命令便可删除资源 31 的区域 2 的字段 1。

▼ 检查特定资源的特定区域中的字段数量

- 键入:

```
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
```

输出结果中显示只有一个字段。

使用事件日志和事件表

可以通过标准 IPMI 事件消息，将有关机框中的运行状况和管理状态变化通知给机框管理器。标准 IPMI 事件消息会记录在 IPMI 系统事件日志中，并转发到活动机框管理器。IPMI 控制器配置为当检测到在系统中声明的或取消声明的重要条件时生成事件消息。这包括如下事件消息：

- 超过温度阈值
- 超过电压阈值
- 电源故障
- 监视程序到期

IPMI 事件消息通常与 SDR 中定义的传感器有关。与事件相关的传感器的类型和事件类型可帮助机框管理器和 HPI 用户决定针对该事件执行哪些操作。

saHpiEventTable

saHpiEventTable 显示 HPI 系统中出现的所有事件的列表。此表包含：

- 事件类型
- 生成事件时的时间戳
- 事件严重性
- 指向包含事件详细信息的子表的指针

子表取决于事件类型。例如，如果事件类型为传感器，则子表为 saHpiSensorEventTable，如果事件类型为热交换，则子表为 saHpiHotSwapEventTable。

▼ 查看事件表中的所有信息

- 键入：

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

▼ 查看事件表中的列

- 键入：

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

saHpiSensorEventTable

根据事件类型，事件详细信息存储在多个表中的一个表中，例如 saHpiSensorEventTable 或 saHpiHotSwapEventTable。本节中的示例与 saHpiSensorEventTable 有关，但访问其他事件表的方法是相同的。

saHpiSensorEventTable 包含有关事件类型、事件类别和事件时间戳的信息。该表的索引是 domainID.resourceID.sensorNum.eventSeverity.eventEntryID。

▼ 查看传感器事件表中的所有信息

- 键入:

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

其中 1 是域 ID。

▼ 查看传感器事件表的列

- 键入:

```
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)
.....
```

此命令返回所有事件的事件类型。

▼ 查看某资源的传感器事件表的列

- 键入:

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

此命令返回源自资源 44 的所有传感器事件的事件类型。

▼ 查看某资源的传感器的传感器事件表的列

- 键入:

```
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)  
.....
```

此命令返回源自资源 44 传感器 7 的所有传感器事件的事件类型。

saHpiEventLogInfoTable

saHpiEventLogInfo 包含具有 EVENT_LOG 功能的资源的事件日志概述信息。这些信息包括:

- 事件日志的大小
- 事件日志中的当前条目数
- 事件日志是否溢出

该表还具有可用于清除日志中所有事件的列。该表的索引是 domainID.resourceID。

▼ 查看域的所有资源的事件日志信息

- 键入:

```
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)
```

输出结果中显示 ID 为 27 和 4294967295 的两个资源具有事件日志。保留 4294967295 值以指示域事件日志 (domain event log, DEL)。其他资源指示系统事件日志 (system event log, SEL)。

SEL 当前具有 42 个条目。它最多可容纳 65535 个条目，当发生溢出时，将放弃新事件。

▼ 查看事件日志信息表的列

- 键入:

```
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
```

此命令返回所有资源的事件日志中的当前条目数。

▼ 查看某资源的事件日志信息表的列

- 键入:

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

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

此命令返回资源 27 的事件日志中的当前条目数。

saHpiEventLogTable

saHpiEventLogTable 包含所有资源的事件日志记录。该表包含以下信息:

- 事件类型
- 添加事件时的时间戳
- 指向包含事件详细信息的其他事件表的指针

该指针取决于事件类型。例如，如果事件类型为传感器，则指针指向 saHpiSensorEventLogTable，如果事件类型为热交换，则指针指向 saHpiHotSwapEventLogTable。该表的索引是 domainID.resourceID.EventLogIndex。

▼ 查看所有资源的 saHpiEventLogTable 中的信息

- 键入:

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

其中 1 是域 ID。

▼ 查看所有资源的 saHpiEventLogTable 的列

- 键入:

```
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)
....
```

此命令返回所有资源的所有事件的事件类型。对于资源 27，第一个事件日志条目为传感器类型。第二个条目也是传感器，第三个是热交换类型。

▼ 查看所有资源的所有事件的事件日志行指针

- 键入:

```
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
.....
```

该行指针基于事件类型。前两个事件为传感器事件，指向 saHpiSensorEventLog 中的条目。第三个事件为热交换事件，指向 saHPIHotSwapLog 中的条目。这些指针可用于访问事件详细信息。

▼ 查看某资源的事件日志表的列

- 键入:

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

此命令返回资源 27 中记录的所有事件的事件类型。

▼ 查看某资源和事件的事件日志表的列

- 键入:

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

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

此命令返回资源 27 中记录的事件日志 ID 2 的事件类型。

saHpiSensorEventLogTable

saHpiSensorEventLogTable 基于事件类型。事件详细信息存储在多个表（例如 saHpiSensorEventLogTable 和 saHpiHotSwapEventLogTable）中的一个表中。本节中的示例与 saHPISensorEventLogTable 有关，但访问其他事件日志表的方法是相同的。saHpiSensorEventLogTable 包含如下信息：

- 事件类型
- 事件类别
- 事件时间戳

该表的索引是

domainID.resourceID.sensorNum.eventSeverity.eventEntryID。

在本节的示例中，资源 ID 代表作为事件源的资源，而不是记录事件的资源。

▼ 查看传感器事件日志表中的所有信息

- 键入:

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

其中 1 是域 ID。

▼ 查看传感器事件日志表的列

- 键入:

```
snmpwalk -v 2c -c public $hMMIP 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)  
.....
```

此命令返回所有事件的事件类型。

▼ 查看某资源的传感器事件日志表的列

- 键入:

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

此命令返回源自资源 40 的所有传感器事件的事件类型。

▼ 查看某资源的传感器的传感器事件日志表的列

- 键入:

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

此命令返回源自资源 40 传感器 4 的所有传感器事件的事件类型。

清除事件日志条目

saHpiEventLogInfoTable 包含名为 saHpiEventLogClear 的列。可将此列设置为 1 以删除所有事件日志条目。

▼ 从系统事件日志中清除特定资源的条目

1. 键入:

```
snmpset -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiEventLogClear.1.27 i 1  
HPI-B0101-MIB::saHpiEventLogClear.1.27 = INTEGER: true(1)
```

这将清除资源 27 的系统事件日志。

2. 通过检查事件日志中的条目数确认删除:

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiEventLogInfoEntries.1.27  
HPI-B0101-MIB::saHpiEventLogInfoEntries.1.27 = Gauge32: 0
```

输出结果中显示条目数为 0。

▼ 从域事件日志中清除特定资源的事件日志

1. 键入:

```
snmpset -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiEventLogClear.1.4294967295  
i 1  
HPI-B0101-MIB::saHpiEventLogClear.1.4294967295 = INTEGER: true(1)
```

此命令可清除资源 4294967295 的事件日志。

2. 通过检查域事件日志中的条目数确认删除:

```
snmpwalk -v 2c -c public ShMMIP HPI-B0101-MIB::saHpiEventLogInfoEntries.1.4294967295  
HPI-B0101-MIB::saHpiEventLogInfoEntries.1.4294967295 = Gauge32: 0
```

配置陷阱和处理通知

通过使用 SNMP 版本 1 和 2 陷阱将系统中的相关异步事件传递到 SNMP 管理器。

除了设置 `hpiSubagent`（基于 OpenHPI 事件，自 R3 起在 HPI-B0101-MIB 中定义）支持的通知外，主代理 (`snmpd`) 还会在守护进程启动期间生成诸如冷启动之类的陷阱。

本节提供了与 SNMP 守护进程陷阱生成配置相关的信息，以及 Sun Netra CT900 服务器 ShMM 支持的 SNMP 通知的概述。

陷阱配置

您可以通过编辑 `/etc/snmpd.conf` 文件来配置 ShMM 上的 SNMP 陷阱。

有关编辑 `snmpd.conf` 文件的其他信息，另请参阅《Sun Netra CT900 服务器管理和参考手册》。

▼ 配置 SNMP 版本 1 的陷阱

- 在 `snmpd.conf` 文件中插入以下行：

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

▼ 配置 SNMP 版本 2 的陷阱

- 在 `snmpd.conf` 文件中插入以下行：

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

以下示例显示了这两种版本的语法：

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

可以使用多个 `trapsink` 或 `trap2sink` 条目来指定多个陷阱目标。

在双 ShMM 配置中，可以配置活动 ShMM 和备用 ShMM 以基于传入事件生成陷阱。自 ATCA R3 起，SNMP 管理器应能够处理源自事件的活动 ShMM 和备用 ShMM 的任何重复陷阱的过滤。

通知处理

以下信息提供了 hpiSubagent 通知的说明，以及处理 hpiSubagent 通知的一些示例。

表 2-1 包含由 HPI-B0101-MIB 定义的 SNMP 通知。

表 2-1 SNMP 通知

通知	说明
saHpiSensorNotification	传感器事件通知。收到该通知之后，管理应用程序应刷新通知中指定的传感器的所有缓存信息。
saHpiSensorEnableChangeNotification	传感器启用更改事件通知。
saHpiResourceNotification	资源故障或恢复事件通知。收到该通知之后，管理应用程序应刷新缓存的所有资源信息。
saHpiDomainNotification	域事件用于向 DRT 宣布域引用的添加和删除情况。
saHpiWatchdogNotification	监视程序通知。
saHpiHotSwapNotification	热交换通知。收到该通知之后，管理应用程序应刷新缓存的所有资源信息，以及与指定资源关联的传感器的所有缓存信息。
saHpiSoftwareNotification	审计事件报告审计过程中的差异。通常由高可用性软件执行审计以检测问题。审计可能查找诸如以下的情况：损坏的数据存储、不一致的 RPT 信息或未正确管理的队列。启动事件报告无法正常启动，或持久保存的数据中出现不一致。
saHpiOemNotification	OEM 事件通知。有关引用，请参阅 HPI 与 ATCA 对应关系规范中定义的状态更改的 OEM 事件原因。
saHpiUserNotification	使用 saHpiEventLogEntryAdd() 将事件添加到事件日志时，用户事件可用于存储 HPI 用户创建的自定义事件。

自版本 ATCA R3 起，hpiSubagent 不支持以下通知：

- saHpiSensorEnableChangeNotification
- saHpiDomainNotification, saHpiWatchdogNotification
- saHpiSoftwareNotification 和 saHpiUserNotification

示例：冷启动陷阱

以下是 SNMP 版本 1 冷启动陷阱的示例：

```
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
```

以下是 SNMP 版本 2 冷启动陷阱的示例：

```
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:
```

示例：热交换 1

以下示例说明了释放锁定且板处于非活动状态（LED 指示灯呈稳定蓝色）后从 Sun Netra CT900 服务器中提取板的过程。

显示的输出来自 Net-SNMP 命令行工具 `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)
```

处理陷阱的方式如下。

1. 从陷阱中过滤重要信息：

- 陷阱的源 IP。
在该示例中，源 IP 为 10.4.77.67。
- 陷阱的字段 3（资源 ID）。
在该示例中，资源 ID 为 39。
- 陷阱的字段 5 和字段 6（资源的当前热交换状态和上一个热交换状态）。
在该示例中，上一个热交换状态为 `extractionPending(4)`，当前热交换状态为 `inactive(1)`。

2. 检查陷阱的源 IP 地址是否为正在监视的活动 ShMM 的 IP 地址。
这样可确保不处理来自备用 ShMM 的陷阱。
3. 检查当前热交换状态和上一个热交换状态。
 - 如果当前热交换状态或上一个热交换状态中任一个为 `notPresent(5)`，则管理应用程序应刷新 `hpiSubagent` 中的所有传感器信息，因为向系统中添加或从系统中删除的 FRU 可能具有关联的传感器。
 - 如果当前热交换状态或上一个热交换状态都不是 `notPresent(5)`，则管理应用程序只需要刷新缓存的电压传感器信息。
4. 刷新缓存的资源信息。

示例：热交换 2

以下示例说明了释放锁定且板处于非活动状态（LED 指示灯呈稳定蓝色）后从 Sun Netra CT900 服务器中提取 Sun Netra CP3020 板的过程。

显示的输出来自 Net-SNMP 命令行工具 `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)
```

从该输出可以看出 `saHpiHotSwapNotification` 包含以下对象，并具有相应的值：

- `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)`

这些对象表示资源 37 的状态已从 `inactive` 转换为 `notPresent`。

可基于资源 ID 从 `saHpiResourceTable` 中检索其他信息，但是，必须在从系统中提取板之前执行此操作。

示例：超过温度传感器阈值

该示例显示超过温度传感器的非临界阈值上限值时生成的通知。

显示的输出来自 Net-SNMP 命令行工具 `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) HPI-B0101-
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 = ""
```

从该输出可以看出 `saHpiSensorNotification` 包含以下对象：

- `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 = ""`

这些对象表示资源 44 上的传感器 5 测量的温度已超过其次要阈值上限值（非临界上限值）。该事件的严重性较低。

传感器编号嵌入在变量绑定的索引值中。根据 MIB 中的定义，索引的第三个值是指 saHpiSensorNum。在上一示例中，使用 ?saHpiSensorEventType.0.44.5.minor.2?，我们可以看出第三个值（传感器编号）为 5。

可基于资源和传感器的 ID 从 saHpiSensorTable 和 saHpiRdrTable 中检索有关资源/传感器的其他信息。

第3章

智能平台管理接口驱动程序

IPMI 是一种消息传送协议，定义了如何监视系统硬件、控制系统组件以及检索硬件事件日志。IPMI 描述了多个嵌入式管理控制器如何协同工作。最新修订版 IPMI v2.0 添加了标准化的控制台访问（称为 Serial-over-LAN (SOL) 重定向），通过 AES 加密提高了安全性，还增强了对刀片和模块化系统的支持。

您可以通过在 ATCA 机框中使用独立管理子系统而受益，因为管理子系统不会受主 CUP 或 OS 故障的影响。因此，可以实现更高级别的系统可管理性。

在 ATCA 体系结构中，IPMI 是管理系统资源的关键要素。本章提供了刀片上使用 IPMI 驱动程序的应用程序示例。

本章包括以下主题：

- [第 68 页的“IPMI 概述”](#)
- [第 68 页的“操作系统支持和 IPMI 安装”](#)
- [第 69 页的“IPMI 用户界面”](#)
- [第 70 页的“IPMI 编程示例”](#)
- [第 77 页的“IPMI 命令”](#)

IPMI 概述

IPMB 是 ATCA 系统中的管理总线。每个刀片都有一个与 IPMB 连接的 IPMI 控制器。Sun Netra CP3xxx 刀片有一个板载 IPMI 控制器，符合 PICMG 标准。Solaris OS IPMI 驱动程序是主机或刀片上的 IPMI 控制器的接口。

您需要使用 IPMI 驱动程序与本地 IPMI 控制器或其他 IPMI 客户机进行通信。例如，使用 IPMI 驱动程序，您可以：

- 对刀片前面板 LED 指示灯进行编程。
- 对 IPMI 控制器中的监视程序计时器进行编程。
- 从其他 IPMI 客户机（通常为机框管理器）接收一条诸如关机请求之类的消息。

操作系统支持和 IPMI 安装

下列配置支持 IPMI 驱动程序：

- CP3010 刀片上的 Solaris 10 和 Solaris 10 1/06 OS
- CP3020 刀片上的 Solaris 10 和 Solaris 10 6/06 OS
- CP3060 刀片上的 Solaris 10 和 Solaris 10 6/06 OS

每个平台需要两个软件包：

- 在 CP3010 刀片上为 SUNWctipmi.u 和 SUNWctipmic
- 在 CP3020 和 CP3060 刀片上为 SUNWctipmi.v 和 SUNWctipmic

您可以从 Oracle 支持站点获取这些软件包：

<https://support.oracle.com>

▼ 安装 IPMI 驱动程序

1. 添加 SUNWctipmi.v 软件包:

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

2. 添加 SUNWctipmic 软件包:

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

3. 重新引导系统:

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

注 – 安装期间, 对任何问题均回答 “是”。

IPMI 用户界面

对于支持的功能, IPMI 驱动程序用户界面与 Linux OpenIPMI 驱动程序用户界面兼容。

IPMI 设备节点具有以下接口:

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

IPMI 驱动程序具有以下 poll(2) 标志:

- POLLPRI
- POLLIN

/usr/include/sys 目录中的 ipmi.h 和 ipmi_msgdef.h 头文件定义了接口。

IPMI 编程示例

本节包含两个有关如何使用 IPMI 驱动程序的编程示例。第一个示例说明了如何获取设备 ID，第二个示例说明了如何对 LED 指示灯进行编程。

获取设备 ID

以下示例说明如何使用 IPMI 驱动程序获取设备 ID。

示例 3-1 IPMI 设备 ID 示例

```
#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;
```

示例 3-1 IPMI 设备 ID 示例 (续)

```
/* 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");
}
```

示例 3-1 IPMI 设备 ID 示例 (续)

```
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);
}
```

对 LED 指示灯进行编程

以下示例说明如何使用 IPMI 驱动程序对系统的 LED 指示灯进行编程。

示例 3-2 IPMI LED 指示灯编程示例

```
/*
 * 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
```

示例 3-2 IPMI LED 指示灯编程示例 (续)

```
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;
```

示例 3-2 IPMI LED 指示灯编程示例 (续)

```
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;
```

示例 3-2 IPMI LED 指示灯编程示例 (续)

```

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);
    }
}

```

示例 3-2 IPMI LED 指示灯编程示例 (续)

```
/* 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 命令

本节列出了 ATCA 刀片上支持的所有 IPMI/ATCA 命令和 Sun OEM 命令。提供了对适用规范的引用，以便提供更多信息。

Sun ATCA 板上支持的 IPMI/ATCA 命令

表 3-1 IPMI 全局设备命令，网络功能：应用程序 (0x06/0x07)

命令	操作码	支持的平台	支持的接口	备注
Get Device ID (获取设备 ID)	0x1	所有	有效载荷, IPMB	
Cold Reset (冷复位)	0x2	所有	有效载荷, IPMB	冷复位命令可复位 IPMC。复位之后会保留节点状态；但是，发出此命令可能会对系统产生不利影响。参考：IPMI 1.5，第 17.3 节
Warm Reset (热复位)	0x3	所有	有效载荷, IPMB	热复位命令可复位 IPMC。复位之后会保留节点状态；但是，发出此命令可能会对系统产生不利影响。参考：IPMI 1.5，第 17.3 节
Get Self Test Results (获取自测试结果)	0x4	所有	有效载荷, IPMB	对于所有的板，从 R3U1 开始均支持此命令。在 R3U1 之前的发行版中，不支持此命令。
Broadcast 'Get Device ID' (广播“获取设备 ID”)	0x1	所有	仅 IPMB	此命令仅用于在 IPMB 总线上搜索板。不会从有效载荷发送此命令。

表 3-2 BMC 监视程序计时器命令，网络功能：应用程序 (0x06/0x07)

命令	操作码	支持的平台	支持的接口	备注
Reset Watchdog Timer (复位监视程序计时器)	0x22	所有	有效载荷, IPMB	使用 Set Watchdog Timer (设置监视程序计时器) 命令设置了监视程序参数之后, 此命令将启动并轻击 (pat) 监视程序。必须在正确设置监视程序参数之后再使用该命令。参考: IPMI 1.5, 第 21.5 节
Set Watchdog Timer (设置监视程序计时器)	0x24	所有	有效载荷, IPMB	不支持“预超时中断”和“关开机循环”这两项计时器操作。参考 IPMI 1.5, 第 21.6 节
Get Watchdog Timer (获取监视程序计时器)	0x25	所有	有效载荷, IPMB	参考 IPMI 1.5, 第 21.7 节

表 3-3 BMC 设备和消息传送命令，网络功能：应用程序，(0x06/0x07)

命令	操作码	支持的平台	支持的接口	备注
Send Message (发送消息)	0x34	所有	有效载荷, IPMB	参考 IPMI 1.5, 第 18.7 节
Master Write-Read (主读写)	0x52	所有	有效载荷, IPMB	用户必须了解所访问的设备的特性。不应发出此命令对 IPMI 总线进行寻址。参考 IPMI 1.5, 第 18.10 节

表 3-4 事件命令，网络功能：传感器/事件，(0x04/0x05)

命令	操作码	支持的平台	支持的接口	备注
Set Event Receiver (设置事件接收器)	0x00	所有	有效载荷，IPMB	此命令用于设置事件接收器的地址和 LUN。默认情况下，事件接收器地址为 0x20 (即 ShMM)。不应更改此地址，因为不会记录事件。参考 IPMI 1.5，第 23.1 节。获取此命令时，IPMC 应该会在板上重新发送声明的事件 (如果支持)，但 IPMC 复位事件除外。执行此操作是为了确保 NetConsole 操作能够顺利进行。
Get Event Receiver (获取事件接收器)	0x01	所有	有效载荷，IPMB	参考 IPMI 1.5，第 23.2 节
Platform Event (平台事件)	0x02	所有	有效载荷，IPMB	此命令用于在 SEL 中记录事件。如果 IPMC 从有效载荷获取此命令，它会将此命令发送到 ShMM，以便在 SEL 中记录事件；但是，从 ShMM 发送此命令没有任何意义。

表 3-5 传感器设备命令，网络功能：传感器/事件，(0x04/0x05)

命令	操作码	支持的平台	支持的接口	备注
Get Device SDR Info (获取设备 SDR 信息)	0x20	所有	有效载荷，IPMB	参考：IPMI 1.5，第 29.2 节
Get Device SDR (获取设备 SDR)	0x21	所有	有效载荷，IPMB	参考：IPMI 1.5，第 29.3 节
Reserve Device SDR Repository (保留设备 SDR 系统信息库)	0x22	所有	有效载荷，IPMB	参考：IPMI 1.5，第 29.4 节
Set Sensor Hysteresis (设置传感器磁滞)	0x24	所有	有效载荷，IPMB	参考：IPMI 1.5，第 29.6 节

表 3-5 传感器设备命令，网络功能：传感器/事件，(0x04/0x05) (续)

命令	操作码	支持的平台	支持的接口	备注
Get Sensor Hysteresis (获取传感器磁滞)	0x25	所有	有效载荷，IPMB	参考：IPMI 1.5，第 29.7 节
Set Sensor Threshold (设置传感器阈值)	0x26	所有	有效载荷，IPMB	参考：IPMI 1.5，第 29.8 节
Get Sensor Threshold (获取传感器阈值)	0x27	所有	有效载荷，IPMB	参考：IPMI 1.5，第 29.9 节
Set Sensor Event Enable (设置传感器事件启用)	0x28	所有	有效载荷，IPMB	参考：IPMI 1.5，第 29.10 节
Get Sensor Event Enable (获取传感器事件启用)	0x29	所有	有效载荷，IPMB	参考：IPMI 1.5，第 29.11 节
Get Sensor Event Status (获取传感器事件状态)	0x2B	所有	有效载荷，IPMB	参考：IPMI 1.5，第 29.13 节
Get Sensor Reading (获取传感器读数)	0x2D	所有	有效载荷，IPMB	参考：IPMI 1.5，第 29.14 节

表 3-6 FRU 设备命令，网络功能：存储，(0xA/0xB)

命令	操作码	支持的平台	支持的接口	备注
Get FRU Inventory Area Info (获取 FRU 库存区信息)	0x10	所有	有效载荷, IPMB	参考: IPMI 1.5, 第 28.1 节
Read FRU Data (读取 FRU 数据)	0x11	所有	有效载荷, IPMB	参考: IPMI 1.5, 第 28.2 节
Write FRU Data (写入 FRU 数据)	0x12	所有	有效载荷, IPMB	参考: IPMI 1.5, 第 28.3 节

表 3-7 ATCA 命令，网络功能：ATCA (0x2C/0x2D)

命令	操作码	支持的平台	支持的接口	备注
Get PICMG Properties (获取 PICMG 属性)	0x00	所有	有效载荷, IPMB	参考: PICMG 3.0R2.0ECN002, 第 3-10 节
Get Address Info (获取地址信息)	0x01	所有	有效载荷, IPMB	参考: PICMG 3.0R2.0ECN002, 第 3-9 节
FRU Control (FRU 控制)	0x04	所有	有效载荷, IPMB	参考: PICMG 3.0R2.0ECN002, 第 3-25 节
Get FRU LED Properties (获取 FRU LED 指示灯属性)	0x5	所有	有效载荷, IPMB	参考: PICMG 3.0R2.0ECN002, 第 3-27 节
Get LED Color Capabilities (获取 LED 指示灯颜色功能)	0x6	所有	有效载荷, IPMB	参考: PICMG 3.0R2.0ECN002, 第 3-28 节
Set FRU LED State (设置 FRU LED 指示灯状态)	0x7	所有	有效载荷, IPMB	参考: PICMG 3.0R2.0ECN002, 第 3-29 节

表 3-7 ATCA 命令，网络功能：ATCA (0x2C/0x2D) (续)

命令	操作码	支持的平台	支持的接口	备注
Get FRU LED State (获取 FRU LED 指示灯状态)	0x8	所有	有效载荷, IPMB	参考: PICMG 3.0R2.0ECN002, 第 3-30 节
Set IPMB State (设置 IPMB 状态)	0x9	所有	IPMB	参考: PICMG 3.0R2.0ECN002, 第 3-65 节
Set FRU Activation Policy (设置 FRU 激活策略)	0xA	所有	有效载荷, IPMB	参考: PICMG 3.0R2.0ECN002, 第 3-19 节
Get FRU Activation Policy (获取 FRU 激活策略)	0xB	所有	有效载荷, IPMB	参考: PICMG 3.0R2.0ECN002, 第 3-20 节
Set FRU Activation (设置 FRU 激活)	0xC	所有	有效载荷, IPMB	参考: PICMG 3.0R2.0ECN002, 第 3-18 节
Get Device Locator Record ID (获取设备定位器记录 ID)	0xD	所有	有效载荷, IPMB	参考: PICMG 3.0R2.0ECN002, 第 3-35 节
Set Port State (设置端口状态)	0xE	所有	IPMB	参考: PICMG 3.0R2.0ECN002, 第 3-54 节
Get Port State (获取端口状态)	0xF	所有	有效载荷, IPMB	参考: PICMG 3.0R2.0ECN002, 第 3-55 节
Compute Power Properties (计算电源属性)	0x10	所有	IPMB	参考: PICMG 3.0R2.0ECN002, 第 3-77 节
Set Power Level (设置电源级别)	0x11	所有	IPMB	参考: PICMG 3.0R2.0ECN002, 第 3-79 节

表 3-7 ATCA 命令，网络功能：ATCA (0x2C/0x2D) (续)

命令	操作码	支持的平台	支持的接口	备注
Get Power Level (获取电源级别)	0x12	所有	有效载荷, IPMB	参考: PICMG 3.0R2.0ECN002, 第 3-78 节
Get IPMB Link info (获取 IPMB 链路信息)	0x18	所有	有效载荷, IPMB	参考: PICMG 3.0R2.0ECN002, 第 3-63 节
FRU control capabilities (FRU 控制功能)	0x1E	所有	有效载荷, IPMB	在 IPMC 固件的某些版本中, 可能支持返回正常重新引导选项, 但是如果 OS 中不支持此功能, 此功能将不起作用。 参考: PICMG 3.0R2.0ECN002, 第 3-24 节

Sun 和 OEM IPMI 命令

表 3-8 Sun OEM 命令，网络功能：OEM, (0x2E/0x2F)

命令	操作码	支持的平台	支持的接口	备注
Set AMC timeout params (设置 AMC 超时参数)	0xF1	CP3220 CP3260 CP3270 T3-1BA	有效载荷, IPMB	可从 ShMM、有效载荷或调试接口发送此命令, 以设置启动 AMC 的超时值。直到所有 AMC 均变为 M4 状态或者此超时时间已到, IPMC 才会使有效载荷复位。超时值以秒为单位。IPMC 在永久存储中存储此超时值, 并且板复位后会保留该值。
Get AMC timeout parameter (获取 AMC 超时参数)	0xF0	CP3220 CP3260 CP3270 T3-1BA	有效载荷, IPMB	可从 ShMM、有效载荷或调试端口发送此命令, 以读取默认 AMC 超时值。
Set boot page (设置引导页面)	0x81	CP3020 CP3060 CP3220 CP3250 CP3260 CP3270	有效载荷, IPMB	可从 ShMM、有效载荷或调试接口发送此命令, 以设置 BIOS 引导页面。引导页面的默认值为 0。用户设置的值存储在 SEEPROM 中。下一次引导时, 将使用引导页面的同一个值。

表 3-8 Sun OEM 命令，网络功能：OEM，(0x2E/0x2F)（续）

命令	操作码	支持的平台	支持的接口	备注
Get boot page (获取引导页面)	0x82	CP3020 CP3060 CP3220 CP3250 CP3260 CP3270	有效载荷, IPMB	可从 ShMM、有效载荷或调试接口发送此命令，以读取 BIOS 引导的引导页面设置。
Set front panel reset button state (设置前面板复位按钮状态)	0x83	CP3010 CP3220 CP3020 CP3270	有效载荷, IPMB	软件可使用此命令来更改按下前面板复位按钮时 CPLD 处理前面板复位的方式。CPLD 电源打开时的默认值为 10。
Get front panel reset button state (获取前面板复位按钮状态)	0x84	CP3220 CP3010 CP3020 CP3270	有效载荷, IPMB	此命令可返回前面板复位按钮处理方式的当前设置。默认情况下，CPLD 电源打开时会显示 10（即按下此按钮会导致 CPU 加电复位）。
Set IPMC control bits (设置 IPMC 控制位)	0xE9	CP3220 CP3260 CP3270 T3-1BA	有效载荷, IPMB	此命令可给予 IPMC 控制权或者从 IPMC 获取控制权，以控制可能受 IPMC 或外部实体控制的各种功能。在更改控制字节中的任何位时，用户必须始终执行读取、修改、写入的顺序。
Get IPMC control bits (获取 IPMC 控制位)	0xE8	CP3220 CP3260 CP3270 T3-1BA	有效载荷, IPMB	此命令可返回 IPMC 控制位的当前设置。位 0 控制绿色 LED 指示灯的行为。
Set management port (设置管理端口)	0x9B	T3-1BA	有效载荷, IPMB	此命令可将管理端口访问权限路由到前面板或后面板。
Get management port (获取管理端口)	0x9C	T3-1BA	有效载荷, IPMB	此命令可返回管理端口访问权限的当前设置。
Get NIC IPMI PT firmware version (获取 NIC IPMI PT 固件版本)	0x87	CP3010 CP3020 CP3220	有效载荷, IPMB	此命令可返回在 Broadcom NIC 芯片中运行的 IPMI-PT 固件的版本字符串。

表 3-8 Sun OEM 命令，网络功能：OEM，(0x2E/0x2F) (续)

命令	操作码	支持的平台	支持的接口	备注
Get version (获取版本)	0x80	CP3270 T3-1BA	有效载荷, IPMB	此命令可返回 IPMC 固件版本和备用 CPLD 版本。尽管此命令会返回 IPMC 固件版本及 CPLD 版本，但此命令的主要目的是为 IPMC 版本提供 CPLD 版本。要替代此命令，请使用 IPMI “获取设备 ID” 命令。
Get Status (获取状态)	0x00	CP3020 CP3060 CP3220 CP3250 CP3260 CP3270 T3-1BA	有效载荷, IPMB	此命令可返回当前 IPMC 警报状态。
Graceful Payload Reset (正常 有效载荷复位)	0x11	CP3220 CP3250 CP3260 CP3270 T3-1BA	有效载荷, IPMB	此命令用于通知载体 IPMC 有效载荷关闭已完成。
Set SOL fail over link change timeouts (设置 SOL 故 障转移链路更 改超时)	0xE7	CP3270 T3-1BA	有效载荷, IPMB	此命令可设置主链路失败时 IPMC 等待切换至辅助链路的时间，以及主通道链路恢复时 IPMC 等待切换回主通道的时间。等待时间对于避免链路打开/关闭跳转非常有用。
Get SOL fail over link change timeouts (获取 SOL 故 障转移链路更 改超时)	0xE6	CP3270 T3-1BA	有效载荷, IPMB	此命令可返回 IPMC 控制位的当前设置。位 0 控制绿色 LED 指示灯的行为，位 1 控制故障 LED 指示灯的行为。
Set Payload Shutdown Timeout (设置有效载荷 关闭超时)	0x16	CP3220 CP3250 CP3260 CP3270 T3-1BA	有效载荷, IPMB	此命令设置有效载荷关闭的超时值。
Get Payload Shutdown Timeout (获取有效载荷 关闭超时)	0x15	CP3220 CP3250 CP3260 CP3270 T3-1BA	有效载荷, IPMB	此命令返回有效载荷关闭超时的当前值。

表 3-8 Sun OEM 命令，网络功能：OEM，(0x2E/0x2F)（续）

命令	操作码	支持的平台	支持的接口	备注
Set Thermal Trip（设置极限温度）	E5	T3-1BA	有效载荷，IPMB	此命令可启用或禁用决定何时关闭刀片服务器的极限温度阈值。
Get Thermal Trip（获取极限温度）	0xE4	T3-1BA	有效载荷，IPMB	此命令可返回极限温度的值。
Set XAUI mux control（设置 XAUI mux 控制）	0x95	CP3260 T3-1BA	有效载荷，IPMB	此命令用于将 XAUI1 和 XAUI2 接口路由到区域 2 或区域 3。
Get XAUI mux control（获取 XAUI mux 控制）	0x96	CP3260 T3-1BA	有效载荷，IPMB	此命令可返回板的 XAUI1 和 XAUI2 接口路由（区域 2 或区域 3）的当前设置。

提示 – 以下各部分提供了有关这些命令的更多详细信息。

Set AMC timeout params（设置 AMC 超时参数），操作码：0xF1，网络功能：0x2E

可从 ShMM、有效载荷或调试接口发送此命令，以设置启动 AMC 的超时值。

```

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 (获取 AMC 超时参数), 操作码 0xF0, 网络功能: 0x2E

可从 ShMM、有效载荷或调试端口发送此命令, 以读取默认 AMC 超时值。

```
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 (设置引导页面), 操作码 0x82, 网络 功能: 0x2E

可从 ShMM、有效载荷或调试接口发送此命令, 以设置 BIOS 引导页面。引导页面的默认值为 0。位 7 到 1 应设置为零。用户设置的值存储在 SEEPROM 中。下一次引导时, 将使用引导页面的同一个值。

```
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（获取引导页面），操作码 0x81，网络功能：0x2E

可从 ShMM、有效载荷或调试接口发送此命令，以读取 BIOS 引导页面。

```
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（设置前面板复位按钮状态），操作码 0x83，网络功能：0x2e

软件可使用此命令来更改按下前面板复位按钮时 CPLD 处理前面板复位的方式。CPLD 电源打开时的默认值为 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 state (获取前面板复位按钮状态), 操作码 0x84, 网络功能: 0x2E

此命令可返回前面板复位按钮处理方式的当前设置。默认情况下, CPLD 电源打开时会显示 10, 即, 按下此按钮会导致 CPU 加电复位。

```
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 (设置 IPMC 控制位), 操作码 0xE9, 网络功能: 0x2E

此命令可用于设置刀片服务器的 LED 指示灯和 AMC 关闭行为的配置。

注 – 在更改控制字节中的任何位时, 用户必须始终执行读取、修改、写入的顺序。

```
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) ontrl 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)

```

注 – 如果尝试将 0 写入任何保留位（3 到 7），IPMC 将拒绝命令，且完成代码为 0xCC。

Get IPMC control bits（获取 IPMC 控制位），操作码 0xE8，网络功能：0x2E

此命令可返回刀片服务器的 LED 指示灯和 AMC 关闭行为的当前配置。

```

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 (设置管理端口), 操作码 0x9B, 网络功能: 0x2E

此命令可用于将管理端口访问权限路由到前面板或后面板。

```
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 (获取管理端口), 操作码 0x9C, 网络功能: 0x2E

此命令可返回管理端口访问权限的当前设置。

```
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 (获取 NIC IPMI PT 固件版本), 操作码 0x87, 网络功能: 0x2E

此命令返回 IPMI PT 固件版本字符串。

```

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 (获取版本), 操作码 0x80, 网络功能: 0x2E

此命令可返回 IPMC 固件版本和备用 CPLD 版本。尽管此命令会返回 IPMC 固件版本及 CPLD 版本, 但此命令的主要目的是为 IPMC 版本提供 CPLD 版本。要替代此命令, 请使用 IPMI “获取设备 ID” 命令。

```

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)

```

注 – IPMC 版本读为 REV1 的低半字节、REV2 的高半字节，以及 REV2 的低半字节。

Get Status（获取状态），操作码 0x00，网络功能：0x2E

此命令可返回当前 IPMC 警报状态。

```

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（正常有效载荷复位）， 操作码 0x11，网络功能：0x2E

此命令用于通知载体 IPMC 有效载荷关闭已完成。从有效载荷获取此命令时以及关闭计时器过期之前，会继续执行后续操作。

```

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（设置有效载荷关闭超时），操作码 0x16，网络功能：0x2E

此命令设置有效载荷关闭的超时值。获取关闭请求时，IPMC 会向有效载荷发送警报，以便为关闭电源做好准备，此次超时以后，IPMC 将关闭电源。IPMC 复位后会保留该值。超时值以 100 毫秒滴答数为单位，也就是说，值 0x32（十进制数 50）表示 100 毫秒滴答 50 次，即 5 秒。

```
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（获取有效载荷关闭超时），操作码 0x15，网络功能：0x2E

此命令将返回有效载荷关闭超时的当前值。超时值以 100 毫秒滴答数为单位，也就是说，值 0x32（十进制数 50）表示 100 毫秒滴答 50 次，即 5 秒。

```
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 (设置 SOL 故障转移链路更改超时), 操作码 0xE7, 网络功能: 0x2E

此命令可设置主链路失败时 IPMC 等待切换至辅助 Serial over LAN (SOL) 链路的时间, 以及主通道链路恢复时 IPMC 等待切换回主通道的时间。等待时间对于避免链路打开/关闭跳转非常有用。

等待时间以秒为单位。例如, 字节 4 中的数字 10 (0xA) 表示, 在将链路切换到辅助通道之前, IPMC 将等待 10 秒。数字 15 (0xf) 则表示, 主通道恢复以后, IPMC 将等待 15 秒再切换回主通道。

```
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 (获取 SOL 故障转移链路更改超时), 操作码 0xE6, 网络功能: 0x2E

此命令可返回 Serial over LAN (SOL) 的 IPMC 控制位的当前设置。位 0 控制绿色 LED 指示灯的行为, 位 1 控制故障 LED 指示灯的行为。

```
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（设置极限温度），操作码 E5， 网络功能：0x2E

此命令可用于启用或禁用极限温度。极限温度设置将决定刀片服务器是否由于达到最高温度而关闭。此功能仅在 Netra SPARC T3-1BA 刀片服务器上可用。



注意 – 如果达到了温度阈值而没有关闭，可能会对刀片和系统造成损坏。除非运行环境允许覆盖默认值，否则请使用默认值。

在极端情况下（例如在战区使用时），用户可能需要覆盖最高温度阈值以防止刀片服务器关闭。这称为“战区模式”，用户可以覆盖极限温度值，以使刀片和系统保持运行（即使达到最高温度阈值）。即使禁用了关闭，传感器仍将记录阈值违规事件。

```

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（获取极限温度），操作码 0xE4，网络功能：0x2E

此命令返回极限温度的当前设置。

```
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（设置 XAUI mux 控制），操作码 0x95，网络功能：0x2E

此命令可用于将 XAUI1 和 XAUI2 接口路由到区域 2 或区域 3。仅适用于 Sun Netra CP3260 板。

```
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 (获取 XAUI mux 控制), 操作码 0x96, 网络功能: 0x2E

此命令可返回 XAUI1 和 XAUI2 接口路由的当前设置 (区域 2 或区域 3)。仅适用于 Sun Netra CP3260 板。

```
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.
```


附录 A

实体路径

实体表示系统中的物理组件。每个实体都具有一个称为实体路径的唯一标识符。实体路径由组件在系统的物理包含分层结构中的位置进行定义。一个实体路径包含一系列 {实体类型, 实体位置} 对，从实体开始，到系统分层结构的根结束。

例如，位于位置 3 的 ATCA 机箱的插槽 4 中的刀片的实体路径为：

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

其中 SAHPI_ENT_ROOT 为实体类型，0 为实体位置。

表 A-1 包含 Sun Netra CT900 服务器资源表的一个简短示例。在本示例中，系统包含两个 ShMM 500 机框管理器、两个 CP3140 交换机刀片（插槽 7 和 8）、一个 CP3010 刀片（插槽 3）、一个 CP3020 刀片（插槽 14）以及一个 CP3060 刀片（插槽 12）。

表 A-1 资源表

资源标记	实体路径
机框资源	{SYSTEM_CHASSIS, 1}
OEM 插槽 1	{SYSTEM_CHASSIS, 1} {OEM_SYSINT_SPECIFIC, 1}
ATCA 板插槽 1	{SYSTEM_CHASSIS, 1} {PHYSICAL_SLOT, 1}
ATCA 板插槽 2	{SYSTEM_CHASSIS, 1} {PHYSICAL_SLOT, 2}
ATCA 板插槽 3	{SYSTEM_CHASSIS, 1} {PHYSICAL_SLOT, 3}
ATCA 板插槽 4	{SYSTEM_CHASSIS, 1} {PHYSICAL_SLOT, 4}
ATCA 板插槽 5	{SYSTEM_CHASSIS, 1} {PHYSICAL_SLOT, 5}
ATCA 板插槽 6	{SYSTEM_CHASSIS, 1} {PHYSICAL_SLOT, 6}
ATCA 板插槽 7	{SYSTEM_CHASSIS, 1} {PHYSICAL_SLOT, 7}

表 A-1 资源表 (续)

资源标记	实体路径
ATCA 板插槽 8	{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,8}
ATCA 板插槽 9	{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,9}
ATCA 板插槽 10	{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,10}
ATCA 板插槽 11	{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,11}
ATCA 板插槽 12	{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,12}
ATCA 板插槽 13	{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,13}
ATCA 板插槽 14	{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,14}
电源输入模块插槽 1	{SYSTEM_CHASSIS,1}{POWER_ENTRY_MODULE_SLOT,1}
电源输入模块插槽 2	{SYSTEM_CHASSIS,1}{POWER_ENTRY_MODULE_SLOT,2}
机框 FRU 信息插槽 1	{SYSTEM_CHASSIS,1}{SHELF_FRU_DEVICE_SLOT,1}
机框 FRU 信息插槽 2	{SYSTEM_CHASSIS,1}{SHELF_FRU_DEVICE_SLOT,2}
专用 ShMC 插槽 1	{SYSTEM_CHASSIS,1}{SHELF_MANAGER_SLOT,1}
专用 ShMC 插槽 2	{SYSTEM_CHASSIS,1}{SHELF_MANAGER_SLOT,2}
风扇托盘插槽 1	{SYSTEM_CHASSIS,1}{FAN_TRAY_SLOT,1}
风扇托盘插槽 2	{SYSTEM_CHASSIS,1}{FAN_TRAY_SLOT,2}
风扇托盘插槽 3	{SYSTEM_CHASSIS,1}{FAN_TRAY_SLOT,3}
报警插槽 1	{SYSTEM_CHASSIS,1}{ALARM_SLOT,1}
PPS BMC	{SYSTEM_CHASSIS,1}{SHELF_MANAGER,0}
机框 EEPROM 1	{SYSTEM_CHASSIS,1}{SHELF_FRU_DEVICE_SLOT,1}{SHELF_FRU_DEVICE,1}
机框 EEPROM 2	{SYSTEM_CHASSIS,1}{SHELF_FRU_DEVICE_SLOT,2}{SHELF_FRU_DEVICE,2}
SAP 板	{SYSTEM_CHASSIS,1}{ALARM_SLOT,1}{ALARM_MANAGER,1}
风扇托盘 0	{SYSTEM_CHASSIS,1}{FAN_TRAY_SLOT,1}{COOLING_UNIT,1}
风扇托盘 1	{SYSTEM_CHASSIS,1}{FAN_TRAY_SLOT,2}{COOLING_UNIT,2}
风扇托盘 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}
CP3140H-BEG	{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,7}{PICMG_FRONT_BLADE,7}
CP3140H-BEG	{SYSTEM_CHASSIS,1}{PHYSICAL_SLOT,8}{PICMG_FRONT_BLADE,8}

表 A-1 资源表 (续)

资源标记	实体路径
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 插槽	{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 插槽	{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 插槽	{SYSTEM_CHASSIS, 1} {PHYSICAL_SLOT, 12} {PICMG_FRONT_BLADE, 12} {RTM_SLOT, 1} {BACK_PANEL_BOARD, 1}

附录 B

资源数据记录

资源数据记录 (resource data record, RDR) 定义了与资源关联的管理工具（传感器、控制机制、监视程序计时器、库存数据系统信息库或报警器）。

本附录包括以下 RDR:

- 第 106 页的 “Sun Netra CP3010 板资源数据记录”
- 第 108 页的 “Sun Netra CP3020 板资源数据记录”
- 第 110 页的 “Sun Netra CP3060 板资源数据记录”
- 第 112 页的 “Sun Netra CP3140 交换机资源数据记录”
- 第 114 页的 “Sun Netra CP3240 交换机资源数据记录”
- 第 118 页的 “Sun Netra CP3220 板资源数据记录”
- 第 120 页的 “Sun Netra CP3260 板资源数据记录”
- 第 122 页的 “Sun Netra CP32x0 双 SAS 存储高级后部转换模块 (ARTM-HD) 资源数据记录”

表 B-1 包含 Sun Netra CP3010 板的资源数据记录。

表 B-1 Sun Netra CP3010 板资源数据记录

ID 字符串	类型
蓝色 LED 指示灯	ctrlRdr (2)
LED 指示灯 1	ctrlRdr (2)
LED 指示灯 2	ctrlRdr (2)
FRU 所需电源	ctrlRdr (2)
IPMB-A 状态控制	ctrlRdr (2)
IPMB-B 状态控制	ctrlRdr (2)
FRU 重新引导和诊断控制	ctrlRdr (2)
FRU IPM 控制器复位控制	ctrlRdr (2)
FRU 0 热交换	sensorRdr (3)
系统事件	sensorRdr (3)
RTM 存在	sensorRdr (3)
E 键控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 1	sensorRdr (3)
E 键控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 2	sensorRdr (3)
E 键控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 1	sensorRdr (3)
E 键控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 2	sensorRdr (3)
IPMB 物理	sensorRdr (3)
NetraCP-3010	inventoryRdr (4)
{RTM_SLOT,1}	
FRU 激活控制	ctrlRdr (2)
插槽状态传感器	sensorRdr (3)
已分配功率传感器	sensorRdr (3)
最大功率容量传感器	sensorRdr (3)
{PROCESSOR,0}	
BMC 监视程序	sensorRdr (3)
CPU1 温度	sensorRdr (3)
CPU2 温度	sensorRdr (3)
进气口温度	sensorRdr (3)
版本更改	sensorRdr (3)

表 B-1 Sun Netra CP3010 板资源数据记录 (续)

ID 字符串	类型
{POWER_MODULE,0}	
+12.0V	sensorRdr (3)
-12.0V	sensorRdr (3)
+5.0V VCC	sensorRdr (3)
+3.3V 主电源	sensorRdr (3)
+3.3V 备用电源	sensorRdr (3)
VBAT	sensorRdr (3)
VDD 核心 0	sensorRdr (3)
VDD 核心 1	sensorRdr (3)
VTI 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)

表 B-2 包含 Sun Netra CP3020 板的资源数据记录。

表 B-2 Sun Netra CP3020 板资源数据记录

ID 字符串	类型
蓝色 LED 指示灯	ctrlRdr (2)
LED 指示灯 1	ctrlRdr (2)
LED 指示灯 2	ctrlRdr (2)
FRU 所需电源	ctrlRdr (2)
IPMB-A 状态控制	ctrlRdr (2)
IPMB-B 状态控制	ctrlRdr (2)
FRU 重新引导和诊断控制	ctrlRdr (2)
FRU IPM 控制器复位控制	ctrlRdr (2)
FRU 0 热交换	sensorRdr (3)
系统事件	sensorRdr (3)
RTM 存在	sensorRdr (3)
E 键控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 1	sensorRdr (3)
E 键控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 2	sensorRdr (3)
E 键控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 1	sensorRdr (3)
E 键控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 2	sensorRdr (3)
IPMB 物理	sensorRdr (3)
NetraCP-3020	inventoryRdr (4)
{PROCESSOR,0}	
BMC 监视程序	sensorRdr (3)
CPU Tcontrol	sensorRdr (3)
板温度	sensorRdr (3)
ADM 内部温度	sensorRdr (3)
版本更改	sensorRdr (3)
{POWER_MODULE,0}	
+12.0V 运行	sensorRdr (3)
-12.0V 运行	sensorRdr (3)
VCC 5V 运行	sensorRdr (3)
+3.3V 运行	sensorRdr (3)
+3.3V ALW	sensorRdr (3)

表 B-2 Sun Netra CP3020 板资源数据记录 (续)

ID 字符串	类型
VCC RTC	sensorRdr (3)
VDD 核心运行	sensorRdr (3)
VCC 1.8V 双	sensorRdr (3)
DDR VTT 1.3V 运行	sensorRdr (3)
VCC 1.2V 运行	sensorRdr (3)
VCC 5V ALW	sensorRdr (3) r
VDD PU 2.5V 运行	sensorRdr (3)
DDR VDD 2.6V 运行	sensorRdr (3)
VCC 1.8V 运行	sensorRdr (3)
{RTM_SLOT,1}	
FRU 激活控制	ctrlRdr (2)
插槽状态传感器	sensorRdr (3)
已分配功率传感器	sensorRdr (3)
最大功率容量传感器	sensorRdr (3)

表 B-3 包含 Sun Netra CP3060 板的资源数据记录。

表 B-3 Sun Netra CP3060 板资源数据记录

ID 字符串	类型
蓝色 LED 指示灯	ctrlRdr (2)
LED 指示灯 1	ctrlRdr (2)
LED 指示灯 2	ctrlRdr (2)
FRU 所需电源	ctrlRdr (2)
IPMB-A 状态控制	ctrlRdr (2)
IPMB-B 状态控制	ctrlRdr (2)
FRU 重新引导和诊断控制	ctrlRdr (2)
FRU IPM 控制器复位控制	ctrlRdr (2)
AMC 通电顺序提交	ctrlRdr (2)
AMC 通电顺序 #0	ctrlRdr (2)
FRU 0 热交换	sensorRdr (3)
RTM 存在	sensorRdr (3)
E 键控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 1	sensorRdr (3)
E 键控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 2	sensorRdr (3)
E 键控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 1	sensorRdr (3)
E 键控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 2	sensorRdr (3)
IPMB 物理	sensorRdr (3)
AMC 通电顺序提交状态	sensorRdr (3)
NetraCP-3060	inventoryRdr (4)
{PROCESSOR,0}	
BMC 监视程序	sensorRdr (3)
CPU 温度 1	sensorRdr (3)
CPU 温度 2	sensorRdr (3)
板温度	sensorRdr (3)
版本更改	sensorRdr (3)
{POWER_MODULE,0}	
12.0V	sensorRdr (3)
5.0V	sensorRdr (3)
3.3V	sensorRdr (3)

表 B-3 Sun Netra CP3060 板资源数据记录 (续)

ID 字符串	类型
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 所需电源	ctrlRdr (2)
FRU 重新引导和诊断控制	ctrlRdr (2)
RTM 热交换	sensorRdr (3)

表 B-4 包含 Sun Netra CP3140 交换机的资源数据记录。

表 B-4 Sun Netra CP3140 交换机资源数据记录

ID 字符串	类型
蓝色 LED 指示灯	ctrlRdr (2)
LED 指示灯 1	ctrlRdr (2)
LED 指示灯 2	ctrlRdr (2)
FRU 所需电源	ctrlRdr (2)
IPMB-A 状态控制	ctrlRdr (2)
IPMB-B 状态控制	ctrlRdr (2)
FRU 重新引导和诊断控制	ctrlRdr (2)
FRU IPM 控制器复位控制	ctrlRdr (2)
FRU 0 热交换	sensorRdr (3)
-48V 报警	sensorRdr (3)
RTM 存在	sensorRdr (3)
OOS LED 指示灯	sensorRdr (3)
活动 LED 指示灯	sensorRdr (3)
5V	sensorRdr (3)
3.3V	sensorRdr (3)
2.5V	sensorRdr (3)
1.5V	sensorRdr (3)
1.25V	sensorRdr (3)
板温度 1	sensorRdr (3)
板温度 2	sensorRdr (3)
IPMC 固件	sensorRdr (3)
BMC 监视程序	sensorRdr (3)
E 钥控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 1 通道 1	sensorRdr (3)
E 钥控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 1 通道 2	sensorRdr (3)
E 钥控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 1	sensorRdr (3)
E 钥控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 2	sensorRdr (3)
E 钥控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 3	sensorRdr (3)
E 钥控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 4	sensorRdr (3)
E 钥控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 5	sensorRdr (3)

表 B-4 Sun Netra CP3140 交换机资源数据记录 (续)

ID 字符串	类型
E 钥控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 6	sensorRdr (3)
E 钥控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 7	sensorRdr (3)
E 钥控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 8	sensorRdr (3)
E 钥控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 9	sensorRdr (3)
E 钥控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 10	sensorRdr (3)
E 钥控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 11	sensorRdr (3) t
E 钥控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 12	sensorRdr (3)
E 钥控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 13	sensorRdr (3)
E 钥控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 14	sensorRdr (3)
E 钥控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 15	sensorRdr (3)
E 钥控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 16	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 1	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 2	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 3	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 4	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 5	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 6	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 7	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 8	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 9	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 10	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 11	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 12	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 13	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 14	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 15	sensorRdr (3)
IPMB 链路	sensorRdr (3)
CP3140H-BEG	inventoryRdr (4) t

表 B-5 包含 Sun Netra CP3240 交换机的资源数据记录。

表 B-5 Sun Netra CP3240 交换机资源数据记录

ID 字符串	类型
蓝色 LED 指示灯	ctrlRdr (2)
LED 指示灯 1	ctrlRdr (2)
LED 指示灯 2	ctrlRdr (2)
FRU 所需电源	ctrlRdr (2)
IPMB-A 状态控制	ctrlRdr (2)
IPMB-B 状态控制	ctrlRdr (2)
FRU 重新引导和诊断控制	ctrlRdr (2)
FRU IPM 控制器复位控制	ctrlRdr (2)
热交换	sensorRdr (3)
热交换 AMC 0	sensorRdr (3)
热交换 AMC 1	sensorRdr (3)
热交换 AMC 2	sensorRdr (3)
站点 1 PWR cur	sensorRdr (3)
站点 1 PWR	sensorRdr (3)
站点 1 MP	sensorRdr (3)
站点 2 PWR cur	sensorRdr (3)
站点 2 PWR	sensorRdr (3)
站点 2 MP	sensorRdr (3)
站点 3 PWR cur	sensorRdr (3)
站点 3 PWR	sensorRdr (3)
站点 3 MP	sensorRdr (3)
E 键控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 1 通道 1	sensorRdr (3)
E 键控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 1 通道 2	sensorRdr (3)
E 键控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 1	sensorRdr (3)
E 键控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 2	sensorRdr (3)
E 键控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 3	sensorRdr (3)
E 键控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 4	sensorRdr (3)
E 键控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 5	sensorRdr (3)
E 键控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 6	sensorRdr (3)

表 B-5 Sun Netra CP3240 交换机资源数据记录 (续)

ID 字符串	类型
E 钥控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 7	sensorRdr (3)
E 钥控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 8	sensorRdr (3)
E 钥控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 9	sensorRdr (3)
E 钥控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 10	sensorRdr (3)
E 钥控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 11	sensorRdr (3)
E 钥控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 12	sensorRdr (3)
E 钥控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 13	sensorRdr (3)
E 钥控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 14	sensorRdr (3)
E 钥控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 15	sensorRdr (3)
E 钥控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 16	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 1 通道 1	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 1	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 1 通道 2	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 2	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 1 通道 3	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 3	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 1 通道 4	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 4	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 1 通道 5	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 5	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 1 通道 6	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 6	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 1 通道 7	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 7	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 1 通道 8	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 8	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 1 通道 9	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 9	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 1 通道 10	sensorRdr (3)
E 钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 10	sensorRdr (3)

表 B-5 Sun Netra CP3240 交换机资源数据记录 (续)

ID 字符串	类型
E 键控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 1 通道 11	sensorRdr (3)
E 键控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 11	sensorRdr (3)
E 键控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 1 通道 12	sensorRdr (3)
E 键控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 12	sensorRdr (3)
E 键控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 1 通道 13	sensorRdr (3)
E 键控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 13	sensorRdr (3)
E 键控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 1 通道 14	sensorRdr (3)
E 键控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 14	sensorRdr (3)
E 键控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 1 通道 15	sensorRdr (3)
E 键控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 15	sensorRdr (3)
IPMB 物理	sensorRdr (3)
CP3240H-BEX-Z	inventoryRdr (4)
{PROCESSOR,0}	
BMC 监视程序	sensorRdr (3)
基本 CPU 温度	sensorRdr (3) n
光纤网络 CPU 温度	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 热交换	sensorRdr (3)
RTM 存在	sensorRdr (3)
RTM 温度	sensorRdr (3)

表 B-5 Sun Netra CP3240 交换机资源数据记录 (续)

ID 字符串	类型
{OPERATING_SYSTEM,0}	
基接口早期	sensorRdr (3)
基接口完整	sensorRdr (3)
基接口良好	sensorRdr (3)
交换接口早期	sensorRdr (3)
交换接口完整	sensorRdr (3)
交换接口良好	sensorRdr (3)
{RTM_SLOT,1}{BACK_PANEL_BOARD,1}	
蓝色 LED 指示灯	ctrlRdr (2)
FRU 所需电源	ctrlRdr (2)
FRU 重新引导和诊断控制	ctrlRdr (2)
XCP3240H-RTM-CUZ	inventoryRdr (4) E

表 B-6 包含 Sun Netra CP3220 板的资源数据记录。

表 B-6 Sun Netra CP3220 板资源数据记录

ID 字符串	类型
蓝色 LED 指示灯	ctrlRdr (2)
LED 指示灯 1	ctrlRdr (2)
LED 指示灯 2	ctrlRdr (2)
FRU 所需电源	ctrlRdr (2)
IPMB-A 状态控制	ctrlRdr (2)
IPMB-B 状态控制	ctrlRdr (2)
FRU 重新引导和诊断控制	ctrlRdr (2)
FRU IPM 控制器复位控制	ctrlRdr (2)
FRU 0 热交换	sensorRdr (3)
热交换 AMC 5	sensorRdr (3)
热交换 AMC 6	sensorRdr (3)
板进气口温度	sensorRdr (3)
E 键控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 1	sensorRdr (3)
E 键控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 2	sensorRdr (3)
E 键控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 1	sensorRdr (3)
E 键控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 2	sensorRdr (3)
IPMB 物理	sensorRdr (3)
NetraCP-3220	inventoryRdr (4)
{PROCESSOR,0}	
BMC 监视程序	sensorRdr (3)
CPU 表面温度	sensorRdr (3)
区域 3 温度	sensorRdr (3)
AMC 区域温度	sensorRdr (3)
版本更改	sensorRdr (3)
{POWER_MODULE,0}	
12.0V	sensorRdr (3)
5.0V	sensorRdr (3)
3.3V	sensorRdr (3)
3.3V STBY	sensorRdr (3)

表 B-6 Sun Netra CP3220 板资源数据记录 (续)

ID 字符串	类型
电池电压	sensorRdr (3)
VCC 1.15V M 双	sensorRdr (3)
Proc0 0.9V DDR	sensorRdr (3)
VCC 1.2V HT	sensorRdr (3)
Proc0 核心 NB	sensorRdr (3)
VCC 1.15V M 运行	sensorRdr (3)
VCC 1.2V 运行	sensorRdr (3)
Proc0 1.8V DDR	sensorRdr (3)
VCC 1.5V 运行	sensorRdr (3)
Proc0 核心	sensorRdr (3)
PM 主端温度	sensorRdr (3)
PM 次端温度	sensorRdr (3)
-48V A 干线	sensorRdr (3)
-48V B 干线	sensorRdr (3)
-48V 电压	sensorRdr (3)
-48V 电流	sensorRdr (3)
12V 电流	sensorRdr (3)

表 B-7 包含 Sun Netra CP3260 板的资源数据记录。

表 B-7 Sun Netra CP3260 板资源数据记录

ID 字符串	类型
蓝色 LED 指示灯	ctrlRdr(2)
LED 指示灯 1	ctrlRdr(2)
LED 指示灯 2	ctrlRdr(2)
FRU 所需电源	ctrlRdr(2)
IPMB-A 状态控制	ctrlRdr(2)
IPMB-B 状态控制	ctrlRdr(2)
FRU 重新引导和诊断控制	ctrlRdr(2)
FRU IPM 控制器复位控制	ctrlRdr(2)
AMC 通电顺序提交	ctrlRdr(2)
FRU 0 热交换	sensorRdr(3)
E 键控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 1	sensorRdr(3)
E 键控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 2	sensorRdr(3)
E 键控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 1 通道 1	sensorRdr(3)
E 键控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 1 通道 2	sensorRdr(3)
IPMB 物理	sensorRdr(3)
AMC 通电顺序提交状态	sensorRdr(3)
Netra CP3260	inventoryRdr(4)
{PROCESSOR,0}	
BMC 监视程序	sensorRdr(3)
CPU 温度 1	sensorRdr(3)
CPU 温度 2	sensorRdr(3)
板温度	sensorRdr(3)
{POWER_MODULE,0}	
12.0V	sensorRdr(3)
5.0V	sensorRdr(3)
3.3V	sensorRdr(3)
3.3V STBY	sensorRdr(3)

表 B-7 Sun Netra CP3260 板资源数据记录 (续)

ID 字符串	类型
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)

表 B-8 包含 Sun Netra CP32x0 ARTM-HD 的资源数据记录。

表 B-8 Sun Netra CP32x0 双 SAS 存储高级后部转换模块 (ARTM-HD) 资源数据记录

ID 字符串	类型
蓝色 LED 指示灯	ctrlRdr(2)
LED 指示灯 1	ctrlRdr(2)
LED 指示灯 2	ctrlRdr(2)
应用程序 LED 指示灯	1 ctrlRdr(2)
应用程序 LED 指示灯	2 ctrlRdr(2)
FRU 所需电源	ctrlRdr(2)
FRU 重新引导和诊断控制	ctrlRdr(2)
ARTM 热交换	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

表 B-9 包含 Sun Netra CP3250 板的资源数据记录。

表 B-9 Sun Netra CP3250 板资源数据记录

ID 字符串	类型
蓝色 LED 指示灯	ctrlRdr(2)
LED 指示灯 1	ctrlRdr(2)
LED 指示灯 2	ctrlRdr(2)
FRU 所需电源	ctrlRdr(2)
IPMB-A 状态控制	ctrlRdr(2)
IPMB-B 状态控制	ctrlRdr(2)
FRU 重新引导和诊断控制	ctrlRdr(2)
FRU IPM 控制器复位控制	ctrlRdr(2)
AMC 通电顺序提交	ctrlRdr(2)
AMC 通电顺序提交状态	sensorRdr(3)
FRU 0 热交换	sensorRdr(3)
ARTM 热交换	sensorRdr(3)
版本更改	sensorRdr(3)
P48V 报警	sensorRdr(3)
IPMB 物理	sensorRdr(3)
E 钥控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 1	sensorRdr(3)
钥控链路状态: 0 接口, 链路类型 1, 链路类型 Ext 0 通道 2	sensorRdr(3)
钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 1	sensorRdr(3)
钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 0 通道 2	sensorRdr(3)
钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 1 通道 1	sensorRdr(3)
钥控链路状态: 1 接口, 链路类型 2, 链路类型 Ext 1 通道 2	sensorRdr(3)
Netra CP3250	inventoryRdr(4)
{PROCESSOR,0}	
BMC 监视程序	sensorRdr(3)
CPU 温度 1	sensorRdr(3)
CPU 温度 2	sensorRdr(3)
板温度	sensorRdr(3)
系统固件进度	sensorRdr(3)
正常重新引导	sensorRdr(3)

表 B-9 Sun Netra CP3250 板资源数据记录 (续)

ID 字符串	类型
{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)

附录 C

Sun Netra CP3140 SNMP MIB 对象和陷阱

本附录包含 Sun Netra CP3140 交换机刀片上支持或不支持的 SNMP MIB 对象和陷阱。FASTPATH 4.2 在 Sun Netra CP3140 交换机刀片上使用。FASTPATH 4.2 支持或不支持本附录中介绍的对象和陷阱。每个表都包括对象的名称、对象的支持状态以及访问控制。

有关 Netra CP3140 交换机刀片上 SNMP 的更多信息，请参阅《Sun Netra CT900 Server Switch Software Reference Manual》。可以从以下网址获得此手册：

<http://www.sun.com/documentation/>

表 C-1 802.3AD 链路聚合 MIB

对象	支持	访问
lagMIBObjects 组		
dot3adTablesLastChanged	是	RO
dot3adAggTable		
索引: dot3adAggIndex		
dot3adAggMACAddress	是	RO
dot3adAggActorSystemPriority	是	RW
dot3adAggActorSystemID	是	RO
dot3adAggAggregateOrIndividual	是	RO
dot3adAggActorAdminKey	是	RW
dot3adAggActorOperKey	是	RO
dot3adAggPartnerSystemID	是	RO

表 C-1 802.3AD 链路聚合 MIB (续)

对象	支持	访问
dot3adAggPartnerSystemPriority	是	RO
dot3adAggPartnerOperKey	是	RO
dot3adAggCollectorMaxDelay	是	RW
dot3adAggPortListTable		
索引: dot3adAggIndex		
dot3adAggPortListPorts	是	RO
dot3adAggPortTable		
索引: dot3adAggPortIndex		
dot3adAggPortActorSystemPriority	是	RW
dot3adAggPortActorSystemID	是	RO
dot3adAggPortActorAdminKey	是	RW
dot3adAggPortActorOperKey	是	RW
dot3adAggPortPartnerAdminSystemPriority	是	RW
dot3adAggPortPartnerOperSystemPriority	是	RO
dot3adAggPortPartnerAdminSystemID	是	RW
dot3adAggPortPartnerOperSystemID	是	RO
dot3adAggPortPartnerAdminKey	是	RW
dot3adAggPortPartnerOperKey	是	RO
dot3adAggPortSelectedAggID	是	RO
dot3adAggPortAttachedAggID	是	RO
dot3adAggPortActorPort	是	RO
dot3adAggPortActorPortPriority	是	RW
dot3adAggPortPartnerAdminPort	是	RW
dot3adAggPortPartnerOperPort	是	RO
dot3adAggPortPartnerAdminPortPriority	是	RW
dot3adAggPortPartnerOperPortPriority	是	RO
dot3adAggPortActorAdminState	是	RW
dot3adAggPortActorOperState	是	RO

表 C-1 802.3AD 链路聚合 MIB (续)

对象	支持	访问
dot3adAggPortPartnerAdminState	是	RW
dot3adAggPortPartnerOperState	是	RO
dot3adAggPortAggregateOrIndividual	是	RO
dot3adAggPortStatsTable		
索引: dot3adAggPortIndex		
dot3adAggPortStatsLACPDUrx	是	RO
dot3adAggPortStatsMarkerPDUsRx	是	RO
dot3adAggPortStatsMarkerResponsePDUsRx	否	N/A
dot3adAggPortStatsUnknownRx	是	RO
dot3adAggPortStatsIllegalRx	是	RO
dot3adAggPortStatsLACPDUtx	是	RO
dot3adAggPortStatsMarkerPDUsTx	否	N/A
dot3adAggPortStatsMarkerResponsePDUsTx	是	RO
dot3adAggPortDebugTable		
索引: dot3adAggPortIndex		
dot3adAggPortDebugRxState	否	N/A
dot3adAggPortDebugLastRxTime	否	N/A
dot3adAggPortDebugMuxState	否	N/A
dot3adAggPortDebugMuxReason	否	N/A
dot3adAggPortDebugActorChurnState	否	N/A
dot3adAggPortDebugPartnerChurnState	否	N/A
dot3adAggPortDebugActorChurnCount	否	N/A
dot3adAggPortDebugPartnerChurnCount	否	N/A
dot3adAggPortDebugActorSyncTransitionCount	否	N/A
dot3adAggPortDebugPartnerSyncTransitionCount	否	N/A
dot3adAggPortDebugActorChangeCount	否	N/A
dot3adAggPortDebugPartnerChangeCount	否	N/A

表 C-2 RFC 2934 PIM-SM/DM MIB

对象	支持	访问
pim		
pimJoinPruneInterval	是	RW
pimInterfaceTable		
索引: pimInterfaceIfIndex		
pimInterfaceAddress	是	RO
pimInterfaceNetMask	是	RO
pimInterfaceMode	是	RC
pimInterfaceDR	是	RO
pimInterfaceHelloInterval	是	RC
pimInterfaceStatus	是	RC
pimInterfaceJoinPruneInterval	否	N/A
pimInterfaceCBSRPreference	是	RC
pimNeighborTable		
索引: pimNeighborAddress		
pimNeighborIfIndex	是	RO
pimNeighborUpTime	是	RO
pimNeighborExpiryTime	是	RO
pimNeighborMode	是	RO
pimIpMRouteTable		
索引: ipMRouteGroup、ipMRouteSource、ipMRouteSourceMask		
pimIpMRouteUpstreamAssertTimer	是	RO
pimIpMRouteAssertMetric	是	RO
pimIpMRouteAssertMetricPref	是	RO
pimIpMRouteAssertRPTBit	是	RO
pimIpMRouteFlags	是	RO

表 C-2 RFC 2934 PIM-SM/DM MIB (续)

对象	支持	访问
pimIpMRouteNextHopTable		
索引: ipMRouteNextHopGroup、ipMRouteNextHopSource、ipMRouteNextHopSourceMask、ipMRouteNextHopIfIndex、ipMRouteNextHopAddress		
pimIpMRouteNextHopPruneReason	是	RO
pimRPTable		
索引: pimRPGroupAddress、pimRPAddress		
pimRPState	否	N/A
pimRPStateTimer	否	N/A
pimRPLastChange	否	N/A
pimRPRowStatus	否	N/A
pimRPSetTable		
索引: pimRPSetComponent、pimRPSetGroupAddress、pimRPSetGroupMask、pimRPSetAddress		
pimRPSetHoldTime	是	RO
pimRPSetExpiryTime	是	RO
pimCandidateRPTable		
索引: pimCandidateRPGroupAddress、pimCandidateRPGroupMask		
pimCandidateRPAddress	是	RO
pimCandidateRPRowStatus	是	RO
pimComponentTable		
索引: pimComponentIndex		
pimComponentBSRAddress	是	RO
pimComponentBSRExpiryTime	是	RO
pimComponentCRPHoldTime	是	RO
pimComponentStatus	是	RO

表 C-3 RFC 2933 IGMP MIB

对象	支持	访问
igmpInterfaceTable		
索引: igmpInterfaceIfIndex		
igmpInterfaceQueryInterval	是	RC
igmpInterfaceStatus	是	RC
igmpInterfaceVersion	是	RC
igmpInterfaceQuerier	是	RO
igmpInterfaceQueryMaxResponseTime	是	RC
igmpInterfaceQuerierUpTime	是	RO
igmpInterfaceQuerierExpiryTime	是	RO
igmpInterfaceVersion1QuerierTimer	否	N/A
igmpInterfaceWrongVersionQueries	是	RO
igmpInterfaceJoins	是	RO
igmpInterfaceProxyIfIndex	否	N/A
igmpInterfaceGroups	是	RO
igmpInterfaceRobustness	是	RC
igmpInterfaceLastMembQueryIntvl	是	RC
igmpCacheTable		
索引: igmpCacheAddress、igmpCacheIfIndex		
igmpCacheSelf	否	N/A
igmpCacheLastReporter	是	RO
igmpCacheUpTime	是	RO
igmpCacheExpiryTime	是	RO
igmpCacheStatus	是	RO
igmpCacheVersion1HostTimer	是	RO

表 C-4 RFC 2932 IPv4 多播路由 MIB

对象	支持	访问
ipMRoute		
ipMRouteEnable	是	RW
ipMRouteEntryCount	是	RO
ipMRouteTable		
索引: ipMRouteGroup、ipMRouteSource、ipMRouteSourceMask		
ipMRouteUpstreamNeighbor	是	RO
ipMRouteInIfIndex	是	RO
ipMRouteUpTime	是	RO
ipMRouteExpiryTime	是	RO
ipMRoutePkts	否	N/A
ipMRouteDifferentInIfPackets	否	N/A
ipMRouteOctets	否	N/A
ipMRouteProtocol	是	RO
ipMRouteRtProto	否	N/A
ipMRouteRtAddress	是	RO
ipMRouteRtMask	是	RO
ipMRouteRtType	是	RO
ipMRouteHCOctets	否	N/A
ipMRouteNextHopTable		
索引: ipMRouteNextHopGroup、ipMRouteNextHopSource、ipMRouteNextHopSourceMask、ipMRouteNextHopIfIndex、ipMRouteNextHopAddress		
ipMRouteNextHopState	否	N/A
ipMRouteNextHopUpTime	否	N/A
ipMRouteNextHopExpiryTime	否	N/A
ipMRouteNextHopClosestMemberHops	否	N/A
ipMRouteNextHopProtocol	否	N/A
ipMRouteNextHopPkts	否	N/A

表 C-4 RFC 2932 IPv4 多播路由 MIB (续)

对象	支持	访问
ipMRouteInterfaceTable		
索引: ipMRouteInterfaceIfIndex		
ipMRouteInterfaceTtl	是	RW
ipMRouteInterfaceProtocol	是	RO
ipMRouteInterfaceRateLimit	否	N/A
ipMRouteInterfaceInMcastOctets	否	N/A
ipMRouteInterfaceOutMcastOctets	否	N/A
ipMRouteInterfaceHCInMcastOctets	否	N/A
ipMRouteInterfaceHCOutMcastOctets	否	N/A
ipMRouteBoundaryTable		
索引: ipMRouteBoundaryIfIndex、ipMRouteBoundaryAddress、ipMRouteBoundaryAddressMask		
ipMRouteBoundaryStatus	是	RC
ipMRouteScopeNameTable		
索引: ipMRouteScopeNameAddress、ipMRouteScopeNameAddressMask、ipMRouteScopeNameLanguage		
ipMRouteScopeNameString	否	N/A
ipMRouteScopeNameDefault	否	N/A
ipMRouteScopeNameStatus	否	N/A

表 C-5 RFC 2819 RMON MIB

对象	支持	访问
etherStatsTable		
索引: etherStatsIndex		
etherStatsDataSource	是	RC
etherStatsDropEvents	是	RO
etherStatsOctets	是	RO
etherStatsPkts	是	RO

表 C-5 RFC 2819 RMON MIB (续)

对象	支持	访问
etherStatsBroadcastPkts	是	RO
etherStatsMulticastPkts	是	RO
etherStatsCRCAlignErrors	是	RO
etherStatsUndersizePkts	是	RO
etherStatsOversizePkts	是	RO
etherStatsFragments	是	RO
etherStatsJabbers	是	RO
etherStatsCollisions	是	RO
etherStatsPkts64Octets	是	RO
etherStatsPkts65to127Octets	是	RO
etherStatsPkts128to255Octets	是	RO
etherStatsPkts256to511Octets	是	RO
etherStatsPkts512to1023Octets	是	RO
etherStatsPkts1024to1518Octets	是	RO
etherStatsOwner	是	RC
etherStatsStatus	是	RC
historyControlTable		
索引: historyControlIndex		
historyControlDataSource	是	RC
historyControlBucketsRequested	是	RC
historyControlBucketsGranted	是	RO
historyControlInterval	是	RC
historyControlOwner	是	RC
historyControlStatus	是	RC
etherHistoryTable		
索引: etherHistoryIndex、 etherHistorySampleIndex		
etherHistoryIntervalStart	是	RO
etherHistoryDropEvents	是	RO

表 C-5 RFC 2819 RMON MIB (续)

对象	支持	访问
etherHistoryOctets	是	RO
etherHistoryPkts	是	RO
etherHistoryBroadcastPkts	是	RO
etherHistoryMulticastPkts	是	RO
etherHistoryCRCAlignErrors	是	RO
etherHistoryUndersizePkts	是	RO
etherHistoryOversizePkts	是	RO
etherHistoryFragments	是	RO
etherHistoryJabbers	是	RO
etherHistoryCollisions	是	RO
etherHistoryUtilization	是	RO
alarmTable		
索引: alarmIndex		
alarmInterval	是	RC
alarmVariable	是	RC
alarmSampleType	是	RC
alarmValue	是	RO
alarmStartupAlarm	是	RC
alarmRisingThreshold	是	RC
alarmFallingThreshold	是	RC
alarmRisingEventIndex	是	RC
alarmFallingEventIndex	是	RC
alarmOwner	是	RC
alarmStatus	是	RC
hostControlTable		
索引: hostControlIndex		
hostControlDataSource	否	N/A
hostControlTableSize	否	N/A

表 C-5 RFC 2819 RMON MIB (续)

对象	支持	访问
hostControlLastDeleteTime	否	N/A
hostControlOwner	否	N/A
hostControlStatus	否	N/A
hostTable		
索引: hostIndex、hostAddress		
hostCreationOrder	否	N/A
hostInPkts	否	N/A
hostOutPkts	否	N/A
hostInOctets	否	N/A
hostOutOctets	否	N/A
hostOutErrors	否	N/A
hostOutBroadcastPkts	否	N/A
hostOutMulticastPkts	否	N/A
hostTimeTable		
索引: hostTimeIndex、hostTimeCreationOrder		
hostTimeAddress	否	N/A
hostTimeInPkts	否	N/A
hostTimeOutPkts	否	N/A
hostTimeInOctets	否	N/A
hostTimeOutOctets	否	N/A
hostTimeOutErrors	否	N/A
hostTimeOutBroadcastPkts	否	N/A
hostTimeOutMulticastPkts	否	N/A
hostTopNControlTable		
索引: hostTopNControlIndex		
hostTopNHostIndex	否	N/A
hostTopNRateBase	否	N/A

表 C-5 RFC 2819 RMON MIB (续)

对象	支持	访问
hostTopNTimeRemaining	否	N/A
hostTopNDuration	否	N/A
hostTopNRequestedSize	否	N/A
hostTopNGrantedSize	否	N/A
hostTopNStartTime	否	N/A
hostTopNOwner	否	N/A
hostTopNStatus	否	N/A
hostTopNTable		
索引: hostTopNReport、hostTopNIndex		
hostTopNAddress	否	N/A
hostTopNRate	否	N/A
matrixControlTable		
索引: matrixControlIndex		
matrixControlDataSource	否	N/A
matrixControlTableSize	否	N/A
matrixControlLastDeleteTime	否	N/A
matrixControlOwner	否	N/A
matrixControlStatus	否	N/A
matrixSDTable		
索引: matrixSDIndex、matrixSDSourceAddress、 matrixSDDestAddress		
matrixSDPkts	否	N/A
matrixSDOctets	否	N/A
matrixSDErrors	否	N/A
matrixDSTable		
索引: matrixDSIndex、matrixDSDestAddress、 matrixDSSourceAddress		

表 C-5 RFC 2819 RMON MIB (续)

对象	支持	访问
matrixDSPkts	否	N/A
matrixDSOctets	否	N/A
matrixDSErrors	否	N/A
filterTable		
索引: filterIndex		
filterChannelIndex	否	N/A
filterPktDataOffset	否	N/A
filterPktData	否	N/A
filterPktDataMask	否	N/A
filterPktDataNotMask	否	N/A
filterPktStatus	否	N/A
filterPktStatusMask	否	N/A
filterPktStatusNotMask	否	N/A
filterOwner	否	N/A
filterStatus	否	N/A
channelTable		
索引: channelIndex		
channelIfIndex	否	N/A
channelAcceptType	否	N/A
channelDataControl	否	N/A
channelTurnOnEventIndex	否	N/A
channelTurnOffEventIndex	否	N/A
channelEventIndex	否	N/A
channelEventStatus	否	N/A
channelMatches	否	N/A
channelDescription	否	N/A
channelOwner	否	N/A
channelStatus	否	N/A

表 C-5 RFC 2819 RMON MIB (续)

对象	支持	访问
bufferControlTable		
索引: bufferControlIndex		
bufferControlChannelIndex	否	N/A
bufferControlFullStatus	否	N/A
bufferControlFullAction	否	N/A
bufferControlCaptureSliceSize	否	N/A
bufferControlDownloadSliceSize	否	N/A
bufferControlDownloadOffset	否	N/A
bufferControlMaxOctetsRequested	否	N/A
bufferControlMaxOctetsGranted	否	N/A
bufferControlCapturedPackets	否	N/A
bufferControlTurnOnTime	否	N/A
bufferControlOwner	否	N/A
bufferControlStatus	否	N/A
captureBufferTable		
索引: captureBufferControlIndex、captureBufferIndex		
captureBufferPacketID	否	N/A
captureBufferPacketData	否	N/A
captureBufferPacketLength	否	N/A
captureBufferPacketTime	否	N/A
captureBufferPacketStatus	否	N/A
eventTable		
索引: eventIndex		
eventDescription	是	RC
eventType	是	RC
eventCommunity	是	RC
eventLastTimeSent	是	RO

表 C-5 RFC 2819 RMON MIB (续)

对象	支持	访问
eventOwner	是	RC
eventStatus	是	RC
logTable		
索引: logEventIndex、logIndex		
logTime	是	RO
logDescription	是	RO

表 C-6 RFC 2787 VRRP MIB

对象	支持	访问
vrrpOperationsGroup		
vrrpNodeVersion	是	RO
vrrpNotificationCntl	是	RW
vrrpOperTable		
索引: ifIndex、vrrpOperVrId		
vrrpOperVirtualMacAddr	是	RO
vrrpOperState	是	RO
vrrpOperAdminState	是	RC
vrrpOperPriority	是	RC
vrrpOperIpAddrCount	是	RO
vrrpOperMasterIpAddr	是	RO
vrrpOperPrimaryIpAddr	是	RC
vrrpOperAuthType	是	RC
vrrpOperAuthKey	是	RC
vrrpOperAdvertisementInterval	是	RC
vrrpOperPreemptMode	是	RC
vrrpOperVirtualRouterUpTime	是	RO
vrrpOperProtocol	是	RC

表 C-6 RFC 2787 VRRP MIB (续)

对象	支持	访问
vrrpOperRowStatus	是	RC
vrrpAssoIpTable		
索引: vrrpAssoIpAddr		
vrrpAssoIpAddrRowStatus	否	RC
vrrpStatisticsGroup		
vrrpRouterChecksumErrors	是	RO
vrrpRouterVersionErrors	是	RO
vrrpRouterVrIdErrors	是	RO
vrrpRouterStatsTable		
扩充: vrrpOperTable		
vrrpStatsBecomeMaster	是	RO
vrrpStatsAdvertiseRcvd	是	RO
vrrpStatsAdvertiseIntervalErrors	是	RO
vrrpStatsAuthFailures	是	RO
vrrpStatsIpTtlErrors	是	RO
vrrpStatsPriorityZeroPktsRcvd	是	RO
vrrpStatsPriorityZeroPktsSent	是	RO
vrrpStatsInvalidTypePktsRcvd	是	RO
vrrpStatsAddressListErrors	是	RO
vrrpStatsInvalidAuthType	是	RO
vrrpStatsAuthTypeMismatch	是	RO
vrrpStatsPacketLengthErrors	是	RO

表 C-7 RFC 2737 实体 MIB (版本 2)

对象	支持	访问
entPhysicalTable		
索引: entPhysicalIndex		
entPhysicalDescr	是	RO
entPhysicalVendorType	是	RO
entPhysicalContainedIn	是	RO
entPhysicalClass	是	RO
entPhysicalParentRelPos	是	RO
entPhysicalName	是	RO
entPhysicalHardwareRev	是	RO
entPhysicalFirmwareRev	是	RO
entPhysicalSoftwareRev	是	RO
entPhysicalSerialNum	是	RO
entPhysicalMfgName	是	RO
entPhysicalModelName	是	RO
entPhysicalAlias	是	RO
entPhysicalAssetID	是	RO
entPhysicalIsFRU	是	RO
entLogicalTable		
索引: entLogicalIndex		
entLogicalDescr	否	N/A
entLogicalType	否	N/A
entLogicalCommunity	否	N/A
entLogicalTAddress	否	N/A
entLogicalTDomain	否	N/A
entLogicalContextEngineID	否	N/A
entLogicalContextName	否	N/A

表 C-7 RFC 2737 实体 MIB (版本 2) (续)

对象	支持	访问
entLPMappingTable		
索引: entLogicalIndex、entLPPhysicalIndex		
entLPPhysicalIndex	否	N/A
entAliasMappingTable		
索引: entPhysicalIndex、entAliasLogicalIndexOrZero		
entAliasMappingIdentifier	否	N/A
entPhysicalContainsTable		
entPhysicalChildIndex	是	RO
entityGeneral		
entLastChangeTime	是	RO
陷阱		
entConfigChange	是	

表 C-8 RFC 2674 VLAN MIB (P 网桥 MIB、Q 网桥 MIB)

对象	支持	访问
dot1dTpHCPortTable		
索引: dot1dTpPort		
dot1dTpHCPortInFrames	是	RO
dot1dTpHCPortOutFrames	是	RO
dot1dTpHCPortInDiscards	是	RO
dot1dTpPortOverflowTable		
索引: dot1dTpPort		
dot1dTpPortInOverflowFrames	是	RO
dot1dTpPortOutOverflowFrames	是	RO
dot1dTpPortInOverflowDiscards	是	RO

表 C-8 RFC 2674 VLAN MIB (P 网桥 MIB、Q 网桥 MIB) (续)

对象	支持	访问
dot1dExtBaseGroup		
dot1dDeviceCapabilities	是	RO
dot1dTrafficClassesEnabled	是	RW
dot1dGmrpStatus	是	RO
dot1dPortCapabilitiesTable		
扩充: dot1dBasePort 表		
dot1dPortCapabilities	是	RO
dot1dPortPriorityTable		
扩充: dot1dBasePort 表		
dot1dPortDefaultUserPriority	是	RW
dot1dPortNumTrafficClasses	是	RO
dot1dUserPriorityRegenTable		
索引: dot1dBasePort、dot1dUserPriority		
dot1dRegenUserPriority	否	N/A
dot1dTrafficClassTable		
索引: dot1dBasePort、dot1dTrafficClassPriority		
dot1dTrafficClass	是	RW
dot1dPortOutboundAccessPriorityTable		
索引: dot1dBasePort		
dot1dPortOutboundAccessPriority	否	N/A
dot1dPortGarpTable		
扩充: dot1dBasePort 表		
dot1dPortGarpJoinTime	是	RW
dot1dPortGarpLeaveTime	是	RW
dot1dPortGarpLeaveAllTime	是	RW

表 C-8 RFC 2674 VLAN MIB (P 网桥 MIB、Q 网桥 MIB) (续)

对象	支持	访问
dot1dPortGmrpTable		
扩充: dot1dBasePort 表		
dot1dPortGmrpStatus	是	RW
dot1dPortGmrpFailedRegistrations	是	RO
dot1dPortGmrpLastPduOrigin	是	RO
dot1qGroup		
dot1qVlanVersionNumber	是	RO
dot1qMaxVlanId	是	RO
dot1qMaxSupportedVlans	是	RO
dot1qNumVlans	是	RO
dot1qGvrpStatus	是	RW
dot1qFdbTable		
索引: dot1qFdbId		
dot1qFdbDynamicCount	是	RO
dot1qTpFdbTable		
索引: dot1qFdbId、dot1qTpFdbAddress		
dot1qTpFdbPort	是	RO
dot1qTpFdbStatus	是	RO
dot1qTpGroupTable		
索引: dot1qVlanIndex、dot1qTpGroupAddress		
dot1qTpGroupEgressPorts	否	N/A
dot1qTpGroupLearnt	否	N/A
dot1qForwardAllTable		
索引: dot1qVlanIndex		
dot1qForwardAllPorts	否	N/A

表 C-8 RFC 2674 VLAN MIB (P 网桥 MIB、Q 网桥 MIB) (续)

对象	支持	访问
dot1qForwardAllStaticPorts	否	N/A
dot1qForwardAllForbiddenPorts	否	N/A
dot1qForwardUnregisteredTable		
索引: dot1qVlanIndex		
dot1qForwardUnregisteredPorts	否	N/A
dot1qForwardUnregisteredStaticPorts	否	N/A
dot1qForwardUnregisteredForbiddenPorts	否	N/A
dot1qStaticUnicastTable		
索引: dot1qFdbId、dot1qStaticUnicastAddress、 dot1qStaticUnicastReceivePort		
dot1qStaticUnicastAllowedToGoTo	否	N/A
dot1qStaticUnicastStatus	否	N/A
dot1qStaticMulticastTable		
索引: dot1qVlanIndex、dot1qStaticMulticastAddress、 dot1qStaticMulticastReceivePort		
dot1qStaticMulticastStaticEgressPorts	否	N/A
dot1qStaticMulticastForbiddenEgressPorts	否	N/A
dot1qStaticMulticastStatus	否	N/A
dot1qVlanGroup		
dot1qVlanNumDeletes	是	RO
dot1qNextFreeLocalVlanIndex	是	RO
dot1qConstraintSetDefault	否	N/A
dot1qConstraintTypeDefault	否	N/A
dot1qVlanCurrentTable		
索引: dot1qVlanTimeMark、dot1qVlanIndex		
dot1qVlanFdbId	是	RO

表 C-8 RFC 2674 VLAN MIB (P 网桥 MIB、Q 网桥 MIB) (续)

对象	支持	访问
dot1qVlanCurrentEgressPorts	是	RO
dot1qVlanCurrentUntaggedPorts	是	RO
dot1qVlanStatus	是	RO
dot1qVlanCreationTime	是	RO
dot1qVlanStaticTable		
索引: dot1qVlanIndex		
dot1qVlanStaticName	是	RC
dot1qVlanStaticEgressPorts	是	RC
dot1qVlanForbiddenEgressPorts	是	RC
dot1qVlanStaticUntaggedPorts	是	RC
dot1qVlanStaticRowStatus	是	RC
dot1qPortVlanTable		
扩充: dot1dBasePortEntry		
dot1qPvid	是	RW
dot1qPortAcceptableFrameTypes	是	RW
dot1qPortIngressFiltering	是	RW
dot1qPortGvrpStatus	是	RW
dot1qPortGvrpFailedRegistrations	是	RO
dot1qPortGvrpLastPduOrigin	是	RO
dot1qPortVlanStatisticsTable		
索引: dot1dBasePort、dot1qVlanIndex		
dot1qTpVlanPortInFrames	否	N/A
dot1qTpVlanPortOutFrames	否	N/A
dot1qTpVlanPortInDiscards	否	N/A
dot1qTpVlanPortInOverflowFrames	否	N/A
dot1qTpVlanPortOutOverflowFrames	否	N/A
dot1qTpVlanPortInOverflowDiscards	否	N/A

表 C-8 RFC 2674 VLAN MIB (P 网桥 MIB、Q 网桥 MIB) (续)

对象	支持	访问
dot1qPortVlanHCStatisticsTable		
索引: dot1dBasePort、dot1qVlanIndex		
dot1qTpVlanPortHCInFrames	否	N/A
dot1qTpVlanPortHCOutFrames	否	N/A
dot1qTpVlanPortHCInDiscards	否	N/A
dot1qLearningConstraintsTable		
索引: dot1qConstraintVlan、dot1qConstraintSet		
dot1qConstraintType	否	N/A
dot1qConstraintStatus	否	N/A

表 C-9 RFC 2620 Radius 记帐客户机 MIB

对象	支持	访问
radiusAccClient 组		
radiusAccClientInvalidServerAddresses	是	RO
radiusAccClientIdentifier	是	RO
radiusAccServerTable		
索引: radiusAccServerIndex		
radiusAccServerAddress	是	RO
radiusAccClientServerPortNumber	是	RO
radiusAccClientRoundTripTime	是	RO
radiusAccClientRequests	是	RO
radiusAccClientRetransmissions	是	RO
radiusAccClientResponses	是	RO
radiusAccClientMalformedResponses	是	RO
radiusAccClientBadAuthenticators	是	RO
radiusAccClientPendingRequests	是	RO

表 C-9 RFC 2620 Radius 记帐客户机 MIB (续)

对象	支持	访问
radiusAccClientTimeouts	是	RO
radiusAccClientUnknownTypes	是	RO
radiusAccClientPacketsDropped	是	RO

表 C-10 RFC 2618 Radius 验证客户机 MIB

对象	支持	访问
radiusAuthClient 组		
radiusAuthClientInvalidServerAddresses	是	RO
radiusAuthClientIdentifier	是	RO
radiusAuthServerTable		
索引: radiusAuthServerIndex		
radiusAuthServerAddress	是	RO
radiusAuthClientServerPortNumber	是	RO
radiusAuthClientRoundTripTime	是	RO
radiusAuthClientAccessRequests	是	RO
radiusAuthClientAccessRetransmissions	是	RO
radiusAuthClientAccessAccepts	是	RO
radiusAuthClientAccessRejects	是	RO
radiusAuthClientAccessChallenges	是	RO
radiusAuthClientMalformedAccessResponses	是	RO
radiusAuthClientBadAuthenticators	是	RO
radiusAuthClientPendingRequests	是	RO
radiusAuthClientTimeouts	是	RO
radiusAuthClientUnknownTypes	是	RO
radiusAuthClientPacketsDropped	是	RO

表 C-11 RFC 2233 接口 MIB

对象	支持	访问
接口		
ifNumber	否	N/A
ifMIBObjects		
ifTableLastChange	否	N/A
ifStackLastChange	否	N/A
ifTable		
索引: ifIndex		
ifDescr	是	RO
ifType	是	RO
ifMtu	是	RO
ifSpeed	是	RO
ifPhysAddress	是	RO
ifAdminStatus	是	RW
ifOperStatus	是	RO
ifLastChange	是	RO
ifInOctets	是	RO
ifInUcastPkts	是	RO
ifInNUcastPkts	是	RO
ifInDiscards	是	RO
ifInErrors	是	RO
ifInUnknownProtos	是	RO
ifOutOctets	是	RO
ifOutUcastPkts	是	RO
ifOutNUcastPkts	是	RO
ifOutDiscards	是	RO
ifOutErrors	是	RO
ifOutQLen	否	N/A
ifSpecific	否	N/A

表 C-11 RFC 2233 接口 MIB (续)

对象	支持	访问
ifXTable		
索引: ifIndex		
ifName	是	RO
ifInMulticastPkts	是	RO
ifInBroadcastPkts	是	RO
ifOutMulticastPkts	是	RO
ifOutBroadcastPkts	是	RO
ifHCInOctets	是	RO
ifHCInUcastPkts	是	RO
ifHCInMulticastPkts	是	RO
ifHCInBroadcastPkts	是	RO
ifHCOctets	是	RO
ifHCOUcastPkts	是	RO
ifHCOMulticastPkts	是	RO
ifHCOBroadcastPkts	是	RO
ifLinkUpDownTrapEnable	是	RW
ifHighSpeed	是	RO
ifPromiscuousMode	是	RW
ifConnectorPresent	是	RO
ifAlias	否	N/A
ifCounterDiscontinuityTime	是	RO
ifStackTable		
索引: ifStackHigherLayer、ifStackLowerLayer		
ifStackStatus	否	N/A
ifRcvAddressTable		
索引: ifIndex、ifRcvAddressAddress		
ifRcvAddressStatus	否	N/A
ifRcvAddressType	否	N/A

表 C-11 RFC 2233 接口 MIB (续)

对象	支持	访问
ifTestTable		
索引: ifTestId		
ifTestStatus	否	N/A
ifTestType	否	N/A
ifTestResult	否	N/A
ifTestCode	否	N/A
ifTestOwner	否	N/A

表 C-12 RFC 1850 OSPF MIB

对象	支持	访问
ospfGeneralGroup		
ospfRouterId	是	RW
ospfAdminStat	是	RW
ospfVersionNumber	是	RO
ospfAreaBdrRtrStatus	是	RO
ospfASBdrRtrStatus	是	RW
ospfExternLsaCount	是	RO
ospfExternLsaCksumSum	是	RO
ospfTOSupport	是	RW
ospfOriginateNewLsas	是	RO
ospfRxNewLsas	是	RO
ospfExtLsdbLimit	是	RW
ospfMulticastExtensions	是	RO
ospfExitOverflowInterval	是	RW
ospfDemandExtensions	是	RO
ospfAreaTable		
索引: ospfAreaId		
ospfAuthType	否	N/A

表 C-12 RFC 1850 OSPF MIB (续)

对象	支持	访问
ospfImportAsExtern	是	RC
ospfSpfRuns	是	RO
ospfAreaBdrRtrCount	是	RO
ospfAsBdrRtrCount	是	RO
ospfAreaLsaCount	是	RO
ospfAreaLsaCksumSum	是	RO
ospfAreaSummary	是	RC
ospfAreaStatus	是	RO
ospfStubAreaTable		
索引: ospfStubAreaId、ospfStubTOS		
ospfStubMetric	是	RC
ospfStubStatus	是	RC
ospfStubMetricType	是	RC
ospfLsdbTable		
索引: ospfLsdbAreaId、ospfLsdbType、ospfLsdbLsid、ospfLsdbRouterId		
ospfLsdbSequence	是	RO
ospfLsdbAge	是	RO
ospfLsdbChecksum	是	RO
ospfLsdbAdvertisement	是	RO
ospfAreaRangeTable		
索引: ospfAreaRangeAreaId、ospfAreaRangeNet		
ospfAreaRangeMask	不再使用	
ospfAreaRangeStatus	不再使用	
ospfAreaRangeEffect	不再使用	

表 C-12 RFC 1850 OSPF MIB (续)

对象	支持	访问
ospfHostTable		
索引: ospfHostIpAddress、ospfHostTOS		
ospfHostMetric	否	N/A
ospfHostStatus	否	N/A
ospfHostAreaID	否	N/A
ospfIfTable		
索引: ospfIfIpAddress、ospfAddressLessIf		
ospfIfAreaId	是	RC
ospfIfType	是	RO
ospfIfAdminStat	是	RO
ospfIfRtrPriority	是	RC
ospfIfTransitDelay	是	RC
ospfIfRetransInterval	是	RC
ospfIfHelloInterval	是	RC
ospfIfRtrDeadInterval	是	RC
ospfIfPollInterval	否	N/A
ospfIfState	是	RO
ospfIfDesignatedRouter	是	RO
ospfIfBackupDesignatedRouter	是	RO
ospfIfEvents	是	RO
ospfIfAuthKey	是	RC
ospfIfStatus	是	RC
ospfIfMulticastForwarding	是	RO
ospfIfDemand	是	RO
ospfIfAuthType	是	RW

表 C-12 RFC 1850 OSPF MIB (续)

对象	支持	访问
ospfIfMetricTable		
索引: ospfIfMetricIpAddress、ospfIfMetricAddressLessIf、ospfIfMetricTOS		
ospfIfMetricValue	是	RW
ospfIfMetricStatus	是	RO
ospfVirtIfTable		
索引: ospfVirtIfAreaId、ospfVirtIfNeighbor		
ospfVirtIfTransitDelay	是	RW
ospfVirtIfRetransInterval	是	RW
ospfVirtIfHelloInterval	是	RW
ospfVirtIfRtrDeadInterval	是	RW
ospfVirtIfState	是	RO
ospfVirtIfEvents	是	RO
ospfVirtIfAuthKey	是	RO
ospfVirtIfStatus	是	RC
ospfVirtIfAuthType	是	RW
ospfNbrTable		
索引: ospfNbrIpAddr、ospfNbrAddressLessIndex		
ospfNbrRtrId	是	RO
ospfNbrOptions	是	RO
ospfNbrPriority	是	RO
ospfNbrState	是	RO
ospfNbrEvents	是	RO
ospfNbrLsRetransQLen	是	RO
ospfNbmaNbrStatus	是	RO
ospfNbmaNbrPermanence	是	RO
ospfNbrHelloSuppressed	是	RO

表 C-12 RFC 1850 OSPF MIB (续)

对象	支持	访问
ospfVirtNbrTable		
索引: ospfVirtNbrArea、ospfVirtNbrRtrId		
ospfVirtNbrIpAddr	是	RO
ospfVirtNbrOptions	是	RO
ospfVirtNbrState	是	RO
ospfVirtNbrEvents	是	RO
ospfVirtNbrLsRetransQLen	是	RO
ospfVirtNbrHelloSuppressed	是	RO
ospfExtLsdbTable		
索引: ospfExtLsdbType、ospfExtLsdbLsid、ospfExtLsdbRouterId		
ospfExtLsdbSequence	是	RO
ospfExtLsdbAge	是	RO
ospfExtLsdbChecksum	是	RO
ospfExtLsdbAdvertisement	是	RO
ospfAreaAggregateTable		
索引: ospfAreaAggregateAreaID、ospfAreaAggregateLsdbType、ospfAreaAggregateNet、ospfAreaAggregateMask		
ospfAreaAggregateStatus	是	RO
ospfAreaAggregateEffect	是	RW

表 C-13 RFC 1724 RIPv2 MIB

对象	支持	访问
rip2GlobalGroup		
rip2GlobalRouteChanges	是	RO
rip2GlobalQueries	是	RO
rip2IfStatTable		

表 C-13 RFC 1724 RIPv2 MIB (续)

对象	支持	访问
索引: rip2IfStatAddress		
rip2IfStatRcvBadPackets	是	RO
rip2IfStatRcvBadRoutes	是	RO
rip2IfStatSentUpdates	是	RO
rip2IfStatStatus	是	RC
rip2IfConfTable		
索引: rip2IfConfAddress		
rip2IfConfDomain	否	
rip2IfConfAuthType	是	RC
rip2IfConfAuthKey	是	RC
rip2IfConfSend	是	RC
rip2IfConfReceive	是	RC
rip2IfConfDefaultMetric	否	N/A
rip2IfConfStatus	是	RC
rip2IfConfSrcAddress	是	RO
rip2PeerTable		
索引: rip2PeerAddress、rip2PeerDomain		
rip2PeerLastUpdate	否	RO
rip2PeerVersion	否	RO
rip2PeerRcvBadPackets	否	RO
rip2PeerRcvBadRoutes	否	RO

表 C-14 RFC 1657 BGP4 MIB

对象	支持	访问
bgp		
bgpVersion	是	RO
bgpLocalAs	是	RO

表 C-14 RFC 1657 BGP4 MIB (续)

对象	支持	访问
bgpIdentifier	是	RO
bgpPeerTable		
索引: bgpPeerRemoteAddr		
bgpPeerIdentifier	是	RO
bgpPeerState	是	RO
bgpPeerAdminStatus	是	RW
bgpPeerNegotiatedVersion	是	RO
bgpPeerLocalAddr	是	RO
bgpPeerLocalPort	是	RO
bgpPeerRemotePort	是	RO
bgpPeerRemoteAs	是	RO
bgpPeerInUpdates	是	RO
bgpPeerOutUpdates	是	RO
bgpPeerInTotalMessages	是	RO
bgpPeerOutTotalMessages	是	RO
bgpPeerLastError	是	RO
bgpPeerFsmEstablishedTransitions	是	RO
bgpPeerFsmEstablishedTime	是	RO
bgpPeerConnectRetryInterval	是	RW
bgpPeerHoldTime	是	RO
bgpPeerKeepAlive	是	RO
bgpPeerHoldTimeConfigured	是	RW
bgpPeerKeepAliveConfigured	是	RW
bgpPeerMinASOriginationInterval	否	RW
bgpPeerMinRouteAdvertisementInterval	否	RW
bgpPeerInUpdateElapsedTime	是	RO
bgpRcvdPathAttrTable		
索引: bgpPathAttrDestNetwork、bgpPathAttrPeer		
bgpPathAttrOrigin	不再使用	
bgpPathAttrASPath	不再使用	

表 C-14 RFC 1657 BGP4 MIB (续)

对象	支持	访问
bgpPathAttrNextHop	不再使用	
bgpPathAttrInterASMetric	不再使用	
bgp4PathAttrTable		
索引: bgp4PathAttrIpAddressPrefix、 bgp4PathAttrIpAddressPrefixLen、bgp4PathAttrPeer		
bgp4PathAttrOrigin	是	RO
bgp4PathAttrASPathSegment	是	RO
bgp4PathAttrNextHop	是	RO
bgp4PathAttrMultiExitDisc	是	RO
bgp4PathAttrLocalPref	是	RO
bgp4PathAttrAtomicAggregate	是	RO
bgp4PathAttrAggregatorAS	是	RO
bgp4PathAttrAggregatorAddr	是	RO
bgp4PathAttrCalcLocalPref	是	RO
bgp4PathAttrBest	是	RO
bgp4PathAttrUnknown	是	RO

表 C-15 RFC 1643 以太网 MIB

对象	支持	访问
dot3StatsTable		
索引: dot3StatsIndex		
dot3StatsAlignmentErrors	是	RO
dot3StatsFCSErrors	是	RO
dot3StatsSingleCollisionFrames	是	RO
dot3StatsMultipleCollisionFrames	是	RO
dot3StatsSQETestErrors	是	RO
dot3StatsDeferredTransmissions	是	RO
dot3StatsLateCollisions	是	RO

表 C-15 RFC 1643 以太网 MIB (续)

对象	支持	访问
dot3StatsExcessiveCollisions	是	RO
dot3StatsInternalMacTransmitErrors	是	RO
dot3StatsCarrierSenseErrors	是	RO
dot3StatsFrameTooLongs	是	RO
dot3StatsInternalMacReceiveErrors	是	RO
dot3StatsEtherChipSet	否	N/A
dot3CollTable		
索引: ifIndex、dot3CollCount		
dot3CollFrequencies	否	

表 C-16 RFC 1493 网桥 MIB

对象	支持	访问
dot1dBase		
dot1dBaseBridgeAddress	是	RO
dot1dBaseNumPorts	是	RO
dot1dBaseType	是	RO
dot1dBasePortTable		
索引: dot1dBasePort		
dot1dBasePortIfIndex	是	RO
dot1dBasePortCircuit	是	RO
dot1dBasePortDelayExceededDiscards	否	N/A
dot1dBasePortMtuExceededDiscards	否	N/A
dot1dStp		
dot1dStpProtocolSpecification	是	RO
dot1dStpPriority	是	RW

表 C-16 RFC 1493 网桥 MIB (续)

对象	支持	访问
dot1dStpTimeSinceTopologyChange	是	RO
dot1dStpTopChanges	是	RO
dot1dStpDesignatedRoot	是	RO
dot1dStpRootCost	是	RO
dot1dStpRootPort	是	RO
dot1dStpMaxAge	是	RO
dot1dStpHelloTime	是	RO
dot1dStpHoldTime	是	RO
dot1dStpForwardDelay	是	RO
dot1dStpBridgeMaxAge	是	RW
dot1dStpBridgeHelloTime	是	RW
dot1dStpBridgeForwardDelay	是	RW
dot1dStpPortTable		
索引: dot1dStpPort		
dot1dStpPortPriority	是	RW
dot1dStpPortState	是	RO
dot1dStpPortEnable	是	RW
dot1dStpPortPathCost	是	RW
dot1dStpPortDesignatedRoot	是	RO
dot1dStpPortDesignatedCost	是	RO
dot1dStpPortDesignatedBridge	是	RO
dot1dStpPortDesignatedPort	是	RO
dot1dStpPortForwardTransitions	是	RO
dot1dTp		
dot1dTpLearnedEntryDiscards	否	N/A
dot1dTpAgingTime	是	RW

表 C-16 RFC 1493 网桥 MIB (续)

对象	支持	访问
dot1dTpFdbTable		
索引: dot1dTpFdbAddress		
dot1dTpFdbPort	是	RO
dot1dTpFdbStatus	是	RO
dot1dTpPortTable		
索引: dot1dTpPort		
dot1dTpPortMaxInfo	是	RO
dot1dTpPortInFrames	是	RO
dot1dTpPortOutFrames	是	RO
dot1dTpPortInDiscards	是	RO
dot1dStaticTable		
索引: dot1dStaticAddress、dot1dStaticReceivePort		
dot1dStaticAllowedToGoTo	否	N/A
dot1dStaticStatus	否	N/A

表 C-17 RFC 1213 Mib-2 MIB

对象	支持	访问
系统		
sysDescr	是	RO
sysObjectID	是	RO
sysUpTime	是	RO
sysContact	是	RW
sysName	是	RW
sysLocation	是	RW
sysServices	是	RO

表 C-17 RFC 1213 Mib-2 MIB (续)

对象	支持	访问
接口		
ifNumber	是	RO
ifTable		
索引: ifIndex		
ifDescr	是	RO
ifType	是	RO
ifMtu	是	RO
ifSpeed	是	RO
ifPhysAddress	是	RO
ifAdminStatus	是	RW
ifOperStatus	是	RO
ifLastChange	是	RO
ifInOctets	是	RO
ifInUcastPkts	是	RO
ifInNUcastPkts	是	RO
ifInDiscards	是	RO
ifInErrors	是	RO
ifInUnknownProtos	是	RO
ifOutOctets	是	RO
ifOutUcastPkts	是	RO
ifOutNUcastPkts	是	RO
ifOutDiscards	是	RO
ifOutErrors	是	RO
ifOutQLen	否	N/A
ifSpecific	否	N/A
atTable		
索引: atIfIndex、atNetAddress		
atPhysAddress	使用价值已降低	

表 C-17 RFC 1213 Mib-2 MIB (续)

对象	支持	访问
ip		
ipForwarding	是	RW
ipDefaultTTL	是	RO
ipInReceives	是	RO
ipInHdrErrors	是	RO
ipInAddrErrors	是	RO
ipForwDatagrams	是	RO
ipInUnknownProtos	是	RO
ipInDiscards	是	RO
ipInDelivers	是	RO
ipOutRequests	是	RO
ipOutDiscards	是	RO
ipOutNoRoutes	是	RO
ipReasmTimeout	是	RO
ipReasmReqds	是	RO
ipReasmOKs	是	RO
ipReasmFails	是	RO
ipFragOKs	是	RO
ipFragFails	是	RO
ipFragCreates	是	RO
ipRoutingDiscards	是	RO
ipAddrTable		
索引: ipAdEntAddr		
ipAdEntIfIndex	是	RO
ipAdEntNetMask	是	RO
ipAdEntBcastAddr	是	RO
ipAdEntReasmMaxSize	是	RO

表 C-17 RFC 1213 Mib-2 MIB (续)

对象	支持	访问
ipRouteTable		
索引: ipRouteDest		
ipRouteIfIndex	是	RO
ipRouteMetric1	是	RO
ipRouteMetric2	是	RO
ipRouteMetric3	是	RO
ipRouteMetric4	是	RO
ipRouteNextHop	是	RO
ipRouteType	是	RO
ipRouteProto	是	RO
ipRouteAge	否	N/A
ipRouteMask	是	RO
ipRouteMetric5	是	RO
ipRouteInfo	是	RO
ipNetToMedia 表		
索引: ipNetToMediaIfIndex、 ipNetToMediaNetAddress		
ipNetToMediaPhysAddress	是	RO
ipNetToMediaType	是	RO
icmp 组		
icmpInMsgs	是	RO
icmpInErrors	是	RO
icmpInDestUnreachs	是	RO
icmpInTimeExcds	是	RO
icmpInParmProbs	是	RO
icmpInSrcQuenchs	是	RO
icmpInRedirects	是	RO
icmpInEchos	是	RO
icmpInEchoReps	是	RO

表 C-17 RFC 1213 Mib-2 MIB (续)

对象	支持	访问
icmpInTimestamps	是	RO
icmpInTimestampReps	是	RO
icmpInAddrMasks	是	RO
icmpInAddrMaskReps	是	RO
icmpOutMsgs	是	RO
icmpOutErrors	是	RO
icmpOutDestUnreachs	是	RO
icmpOutTimeExcds	是	RO
icmpOutParmProbs	是	RO
icmpOutSrcQuenchs	是	RO
icmpOutRedirects	是	RO
icmpOutEchos	是	RO
icmpOutEchoReps	是	RO
icmpOutTimestamps	是	RO
icmpOutTimestampReps	是	RO
icmpOutAddrMasks	是	RO
icmpOutAddrMaskReps	是	RO
tcp 组		
tcpRtoAlgorithm	是	RO
tcpRtoMin	是	RO
tcpRtoMax	是	RO
tcpMaxConn	是	RO
tcpActiveOpens	是	RO
tcpPassiveOpens	是	RO
tcpAttemptFails	是	RO
tcpEstabResets	是	RO
tcpCurrEstab	是	RO
tcpInSegs	是	RO
tcpOutSegs	是	RO

表 C-17 RFC 1213 Mib-2 MIB (续)

对象	支持	访问
tcpRetransSegs	是	RO
tcpInErrs	是	RO
tcpOutRsts	是	RO
tcpConn 表		
索引: tcpConnLocalAddress、tcpConnLocalPort、tcpConnRemAddress、tcpConnRemPort		
tcpConnState	是	RO
udp 组		
udpInDatagrams	是	RO
udpNoPorts	是	RO
udpInErrors	是	RO
udpOutDatagrams	是	RO
udp 表		
索引: udpLocalAddress、udpLocalPort		
udpLocalAddress	是	RO
udpLocalPort	是	RO
egp 组		
egpInMsgs	否	N/A
egpInErrors	否	N/A
egpOutMsgs	否	N/A
egpOutErrors	否	N/A
egpAs	否	N/A
egpNeighTable		
索引: egpNeighAddr		
egpNeighState	否	N/A

表 C-17 RFC 1213 Mib-2 MIB (续)

对象	支持	访问
egpNeighAs	否	N/A
egpNeighInMsgs	否	N/A
egpNeighInErrs	否	N/A
egpNeighOutMsgs	否	N/A
egpNeighOutErrs	否	N/A
egpNeighInErrMsgs	否	N/A
egpNeighOutErrMsgs	否	N/A
egpNeighStateUps	否	N/A
egpNeighStateDowns	否	N/A
egpNeighIntervalHello	否	N/A
egpNeighIntervalPoll	否	N/A
egpNeighMode	否	N/A
egpNeighEventTrigger	否	N/A
snmp 组		
snmpInPkts	是	RO
snmpOutPkts	不再使用	
snmpInBadVersions	是	RO
snmpInBadCommunityNames	是	RO
snmpInBadCommunityUses	是	RO
snmpInASNParseErrs	是	RO
snmpInTooBig	不再使用	
snmpInNoSuchNames	不再使用	
snmpInBadValues	不再使用	
snmpInReadOnly	不再使用	
snmpInGenErrs	不再使用	
snmpInTotalReqVars	不再使用	
snmpInTotalSetVars	不再使用	
snmpInGetRequests	不再使用	
snmpInGetNexts	不再使用	

表 C-17 RFC 1213 Mib-2 MIB (续)

对象	支持	访问
snmpInSetRequests	不再使用	
snmpInGetResponses	不再使用	
snmpInTraps	不再使用	
snmpOutTooBig	不再使用	
snmpOutNoSuchNames	不再使用	
snmpOutBadValues	不再使用	
snmpOutGenErrs	不再使用	
snmpOutGetRequests	不再使用	
snmpOutGetNexts	不再使用	
snmpOutSetRequests	不再使用	
snmpOutGetResponses	不再使用	
snmpOutTraps	不再使用	
snmpEnableAuthenTraps	是	RW
snmpSilentDrops	是	RO
snmpProxyDrops	是	RO

表 C-18 POWER-ETHERNET-MIB

对象	支持	访问
pethPsePortTable		
索引: pethPsePortGroupIndex、pethPsePortIndex		
pethPsePortAdminEnable	是	RW
pethPsePortPowerPairsControlAbility	是	RO
pethPsePortPowerPairs	是	RW
pethPsePortDetectionStatus	是	RO
pethPsePortPowerPriority	是	RW
pethPsePortMPSAbsentCounter	是	RO
pethPsePortType	是	RW
pethPsePortPowerClassifications	是	RO
pethPsePortInvalidSignatureCounter	是	RO

表 C-18 POWER-ETHERNET-MIB (续)

对象	支持	访问
pethPsePortPowerDeniedCounter	是	RO
pethPsePortOverLoadCounter	是	RO
pethPsePortShortCounter	是	RO
pethMainPseTable		
索引: pethMainPseGroupIndex		
pethMainPsePower	是	RO
pethMainPseOperStatus	是	RO
pethMainPseConsumptionPower	是	RO
pethMainPseUsageThreshold	是	RW
pethNotificationControlTable		
索引: pethNotificationControlGroupIndex		
pethNotificationControlEnable	是	RW

表 C-19 LVL7-POWER-ETHERNET-MIB

对象	支持	访问
agentPethPsePortTable		
扩充: pethPsePortEntry		
agentPethPowerLimit	是	RW
agentPethOutputPower	是	RO
agentPethOutputCurrent	是	RO
agentPethOutputVolts	是	RO

表 C-20 IEEE8021-PAE-MIB dot1x MIB

对象	支持	访问
dot1xPaeSystem 组		
dot1xPaeSystemAuthControl	是	RW
dot1xPaePortTable		
索引: dot1xPaePortNumber		
dot1xPaePortProtocolVersion	是	RO
dot1xPaePortCapabilities	是	RO
dot1xPaePortInitialize	是	RW
dot1xPaePortReauthenticate	是	RW
dot1xAuthConfigTable		
索引: dot1xPaePortNumber		
dot1xAuthPaeState	是	RO
dot1xAuthBackendAuthState	是	RO
dot1xAuthAdminControlledDirections	是	RO
dot1xAuthOperControlledDirections	是	RO
dot1xAuthAuthControlledPortStatus	是	RO
dot1xAuthAuthControlledPortControl	是	RW
dot1xAuthQuietPeriod	是	RW
dot1xAuthTxPeriod	是	RW
dot1xAuthSuppTimeout	是	RW
dot1xAuthServerTimeout	是	RW
dot1xAuthMaxReq	是	RW
dot1xAuthReAuthPeriod	是	RW
dot1xAuthReAuthEnabled	是	RW
dot1xAuthKeyTxEnabled	是	RO

表 C-20 IEEE8021-PAE-MIB dot1x MIB (续)

对象	支持	访问
dot1xAuthStatsTable		
索引: dot1xPaePortNumber		
dot1xAuthEapolFramesRx	是	RO
dot1xAuthEapolFramesTx	是	RO
dot1xAuthEapolStartFramesRx	是	RO
dot1xAuthEapolLogoffFramesRx	是	RO
dot1xAuthEapolRespIdFramesRx	是	RO
dot1xAuthEapolRespFramesRx	是	RO
dot1xAuthEapolReqIdFramesTx	是	RO
dot1xAuthEapolReqFramesTx	是	RO
dot1xAuthInvalidEapolFramesRx	是	RO
dot1xAuthEapLengthErrorFramesRx	是	RO
dot1xAuthLastEapolFrameVersion	是	RO
dot1xAuthLastEapolFrameSource	是	RO
dot1xAuthDiagTable		
索引: dot1xPaePortNumber		
dot1xAuthEntersConnecting	是	RO
dot1xAuthEapLogoffsWhileConnecting	是	RO
dot1xAuthEntersAuthenticating	是	RO
dot1xAuthAuthSuccessWhileAuthenticating	是	RO
dot1xAuthAuthTimeoutsWhileAuthenticating	是	RO
dot1xAuthAuthFailWhileAuthenticating	是	RO
dot1xAuthAuthReauthsWhileAuthenticating	是	RO
dot1xAuthAuthEapStartsWhileAuthenticating	是	RO
dot1xAuthAuthEapLogoffWhileAuthenticating	是	RO
dot1xAuthAuthReauthsWhileAuthenticated	是	RO
dot1xAuthAuthEapStartsWhileAuthenticated	是	RO
dot1xAuthAuthEapLogoffWhileAuthenticated	是	RO
dot1xAuthBackendResponses	是	RO

表 C-20 IEEE8021-PAE-MIB dot1x MIB (续)

对象	支持	访问
dot1xAuthBackendAccessChallenges	是	RO
dot1xAuthBackendOtherRequestsToSupplicant	是	RO
dot1xAuthBackendNonNakResponsesFromSupplicant	是	RO
dot1xAuthBackendAuthSuccesses	是	RO
dot1xAuthBackendAuthFails	是	RO
dot1xAuthSessionStatsTable		
索引: dot1xPaePortNumber		
dot1xAuthSessionOctetsRx	否	N/A
dot1xAuthSessionOctetsTx	否	N/A
dot1xAuthSessionFramesRx	否	N/A
dot1xAuthSessionFramesTx	否	N/A
dot1xAuthSessionId	否	N/A
dot1xAuthSessionAuthenticMethod	否	N/A
dot1xAuthSessionTime	否	N/A
dot1xAuthSessionTerminateCause	否	N/A
dot1xAuthSessionUserName	否	N/A
dot1xSuppConfigTable		
索引: dot1xPaePortNumber		
dot1xSuppPaeState	否	N/A
dot1xSuppHeldPeriod	否	N/A
dot1xSuppAuthPeriod	否	N/A
dot1xSuppStartPeriod	否	N/A
dot1xSuppMaxStart	否	N/A
dot1xSuppStatsTable		
索引: dot1xPaePortNumber		
dot1xSuppEapolFramesRx	否	N/A
dot1xSuppEapolFramesTx	否	N/A

表 C-20 IEEE8021-PAE-MIB dot1x MIB (续)

对象	支持	访问
dot1xSuppEapolStartFramesTx	否	N/A
dot1xSuppEapolLogoffFramesTx	否	N/A
dot1xSuppEapolRespIdFramesTx	否	N/A
dot1xSuppEapolRespFramesTx	否	N/A
dot1xSuppEapolReqIdFramesRx	否	N/A
dot1xSuppEapolReqFramesRx	否	N/A
dot1xSuppInvalidEapolFramesRx	否	N/A
dot1xSuppEapLengthErrorFramesRx	否	N/A
dot1xSuppLastEapolFrameVersion	否	N/A
dot1xSuppLastEapolFrameSource	否	N/A

表 C-21 FASTPATH-SECURITY-MIB

对象	支持	访问
agentSSLConfigGroup		
agentSSLAdminMode	是	RW
agentSSLSecurePort	是	RW
agentSSLProtocolLevel	是	RW
agentSSHConfigGroup		
agentSSHAdminMode	是	RW
agentSSHProtocolLevel	是	RW
agentSSHSessionsCount	是	RW

表 C-22 FASTPATH-MULTICAST-MIB

对象	支持	访问
agentMulticastIGMPConfigGroup		
agentMulticastIGMPAdminMode	是	RW

表 C-22 FASTPATH-MULTICAST-MIB (续)

对象	支持	访问
agentMulticastIGMPInterfaceTable		
索引: agentMulticastIGMPInterfaceIfIndex		
agentMulticastIGMPInterfaceAdminMode	是	RW
agentMulticastPIMConfigGroup		
agentMulticastPIMConfigMode	是	RW
agentMulticastPIMSMConfigGroup		
agentMulticastPIMSMAdminMode	是	RW
agentMulticastPIMSMDataThresholdRate	是	RW
agentMulticastPIMSMRegThresholdRate	是	RW
agentMulticastPIMSMStaticRPTable		
索引: agentMulticastPIMSMStaticRPIpAddr、 agentMulticastPIMSMStaticRPGroupIpAddr、 agentMulticastPIMSMStaticRPGroupIpMask		
agentMulticastPIMSMStaticRPStatus	是	RW
agentMulticastPIMSMInterfaceTable		
索引: agentMulticastPIMSMInterfaceIndex		
agentMulticastPIMSMInterfaceCBRSRHashMaskLength	是	RW
agentMulticastPIMSMInterfaceCRPPreference	是	RW
agentMulticastPIMDMConfigGroup		
agentMulticastPIMDMAdminMode	是	RW
agentMulticastRoutingConfigGroup		
agentMulticastRoutingAdminMode	是	RW
agentMulticastDVMRPConfigGroup		
agentMulticastDVMRPAdminMode	是	RW

表 C-23 FASTPATH-MGMT-SECURITY-MIB

对象	支持	访问
agentSSLConfigGroup		
agentSSLAdminMode	是	RW
agentSSLSecurePort	是	RW
agentSSLProtocolLevel	是	RW
agentSSHConfigGroup		
agentSSHAdminMode	是	RW
agentSSHProtocolLevel	是	RW
agentSSHSessionsCount	是	RW

表 C-24 FASTPATH-DHCPSEVER-PRIVATE-MIB

对象	支持	访问
agentDhcpServerGroup		
agentDhcpServerAdminMode	是	RW
agentDhcpServerPingPktNos	是	RW
agentDhcpServerAutomaticBindingsNos	是	RO
agentDhcpServerExpiredBindingsNos	是	RO
agentDhcpServerMalformedMessagesReceived	是	RO
agentDhcpServerDISCOVERMessagesReceived	是	RO
agentDhcpServerREQUESTMessagesReceived	是	RO
agentDhcpServerDECLINEMessagesReceived	是	RO
agentDhcpServerRELEASEMessagesReceived	是	RO
agentDhcpServerINFORMMessagesReceived	是	RO
agentDhcpServerOFFERMessagesSent	是	RO
agentDhcpServerACKMessagesSent	是	RO
agentDhcpServerNAKMessagesSent	是	RO
agentDhcpServerClearStatistics	是	RW

表 C-24 FASTPATH-DHCPSEVER-PRIVATE-MIB (续)

对象	支持	访问
agentDhcpServerBootpAutomatic	是	RW
agentDhcpServerPoolConfigGroup		
agentDhcpServerPoolNameCreate	是	RW
agentDhcpServerPoolConfigTable		
索引: agentDhcpServerPoolIndex		
agentDhcpServerPoolName	是	RO
agentDhcpServerPoolDefRouter	是	RW
agentDhcpServerPoolDNSServer	是	RW
agentDhcpServerPoolLeaseTime	是	RW
agentDhcpServerPoolType	是	RO
agentDhcpServerPoolNetbiosNameServer	是	RW
agentDhcpServerPoolNetbiosNodeType	是	RW
agentDhcpServerPoolNextServer	是	RW
agentDhcpServerPoolDomainName	是	RW
agentDhcpServerPoolBootfile	是	RW
agentDhcpServerPoolRowStatus	是	RW
agentDhcpServerPoolAllocationTable		
扩充: agentDhcpServerPoolConfigEntry		
agentDhcpServerPoolAllocationName	是	RO
agentDhcpServerDynamicPoolIpAddress	是	RW
agentDhcpServerDynamicPoolIpMask	是	RW
agentDhcpServerDynamicPoolIpPrefixLength	是	RW
agentDhcpServerPoolAllocationType	是	RO
agentDhcpServerManualPoolClientIdentifier	是	RW
agentDhcpServerManualPoolClientName	是	RW
agentDhcpServerManualPoolClientHWAddr	是	RW
agentDhcpServerManualPoolClientHWType	是	RW
agentDhcpServerManualPoolIpAddress	是	RW
agentDhcpServerManualPoolIpMask	是	RW

表 C-24 FASTPATH-DHCPSEVER-PRIVATE-MIB (续)

对象	支持	访问
agentDhcpServerManualPoolIpPrefixLength	是	RW
agentDhcpServerPoolConfigGroup		
agentDhcpServerExcludedAddressRangeCreate	是	RW
agentDhcpServerExcludedAddressRangeTable		
索引: agentDhcpServerExcludedRangeIndex		
agentDhcpServerExcludedStartIpAddress	是	RO
agentDhcpServerExcludedEndIpAddress	是	RO
agentDhcpServerExcludedAddressRangeStatus	是	RW
agentDhcpServerPoolConfigGroup		
agentDhcpServerPoolOptionCreate	是	RW
agentDhcpServerPoolOptionTable		
索引: agentDhcpServerPoolOptionIndex、 agentDhcpServerPoolOptionCode		
agentDhcpServerOptionPoolName	是	RO
agentDhcpServerPoolOptionAsciiData	是	RW
agentDhcpServerPoolOptionHexData	是	RW
agentDhcpServerPoolOptionIpAddressData	是	RW
agentDhcpServerPoolOptionStatus	是	RW
agentDhcpServerLeaseGroup		
agentDhcpServerLeaseClearAllBindings	是	RW
agentDhcpServerLeaseTable		
索引: agentDhcpServerLeaseIPAddress		
agentDhcpServerLeaseIPMask	是	RO
agentDhcpServerLeaseHWAddress	是	RO
agentDhcpServerLeaseRemainingTime	是	RO
agentDhcpServerLeaseType	是	RO

表 C-24 FASTPATH-DHCPSEVER-PRIVATE-MIB (续)

对象	支持	访问
agentDhcpServerLeaseStatus	是	RW
agentDhcpServerAddressConflictGroup		
agentDhcpServerClearAllAddressConflicts	是	RW
agentDhcpServerAddressConflictLogging	是	RW
agentDhcpServerAddressConflictTable		
索引: agentDhcpServerAddressConflictIP		
agentDhcpServerAddressConflictDetectionType	是	RO
agentDhcpServerAddressConflictDetectionTime	是	RO
agentDhcpServerAddressConflictStatus	是	RW

表 C-25 FASTPATH-BGP-MIB

对象	支持	访问
agentBGPConfigGroup		
agentBGPAdminMode	是	RW
agentBGPDefaultMetric	是	RW
agentBGPDefaultMetricConfigured	是	RW
agentBGPDefaultInfoOriginate	是	RW
agentBgpPeerTable		
扩充: bgpPeerEntry		
agentBgpPeerAuthType	是	RC
agentBgpPeerAuthKey	是	RC
agentBGPRouteRedistTable		
索引: agentBGPRouteRedistSource		
agentBGPRouteRedistMode	是	RW
agentBGPRouteRedistMetric	是	RW
agentBGPRouteRedistMetricConfigured	是	RW

表 C-25 FASTPATH-BGP-MIB (续)

对象	支持	访问
agentBGPRouteRedistMatchInternal	是	RW
agentBGPRouteRedistMatchExternal1	是	RW
agentBGPRouteRedistMatchExternal2	是	RW
agentBGPRouteRedistMatchNSSAExternal1	是	RW
agentBGPRouteRedistMatchNSSAExternal2	是	RW
agentBGPRouteRedistDistList	是	RW
agentBGPRouteRedistDistListConfigured	是	RW

表 C-26 FASTPATH 交换 MIB

对象	支持	访问
agentInventoryGroup		
agentInventorySysDescription	是	RO
agentInventoryMachineType	是	RO
agentInventoryMachineModel	是	RO
agentInventorySerialNumber	是	RO
agentInventoryFRUNumber	是	RO
agentInventoryMaintenanceLevel	是	RO
agentInventoryPartNumber	是	RO
agentInventoryManufacturer	是	RO
agentInventoryBurnedInMacAddress	是	RO
agentInventoryOperatingSystem	是	RO
agentInventoryNetworkProcessingDevice	是	RO
agentInventoryAdditionalPackages	是	RO
agentInventorySoftwareVersion	是	RO
agentTrapLogGroup		
agentTrapLogTotal	是	RO
agentTrapLogTotalSinceLastViewed	否	RO

表 C-26 FASTPATH 交换 MIB (续)

对象	支持	访问
agentTrapLogTable		
索引: agentTrapLogIndex		
agentTrapLogSystemTime	是	RO
agentTrapLogTrap	是	RO
agentSupportedMibTable		
索引: agentSupportedMibIndex		
agentSupportedMibName	是	RO
agentSupportedMibDescription	是	RO
agentLoginSessionTable		
索引: agentLoginSessionIndex		
agentLoginSessionUserName	是	RO
agentLoginSessionIPAddress	是	RO
agentLoginSessionConnectionType	是	RO
agentLoginSessionIdleTime	是	RO
agentLoginSessionSessionTime	是	RO
agentLoginSessionStatus	是	RW
agentTelnetGroup		
agentTelnetLoginTimeout	是	RW
agentTelnetMaxSessions	是	RW
agentTelnetAllowNew	是	RW
agentUserConfigGroup		
agentUserConfigCreate	是	RW
agentUserConfig 表		
索引: agentUserIndex		

表 C-26 FASTPATH 交换 MIB (续)

对象	支持	访问
agentUserName	是	RW
agentUserPassword	是	RW
agentUserAccessMode	是	RO
agentUserStatus	是	RW
agentUserAuthenticationType	是	RW
agentUserEncryptionType	是	RW
agentUserEncryptionPassword	是	RW
agentSerial 组		
agentSerialBaudrate	是	RW
agentSerialTimeout	是	RW
agentSerialCharacterSize	是	RO
agentSerialHWFlowControlMode	是	RO
agentSerialStopBits	是	RO
agentSerialParityType	是	RO
agentLagConfigGroup		
agentLagConfigCreate	是	RW
agentLagConfigStaticCapability	是	RW
agentLagSummaryConfig 表		
索引: agentLagSummaryLagIndex		
agentLagSummaryName	是	RW
agentLagSummaryFlushTimer	否	N/A
agentLagSummaryLinkTrap	是	RW
agentLagSummaryAdminMode	是	RW
agentLagSummaryStpMode	是	RW
agentLagSummaryAddPort	是	RW
agentLagSummaryDeletePort	是	RW
agentLagSummaryStatus	是	RW

表 C-26 FASTPATH 交换 MIB (续)

对象	支持	访问
agentLagSummaryType	是	RO
agentLagDetailedConfig 表		
索引: agentLagDetailedLagIndex、 agentLagDetailedIfIndex		
agentLagDetailedPortSpeed	是	RO
agentLagDetailedPortStatus	是	RO
agentNetworkConfig 组		
agentNetworkIPAddress	是	RW
agentNetworkSubnetMask	是	RW
agentNetworkDefaultGateway	是	RW
agentNetworkBurnedInMacAddress	是	RO
agentNetworkLocalAdminMacAddress	是	RW
agentNetworkMacAddressType	是	RW
agentNetworkConfigProtocol	是	RW
agentNetworkWebMode	是	RW
agentNetworkJavaMode	是	RW
agentNetworkMgmtVlan	是	RW
agentServicePortConfig 组		
agentServicePortIPAddress	是	RW
agentServicePortSubnetMask	是	RW
agentServicePortDefaultGateway	是	RW
agentServicePortBurnedInMacAddress	是	RO
agentServicePortConfigProtocol	是	RW
agentSnmpConfig 组		
agentSnmpCommunityCreate	是	RW
agentSnmpTrapReceiverCreate	是	RW
agentSnmpCommunityConfig 表		

表 C-26 FASTPATH 交换 MIB (续)

对象	支持	访问
索引: agentSnmpCommunityIndex		
agentSnmpCommunityName	是	RW
agentSnmpCommunityIPAddress	是	RW
agentSnmpCommunityIPMask	是	RW
agentSnmpCommunityAccessMode	是	RW
agentSnmpCommunityStatus	是	RW
agentSnmpTrapReceiverConfig 表		
索引: agentSnmpTrapReceiverIndex		
agentSnmpTrapReceiverCommunityName	是	RW
agentSnmpTrapReceiverIPAddress	是	RW
agentSnmpTrapReceiverStatus	是	RW
agentSnmpTrapFlagsConfig 组		
agentSnmpAuthenticationTrapFlag	是	RW
agentSnmpLinkUpDownTrapFlag	是	RW
agentSnmpMultipleUsersTrapFlag	是	RW
agentSnmpSpanningTreeTrapFlag	是	RW
agentSnmpBroadcastStormTrapFlag	是	RW
agentSpanningTreeConfig 组		
agentSpanningTreeMode	是	RW
agentSwitchConfig 组		
agentSwitchBroadcastStormRecoveryMode	是	RW
agentSwitchDot3FlowControlMode	是	RW
agentSwitchAddressAgingTimeoutTable		
索引: dot1qFdbId		
agentSwitchAddressAgingTimeout	是	RW

表 C-26 FASTPATH 交换 MIB (续)

对象	支持	访问
agentSwitchStaticMacFilteringTable		
索引: agentSwitchStaticMacFilteringVlanId		
agentSwitchStaticMacFilteringAddress	是	RW
agentSwitchStaticMacFilteringSourcePortMask	是	RW
agentSwitchStaticMacFilteringDestPortMask	是	RW
agentSwitchStaticMacFilteringStatus	是	RC
agentSwitchIGMPSnoopingGroup		
agentSwitchIGMPSnoopingAdminMode	是	RW
agentSwitchIGMPSnoopingGroupMembershipInterval	是	RW
agentSwitchIGMPSnoopingMaxResponseTime	是	RW
agentSwitchIGMPSnoopingExpirationTime	是	RW
agentSwitchIGMPSnoopingPortMask	是	RW
agentSwitchIGMPSnoopingMulticastControlFramesProcess	是	RO
agentSwitchMFDBTable		
索引: agentSwitchMFDBVlanId、agentSwitchMFDBMacAddress、 agentSwitchMFDBProtocolType		
agentSwitchMFDBType	是	RO
agentSwitchMFDBDescription	是	RO
agentSwitchMFDBForwardingPortMask	是	RO
agentSwitchMFDBFilteringPortMask	是	RO
agentSwitchMFDBSummaryTable		
索引: agentSwitchMFDBSummaryVlanId、 agentSwitchMFDBSummaryMacAddress		
agentSwitchMFDBSummaryForwardingPortMask	是	RO
agentSwitchMFDBGroup		
agentSwitchMFDBMaxTableEntries	是	RO
agentSwitchMFDBMostEntriesUsed	是	RO

表 C-26 FASTPATH 交换 MIB (续)

对象	支持	访问
agentSwitchMFDBCurrentEntries	是	RO
agentTransferUploadConfig 组		
agentTransferUploadMode	是	RW
agentTransferUploadServerIP	是	RW
agentTransferUploadPath	是	RW
agentTransferUploadFilename	是	RW
agentTransferUploadDataType	是	RW
agentTransferUploadStart	是	RW
agentTransferUploadStatus	是	RO
agentTransferDownloadConfig 组		
agentTransferDownloadMode	是	RW
agentTransferDownloadServerIP	是	RW
agentTransferDownloadPath	是	RW
agentTransferDownloadFilename	是	RW
agentTransferDownloadDataType	是	RW
agentTransferDownloadStart	是	RW
agentTransferDownloadStatus	是	RO
agentPortMirroring 组		
agentMirroredPortIfIndex	是	RW
agentProbePortIfIndex	是	RW
agentPortMirroringMode	是	RW
agentDot3adAggPortTable		
索引: agentDot3adAggPort		
agentDot3adAggPortLACPMode	是	RW
agentPortConfig 表		
索引: agentPortDot1dBasePort		

表 C-26 FASTPATH 交换 MIB (续)

对象	支持	访问
agentPortIfIndex	是	RO
agentPortIanaType	是	RO
agentPortSTPState	是	RO
agentPortSTPMode	是	RW
agentPortAdminMode	是	RW
agentPortPhysicalMode	否	N/A
agentPortPhysicalStatus	否	N/A
agentPortLinkTrapMode	是	RW
agentPortClearStats	是	RW
agentPortDefaultType	是	RW
agentPortType	是	RO
agentPortAutoNegAdminStatus	是	RW
agentPortDot3FlowControlMode	是	RW
agentPortDVlanTagMode	是	RW
agentPortDVlanTagEthertype	是	RW
agentPortDVlanTagCustomerId	是	RW
agentPortMaxFrameSizeLimit	是	RO
agentPortMaxFrameSize	是	RW
agentProtocolConfigGroup		
agentProtocolGroupCreate	是	RW
agentProtocolGroupTable		
索引: agentProtocolGroupId		
agentProtocolGroupName	是	RO
agentProtocolGroupVlanId	是	RW
agentProtocolGroupProtocolIP	是	RW
agentProtocolGroupProtocolARP	是	RW
agentProtocolGroupProtocolIPX	是	RW
agentProtocolGroupStatus	是	RW

表 C-26 FASTPATH 交换 MIB (续)

对象	支持	访问
agentProtocolGroupPortTable		
索引: agentProtocolGroupId、agentProtocolGroupPortIfIndex		
agentProtocolGroupPortStatus	是	RC
agentStpSwitchConfigGroup		
agentStpConfigDigestKey	是	RO
agentStpConfigFormatSelector	是	RO
agentStpConfigName	是	RW
agentStpConfigRevision	是	RW
agentStpForceVersion	是	RW
agentStpAdminMode	是	RW
agentStpPortTable		
索引: ifIndex		
agentStpPortState	是	RW
agentStpPortStatsMstpBpduRx	是	RO
agentStpPortStatsMstpBpduTx	是	RO
agentStpPortStatsRstpBpduRx	是	RO
agentStpPortStatsRstpBpduTx	是	RO
agentStpPortStatsStpBpduRx	是	RO
agentStpPortStatsStpBpduTx	是	RO
agentStpPortUpTime	是	RO
agentStpPortMigrationCheck	是	RW
agentStpCstConfigGroup		
agentStpCstHelloTime	是	RO
agentStpCstMaxAge	是	RO
agentStpCstRegionalRootId	是	RO
agentStpCstRegionalRootPathCost	是	RO

表 C-26 FASTPATH 交换 MIB (续)

对象	支持	访问
agentStpCstRootFwdDelay	是	RO
agentStpCstBridgeFwdDelay	是	RW
agentStpCstBridgeHelloTime	是	RW
agentStpCstBridgeHoldTime	是	RO
agentStpCstBridgeMaxAge	是	RW
agentStpCstPortTable		
索引: ifIndex		
agentStpCstPortOperEdge	是	RO
agentStpCstPortOperPointToPoint	是	RO
agentStpCstPortTopologyChangeAck	是	RO
agentStpCstPortEdge	是	RW
agentStpCstPortForwardingState	是	RO
agentStpCstPortId	是	RO
agentStpCstPortPathCost	是	RW
agentStpCstPortPriority	是	RW
agentStpCstDesignatedBridgeId	是	RO
agentStpCstDesignatedCost	是	RO
agentStpCstDesignatedPortId	是	RO
agentStpMstTable		
索引: agentStpMstId		
agentStpMstBridgePriority	是	RW
agentStpMstBridgeIdentifier	是	RO
agentStpMstDesignatedRootId	是	RO
agentStpMstRootPathCost	是	RO
agentStpMstRootPortId	是	RO
agentStpMstTimeSinceTopologyChange	是	RO
agentStpMstTopologyChangeCount	是	RO
agentStpMstTopologyChangeParm	是	RO

表 C-26 FASTPATH 交换 MIB (续)

对象	支持	访问
agentStpMstRowStatus	是	RC
agentStpMstPortTable		
索引: agentStpMstId、ifIndex		
agentStpMstPortForwardingState	是	RO
agentStpMstPortId	是	RO
agentStpMstPortPathCost	是	RW
agentStpMstPortPriority	是	RW
agentStpMstDesignatedBridgeId	是	RO
agentStpMstDesignatedCost	是	RO
agentStpMstDesignatedPortId	是	RO
agentStpMstVlanTable		
索引: agentStpMstId、dot1qVlanIndex		
agentStpMstVlanRowStatus	是	RC
agentAuthenticationGroup		
agentAuthenticationListCreate	是	RW
agentUserConfigDefaultAuthenticationList	是	RW
agentAuthenticationListTable		
索引: agentAuthenticationListIndex		
agentAuthenticationListName	是	RO
agentAuthenticationListMethod1	是	RW
agentAuthenticationListMethod2	是	RW
agentAuthenticationListMethod3	是	RW
agentAuthenticationListStatus	是	RW
agentUserAuthenticationConfigTable		
扩充: agentUserConfigEntry		
agentUserAuthenticationList	是	RW

表 C-26 FASTPATH 交换 MIB (续)

对象	支持	访问
agentUserPortConfigTable		
扩充: agentUserConfigEntry		
agentUserPortSecurity	是	RW
agentClassOfServicePortTable		
索引: ifIndex、agentClassOfServicePortPriority		
agentClassOfServicePortClass	是	RW
agentSystemConfig 组		
agentSaveConfig	是	RW
agentSaveConfigStatus	是	RW
agentClearConfig	是	RW
agentClearLags	是	RW
agentClearLoginSessions	是	RW
agentClearPasswords	是	RW
agentClearPortStats	是	RW
agentClearSwitchStats	是	RW
agentClearTrapLog	是	RW
agentClearVlan	是	RW
agentResetSystem	是	RO
agentCableTesterGroup		
agentCableTesterStatus	是	RW
agentCableTesterIfIndex	是	RW
agentCableTesterCableStatus	是	RO
agentCableTesterMinimumCableLength	是	RO
agentCableTesterMaximumCableLength	是	RO
agentCableTesterCableFailureLocation	是	RO

表 C-27 FASTPATH 路由 MIB

对象	支持	访问
agentSwitchArpGroup		
agentSwitchArpAgeoutTime	是	RW
agentSwitchArpResponseTime	是	RW
agentSwitchArpMaxRetries	是	RW
agentSwitchArpCacheSize	是	RW
agentSwitchArpDynamicRenew	是	RW
agentSwitchArpTotalEntryCountCurrent	是	RO
agentSwitchArpTotalEntryCountPeak	是	RO
agentSwitchArpStaticEntryCountCurrent	是	RO
agentSwitchArpStaticEntryCountMax	是	RO
agentSwitchArpTable		
索引: agentSwitchArpIpAddress		
agentSwitchArpAge	是	RO
agentSwitchArpMacAddress	是	RC
agentSwitchArpInterface	是	RO
agentSwitchArpType	是	RO
agentSwitchArpStatus	是	RW
agentSwitchIpGroup		
agentSwitchIpRoutingMode	是	RW
agentSwitchIpInterfaceTable		
索引: agentSwitchIpInterfaceIfIndex		
agentSwitchIpInterfaceIpAddress	是	RW
agentSwitchIpInterfaceNetMask	是	RW
agentSwitchIpInterfaceClearIp	是	RW
agentSwitchIpInterfaceRoutingMode	是	RW

表 C-27 FASTPATH 路由 MIB (续)

对象	支持	访问
agentSwitchIpRouterDiscoveryTable		
索引: agentSwitchIpRouterDiscoveryIfIndex		
agentSwitchIpRouterDiscoveryAdvertiseMode	是	RW
agentSwitchIpRouterDiscoveryIpAddress	是	RO
agentSwitchIpRouterDiscoveryMaxAdvertisementInterval	是	RW
agentSwitchIpRouterDiscoveryMinAdvertisementInterval	是	RW
agentSwitchIpRouterDiscoveryAdvertisementLifetime	是	RW
agentSwitchIpRouterDiscoveryPreferenceLevel	是	RW
agentSwitchIpRouterDiscoveryAdvertisementAddress	是	RW
agentSwitchIpVlanTable		
索引: agentSwitchIpVlanId		
agentSwitchIpVlanIfIndex	是	RO
agentSwitchIpVlanRoutingStatus	是	RC
agentRouterRipConfigGroup		
agentRouterRipAdminState	是	RW
agentRouterRipSplitHorizonMode	是	RW
agentRouterRipAutoSummaryMode	是	RW
agentRouterRipHostRoutesAcceptMode	是	RW
agentRouterRipDefaultMetric	是	RW
agentRouterRipDefaultMetricConfigured	是	RW
agentRouterRipDefaultInfoOriginate	是	RW
agentRipRouteRedistTable		
索引: agentRipRouteRedistSource		
agentRipRouteRedistMode	是	RW
agentRipRouteRedistMetric	是	RW
agentRipRouteRedistMetricConfigured	是	RW
agentRipRouteRedistMatchInternal	是	RW

表 C-27 FASTPATH 路由 MIB (续)

对象	支持	访问
agentRipRouteRedistMatchExternal1	是	RW
agentRipRouteRedistMatchExternal2	是	RW
agentRipRouteRedistMatchNSSAExternal1	是	RW
agentRipRouteRedistMatchNSSAExternal2	是	RW
agentRipRouteRedistDistList	是	RW
agentRipRouteRedistDistListConfigured	是	RW
agentRouterOspfConfigGroup		
agentOspfDefaultMetric	是	RW
agentOspfDefaultMetricConfigured	是	RW
agentOspfDefaultInfoOriginate	是	RW
agentOspfDefaultInfoOriginateAlways	是	RW
agentOspfDefaultInfoOriginateMetric	是	RW
agentOspfDefaultInfoOriginateMetricConfigured	是	RW
agentOspfDefaultInfoOriginateMetricType	是	RW
agentRouterOspfRFC1583CompatibilityMode	是	RW
agentOspfRouteRedistTable		
索引: agentOspfRouteRedistSource		
agentOspfRouteRedistMode	是	RW
agentOspfRouteRedistMetric	是	RW
agentOspfRouteRedistMetricConfigured	是	RW
agentOspfRouteRedistMetricType	是	RW
agentOspfRouteRedistTag	是	RW
agentOspfRouteRedistSubnets	是	RW
agentOspfRouteRedistDistList	是	RW
agentOspfRouteRedistDistListConfigured	是	RW

表 C-27 FASTPATH 路由 MIB (续)

对象	支持	访问
agentOspfIfTable		
扩充: ospfIfEntry		
agentOspfIfAuthKeyId	是	RC
agentOspfVirtIfTable		
扩充: ospfVirtIfEntry		
agentOspfVirtIfAuthKeyId	是	RW
agentOspfAreaTable		
扩充: ospfAreaEntry		
agentOspfAuthType	是	RW
agentSnmpTrapFlagsConfigGroupLayer3		
agentSnmpVRRPNewMasterTrapFlag	是	RW
agentSnmpVRRPAuthFailureTrapFlag	是	RW
agentBootpDhcpRelayGroup		
agentBootpDhcpRelayMaxHopCount	是	RW
agentBootpDhcpRelayForwardingIp	是	RW
agentBootpDhcpRelayForwardMode	是	RW
agentBootpDhcpRelayMinWaitTime	是	RW
agentBootpDhcpRelayCircuitIdOptionMode	是	RW
agentBootpDhcpRelayNumOfRequestsReceived	是	RO
agentBootpDhcpRelayNumOfRequestsForwarded	是	RO
agentBootpDhcpRelayNumOfDiscards	是	RO

表 C-28 FASTPATH Radius MIB

对象	支持	访问
agentRadiusConfigGroup		
agentRadiusMaxTransmit	是	RW
agentRadiusTimeout	是	RW
agentRadiusAccountingMode	是	RW
agentRadiusStatsClear	是	RW
agentRadiusAccountingIndexNextValid	是	RO
agentRadiusServerIndexNextValid	是	RO
agentRadiusAccountingConfig 表		
索引: agentRadi		
agentRadiusAccountingServerAddress	是	RW
agentRadiusAccountingPort	是	RW
agentRadiusAccountingSecret	是	RW
agentRadiusAccountingStatus	是	RW
agentRadiusServerConfig 表		
索引: agentRadi		
agentRadiusServerAddress	是	RW
agentRadiusServerPort	是	RW
agentRadiusServerSecret	是	RW
agentRadiusServerPrimaryMode	是	RW
agentRadiusServerCurrentMode	是	RO
agentRadiusServerMsgAuth	是	RW
agentRadiusServerStatus	是	RW

表 C-29 FASTPATH QOS DiffServ MIB

对象	支持	访问
agentDiffServGenStatusGroup		
索引:		
agentDiffServGenStatusAdminMode	是	RW
agentDiffServGenStatusClassTableSize	是	RO
agentDiffServGenStatusClassTableMax	是	RO
agentDiffServGenStatusClassRuleTableSize	是	RO
agentDiffServGenStatusClassRuleTableMax	是	RO
agentDiffServGenStatusPolicyTableSize	是	RO
agentDiffServGenStatusPolicyTableMax	是	RO
agentDiffServGenStatusPolicyInstTableSize	是	RO
agentDiffServGenStatusPolicyInstTableMax	是	RO
agentDiffServGenStatusPolicyAttrTableSize	是	RO
agentDiffServGenStatusPolicyAttrTableMax	是	RO
agentDiffServGenStatusServiceTableSize	是	RO
agentDiffServGenStatusServiceTableMax	是	RO
agentDiffServClassGroup		
索引:		
agentDiffServClassIndexNextFree	是	RO
agentDiffServClassTable		
索引: agentDiffServClassIndex		
agentDiffServClassName	是	RC
agentDiffServClassType	是	RC
agentDiffServClassAclNum	是	RC
agentDiffServClassRuleIndexNextFree	是	RO
agentDiffServClassStorageType	是	RC
agentDiffServClassRowStatus	是	RC

表 C-29 FASTPATH QOS DiffServ MIB (续)

对象	支持	访问
agentDiffServClassRuleTable		
索引: agentDiffServClassIndex、 agentDiffServClassRuleIndex		
agentDiffServClassRuleMatchEntryType	是	RC
agentDiffServClassRuleMatchCos	是	RC
agentDiffServClassRuleMatchDstIpAddr	是	RC
agentDiffServClassRuleMatchDstIpMask	是	RC
agentDiffServClassRuleMatchDstL4PortStart	是	RC
agentDiffServClassRuleMatchDstL4PortEnd	是	RC
agentDiffServClassRuleMatchDstMacAddr	是	RC
agentDiffServClassRuleMatchDstMacMask	是	RC
agentDiffServClassRuleMatchEvery	是	RO
agentDiffServClassRuleMatchIpDscp	是	RC
agentDiffServClassRuleMatchIpPrecedence	是	RC
agentDiffServClassRuleMatchIpTosBits	是	RC
agentDiffServClassRuleMatchIpTosMask	是	RC
agentDiffServClassRuleMatchProtocolNum	是	RC
agentDiffServClassRuleMatchRefClassIndex	是	RC
agentDiffServClassRuleMatchSrcIpAddr	是	RC
agentDiffServClassRuleMatchSrcIpMask	是	RC
agentDiffServClassRuleMatchSrcL4PortStart	是	RC
agentDiffServClassRuleMatchSrcL4PortEnd	是	RC
agentDiffServClassRuleMatchSrcMacAddr	是	RC
agentDiffServClassRuleMatchSrcMacMask	是	RC
agentDiffServClassRuleMatchVlanId	是	RC
agentDiffServClassRuleMatchExcludeFlag	是	RC
agentDiffServClassRuleStorageType	是	RC
agentDiffServClassRuleRowStatus	是	RC

表 C-29 FASTPATH QOS DiffServ MIB (续)

对象	支持	访问
agentDiffServPolicyGroup		
索引: agentDiffServPolicyIndexNextFree	是	RO
agentDiffServPolicyTable		
索引: agentDiffServPolicyIndex		
agentDiffServPolicyName	是	RC
agentDiffServPolicyType	是	RC
agentDiffServPolicyInstIndexNextFree	是	RO
agentDiffServPolicyStorageType	是	RC
agentDiffServPolicyRowStatus	是	RC
agentDiffServPolicyInstTable		
索引: agentDiffServPolicyIndex、 agentDiffServPolicyInstIndex		
agentDiffServPolicyInstClassIndex	是	RC
agentDiffServPolicyInstAttrIndexNextFree	是	RO
agentDiffServPolicyInstStorageType	是	RC
agentDiffServPolicyInstRowStatus	是	RC
agentDiffServPolicyAttrTable		
索引: agentDiffServPolicyIndex、 agentDiffServPolicyInstIndex、 agentDiffServPolicyAttrIndex		
agentDiffServPolicyAttrStmtEntryType	是	RC
agentDiffServPolicyAttrStmtBandwidthCrate	是	RC
agentDiffServPolicyAttrStmtBandwidthCrateUnits	是	RC
agentDiffServPolicyAttrStmtExpediteCrate	是	RC
agentDiffServPolicyAttrStmtExpediteCrateUnits	是	RC
agentDiffServPolicyAttrStmtExpediteCburst	是	RC
agentDiffServPolicyAttrStmtMarkCosVal	是	RC
agentDiffServPolicyAttrStmtMarkIpDscpVal	是	RC

表 C-29 FASTPATH QOS DiffServ MIB (续)

对象	支持	访问
agentDiffServPolicyAttrStmtMarkIpPrecedenceVal	是	RC
agentDiffServPolicyAttrStmtPoliceConformAct	是	RC
agentDiffServPolicyAttrStmtPoliceConformVal	是	RC
agentDiffServPolicyAttrStmtPoliceExceedAct	是	RC
agentDiffServPolicyAttrStmtPoliceExceedVal	是	RC
agentDiffServPolicyAttrStmtPoliceNonconformAct	是	RC
agentDiffServPolicyAttrStmtPoliceNonconformVal	是	RC
agentDiffServPolicyAttrStmtPoliceSimpleCrate	是	RC
agentDiffServPolicyAttrStmtPoliceSimpleCburst	是	RC
agentDiffServPolicyAttrStmtPoliceSinglerateCrate	是	RC
agentDiffServPolicyAttrStmtPoliceSinglerateCburst	是	RC
agentDiffServPolicyAttrStmtPoliceSinglerateEburst	是	RC
agentDiffServPolicyAttrStmtPoliceTworateCrate	是	RC
agentDiffServPolicyAttrStmtPoliceTworateCburst	是	RC
agentDiffServPolicyAttrStmtPoliceTworatePrate	是	RC
agentDiffServPolicyAttrStmtPoliceTworatePburst	是	RC
agentDiffServPolicyAttrStmtRandomdropMinThresh	是	RC
agentDiffServPolicyAttrStmtRandomdropMaxThresh	是	RC
agentDiffServPolicyAttrStmtRandomdropMaxDropProb	是	RC
agentDiffServPolicyAttrStmtRandomdropSamplingRate	是	RC
agentDiffServPolicyAttrStmtRandomdropDecayExponent	是	RC
agentDiffServPolicyAttrStmtShapeAverageCrate	是	RC
agentDiffServPolicyAttrStmtShapePeakCrate	是	RC
agentDiffServPolicyAttrStmtShapePeakPrate	是	RC
agentDiffServPolicyAttrStorageType	是	RC
agentDiffServPolicyAttrRowStatus	是	RC
agentDiffServPolicyPerfInTable		
索引: agentDiffServPolicyIndex、 agentDiffServPolicyInstIndex、 ifIndex		
agentDiffServPolicyPerfInOfferedOctets	是	RO

表 C-29 FASTPATH QOS DiffServ MIB (续)

对象	支持	访问
agentDiffServPolicyPerfInOfferedPackets	是	RO
agentDiffServPolicyPerfInDiscardedOctets	是	RO
agentDiffServPolicyPerfInDiscardedPackets	是	RO
agentDiffServPolicyPerfInHCOfferedOctets	是	RO
agentDiffServPolicyPerfInHCOfferedPackets	是	RO
agentDiffServPolicyPerfInHCDiscardedOctets	是	RO
agentDiffServPolicyPerfInHCDiscardedPackets	是	RO
agentDiffServPolicyPerfInStorageType	是	RO
agentDiffServPolicyPerfInRowStatus	是	RO
agentDiffServPolicyPerfOutTable		
索引: agentDiffServPolicyIndex、 agentDiffServPolicyInstIndex、ifIndex		
agentDiffServPolicyPerfOutTailDroppedOctets	是	RO
agentDiffServPolicyPerfOutTailDroppedPackets	是	RO
agentDiffServPolicyPerfOutRandomDroppedOctets	是	RO
agentDiffServPolicyPerfOutRandomDroppedPackets	是	RO
agentDiffServPolicyPerfOutShapeDelayedOctets	是	RO
agentDiffServPolicyPerfOutShapeDelayedPackets	是	RO
agentDiffServPolicyPerfOutSentOctets	是	RO
agentDiffServPolicyPerfOutSentPackets	是	RO
agentDiffServPolicyPerfOutHCTailDroppedOctets	是	RO
agentDiffServPolicyPerfOutHCTailDroppedPackets	是	RO
agentDiffServPolicyPerfOutHCRandomDroppedOctets	是	RO
agentDiffServPolicyPerfOutHCRandomDroppedPackets	是	RO
agentDiffServPolicyPerfOutHCShapeDelayedOctets	是	RO
agentDiffServPolicyPerfOutHCShapeDelayedPackets	是	RO
agentDiffServPolicyPerfOutHCSentOctets	是	RO
agentDiffServPolicyPerfOutHCSentPackets	是	RO
agentDiffServPolicyPerfOutStorageType	是	RO
agentDiffServPolicyPerfOutRowStatus	是	RO

表 C-29 FASTPATH QOS DiffServ MIB (续)

对象	支持	访问
agentDiffServServiceTable		
索引: agentDiffServServiceIfIndex、 agentDiffServServiceIfDirection		
agentDiffServServicePolicyIndex	是	RC
agentDiffServServiceIfOperStatus	是	RO
agentDiffServServiceStorageType	是	RC
agentDiffServServiceRowStatus	是	RC
agentDiffServServicePerfTable		
索引: agentDiffServServiceIfIndex、 agentDiffServServiceIfDirection		
agentDiffServServicePerfOfferedOctets	是	RO
agentDiffServServicePerfOfferedPackets	是	RO
agentDiffServServicePerfDiscardedOctets	是	RO
agentDiffServServicePerfDiscardedPackets	是	RO
agentDiffServServicePerfSentOctets	是	RO
agentDiffServServicePerfSentPackets	是	RO
agentDiffServServicePerfHCOfferedOctets	是	RO
agentDiffServServicePerfHCOfferedPackets	是	RO
agentDiffServServicePerfHCDiscardedOctets	是	RO
agentDiffServServicePerfHCDiscardedPackets	是	RO
agentDiffServServicePerfHCSentOctets	是	RO
agentDiffServServicePerfHCSentPackets	是	RO

表 C-30 FASTPATH QOS DiffServ 扩展 MIB

对象	支持	访问
agentDiffServClassifier		
agentDiffServAuxMfClfrNextFree	是	RO

表 C-30 FASTPATH QOS DiffServ 扩展 MIB (续)

对象	支持	访问
agentDiffServAuxMfClfrTable		
索引: agentDiffServAuxMfClfrId		
agentDiffServAuxMfClfrDstAddr	是	RO
agentDiffServAuxMfClfrDstMask	是	RO
agentDiffServAuxMfClfrSrcAddr	是	RO
agentDiffServAuxMfClfrSrcMask	是	RO
agentDiffServAuxMfClfrProtocol	是	RO
agentDiffServAuxMfClfrDstL4PortMin	是	RO
agentDiffServAuxMfClfrDstL4PortMax	是	RO
agentDiffServAuxMfClfrSrcL4PortMin	是	RO
agentDiffServAuxMfClfrSrcL4PortMax	是	RO
agentDiffServAuxMfClfrCos	是	RO
agentDiffServAuxMfClfrTos	是	RO
agentDiffServAuxMfClfrTosMask	是	RO
agentDiffServAuxMfClfrDstMac	是	RO
agentDiffServAuxMfClfrDstMacMask	是	RO
agentDiffServAuxMfClfrSrcMac	是	RO
agentDiffServAuxMfClfrSrcMacMask	是	RO
agentDiffServAuxMfClfrVlanId	是	RO
agentDiffServAuxMfClfrStorage	是	RO
agentDiffServAuxMfClfrStatus	是	RO
agentDiffServIpPrecMarkActTable		
索引: agentDiffServIpPrecMarkActPrecedence		
agentDiffServIpPrecMarkActPrecedence	是	RO
agentDiffServCosMarkActTable		
索引: agentDiffServCosMarkActCos		
agentDiffServCosMarkActCos	是	RO

表 C-31 FASTPATH QOS BW MIB

对象	支持	访问
trafficClassGroup		
trafficClassCreate	是	RW
trafficClassTable		
索引: trafficClassIndex		
trafficClassName	是	RO
trafficClassIfIndex	是	RW
trafficClassVlanId	是	RW
trafficClassWeight	是	RW
trafficClassBandwidthAllocation	是	RW
trafficClassAcceptByteCount	是	RO
trafficClassStatus	是	RW
bandwidthAllocationGroup		
bandwidthAllocationCreate	是	RW
bandwidthAllocationTable		
索引: bandwidthAllocationIndex		
bandwidthAllocationName	是	RO
bandwidthAllocationMinBandwidth	是	RW
bandwidthAllocationMaxBandwidth	是	RW
bandwidthAllocationStatus	是	RW

表 C-32 FASTPATH QOS ACL MIB

对象	支持	访问
aclTable		
索引: aclIndex		
aclStatus	是	RC
aclIfTable		
索引: aclIndex、aclIfIndex、aclIfDirection		
aclIfStatus	是	RC
aclRuleTable		
索引: aclIndex、aclRuleIndex		
aclRuleAction	是	RC
aclRuleProtocol	是	RC
aclRuleSrcIpAddress	是	RC
aclRuleSrcIpMask	是	RC
aclRuleSrcL4Port	是	RC
aclRuleSrcL4PortRangeStart	是	RC
aclRuleSrcL4PortRangeEnd	是	RC
aclRuleDestIpAddress	是	RC
aclRuleDestIpMask	是	RC
aclRuleDestL4Port	是	RC
aclRuleDestL4PortRangeStart	是	RC
aclRuleDestL4PortRangeEnd	是	RC
aclRuleIPDSCP	是	RC
aclRuleIpPrecedence	是	RC
aclRuleIpTosBits	是	RC
aclRuleIpTosMask	是	RC
aclRuleStatus	是	RC

表 C-33 FASTPATH-INVENTORY-MIB

对象	支持	访问
agentInventoryStackGroup		
agentInventoryStackReplicateConfig	是	RW
agentInventoryStackReplicateCode	是	RW
agentInventoryStackReplicateCodeStatus	是	RO
agentInventoryStackReplicateSTK	是	RW
agentInventorySupportedUnitTable		
索引: agentInventorySupportedUnitIndex		
agentInventorySupportedUnitModelIdentifier	是	RO
agentInventorySupportedUnitDescription	是	RO
agentInventorySupportedUnitExpectedCodeVer	是	RO
agentInventoryUnitTable		
索引: agentInventoryUnitNumber		
agentInventoryUnitAssignNumber	是	RC
agentInventoryUnitType	是	RO
agentInventoryUnitSupportedUnitIndex	是	RC
agentInventoryUnitMgmtAdmin	是	RC
agentInventoryUnitHWMgmtPref	是	RO
agentInventoryUnitHWMgmtPrefValue	是	RO
agentInventoryUnitAdminMgmtPref	是	RC
agentInventoryUnitAdminMgmtPrefValue	是	RC
agentInventoryUnitStatus	是	RO
agentInventoryUnitDetectedCodeVer	是	RO
agentInventoryUnitDetectedCodeInFlashVer	是	RO
agentInventoryUnitUpTime	是	RO
agentInventoryUnitDescription	是	RW
agentInventoryUnitReplicateSTK	是	RW
agentInventoryUnitRowStatus	是	RC

表 C-33 FASTPATH-INVENTORY-MIB (续)

对象	支持	访问
agentInventorySlotTable		
索引: agentInventoryUnitNumber、agentInventorySlotNumber		
agentInventorySlotStatus	是	RO
agentInventorySlotPowerMode	是	RW
agentInventorySlotAdminMode	是	RW
agentInventorySlotInsertedCardType	是	RO
agentInventorySlotConfiguredCardType	是	RW
agentInventorySlotCapabilities	是	RO
agentInventoryCardTypeTable		
索引: agentInventoryCardIndex		
agentInventoryCardType	是	RO
agentInventoryCardModelIdentifier	是	RO
agentInventoryCardDescription	是	RO

表 C-34 draft-ietf-idmr-dvmp-mib-11 DVMP MIB

对象	支持	访问
dvmpscalar		
dvmpVersionString	是	RO
dvmpGenerationId	否	N/A
dvmpNumRoutes	是	RO
dvmpReachableRoutes	是	RO
dvmpInterfaceTable		
索引: dvmpInterfaceIfIndex		
dvmpInterfaceLocalAddress	是	RO
dvmpInterfaceMetric	是	RC
dvmpInterfaceStatus	是	RC
dvmpInterfaceRcvBadPkts	是	RO

表 C-34 draft-ietf-idmr-dvmrp-mib-11 DVMRP MIB (续)

对象	支持	访问
dvmrpInterfaceRcvBadRoutes	是	RO
dvmrpInterfaceSentRoutes	是	RO
dvmrpInterfaceInterfaceKey	否	N/A
dvmrpInterfaceInterfaceKeyVersion	否	N/A
dvmrpNeighborTable		
索引: dvmrpNeighborIfIndex、dvmrpNeighborAddress		
dvmrpNeighborUpTime	是	RO
dvmrpNeighborExpiryTime	是	RO
dvmrpNeighborGenerationId	是	RO
dvmrpNeighborMajorVersion	是	RO
dvmrpNeighborMinorVersion	是	RO
dvmrpNeighborCapabilities	是	RO
dvmrpNeighborRcvRoutes	是	RO
dvmrpNeighborRcvBadPkts	是	RO
dvmrpNeighborRcvBadRoutes	是	RO
dvmrpNeighborState	是	RO
dvmrpRouteTable		
索引: dvmrpRouteSource、dvmrpRouteSourceMask		
dvmrpRouteUpstreamNeighbor	是	RO
dvmrpRouteIfIndex	是	RO
dvmrpRouteMetric	是	RO
dvmrpRouteExpiryTime	是	RO
dvmrpRouteUpTime	是	RO
dvmrpRouteNextHopTable		
索引: dvmrpRouteNextHopSource、 dvmrpRouteNextHopSourceMask、dvmrpRouteNextHopIfIndex		
dvmrpRouteNextHopType	是	RO
dvmrpPruneTable		

表 C-34 draft-ietf-idmr-dvmrp-mib-11 DVMRP MIB (续)

对象	支持	访问
索引: dvmrpPruneGroup、dvmrpPruneSource、 dvmrpPruneSourceMask		
dvmrpPruneExpiryTime	是	RO
陷阱		
dvmrpNeighborLoss	是	
dvmrpNeighborNotPruning	是	

表 C-35 RFC 3289 DiffServ MIB

对象	支持	访问
diffServDataPathTable		
索引: ifIndex、diffServDataPathIfDirection		
diffServDataPathStart	是	RO
diffServDataPathStorage	是	RO
diffServDataPathStatus	是	RO
diffServClassifier		
diffServClfrNextFree	是	RO
diffServClfrElementNextFree	是	RO
diffServMultiFieldClfrNextFree	是	RO
diffServMeter		
diffServMeterNextFree	是	RO
diffServTBParam		
diffServTBParamNextFree	是	RO
diffServAction		
diffServActionNextFree	是	RO

表 C-35 RFC 3289 DiffServ MIB (续)

对象	支持	访问
diffServCountActNextFree	是	RO
diffServAlgDrop		
diffServAlgDropNextFree	是	RO
diffServRandomDropNextFree	是	RO
diffServQueue		
diffServQNextFree	是	RO
diffServScheduler		
diffServSchedulerNextFree	是	RO
diffServMinRateNextFree	是	RO
diffServMaxRateNextFree	是	RO
diffServClfrTable		
索引: diffServClfrId		
diffServClfrStorage	是	RO
diffServClfrStatus	是	RO
diffServClfrElementTable		
索引: diffServClfrId、diffServClfrElementId		
diffServClfrElementPrecedence	是	RO
diffServClfrElementNext	是	RO
diffServClfrElementSpecific	是	RO
diffServClfrElementStorage	是	RO
diffServClfrElementStatus	是	RO
diffServMultiFieldClfrTable		

表 C-35 RFC 3289 DiffServ MIB (续)

对象	支持	访问
索引: diffServMultiFieldClfrId		
diffServMultiFieldClfrAddrType	是	RO
diffServMultiFieldClfrDstAddr	是	RO
diffServMultiFieldClfrDstPrefixLength	是	RO
diffServMultiFieldClfrSrcAddr	是	RO
diffServMultiFieldClfrSrcPrefixLength	是	RO
diffServMultiFieldClfrDscp	是	RO
diffServMultiFieldClfrFlowId	是	RO
diffServMultiFieldClfrProtocol	是	RO
diffServMultiFieldClfrDstL4PortMin	是	RO
diffServMultiFieldClfrDstL4PortMax	是	RO
diffServMultiFieldClfrSrcL4PortMin	是	RO
diffServMultiFieldClfrSrcL4PortMax	是	RO
diffServMultiFieldClfrStorage	是	RO
diffServMultiFieldClfrStatus	是	RO
diffServMeterTable		
索引: diffServMeterId		
diffServMeterSucceedNext	是	RO
diffServMeterFailNext	是	RO
diffServMeterSpecific	是	RO
diffServMeterStorage	是	RO
diffServMeterStatus	是	RO
diffServTBParamTable		
索引: diffServTBParamId		
diffServTBParamType	是	RO
diffServTBParamRate	是	RO
diffServTBParamBurstSize	是	RO
diffServTBParamInterval	是	RO

表 C-35 RFC 3289 DiffServ MIB (续)

对象	支持	访问
diffServTBParamStorage	是	RO
diffServTBParamStatus	是	RO
diffServActionTable		
索引: diffServActionId		
diffServActionInterface	是	RO
diffServActionNext	是	RO
diffServActionSpecific	是	RO
diffServActionStorage	是	RO
diffServActionStatus	是	RO
diffServDscpMarkActTable		
索引: diffServDscpMarkActDscp		
diffServDscpMarkActDscp	是	RO
diffServCountActTable		
索引: diffServCountActId		
diffServCountActOctets	是	RO
diffServCountActPkts	是	RO
diffServCountActStorage	是	RO
diffServCountActStatus	是	RO
diffServAlgDropTable		
索引: diffServAlgDropId		
diffServAlgDropType	是	RO
diffServAlgDropNext	是	RO
diffServAlgDropQMeasure	是	RO
diffServAlgDropQThreshold	是	RO
diffServAlgDropSpecific	是	RO

表 C-35 RFC 3289 DiffServ MIB (续)

对象	支持	访问
diffServAlgDropOctets	是	RO
diffServAlgDropPkts	是	RO
diffServAlgRandomDropOctets	是	RO
diffServAlgRandomDropPkts	是	RO
diffServAlgDropStorage	是	RO
diffServAlgDropStatus	是	RO
diffServRandomDropTable		
索引: diffServRandomDropId		
diffServRandomDropMinThreshBytes	是	RO
diffServRandomDropMinThreshPkts	是	RO
diffServRandomDropMaxThreshBytes	是	RO
diffServRandomDropMaxThreshPkts	是	RO
diffServRandomDropProbMax	是	RO
diffServRandomDropWeight	是	RO
diffServRandomDropSamplingRate	是	RO
diffServRandomDropStorage	是	RO
diffServRandomDropStatus	是	RO
diffServQTable		
索引: diffServQId		
diffServQNext	是	RO
diffServQMinRate	是	RO
diffServQMaxRate	是	RO
diffServQStorage	是	RO
diffServQStatus	是	RO
diffServSchedulerTable		
索引: diffServSchedulerId		
diffServSchedulerNext	是	RO
diffServSchedulerMethod	是	RO

表 C-35 RFC 3289 DiffServ MIB (续)

对象	支持	访问
diffServSchedulerMinRate	是	RO
diffServSchedulerMaxRate	是	RO
diffServSchedulerStorage	是	RO
diffServSchedulerStatus	是	RO
diffServMinRateTable		
索引: diffServMinRateId		
diffServMinRatePriority	是	RO
diffServMinRateAbsolute	是	RO
diffServMinRateRelative	是	RO
diffServMinRateStorage	是	RO
diffServMinRateStatus	是	RO
diffServMaxRateTable		
索引: diffServMaxRateId		
diffServMaxRateLevel	是	RO
diffServMaxRateAbsolute	是	RO
diffServMaxRateRelative	是	RO
diffServMaxRateThreshold	是	RO
diffServMaxRateStorage	是	RO
diffServMaxRateStatus	是	RO

附录 D

传感器对应表和故障隔离

本附录定义了 CT900 机箱内的传感器，还定义了触发传感器时受影响的系统功能（如果适用）。

机箱传感器

表 D-1 传感器对应表

传感器编号	传感器名称	传感器类型	传感器说明	故障状况（受影响的功能）
0	FRU 0 热交换	离散 (0x6f), “热交换” (0xf0)	活动 ShMM 的热交换	
2	FRU 1 热交换	离散 (0x6f), “热交换” (0xf0)	热交换 RTM	N/A
3	FRU 2 热交换	离散 (0x6f), “热交换” (0xf0)	机框 EEPROM（冗余 PROM）的热交换	
4	FRU 8 热交换	离散 (0x6f), “热交换” (0xf0)	SAP 的热交换	
5	FRU 3 热交换	离散 (0x6f), “热交换” (0xf0)	风扇托盘 0 的热交换	
6	FRU 4 热交换	离散 (0x6f), “热交换” (0xf0)	风扇托盘 1 的热交换	
7	FRU 5 热交换	离散 (0x6f), “热交换” (0xf0)	风扇托盘 2 的热交换	
8	FRU 6 热交换	离散 (0x6f), “热交换” (0xf0)	PEM A 的热交换	
9	FRU 7 热交换	离散 (0x6f), “热交换” (0xf0)	PEM B 的热交换	
10	IPMB 链路 1	离散 (0x6f), “IPMB 链路” (0xf1)	插槽 7 的 IPMI 总线（地址 41h）。冗余对（IPMB_A 和 IPMB_B）	如果同时禁用了 IPMB-A 和 IPMB 链路，机框管理器将不会与插槽 7 中的刀片通信。
11	IPMB 链路 2	离散 (0x6f), “IPMB 链路” (0xf1)	插槽 8 的 IPMI 总线（地址 42h）。冗余对（IPMB_A 和 IPMB_B）	如果同时禁用了 IPMB-A 和 IPMB 链路，机框管理器将不会与插槽 8 中的刀片通信。
12	风扇托盘 0	离散 (0x6f), “实体存在” (0x25)	风扇托盘 0 存在	缺少风扇托盘会影响热完整性。必须安装所有风扇托盘。
13	风扇托盘 1	离散 (0x6f), “实体存在” (0x25)	风扇托盘 1 存在	缺少风扇托盘会影响热完整性。必须安装所有风扇托盘。

表 D-1 传感器对应表 (续)

传感器编号	传感器名称	传感器类型	传感器说明	故障状况 (受影响的功能)
14	风扇托盘 2	离散 (0x6f), “实体存在” (0x25)	风扇托盘 2 存在	缺少风扇托盘会影响热完整性。必须安装所有风扇托盘。
15	IPMB 链路 3	离散 (0x6f), “IPMB 链路” (0xf1)	插槽 6 的 IPMI 总线 (地址 43h)。冗余对 (IPMB_A 和 IPMB_B)	如果同时禁用了 IPMB-A 和 IPMB 链路, 机框管理器将不会与插槽 6 中的刀片通信。
16	IPMB 链路 4	离散 (0x6f), “IPMB 链路” (0xf1)	插槽 9 的 IPMI 总线 (地址 44h)。冗余对 (IPMB_A 和 IPMB_B)	如果同时禁用了 IPMB-A 和 IPMB 链路, 机框管理器将不会与插槽 9 中的刀片通信。
17	IPMB 链路 5	离散 (0x6f), “IPMB 链路” (0xf1)	插槽 5 的 IPMI 总线 (地址 45h)。冗余对 (IPMB_A 和 IPMB_B)	如果同时禁用了 IPMB-A 和 IPMB 链路, 机框管理器将不会与插槽 5 中的刀片通信。
18	IPMB 链路 6	离散 (0x6f), “IPMB 链路” (0xf1)	插槽 10 的 IPMI 总线 (地址 46h)。冗余对 (IPMB_A 和 IPMB_B)	如果同时禁用了 IPMB-A 和 IPMB 链路, 机框管理器将不会与插槽 10 中的刀片通信。
19	IPMB 链路 7	离散 (0x6f), “IPMB 链路” (0xf1)	插槽 4 的 IPMI 总线 (地址 47h)。冗余对 (IPMB_A 和 IPMB_B)	如果同时禁用了 IPMB-A 和 IPMB 链路, 机框管理器将不会与插槽 4 中的刀片通信。
20	IPMB 链路 8	离散 (0x6f), “IPMB 链路” (0xf1)	插槽 11 的 IPMI 总线 (地址 48h)。冗余对 (IPMB_A 和 IPMB_B)	如果同时禁用了 IPMB-A 和 IPMB 链路, 机框管理器将不会与插槽 11 中的刀片通信。
21	IPMB 链路 9	离散 (0x6f), “IPMB 链路” (0xf1)	插槽 3 的 IPMI 总线 (地址 49h)。冗余对 (IPMB_A 和 IPMB_B)	如果同时禁用了 IPMB-A 和 IPMB 链路, 机框管理器将不会与插槽 3 中的刀片通信。
22	IPMB 链路 10	离散 (0x6f), “IPMB 链路” (0xf1)	插槽 12 的 IPMI 总线 (地址 4Ah)。冗余对 (IPMB_A 和 IPMB_B)	如果同时禁用了 IPMB-A 和 IPMB 链路, 机框管理器将不会与插槽 12 中的刀片通信。
23	IPMB 链路 11	离散 (0x6f), “IPMB 链路” (0xf1)	插槽 2 的 IPMI 总线 (地址 4Bh)。冗余对 (IPMB_A 和 IPMB_B)	如果同时禁用了 IPMB-A 和 IPMB 链路, 机框管理器将不会与插槽 2 中的刀片通信。

表 D-1 传感器对应表 (续)

传感器编号	传感器名称	传感器类型	传感器说明	故障状况 (受影响的功能)
24	IPMB 链路 12	离散 (0x6f), “IPMB 链路” (0xf1)	插槽 13 的 IPMI 总线 (地址 4Ch)。冗余对 (IPMB_A 和 IPMB_B)	如果同时禁用了 IPMB-A 和 IPMB 链路, 机框管理器将不会与插槽 13 中的刀片通信。
25	IPMB 链路 13	离散 (0x6f), “IPMB 链路” (0xf1)	插槽 1 的 IPMI 总线 (地址 4Dh)。冗余对 (IPMB_A 和 IPMB_B)	如果同时禁用了 IPMB-A 和 IPMB 链路, 机框管理器将不会与插槽 1 中的刀片通信。
26	IPMB 链路 14	离散 (0x6f), “IPMB 链路” (0xf1)	插槽 14 的 IPMI 总线 (地址 4Eh)。冗余对 (IPMB_A 和 IPMB_B)	如果同时禁用了 IPMB-A 和 IPMB 链路, 机框管理器将不会与插槽 14 中的刀片通信。
27	IPMB 链路 15	离散 (0x6f), “IPMB 链路” (0xf1)	IPMI 底板	
120	中部排气	阈值 (0x01), “温度” (0x01)	排气温度, 中部	如果排气温度超过 UNR 阈值, 刀片可能会过热。
121	左侧排气	阈值 (0x01), “温度” (0x01)	排气温度, 左侧	如果排气温度超过 UNR 阈值, 刀片可能会过热。
122	右侧排气	阈值 (0x01), “温度” (0x01)	排气温度, 右侧	如果排气温度超过 UNR 阈值, 刀片可能会过热。
123	SAP 温度	阈值 (0x01), “温度” (0x01)	SAP 板上的温度传感器	如果 SAP 气温超过 UNR 阈值, 刀片可能会过热。
124	左侧进气温度	阈值 (0x01), “温度” (0x01)	左侧进气温度, 位于风扇托盘	如果进气温度超过 UNR 阈值, 则说明机房的空气调节出现了故障。
125	中部进气温度	阈值 (0x01), “温度” (0x01)	中部进气温度, 位于风扇托盘	如果进气温度超过 UNR 阈值, 则说明机房的空气调节出现了故障。
126	右侧进气温度	阈值 (0x01), “温度” (0x01)	右侧进气温度, 位于风扇托盘	如果进气温度超过 UNR 阈值, 则说明机房的空气调节出现了故障。
131	TELCO 报警	离散 (0x6f), “OEM 已保留” (0xdf)	发生了 Telco 事件	
132	BMC 监视程序	离散 (0x6f), “监视程序 2” (0x23)	ATCA IPMI 监视程序	
133	系统事件	离散 (0x6f), “系统事件” (0x12)	系统重新配置事件	

表 D-1 传感器对应表 (续)

传感器编号	传感器名称	传感器类型	传感器说明	故障状况 (受影响的功能)
135	风扇托盘操作状态	管理 子系统 运行状况 (28h)	<p>机框的当前冷却状态:</p> <ul style="list-style-type: none"> • 00h = 完全冗余 = 在地址表中定义的所有风扇托盘均可操作。 • 01h = 冗余丢失 = 在地址表中定义的某些风扇托盘丢失或不可操作。 <p>使用 HPDL 默认冷却管理策略, 将导致所有其余风扇托盘的风扇转速设置为最大值。</p>	
136	冷却状态	管理 子系统 运行状况 (28h)	<p>00h = 转变为“正常”。 冷却状态为“正常”</p> <ul style="list-style-type: none"> • 01h, 由“正常”转变为“非临界”。冷却状态现在为“次要报警”, 先前的冷却状态为“正常”。 • 02h, 从不严重转变为“临界”。冷却状态现在为“重要报警”, 先前的冷却状态为“正常”或“次要报警”。 • 04h, 从比较严重转变为“非临界”。冷却状态现在为“次要报警”, 先前的冷却状态为“重要报警”或“临界报警”。 • 05h, 从“不可恢复”转变为“临界”。当前的冷却状态为“重要报警”, 先前的冷却状态为“临界报警”。 • 06h, 转变为“不可恢复”。当前的冷却状态现在为“临界报警”。 	
137	风扇状态	管理 子系统 运行状况 (28h)	<ul style="list-style-type: none"> • 00h = 转变为“正常”。风扇状态为“正常”(风扇转速计传感器上未超过阈值)。 • 01h = 由“正常”转变为“非临界”。风扇状态现在为“次要报警”(某个转速计上超过了非临界阈值)。 	

表 D-1 传感器对应表 (续)

传感器编号	传感器名称	传感器类型	传感器说明	故障状况 (受影响的功能)
150	空气过滤器	离散 (0x6f), “实体存在” (0x25)	空气过滤器存在传感器	如果不存在空气过滤器, 刀片将变脏且可能会过热。
152	SAP	离散 (0x6f), “实体存在” (0x25)	SAP 存在	如果没有 SAP, 将不会出现 Telco 报警。SAP 丢失时, 将不会有 SAP 温度和排气温度。
162	PEM A 输入 2	离散 (0x6f), “实体存在” (0x25)	PEM A 输入 2 (保险丝前)	如果失败, 由输入 2 供电的 FRU 将无电源冗余。
163	PEM A 输入 2 熔断	离散 (0x6f), “实体存在” (0x25)	PEM A 输入 2 (保险丝后)	传感器 #162 和 163 可用于确定熔断是否失败或输入是否未连接。请参见表 2。
164	PEM A 输入 1	离散 (0x6f), “实体存在” (0x25)	PEM A 输入 1 (保险丝前)	如果失败, 由输入 1 供电的 FRU 将无电源冗余。
165	PEM A 输入 1 熔断	离散 (0x6f), “实体存在” (0x25)	PEM A 输入 1 (保险丝后)	传感器 #164 和 165 可用于确定熔断是否失败或输入是否未连接。请参见表 2。
166	PEM A 输入 4	离散 (0x6f), “实体存在” (0x25)	PEM A 输入 4 (保险丝前)	如果失败, 由输入 4 供电的 FRU 将无电源冗余。
167	PEM A 输入 4 熔断	离散 (0x6f), “实体存在” (0x25)	PEM A 输入 4 (保险丝后)	传感器 #166 和 167 可用于确定熔断是否失败或输入是否未连接。请参见表 2。
168	PEM A 输入 3	离散 (0x6f), “实体存在” (0x25)	PEM A 输入 3 (保险丝前)	如果失败, 由输入 3 供电的 FRU 将无电源冗余。
169	PEM A 输入 3 熔断	离散 (0x6f), “实体存在” (0x25)	PEM A 输入 3 (保险丝后)	传感器 #168 和 169 可用于确定熔断是否失败或输入是否未连接。请参见表 2。
174	PEM B 输入 2	离散 (0x6f), “实体存在” (0x25)	PEM B 输入 2 (保险丝前)	如果失败, 由输入 2 供电的 FRU 将无电源冗余。
175	PEM B 输入 2 熔断	离散 (0x6f), “实体存在” (0x25)	PEM B 输入 2 (保险丝后)	传感器 #174 和 175 可用于确定熔断是否失败或输入是否未连接。请参见表 2。
176	PEM B 输入 1	离散 (0x6f), “实体存在” (0x25)	PEM B 输入 1 (保险丝前)	如果失败, 由输入 1 供电的 FRU 将无电源冗余。

表 D-1 传感器对应表 (续)

传感器编号	传感器名称	传感器类型	传感器说明	故障状况 (受影响的功能)
177	PEM B 输入 1 熔断	离散 (0x6f), “实体存在” (0x25)	PEM B 输入 1 (保险丝后)	传感器 #176 和 177 可用于确定熔断是否失败或输入是否未连接。请参见表 2。
178	PEM B 输入 4	离散 (0x6f), “实体存在” (0x25)	PEM B 输入 4 (保险丝前)	如果失败, 由输入 4 供电的 FRU 将无电源冗余。
179	PEM B 输入 4 熔断	离散 (0x6f), “实体存在” (0x25)	PEM B 输入 4 (保险丝后)	传感器 #178 和 179 可用于确定熔断是否失败或输入是否未连接。请参见表 2。
180	PEM B 输入 3	离散 (0x6f), “实体存在” (0x25)	PEM B 输入 3 (保险丝前)	如果失败, 由输入 3 供电的 FRU 将无电源冗余。
181	PEM B 输入 3 熔断	离散 (0x6f), “实体存在” (0x25)	PEM B 输入 3 (保险丝后)	传感器 #180 和 181 可用于确定熔断是否失败或输入是否未连接。请参见表 2。
192	PEM A	离散 (0x6f), “实体存在” (0x25)	PEM A 存在	如果不存在 PEM A, 则 PEM B 将为系统加电。
193	PEM B	离散 (0x6f), “实体存在” (0x25)	PEM B 存在	如果不存在 PEM B, 则 PEM A 将为系统加电。
194	机框 EEPROM 1	离散 (0x6f), “实体存在” (0x25)		
195	机框 EEPROM 2	离散 (0x6f), “实体存在” (0x25)		
200	PEM A 温度	阈值 (0x01), “温度” (0x01)	PEM A 中的温度	如果 PEM 温度超过 UNR 阈值, 将出现冷却问题。
201	PEM B 温度	阈值 (0x01), “温度” (0x01)	PEM B 中的温度	如果 PEM 温度超过 UNR 阈值, 将出现冷却问题。
208	24V FT 0	离散 (0x6f), “实体存在” (0x25)	24V DC-DC 转换器输出成功	如果 24V DC-DC 失败, FT0 中的风扇将无法工作。
209	-48A 总线 FT 0	离散 (0x6f), “实体存在” (0x25)	FT 0 A 输入 (保险丝前)	如果 A 和 B 输入馈线均丢失, 则 24V DC-DC 转换器将无法运行。
210	-48A FT 0	离散 (0x6f), “实体存在” (0x25)	FT 0 A 输入 (保险丝后)	如果 A 和 B 输入馈线均丢失, 则 24V DC-DC 转换器将无法运行。
211	-48B 总线 FT 0	离散 (0x6f), “实体存在” (0x25)	FT 0 B 输入 (保险丝前)	如果 A 和 B 输入馈线均丢失, 则 24V DC-DC 转换器将无法运行。

表 D-1 传感器对应表 (续)

传感器编号	传感器名称	传感器类型	传感器说明	故障状况 (受影响的功能)
212	-48B FT 0	离散 (0x6f), “实体存在” (0x25)	FT 0 B 输入 (保险丝后)	如果 A 和 B 输入馈线均丢失, 则 24V DC-DC 转换器将无法运行。
213	-48A FT 0 保险丝	离散 (0x6f), “实体存在” (0x25)	FT 0 A 输入保险丝	传感器 #209 和 210 用于确定熔断是否失败或输入是否不存在。
214	-48B FT 0 保险丝	离散 (0x6f), “实体存在” (0x25)	FT 0 B 输入保险丝	传感器 #211 和 212 用于确定熔断是否失败或输入是否不存在。
215	24V FT 1	离散 (0x6f), “实体存在” (0x25)	24V DC-DC 转换器输出成功	如果 24V DC-DC 失败, FT1 中的风扇将无法工作。
216	-48A 总线 FT 1	离散 (0x6f), “实体存在” (0x25)	FT 1 A 输入 (保险丝前)	如果 A 和 B 输入馈线均丢失, 则 24V DC-DC 转换器将无法运行。
217	-48A FT 1	离散 (0x6f), “实体存在” (0x25)	FT 1 A 输入 (保险丝后)	如果 A 和 B 输入馈线均丢失, 则 24V DC-DC 转换器将无法运行。
218	-48B 总线 FT 1	离散 (0x6f), “实体存在” (0x25)	FT 1 B 输入 (保险丝前)	如果 A 和 B 输入馈线均丢失, 则 24V DC-DC 转换器将无法运行。
219	-48B FT 1	离散 (0x6f), “实体存在” (0x25)	FT 1 B 输入 (保险丝后)	如果 A 和 B 输入馈线均丢失, 则 24V DC-DC 转换器将无法运行。
220	-48A FT 1 保险丝	离散 (0x6f), “实体存在” (0x25)	FT 1 A 输入保险丝	传感器 #209 和 210 用于确定熔断是否失败或输入是否不存在。
221	-48B FT 1 保险丝	离散 (0x6f), “实体存在” (0x25)	FT 1 B 输入保险丝	传感器 #211 和 212 用于确定熔断是否失败或输入是否不存在。
222	24V FT 2	离散 (0x6f), “实体存在” (0x25)	24V DC-DC 转换器输出成功	如果 24V DC-DC 失败, FT2 中的风扇将无法工作。
223	-48A 总线 FT 2	离散 (0x6f), “实体存在” (0x25)	FT 2 A 输入 (保险丝前)	如果 A 和 B 输入馈线均丢失, 则 24V DC-DC 转换器将无法运行。
224	-48A FT 2	离散 (0x6f), “实体存在” (0x25)	FT 2 A 输入 (保险丝后)	如果 A 和 B 输入馈线均丢失, 则 24V DC-DC 转换器将无法运行。
225	-48B 总线 FT 2	离散 (0x6f), “实体存在” (0x25)	FT 2 B 输入 (保险丝前)	如果 A 和 B 输入馈线均丢失, 则 24V DC-DC 转换器将无法运行。

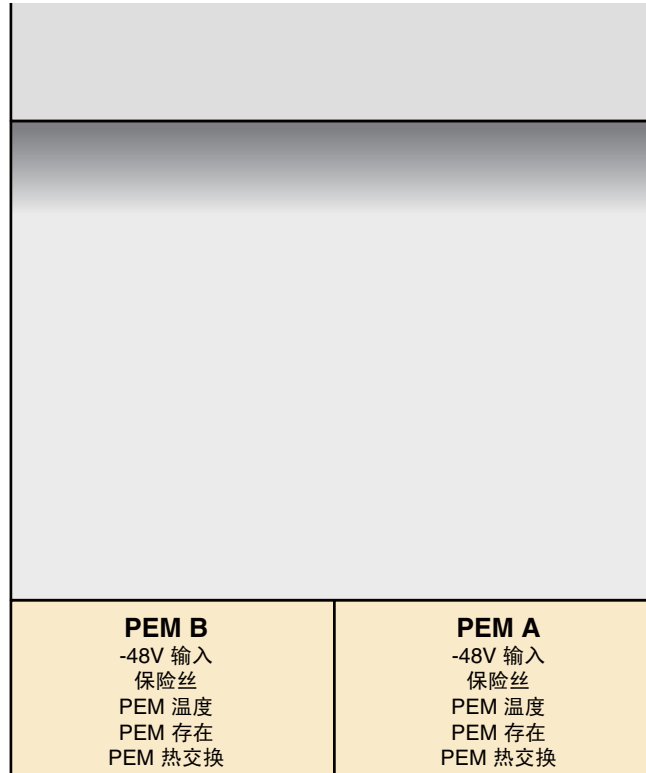
表 D-1 传感器对应表（续）

传感器编号	传感器名称	传感器类型	传感器说明	故障状况（受影响的功能）
226	-48B FT 2	离散 (0x6f), “实体存在” (0x25)	FT 2 B 输入（保险丝后）	如果 A 和 B 输入馈线均丢失，则 24V DC-DC 转换器将无法运行。
227	-48A FT 2 保险丝	离散 (0x6f), “实体存在” (0x25)	FT 2 A 输入保险丝	传感器 #209 和 210 用于确定熔断是否失败或输入是否不存在。
228	-48B FT 2 保险丝	离散 (0x6f), “实体存在” (0x25)	FT 2 B 输入保险丝	传感器 #211 和 212 用于确定熔断是否失败或输入是否不存在。
244	3V3_RAD	离散 (0x6f), “实体存在” (0x25)	向辐射型 IPMB 电路供电	指示 I2C-A 和 I2C-B 电源的故障。机框管理器载板上的辐射型 IPMB 电路将无法工作。

图 D-1 机箱级传感器位置 - 前面

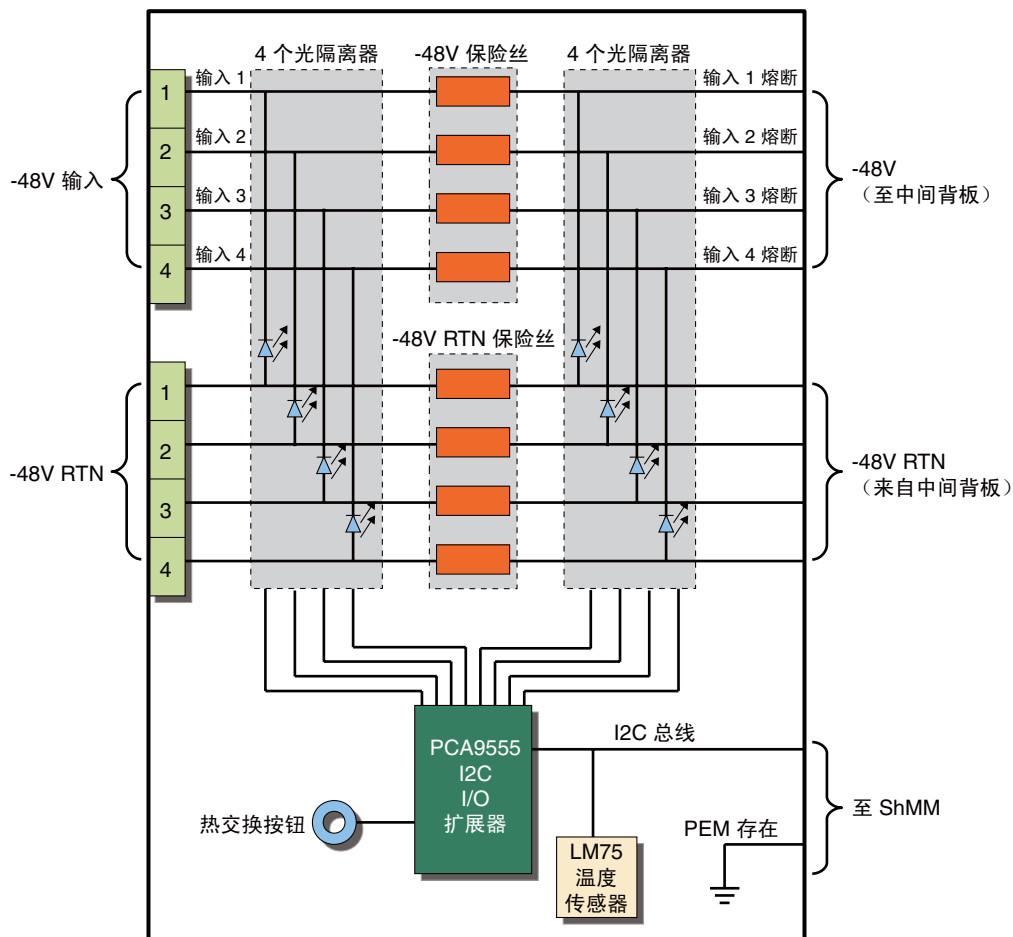


图 D-2 机箱级传感器位置 - 后面



PEM 传感器

图 D-3 PEM 传感器



PEM 传感器故障说明

表 D-2 PEM 传感器故障说明

-48V 输入	-48V 保险丝	-48V RTN 保险丝	-48V RTN	输入 1	输入 1 熔断
存在	正常	正常	存在	1	1
存在	正常	正常	丢失	0	0
存在	正常	已烧断	存在	1	0
存在	正常	已烧断	丢失	0	0
存在	已烧断	正常	存在	1	0
存在	已烧断	正常	丢失	0	0
存在	已烧断	已烧断	存在	1	0
存在	已烧断	已烧断	丢失	0	0
丢失	无关	无关	无关	0	0

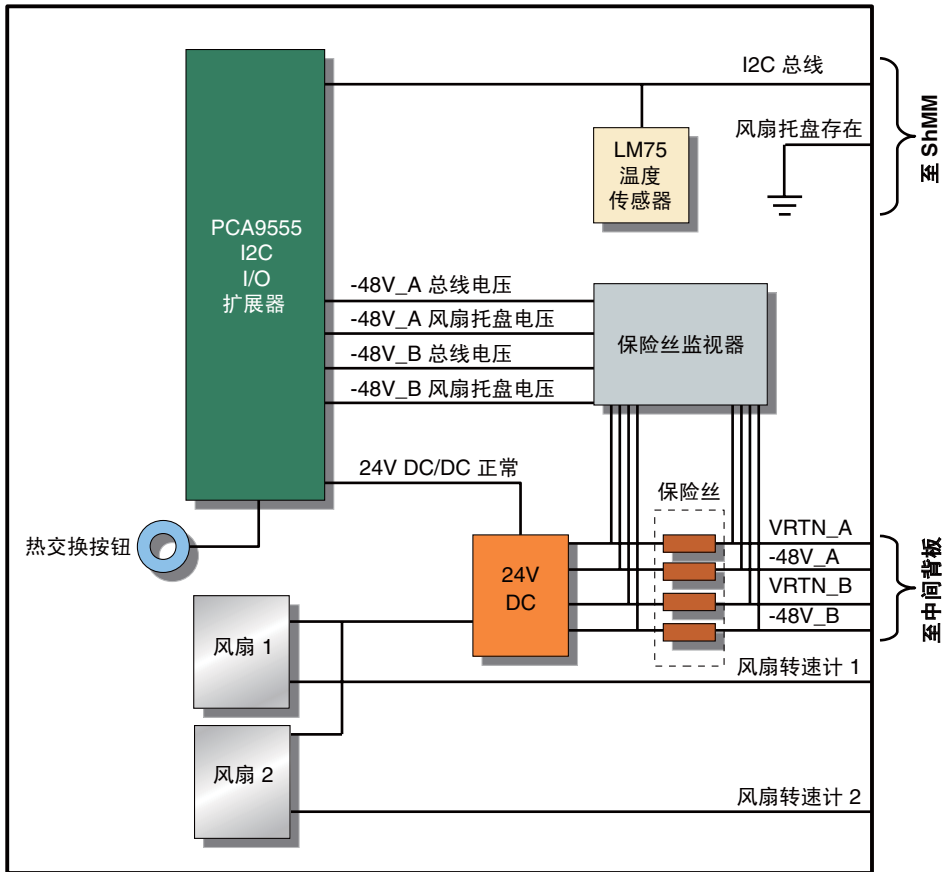
表 D-3 输入 1 的故障状况说明

输入 1	输入 1 熔断	故障状况
0	0	出现一个或多个下列故障：-48V RTN 丢失；-48V RTN 保险丝已烧断；48V 保险丝已烧断；-48V 输入丢失
0	1	无效
1	0	出现一个或多个下列故障：-48V RTN 保险丝已烧断；-48V 保险丝已烧断
1	1	无故障

注 – 如果两个 PEM 上的相同输入均出现故障，则选择的插槽、风扇托盘和/或机框管理器将受到影响。有关哪个输入馈线为系统组件供电的定义，请参阅《CT900 Hardware System Specification》。此故障仅在出现双重故障时发生。

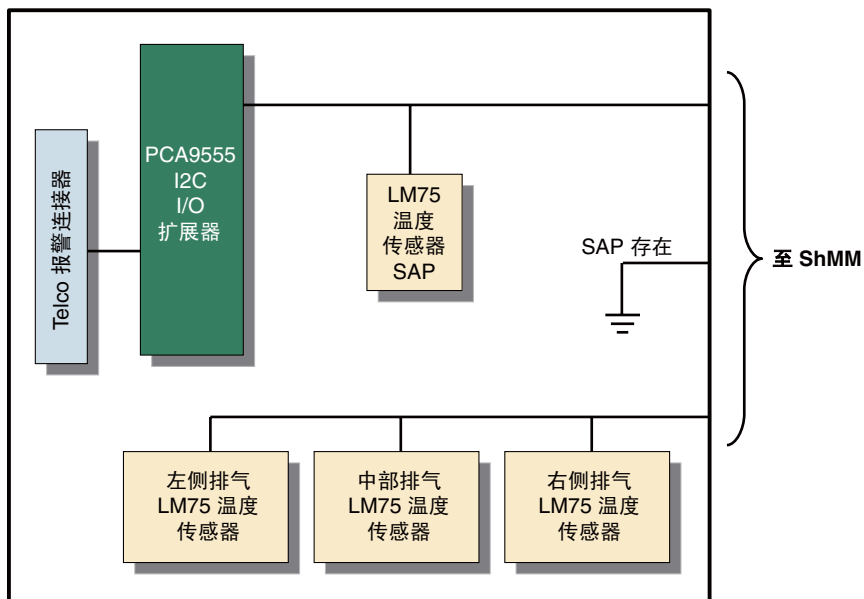
风扇托盘传感器

图 D-4 风扇托盘传感器



SAP 传感器

图 D-5 SAP 传感器



附录 E

ShMM 传感器对应表和故障隔离

本附录定义了 CT900 ShMM 卡内的传感器，还定义了触发传感器时受影响的系统功能（如果适用）。

ShMM 传感器

表 E-1 ShMM 传感器对应表

传感器编号	传感器名称	传感器类型	传感器说明	故障状况（受影响的功能）
0	FRU 0 热交换	离散 (0x6f), “热交换” (0xf0)	活动 ShMM 的热交换	
1	IPMB 链路	离散 (0x6f), “IPMB 链路” (0xf1)	适用于 ShMM 载体的 IPMB 链路	通信将失败。
2	本地温度	阈值 (0x01), “温度” (0x01)	本地温度传感器	如果本地温度超过 UNR 阈值, ShMM 载体可能会过热。
3	3V3_local	阈值 (0x01), “电压” (0x02)	ShMM 载体上的备用 3.3V 信号	将会记录事件。
4	I2C_PWR_A	阈值 (0x01), “电压” (0x02)	ShMM 载体上的 12V 信号 (I2C A 电源)	将会记录事件。
5	I2C_PWR_B	阈值 (0x01), “电压” (0x02)	ShMM 载体上的 5V 信号 (I2C B 电源)	将会记录事件。
6	VBAT	阈值 (0x01), “电压” (0x02)	ShMM 载体上的 VBAT 信号	将会记录事件。
7	风扇转速计 0	阈值 (0x01), “风扇” (0x04)	风扇 1 的 FT0 转速计传感器	如果超过阈值, 将出现风扇故障, 并亮起 FT 红色/SAP LED 指示灯。
8	风扇转速计 1	阈值 (0x01), “风扇” (0x04)	风扇 2 的 FT0 转速计传感器	如果超过阈值, 将出现风扇故障, 并亮起 FT 红色/SAP LED 指示灯。
10	风扇转速计 2	阈值 (0x01), “风扇” (0x04)	风扇 1 的 FT1 转速计传感器	如果超过阈值, 将出现风扇故障, 并亮起 FT 红色/SAP LED 指示灯。
11	风扇转速计 3	阈值 (0x01), “风扇” (0x04)	风扇 2 的 FT1 转速计传感器	如果超过阈值, 将出现风扇故障, 并亮起 FT 红色/SAP LED 指示灯。
13	风扇转速计 4	阈值 (0x01), “风扇” (0x04)	风扇 1 的 FT2 转速计传感器	如果超过阈值, 将出现风扇故障, 并亮起 FT 红色/SAP LED 指示灯。
14	风扇转速计 5	阈值 (0x01), “风扇” (0x04)	风扇 2 的 FT2 转速计传感器	如果超过阈值, 将出现风扇故障, 并亮起 FT 红色/SAP LED 指示灯。

表 E-1 ShMM 传感器对应表 (续)

传感器编号	传感器名称	传感器类型	传感器说明	故障状况 (受影响的功能)
15	-48A 总线电压	离散 (0x6f), “实体存在” (0x25)	载体上存在 GPIO 12	用于查找是否存在输入。
16	-48B 总线电压	离散 (0x6f), “实体存在” (0x25)	载体上存在 GPIO 13	用于查找是否存在输入。
17	-48A ACB 电压	离散 (0x6f), “实体存在” (0x25)	载体上存在 GPIO 14	用于查找是否存在输入。
18	-48B ACB 电压	离散 (0x6f), “实体存在” (0x25)	载体上存在 GPIO 15	用于查找是否存在输入。
19	-48A ACB 保险丝	离散 (0x6f), “实体存在” (0x25)	GPIO12 GPIO 14 不映射到特定信号, 但会说明 48V A 线路的运行状况	用于查找是否存在输入。
20	-48B ACB 保险丝	离散 (0x6f), “实体存在” (0x25)	GPIO13 GPIO 15 不映射到特定信号, 但会说明 48V 线路的运行状况	用于查找是否存在输入。
128	CPLD 状态	离散 (0x6f) “OEM 已保留” (0xde)	<p>CPLD 状态传感器:</p> <ul style="list-style-type: none"> • 0002h - 本地运行状况良好 • 0004h - 切换请求本地 • 0010h - 切换状态 LED 指示灯 1 • 0200h - 远程运行状况良好 (其他 ShMM 的状态: 1 = 运行状况良好, 0 = 运行状况不正常) • 1000h - 本地存在 (ShMM 的状态: 1 = 已连接, 0 = 未连接) • 2000h - 活动 	ShMM CPLD 状态更改 (包括冗余状态更改) 时生成事件。

表 E-1 ShMM 传感器对应表 (续)

传感器编号	传感器名称	传感器类型	传感器说明	故障状况 (受影响的功能)
129	重新引导原因	OEM 已保留 (0xdd)	<p>传感器的状态掩码可指出上一次重新引导的原因。(传感器读数始终为 0, 不具有任何意义。)</p> <ul style="list-style-type: none"> • [1] 切换操作导致重新引导。 • [2] 强制切换操作导致重新引导。 • [3] CLI 命令终止导致重新引导。 • [4] HEALTHY (运行状况良好) 位丢失导致重新引导。 • [5] ACTIVE (活动) 位丢失导致重新引导。 • [6] 由于冗余连接中断但活动 ShMM 仍处于活动状态, 导致备用 ShMM 重新引导。 • [7] 由于机框管理器启动期间出现错误, 导致重新引导。 • [8] ShMM 硬件监视程序导致重新引导。 • [9] 软件启动了重新引导 (reboot() 系统调用)。 • [10] ShMM 进行了开关机循环。 	

附录 F

Sun Netra CP3020 刀片服务器传感器 对应表和故障隔离

本附录定义了 Sun Netra CP3020 刀片服务器的传感器。

Sun Netra CP3020 刀片服务器传感器列表

Sun Netra CP3020 传感器的编号和名称由刀片服务器上的 H8 芯片进行报告。

表 F-1 Sun Netra CP3020 刀片服务器传感器

传感器编号	传感器名称	传感器类型	传感器说明	传感器超出限制时的故障状况
0	FRU 0 热交换	离散 (0x6f), “热交换” (0xf0)	刀片服务器 FRU 的热交换	N/A
1	RTM 热交换	离散 (0x6F), “热交换” (0xf0)	RTM 的热交换传感器	N/A
2	IPMB 物理	离散 (0x6f), “IPMB 链路” (0xf1)	IPMB 的链路状态	未收到来自 IPMB (A 或 B) 的回复。IPMB A 或 B 总线的状态通过监视 IPMB 隔离器上的 READY 信号来报告。
3	BMC 监视程序	离散 (0x6f), “监视程序 2” (0x23)	BMC 的监视程序状态	N/A
4	CPU Tcontrol (最大标准值 = 70)	阈值 (0x01), “温度” (0x01)	刀片服务器温度: Opteron CPU 的表面温度。设备 = ADM 1026, U153 管脚 25/26	IPMB 隔离器未准备好。如果此温度超过 75 摄氏度, H8 将关闭所有电源并打开前面板上的 OOS LED 指示灯。
5	刀片服务器进气口温度	阈值 (0x01), “温度” (0x01)	刀片服务器温度: 刀片服务器进气口处的环境温度。传感器位于刀片服务器的底部边缘 (靠近电源模块)。设备 = ADM 1026, U153 管脚 27/28	如果此温度超过 60 摄氏度, H8 将关闭所有电源并打开前面板上的 OOS LED 指示灯。
6	ADM 内部温度	阈值 (0x01), “温度” (0x01)	刀片服务器温度: 刀片服务器出口处的环境温度。设备 = ADM 1026, U153 内部	如果此温度超过 68 摄氏度, H8 将关闭所有电源并关闭前面板上的所有 LED 指示灯。
7	+12.0V 运行	阈值 (0x01), “电压” (0x02)	+12.0V 电源干线的电压测量值, 设备 = ADM 1026, U153 管脚 32	如果此电压不符合规格, 所有其他电源干线将出现故障 (STBY 除外)。刀片服务器和 RTM 将无法工作。如果 3.3V STBY 存在且符合规格, 则 H8 将可以正常工作。此干线是所有 DC/DC 转换器的电源。
8	-12.0V 运行	阈值 (0x01), “电压” (0x02)	-12.0V 电源干线的电压测量值, 设备 = ADM 1026, U153 管脚 31	这是 PMC 插槽的 -12V 电源干线。如果此电压不符合规格, 所有已安装的 PMC 将无法工作。

表 F-1 Sun Netra CP3020 刀片服务器传感器（续）

传感器编号	传感器名称	传感器类型	传感器说明	传感器超出限制时的故障状况
9	VCC 5.0V 运行	阈值 (0x01), “电压” (0x02)	5.0V 电源干线的电压测量值。设备 = ADM 1026, U153 管脚 30	此电压是内存 VRM、1.2V 转换器、PMC 卡、BIOS 芯片、SAS HDD 和以太网的电源之一。如果此干线不符合规格, 刀片服务器将无法工作。如果 3.3V STBY 存在且符合规格, 则 H8 将可以正常工作。
10	+3.3V 运行	阈值 (0x01), “电压” (0x02)	3.3V 电源干线的电压测量值。设备 = ADM 1026, U153 管脚 7	此电压是处理器、8132 I/O、上拉式电阻器和复位逻辑的电源之一。如果此干线不符合规格, 刀片服务器将无法工作。如果 3.3V STBY 存在且符合规格, 则 H8 将可以正常工作。
11	+3.3V ALW	阈值 (0x01), “电压” (0x02)	3.3V STBY 电源干线的电压测量值。设备 = ADM 1026, U153 管脚 22	如果此电压不符合规格, 刀片服务器和 H8 将无法工作。此干线是刀片服务器上大部分组件 (包括所有 I2C 设备和 H8) 的电源。
12	VCC RTC	阈值 (0x01), “电压” (0x02)	3.0 VBAT 电源干线的电压测量值。设备 = ADM 1026, U153 管脚 29	如果此电压不符合规格或为零, 则说明刀片服务器上的电池已损坏或丢失。当刀片服务器安装在机箱中时, 无需电池也可正常工作, 将使用 -48V 电源。电池的功能是, 当断开输入电源或从机箱中移除刀片服务器时用作 CMOS 和 RTC 的备用电源。
13	VDD 核心运行	阈值 (0x01), “电压” (0x02)	VCC 1.15V M 双电源干线 (3.3V 运行 + 3.3V STBY) 的电压测量值。设备 = ADM 1026, U153 管脚 33	此电压是处理器的电源之一。如果此干线不符合规格, 刀片服务器将无法工作。如果 3.3V STBY 存在且符合规格, 则 H8 将可以正常工作。
14	VCC 1.8V 双	阈值 (0x01), “电压” (0x02)	1.8V CPU 电源干线的电压测量值。设备 = ADM 1026, U153 管脚 34	此电压是 AMD 8111 I/O 集线器的电源之一。如果此干线不符合规格, 刀片服务器将无法工作。如果 3.3V STBY 存在且符合规格, 则 H8 将可以正常工作。
15	DDR VTT 1.3V	阈值 (0x01), “电压” (0x02)	VCC 1.3V 电源干线的电压测量值。设备 = ADM 1026, U153 管脚 35	此电源干线为主内存提供端接电压。如果此干线不符合规格, 刀片服务器将无法工作。如果 3.3V STBY 存在且符合规格, 则 H8 将可以正常工作。

表 F-1 Sun Netra CP3020 刀片服务器传感器（续）

传感器编号	传感器名称	传感器类型	传感器说明	传感器超出限制时的故障状况
16	VCC 1.2V 运行	阈值 (0x01), “电压” (0x02)	VCC 1.2V 电源干线的电压测量值。设备 = ADM 1026, U153 管脚 36	此电压是处理器和 1064 SAS 控制器的电源之一, 并向各种上拉式电阻器提供电源。如果此干线不符合规格, 刀片服务器将无法工作。如果 3.3V STBY 存在且符合规格, 则 H8 将可以正常工作。
17	VCC_5V_ALW	阈值 (0x01), “电压” (0x02)	VCC 5V ALW 电源干线的电压测量值 (传感器读取 1/2 值但 F/W 报告 2 倍的传感器读取值)。设备 = ADM 1026, U153 管脚 38	此电源干线为多个 POK 启用 5V 和 3.3 运行干线电压以及参考电压。如果此电压不符合规格或为零, 刀片服务器将无法工作。如果 3.3V STBY 存在且符合规格, 则 H8 将可以正常工作。
18	VDD 2.5V PU 运行	阈值 (0x01), “电压” (0x02)	VDD 2.5V 运行电源干线的电压测量值。设备 = ADM 1026, U153 管脚 39	此电源干线为多个关键处理器信号提供端接电压。如果此干线不符合规格, 刀片服务器可能无法工作。如果 3.3V STBY 存在且符合规格, 则 H8 将可以正常工作。
19	DDR VDD 2.6V 运行	阈值 (0x01), “电压” (0x02)	2.6 V 电源干线的电压测量值。设备 = ADM 1026, U153 管脚 40	此电压是 CPU 内存控制器和内存的电源之一。如果此干线不符合规格, 内存将无法工作。如果 3.3V STBY 存在且符合规格, 则 H8 将可以正常工作。
20	VCC 1V8 运行	阈值 (0x01), “电压” (0x02)	VCC 1.8V 运行电源干线的电压测量值。设备 = ADM 1026, U153 管脚 41	此电压是 AMD8111 I/O 集线器的电源之一。如果此干线不符合规格, 刀片服务器将无法工作。如果 3.3V STBY 存在且符合规格, 则 H8 将可以正常工作。
21	系统事件	离散 (0x6f), “系统事件” (0x12)		此传感器向 ShMM 报告 IPMC 复位事件。此传感器让 NetConsole 应用程序知道 IPMC 已执行复位操作且 NetConsole 会话必须重新启动。
22	RTM 存在	离散 (0x6f), “实体存在” (0x25)		此传感器指示 RTM 是否存在。
23	版本更改	离散 (0x6f), “已保留” (0x2b) 属于实体: (0x3, 96) [FRU # 0]		IPMC 在固件更新/冷复位后报告事件。

图 F-1 Netra CP3020 电压分配和 H8 传感器对应关系

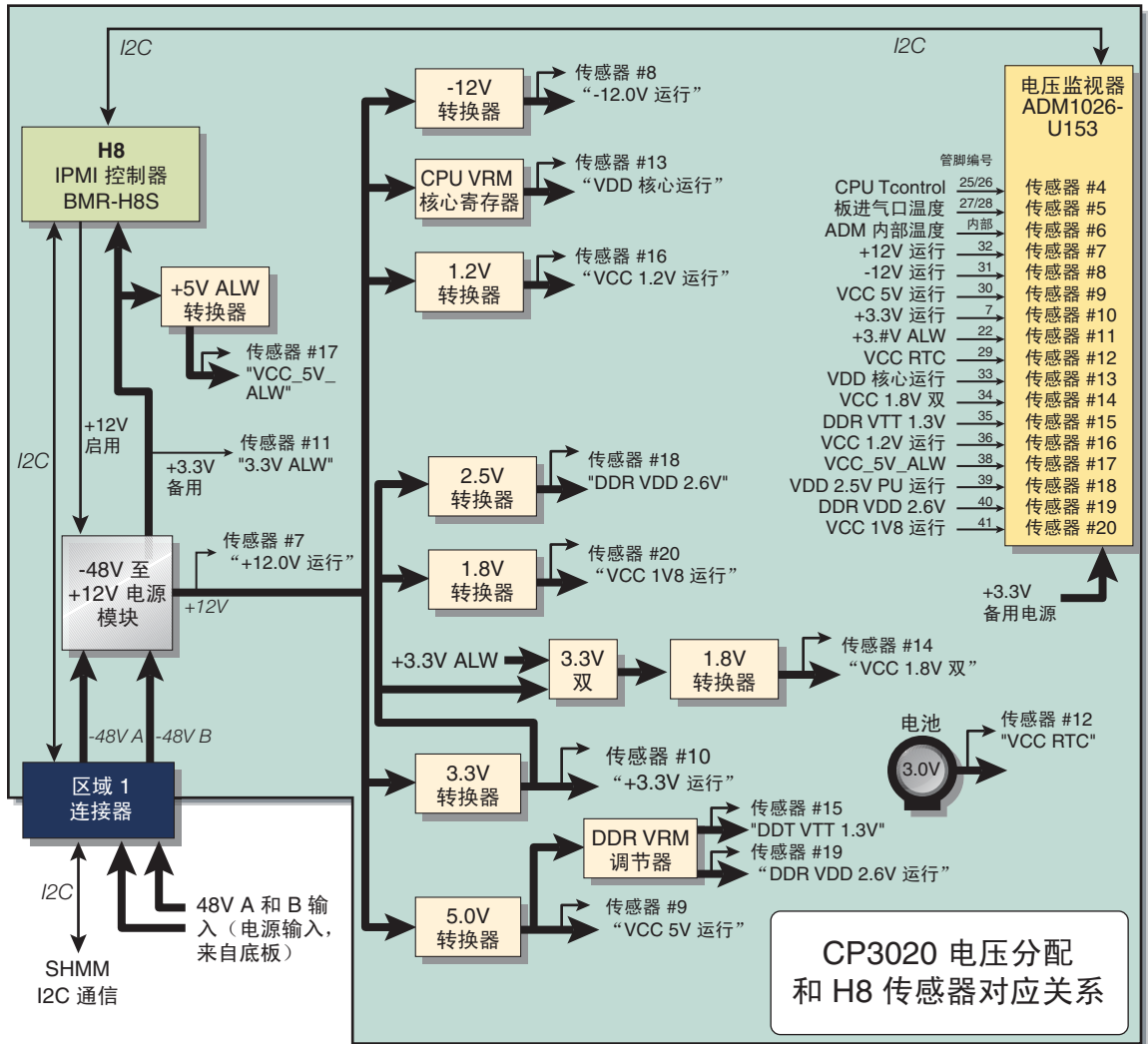
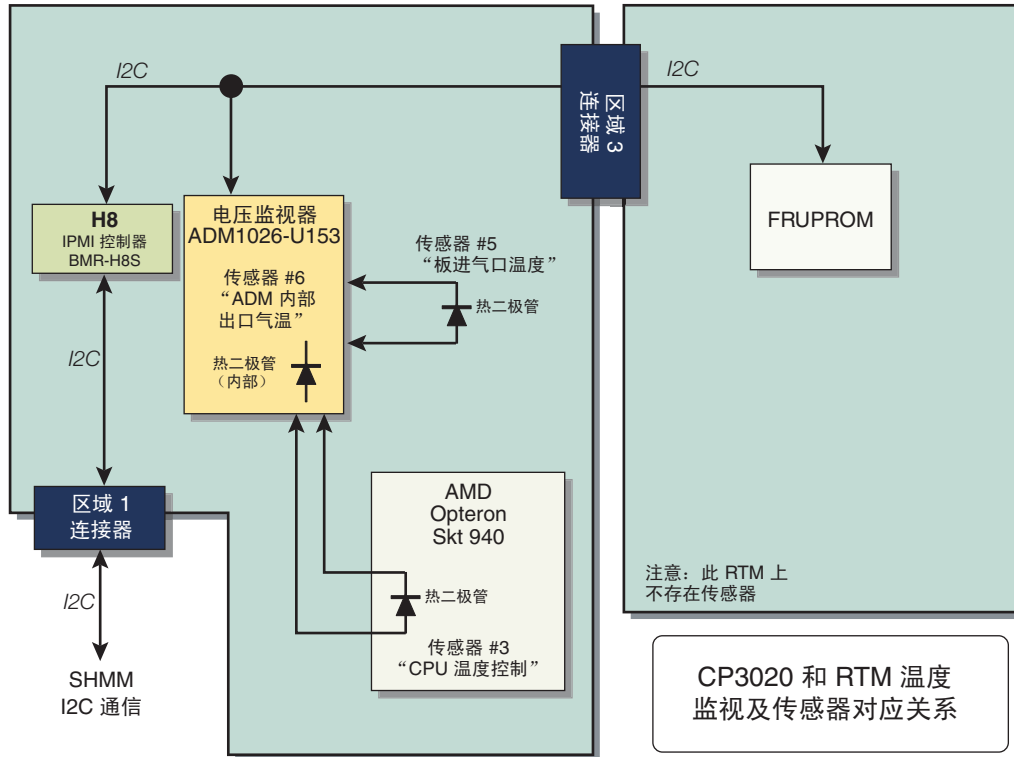


图 F-2 Sun Netra CP3020 刀片服务器和 RTM 温度监视及 H8 传感器对应关系



附录 G

Sun Netra CP3220 刀片服务器传感器 对应表和故障隔离

本附录定义了 Sun Netra CP3220 刀片服务器的传感器。

Sun Netra CP3220 刀片服务器传感器列表

Sun Netra CP3220 传感器的编号和名称由板载 H8 处理器通过 ATCA 机箱中的 ShMM 进行报告。

表 G-1 Sun Netra CP3220 刀片服务器传感器

传感器编号	传感器名称	传感器类型	传感器说明	传感器超出限制时的故障状况
0	FRU 0 热交换	离散 (0x6f), “热交换” (0xf0)	CP3220 FRU 的热交换	N/A
1	AMC 0 热交换	离散 (0x6f), “热交换” (0xf0)	AMC 0 的热交换 (托架 B1)	N/A
2	AMC 1 热交换	离散 (0x6f), “热交换” (0xf0)	AMC 1 的热交换 (托架 B2)	N/A
3	ARTM 热交换	离散 (0x6f), “热交换” (0xf0)	ARTM FRU 的热交换	N/A
4	IPMB 物理	离散 (0x6f), “IPMB 链路” (0xf1)	IPMB 的链路状态	IPMB 隔离器未准备好。IPMB A 或 B 总线的状态通过监视 IPMB 隔离器上的 READY 信号来报告。
5	BMC 监视程序	离散 (0x6f), “监视程序 2” (0x23)	BMC 的监视程序状态	N/A
6	CPU 表面温度	阈值 (0x01), “温度” (0x01)	CP3220 组件温度: Opteron CPU 的表面温度。设备 = ADM 1026, U60 管脚 25/26	如果此温度超过 86 摄氏度, H8 将关闭所有电源, 并打开前面板上的 OOS LED 指示灯。
7	区域 3 温度	“温度” (0x01)	刀片服务器温度: 刀片服务器的顶部 (靠近区域 3 连接器) 的环境温度。设备 = ADM 1026, U60 管脚 27/28	此传感器没有故障状况; 仅供参考之用。
8	AMC 区域温度	“温度” (0x01)	刀片服务器温度: 用于感测 AMC 0 温度的刀片服务器顶部的环境温度。设备 = ADM 1026, U60 内部	此传感器没有故障状况; 仅供参考之用。

表 G-1 Sun Netra CP3220 刀片服务器传感器（续）

传感器编号	传感器名称	传感器类型	传感器说明	传感器超出限制时的故障状况
9	12.0V	阈值 (0x01), “电压” (0x02)	12.0V 电源干线的电压测量值。设备 = ADM 1026, U60 管脚 32	如果此电压不符合规格, 所有其他电源干线将出现故障 (STBY 除外)。CP3220 和 ARTM 将无法工作。如果 3.3V STBY 存在且符合规格, 则 H8 将可以正常工作。此干线是所有 DC/DC 转换器的电源。
10	5.0V	阈值 (0x01), “电压” (0x02)	5.0V 电源干线的电压测量值。设备 = ADM 1026, U60 管脚 30	此电压是处理器、Nvidia I/O、USB 和 CPLD 的电源之一。如果此干线不符合规格, CP3220 将无法工作。如果 3.3V STBY 存在且符合规格, 则 H8 将可以正常工作。
11	3.3V	阈值 (0x01), “电压” (0x02)	3.3V 电源干线的电压测量值。设备 = ADM 1026, U60 管脚 7	此电压是处理器、Nvidia I/O、上拉式电阻器、BIOS 和复位逻辑的电源之一。如果此干线不符合规格, CP3220 将无法工作。如果 3.3V STBY 存在且符合规格, 则 H8 将可以正常工作。
12	3.3V STBY	阈值 (0x01), “电压” (0x02)	3.3V STBY 电源干线的电压测量值。设备 = ADM 1026, U60 管脚 22	如果此电压不符合规格, CP3220 和 H8 将无法工作。此干线是 CP3220 上大部分组件 (包括所有 I2C 设备和 H8) 的电源。
13	电池电压	阈值 (0x01), “电压” (0x02)	3.0 VBAT/STBY 电源干线的电压测量值。设备 = ADM 1026, U60 管脚 29	如果此电压不符合规格或为零, 则说明刀片服务器上的电池已损坏或丢失。当 CP3220 安装在机箱中时, 无需电池也可正常工作, 将使用 -48V 电源。电池的功能是, 当断开输入电源或从机箱中移除 CP3220 时用作 CMOS 和 RTC 的备用电源。
14	VCC 1.15V M 双	阈值 (0x01), “电压” (0x02)	VCC 1.15V M 双电源干线的电压测量值。设备 = ADM 1026, U60 管脚 34	如果此电压不符合规格, NVIDIA MCP55 上的 I/O 部分将无法工作。如果 3.3V STBY 处于打开状态, H8 将可以正常工作。此干线是 NVIDIA MCP55 PRO 的 I/O 部分的电源。

表 G-1 Sun Netra CP3220 刀片服务器传感器（续）

传感器编号	传感器名称	传感器类型	传感器说明	传感器超出限制时的故障状况
15	Proc0 0.9V DDR	阈值 (0x01), “电压” (0x02)	9V CPU 电源干线的电压测量值。设备 = ADM 1026, U60 管脚 35	此电压是处理器内存控制器的电源之一，也是内存终端电源。如果此干线不符合规格，CP3220 将无法工作。如果 3.3V STBY 存在且符合规格，则 H8 将可以正常工作。
16	VCC 1.2V HT	阈值 (0x01), “电压” (0x02)	VCC 1.2V 电源干线的电压测量值。设备 = ADM 1026, U60 管脚 36	此电压是处理器和 Nvidia 超传输总线的电源之一。如果此干线不符合规格，CP3220 将无法工作。如果 3.3V STBY 存在且符合规格，则 H8 将可以正常工作。
17	Proc0 核心 NB	阈值 (0x01), “电压” (0x02)	处理器核心电源干线的电压测量值（电压在 1.1V 到 1.4V 之间变化）。设备 = ADM 1026, U60 管脚 37	此电压是处理器核心的电源之一。如果此干线不符合规格，CP3220 将无法工作。如果 3.3V STBY 存在且符合规格，则 H8 将可以正常工作。
18	VCC 1.15V M 运行	阈值 (0x01), “电压” (0x02)	VCC 1.15V 运行电源干线的电压测量值。设备 = ADM 1026, U60 管脚 38	如果此电压不符合规格或为零，CPU 或主机将无法工作。如果所有其他电源干线均打开，服务处理器和 H8 仍将继续工作。供应：FBDIMM。
19	VCC 1.2V 运行	阈值 (0x01), “电压” (0x02)	VCC 1.2V 运行电源干线的电压测量值。设备 = ADM 1026, U60 管脚 39	此电压是 Nvidia MCP55 PRO 的电源之一。如果此干线不符合规格，CP3220 的大部分 I/O 将无法工作。如果 3.3V STBY 存在且符合规格，则 H8 将可以正常工作。
20	Proc0 1.8V DDR	阈值 (0x01), “电压” (0x02)	处理器 1.8V 电源干线的电压测量值。设备 = ADM 1026, U60 管脚 40	此电压是内存的电源之一。如果此干线不符合规格，存储器总线（处理器）将无法工作。如果 3.3V STBY 存在且符合规格，则 H8 将可以正常工作。

表 G-1 Sun Netra CP3220 刀片服务器传感器（续）

传感器编号	传感器名称	传感器类型	传感器说明	传感器超出限制时的故障状况
21	VCC 1.5V 运行	阈值 (0x01), “电压” (0x02)	VCC 1.5V 运行电源干线的电压测量值。设备 = ADM 1026, U60 管脚 41	此电压是 Nvidia MCP55 PRO 的电源之一。它也是 1.2V 干线的电源。如果此干线不符合规格, CP3220 将无法工作。如果 3.3V STBY 存在且符合规格, 则 H8 将可以正常工作。
22	Proc0 核心	阈值 (0x01), “电压” (0x02)	处理器核心干线的电压测量值 (电压在 1.05V 到 1.4V 之间变化)。设备 = ADM 1026, U60 管脚 33	此电压是 CPU 核心的电源之一。如果此干线不符合规格, CP3220 将无法工作。如果 3.3V STBY 存在且符合规格, 则 H8 将可以正常工作。
23	板进气口温度	“温度” (0x01)	刀片服务器温度: 板进气口处的环境温度。传感器位于刀片服务器的底部边缘 (靠近电源模块)。设备 = ADM 1032, U9 管脚 2/3	此传感器没有故障状况; 仅供参考之用。
24	PM 主端温度	“温度” (0x01)	CP3220 组件温度: 电源模块主端上 FET 的温度。传感器位于电源模块内部。	此传感器没有故障状况; 仅供参考之用。
25	PM 次端温度	“温度” (0x01)	CP3220 组件温度: 电源模块次端上 FET 的温度。传感器位于电源模块 U2 内部。	此传感器没有故障状况; 仅供参考之用。
26	PM -48V A 干线	阈值 (0x01), “电压” (0x02)	-48V A 端与 RTN-A 输入之间的电压测量值。传感器位于电源模块 U2 内部。	此电压传感器位于电源模块的内部。如果 A 输入电源降至 -36V 以下或 -72V 以上, 并且 B 输入符合规格, 电源模块将报告电源过低, 但是仍继续正常工作。
27	PM -48V B 干线	阈值 (0x01), “电压” (0x02)	-48V B 端与和 RTN-B 输入之间的电压测量值。传感器位于电源模块 U2 内部。	此电压传感器位于电源模块的内部。如果 B 输入电源降至 -36V 以下或 -72V 以上, 并且 A 输入符合规格, 电源模块将报告电源过低, 但是仍继续正常工作。
28	PM -48V 电压	阈值 (0x01), “电压” (0x02)	HU- 输入与 HU+ 输入之间的电压测量值。传感器位于电源模块内部。	此电压传感器位于电源模块的内部。如果 A 和 B 输入电源降至 -36V 以下或 -72V 以上, 电源模块将关闭, 强制将所有电源 (备用电池除外) 变为 0V。不会亮起任何 LED 指示灯。

表 G-1 Sun Netra CP3220 刀片服务器传感器（续）

传感器编号	传感器名称	传感器类型	传感器说明	传感器超出限制时的故障状况
29	-48V 电流	阈值 (0x01), “电流” (0x03)	输入 OR-ing 后 -48V 输入的电流测量值。传感器位于电源模块 U2 内部。	此传感器仅用于进行报告。
30	12V 电流	阈值 (0x01), “电流” (0x03)	电源模块 12V 输出的电流测量值。传感器位于电源模块 U2 内部。	此电流传感器位于电源模块的内部。如果 12VDC 的输出电流超过 6.48A, 12V 输出将关闭, CP3220 将无法工作。但是, 3.3V STBY 仍然起作用, 因此 H8 将继续工作。
31	版本更改	离散 (已保留), “版本” (0x2B)	固件更新事件	在固件更新/冷复位后报告事件。
32	系统事件	离散 (0x6f), “系统事件” (0x12)		此传感器向 ShMM 报告 IPMC 复位事件。此传感器让 NetConsole 应用程序知道 IPMC 已执行复位操作且 NetConsole 会话必须重新启动。
33	系统固件进度	离散 (0x6f), “系统固件进度” (0x0f),	监视系统固件进度状态。未使用。	未使用。保留以备将来使用。
34	正常重新引导	离散 (0x6f) “OEM 已保留” (0xc0)	监视正常重新引导的状态。	当正常重新引导计时器启动、停止或过期时, 此传感器会将事件记录到系统事件日志中。

图 G-1 Sun Netra CP3220 电压分配和 H8 传感器对应关系

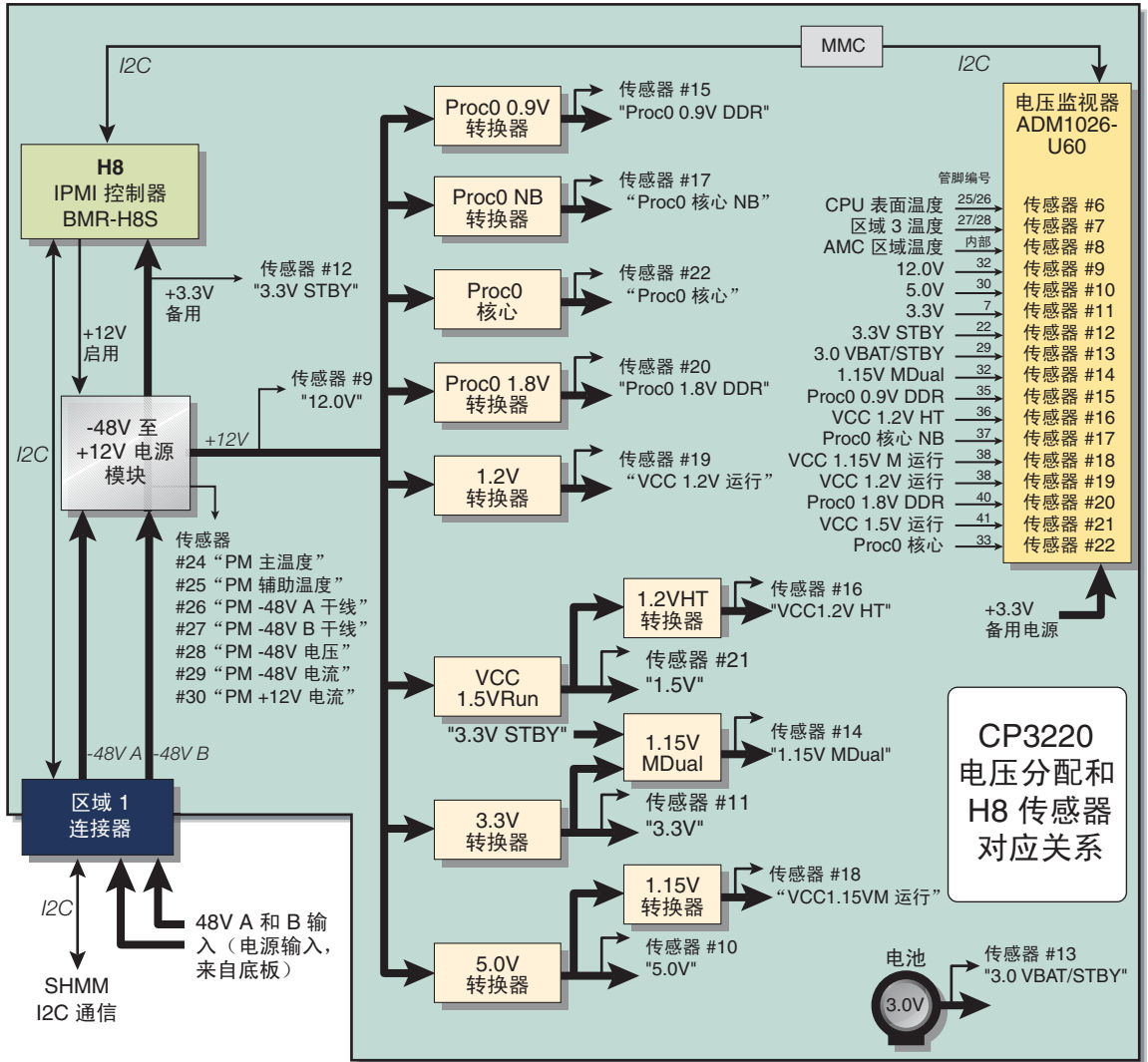
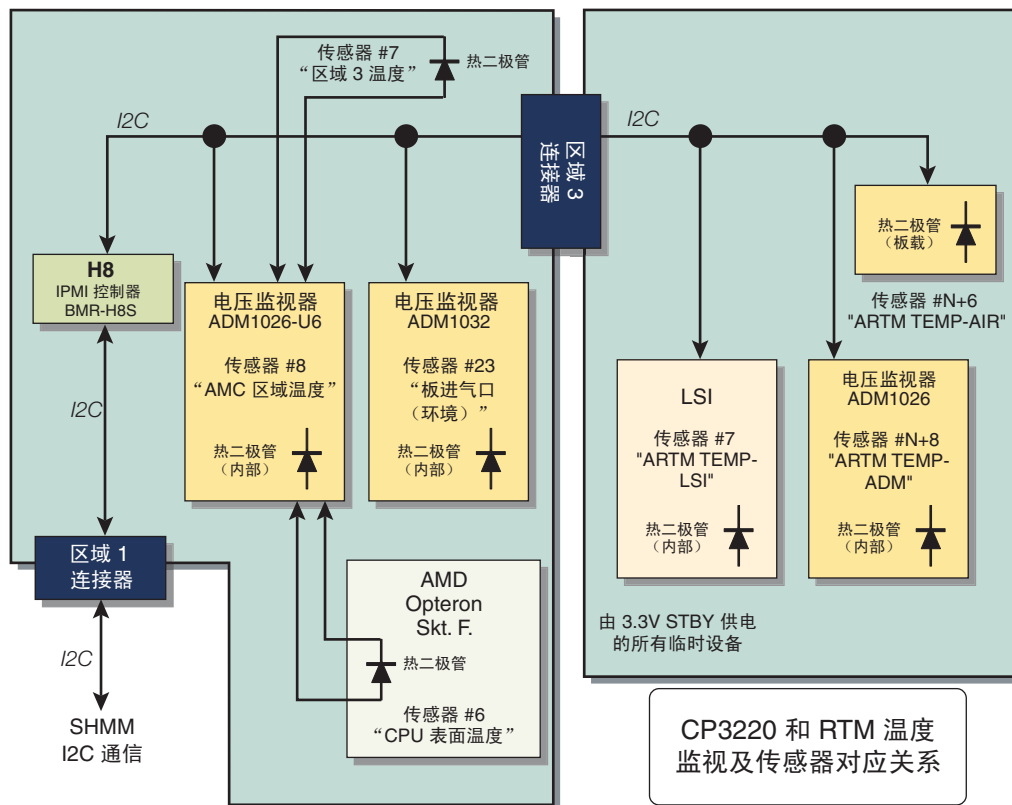


图 G-2 Sun Netra CP3220 刀片服务器和 RTM 温度监视及 H8 传感器对应关系



CP3220 和 RTM 温度
监视及传感器对应关系

附录 H

Sun Netra CP3060 刀片服务器传感器 对应表和故障隔离

本附录定义了 Sun Netra CP3060 刀片服务器的传感器。

Sun Netra CP3060 刀片服务器传感器列表

Sun Netra CP3060 传感器的名称由板载 H8 处理器通过 ATCA 机箱中的机框管理器进行报告。

表 H-1 Sun Netra CP3060 刀片服务器传感器列表

传感器编号	传感器名称	传感器类型	传感器说明	传感器超出限制时的故障状况
0	FRU 0 热交换	离散 (0x6f), “热交换” (0xf0)	CP3060 FRU 的热交换事件	按照 ATCA 规范中的说明监视 FRU 状态
1	RTM 热交换	离散 (0x6f), “热交换” (0xf0)	RTM FRU 的热交换事件	按照 ATCA 规范中的说明监视 FRU 状态
2	热交换 AMC 0	离散 (0x6f), “热交换” (0xf0)	AMC FRU 的热交换事件	按照 ATCA 规范中的说明监视 FRU 状态
3	IPMB 物理	离散 (0x6f), “IPMB 链路” (0xf1)	IPMB 的链路状态	IPMB A 或 B 总线的状态通过监视 IPMB 隔离器上的 READY 信号来报告
4	BMC 监视程序	离散 (0x6f), “监视程序 2” (0x23)	BMC 的监视程序状态	N/A
5	CPU 温度 1	阈值 (0x01), “温度” (0x01)	CPU 的内部管芯温度	将关闭电源/刀片服务器
6	CPU 温度 2	阈值 (0x01), “温度” (0x01)	CPU 的内部管芯温度	将关闭电源/刀片服务器
7	板温度	阈值 (0x01), “温度” (0x01)	刀片服务器温度: ADM1026 的环境温度	将关闭电源/刀片服务器
8	12.0V	阈值 (0x01), “电压” (0x02)	12.0V 电源干线的电压测量值	如果此电压不符合规格或为零, 则所有其他电源干线将出现故障 (STBY 除外)。CP3060 和 RTM 将无法工作。如果 3.3V STBY 处于打开状态, IPMC 将可以正常工作。供应: 所有 DC/DC 转换器
9	5.0V	阈值 (0x01), “电压” (0x02)	5.0V 电源干线的电压测量值	如果此电压不符合规格或为零, 则大部分其他电源干线将出现故障。CP3060 将无法工作。如果 3.3V STBY 处于打开状态, IPMC 将可以正常工作。供应: 大多数 DC/DC 转换器

表 H-1 Sun Netra CP3060 刀片服务器传感器列表 (续)

传感器编号	传感器名称	传感器类型	传感器说明	传感器超出限制时的故障状况
10	3.3V	阈值 (0x01), “电压” (0x02)	3.3V 电源干线的电压测量值	如果此电压不符合规格或为零, CP3060 上的许多组件将无法工作。因此, CP3060 将无法工作。如果 3.3V STBY 处于打开状态, IPMC 将可以正常工作。供应: 多个支持设备
11	3.3V STBY	阈值 (0x01), “电压” (0x02)	3.3V STBY 电源干线的电压测量值	如果此电压不符合规格或为零, CP3060 和 RTM 将无法工作。IPMC 也将无法工作。供应: 所有 I2C 设备、IPMC 域
12	2.5V STBY	阈值 (0x01), “电压” (0x02)	2.5V STBY 电源干线的电压测量值	如果此电压不符合规格或为零, CP3060 和 RTM 将无法工作。IPMC 也将无法工作。供应: IPMC 域
13	1.0V	阈值 (0x01), “电压” (0x02)	1.0V 电源干线的电压测量值	如果此电压不符合规格或为零, CP3060 上的许多组件将无法工作。因此, CP3060 将无法工作。如果 3.3V STBY 处于打开状态, IPMC 将可以正常工作。供应: 多个支持设备、服务处理器
14	1.2V CPU	阈值 (0x01), “电压” (0x02)	1.2V CPU 电源干线的电压测量值	如果此电压不符合规格或为零, CPU 或主机将无法工作。如果所有其他电源干线均已打开, 服务处理器和 IPMC 仍将继续工作。供应: CPU 核心
15	1.2V	阈值 (0x01), “电压” (0x02)	1.2V 电源干线的电压测量值	如果此电压不符合规格或为零, CP3060 上的许多组件将无法工作。因此, CP3060 将无法工作。如果 3.3V STBY 处于打开状态, IPMC 将可以正常工作。供应: 多个支持设备、服务处理器
16	1.5V	阈值 (0x01), “电压” (0x02)	1.5V 电源干线的电压测量值	如果此电压不符合规格或为零, CP3060 上的许多组件将无法工作。因此, CP3060 将无法工作。如果 3.3V STBY 处于打开状态, IPMC 将可以正常工作。供应: 多个支持设备

表 H-1 Sun Netra CP3060 刀片服务器传感器列表 (续)

传感器编号	传感器名称	传感器类型	传感器说明	传感器超出限制时的故障状况
17	0.9V VTTL	阈值 (0x01), “电压” (0x02)	0.9V VTTL 电源干线的电压测量值	如果此电压不符合规格或为零, CPU 或主机将无法工作。如果所有其他电源干线均已打开, 服务处理器和 IPMC 仍将继续工作。供应: DDR DIMM
18	0.9V VTTR	阈值 (0x01), “电压” (0x02)	0.9V VTTR 电源干线的电压测量值	如果此电压不符合规格或为零, CPU 或主机将无法工作。如果所有其他电源干线均已打开, 服务处理器和 IPMC 仍将继续工作。供应: DDR DIMM
19	1.8V DDR2L	阈值 (0x01), “电压” (0x02)	1.8V DDR2L 电源干线的电压测量值	如果此电压不符合规格或为零, CPU 或主机将无法工作。如果所有其他电源干线均已打开, 服务处理器和 IPMC 仍将继续工作。供应: DDR DIMM
20	1.8V DDR2R	阈值 (0x01), “电压” (0x02)	1.8V DDR2R 电源干线的电压测量值	如果此电压不符合规格或为零, CPU 或主机将无法工作。如果所有其他电源干线均已打开, 服务处理器和 IPMC 仍将继续工作。供应: DDR DIMM、多个支持设备
21	2.5V	阈值 (0x01), “电压” (0x02)	VDD 2.5V 电源干线的电压测量值	如果此电压不符合规格或为零, CP3060 上的许多组件将无法工作。因此, CP3060 将无法工作。如果 3.3V STBY 处于打开状态, IPMC 将可以正常工作。供应: 多个支持设备、服务处理器
22	1.2V STBY	阈值 (0x01), “电压” (0x02)	1.2V STBY 电源干线的电压测量值	如果此电压不符合规格或为零, CP3060 和 RTM 将无法工作。IPMC 也将无法工作。供应: 所有 I2C 设备、IPMC 域
23	AMC 12V	阈值 (0x01), “电压” (0x02)	AMC 12V 电源干线的电压测量值	如果此电压不符合规格或为零, AMC 插槽将无法工作。如果此电压为零且 CP3060 正在正常工作, 则表明 IPMC 尚未启用 AMC 插槽。

表 H-1 Sun Netra CP3060 刀片服务器传感器列表（续）

传感器编号	传感器名称	传感器类型	传感器说明	传感器超出限制时的故障状况
24	AMC 3.3V	阈值 (0x01), “电压” (0x02)	AMC 3.3V 电源干线的电压测量值	如果此电压不符合规格或为零, AMC 插槽将无法工作。如果此电压为零且 CP3060 正在正常工作, 则表明 IPMC 尚未启用 AMC 插槽
25	RTM 存在	离散 (0x6f), “实体存在” (0x25)	RTM 的存在情况	指示 RTM 是否连接到 CP3060
26	版本更改	离散 (0x6f), “已保留” (0x2b)	固件更新事件	在固件更新/冷复位后报告事件

图 H-1 Sun Netra CP3060 电压分配和 H8 传感器对应关系

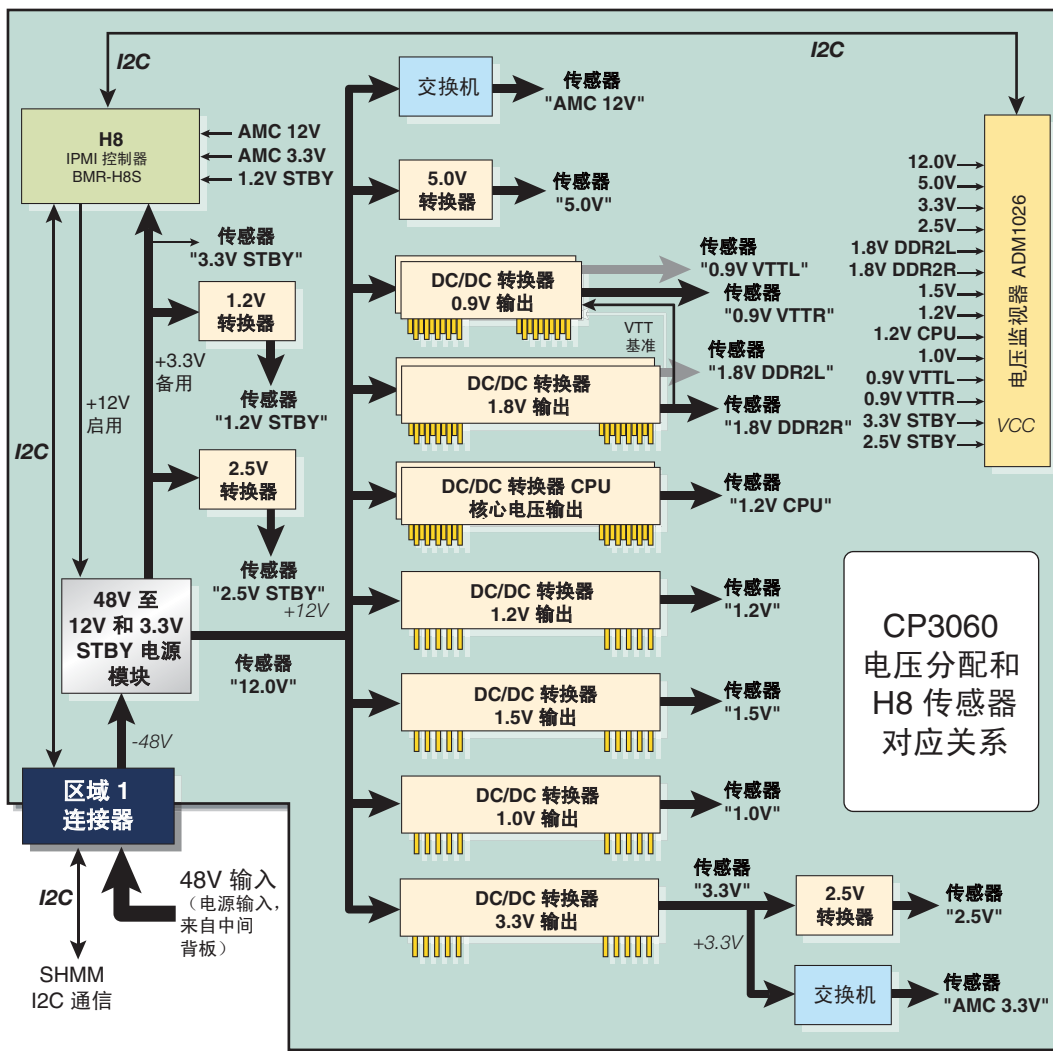
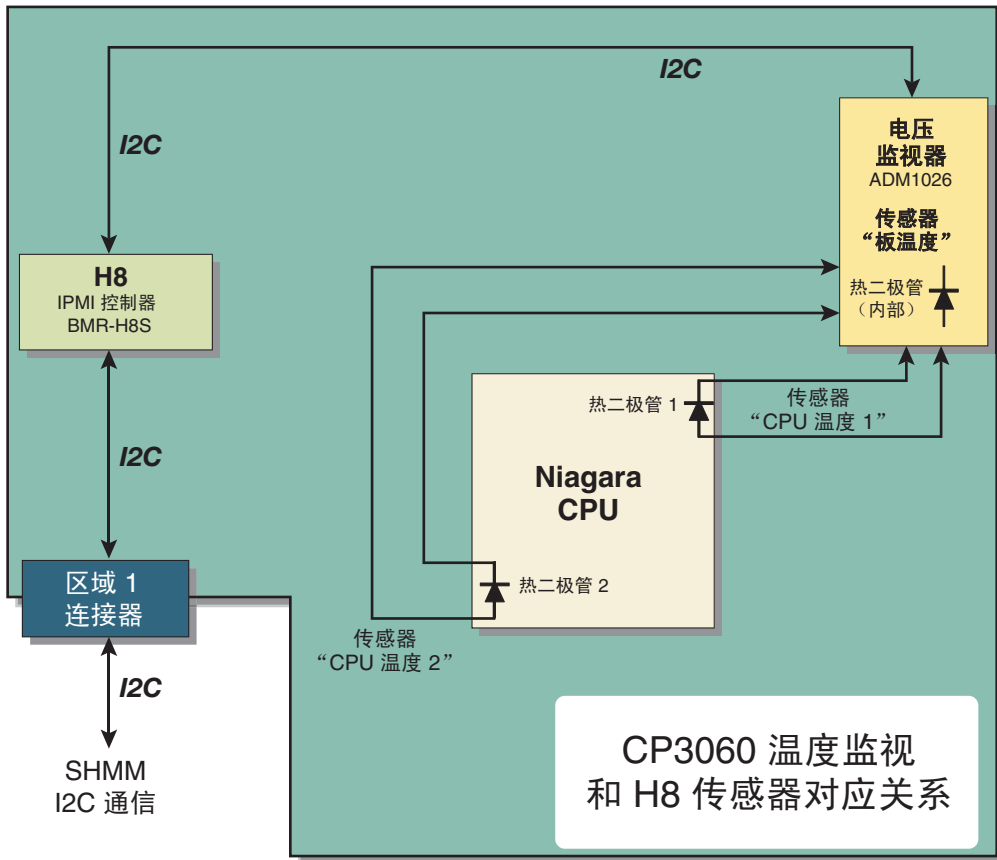


图 H-2 Sun Netra CP3060 温度监视和 H8 传感器对应关系



附录 I

Sun Netra CP3250 刀片服务器传感器 对应表和故障隔离

本附录定义了 Sun Netra CP3250 刀片服务器的传感器。

Sun Netra CP3250 刀片服务器传感器列表

传感器的编号和名称由刀片服务器 Sun Netra CP3250 上的 IPMC 处理器通过 ATCA 机箱中的 ShMM 进行报告。

表 I-1 Sun Netra CP3250 刀片服务器传感器列表

传感器编号	传感器名称	传感器类型	传感器说明	传感器超出限制时的故障状况
0	FRU 0 热交换	离散 (0x6f), “热交换” (0xf0)	CP3250 FRU 的热交换	N/A
1	热交换 AMC 5	离散 (0x6f), “热交换” (0xf0)	AMC FRU 的热交换	N/A
2	热交换 ARTM 15	离散 (0x6f), “热交换” (0xf0)	ARTM FRU 的热交换	N/A
3	IPMB 物理	离散 (0x6f), “热交换” (0xf0)	IPMB 的链路状态	未收到来自 IPMB (A 或 B) 的回复。IPMB A 或 B 总线的状态通过监视 IPMB 隔离器上的 READY 信号来报告。
4	BMC 监视程序	离散 (0x6f), “监视程序 2” (0x23)	BMC 的监视程序状态	N/A
5	12.0V	阈值 (0x01), “电压” (0x02)	12.0V 电源干线的电压测量值	如果此电压不符合规格或为零, 则所有其他电源干线将出现故障 (STBY 除外)。刀片服务器和 RTM 将无法工作。如果 3.3V STBY 处于打开状态, IPMC 将可以正常工作。供应: 所有 DC/DC 转换器
6	5.0V	阈值 (0x01), “电压” (0x02)	5.0V 电源干线的电压测量值	如果此电压不符合规格或为零, 则大部分其他电源干线将出现故障。刀片服务器将无法工作。如果 3.3V STBY 处于打开状态, IPMC 将可以正常工作。供应: 大多数 DC/DC 转换器
7	3.3V	阈值 (0x01), “电压” (0x02)	3.3V 电源干线的电压测量值	如果此电压不符合规格或为零, 刀片服务器上的许多组件将无法工作。因此, 刀片服务器将无法工作。如果 3.3V STBY 处于打开状态, IPMC 将可以正常工作。供应: 多个支持设备

表 I-1 Sun Netra CP3250 刀片服务器传感器列表 (续)

传感器编号	传感器名称	传感器类型	传感器说明	传感器超出限制时的故障状况
8	3.3V STBY	阈值 (0x01), “电压” (0x02)	3.3V STBY 电源干线的电压测量值	如果此电压不符合规格, 刀片服务器和 H8 将无法工作。此干线是刀片服务器上的管理组件 (包括所有 I2C 设备和 H8) 的电源。
9	SuperCAP 电压	阈值 (0x01), “电压” (0x02)	3.0V 电池的电压测量值 (ADM 1026, 管脚 29)	如果此电压超出限制, 则表示电池出现故障或未安装。如果未使用电池, 请忽略消息。
10	1.2V NTune	阈值 (0x01), “电压” (0x02)	1.2V 电源干线的电压测量值	如果此电压不符合规格或为零, 刀片服务器上的许多组件将无法工作。因此, 刀片服务器将无法工作。
11	CPU VTT	阈值 (0x01), “电压” (0x02)	VTT 电源干线的电压测量值	如果此电压不符合规格或为零, 刀片服务器上的许多组件将无法工作。因此, 刀片服务器将无法工作。
12	1.5V	阈值 (0x01), “电压” (0x02)	1.5V 电源干线的电压测量值	如果此电压不符合规格或为零, 刀片服务器上的许多组件将无法工作。因此, 刀片服务器将无法工作。如果 3.3V STBY 处于打开状态, IPMC 将可以正常工作。供应: CPU I/O、DIMM
13	1.8 V	阈值 (0x01), “电压” (0x02)	1.8V 电源干线的电压测量值	如果此电压不符合规格或为零, 刀片服务器上的许多组件将无法工作。因此, 刀片服务器将无法工作。
14	DDR2 VTT	阈值 (0x01), “电压” (0x02)	DDR VTT 电源干线的电压测量值	如果此电压不符合规格或为零, 刀片服务器上的许多组件将无法工作。因此, 刀片服务器将无法工作。设备: DIMM
15	1.05 V 核心	阈值 (0x01), “电压” (0x02)	1.05V 电源干线的电压测量值	如果此电压不符合规格或为零, 刀片服务器上的许多组件将无法工作。因此, 刀片服务器将无法工作。设备: CPU
16	1.5 V NTune	阈值 (0x01), “电压” (0x02)	1.5V 电源干线的电压测量值	如果此电压不符合规格或为零, 刀片服务器上的许多组件将无法工作。因此, 刀片服务器将无法工作。设备: Neptune

表 I-1 Sun Netra CP3250 刀片服务器传感器列表 (续)

传感器编号	传感器名称	传感器类型	传感器说明	传感器超出限制时的故障状况
17	VCC CPU1	阈值 (0x01), “电压” (0x02)	VCC CPU 的电压测量值	如果此电压不符合规格或为零, 刀片服务器上的许多组件将无法工作。因此, 刀片服务器将无法工作。设备: CPU
18	VCC CPU0	阈值 (0x01), “电压” (0x02)	VCC CPU 的电压测量值	如果此电压不符合规格或为零, 刀片服务器上的许多组件将无法工作。因此, 刀片服务器将无法工作。设备: CPU
19	进气口 1 温度传感器	阈值 (0x01), “温度” (0x01)	环境温度, 未使用	N/A。在未来版本中将删除。
20	进气口 3 温度传感器	阈值 (0x01), “温度” (0x01)	环境温度, 未使用	N/A。在未来版本中将删除。
21	进气口 2 温度传感器	阈值 (0x01), “温度” (0x01)	环境温度, 未使用	N/A。在未来版本中将删除。
22	MCH 温度传感器	阈值 (0x01), “温度” (0x01)	结点温度、内存控制器集线器、北桥	此传感器指示 MCH 是否超出温度范围。
23	CPU_TEMP_SK0D0	阈值 (0x01), “温度” (0x01)	CPU 结点温度、插槽 0、域 0	此传感器指示 CPU 是否超出温度范围。超过 94 摄氏度将导致刀片服务器关闭。
24	CPU_TEMP_SK0D1	阈值 (0x01), “温度” (0x01)	CPU 结点温度、插槽 0、域 1	此传感器指示 CPU 是否超出温度范围。超过 94 摄氏度将导致刀片服务器关闭。
25	CPU_TEMP_SK1D0	阈值 (0x01), “温度” (0x01)	CPU 结点温度、插槽 1、域 0	此传感器指示 CPU 是否超出温度范围。超过 94 摄氏度将导致刀片服务器关闭。
26	CPU_TEMP_SK1D1	阈值 (0x01), “温度” (0x01)	CPU 结点温度、插槽 1、域 1	此传感器指示 CPU 是否超出温度范围。超过 94 摄氏度将导致刀片服务器关闭。
27	版本更改	离散 (0x6f), “已保留” (0x2b)	固件更新事件	在固件更新/冷复位后报告事件。
28	系统事件	离散 (0x6f), “系统事件” (0x12)	系统复位事件	此传感器向 ShMM 报告 IPMC 复位事件。此传感器让 NetConsole 应用程序知道 IPMC 已执行复位操作且 NetConsole 会话必须重新启动。
29	CPU 0 存在	离散 (0x6f), “实体存在” (0x25)		指示 CPU0 是否已安装。

表 I-1 Sun Netra CP3250 刀片服务器传感器列表（续）

传感器编号	传感器名称	传感器类型	传感器说明	传感器超出限制时的故障状况
30	CPU 1 存在	离散 (0x6f), “实体存在” (0x25)		指示 CPU1 是否已安装。
31	P48V 报警	离散 (0x70), “OEM 已保留” (0xc0)	48V 电源的电压测量值	检测电源模块的 48V 电源输入 A 或 48V 电源输入 B 是否转换为 12V。
32	系统固件进度	离散 (0x6f), “系统固件进度” (0x0f)	监视系统固件进度状态	此传感器将监视固件进度。系统固件通过 IPMC 将固件进度事件发送到系统事件日志。
33	正常重新引导	离散 (0x6f), “OEM 已保留” (0xc0)	监视正常重新引导的状态	当正常重新引导计时器启动、停止或过期时, 此传感器会将事件记录到系统事件日志中。

附录 J

Sun Netra CP3260 刀片服务器传感器 对应表和故障隔离

本附录定义了 Sun Netra CP3260 刀片服务器的传感器。

Sun Netra CP3260 刀片服务器传感器列表

传感器的编号和名称由板载 Sun Netra CP3260 IPMC 处理器通过 ATCA 机箱中的 ShMM 进行报告。

表 J-1 Sun Netra CP3260 刀片服务器传感器列表

传感器编号	传感器名称	传感器类型	传感器说明	传感器超出限制时的故障状况
0	FRU 0 热交换	离散 (0x6f), “热交换” (0xf0)	CP3260 FRU 的热交换	N/A
1	ARTM 热交换	离散 (0x6f), “热交换” (0xf0)	RTM FRU 的热交换	N/A
2	IPMB 物理	离散 (0x6f), “IPMB 链路” (0xf1)	IPMB 的链路状态	未收到来自 IPMB (A 或 B) 的回复。IPMB A 或 B 总线的状态通过监视 IPMB 隔离器上的 READY 信号来报告。
3	BMC 监视程序	离散 (0x6f), “监视程序 2” (0x23)	BMC 的监视程序状态	N/A
4	CPU 温度 1	阈值 (0x01), “温度” (0x01)	CPU (Niagra 2) 的内部管芯温度	如果此温度超过 112 摄氏度, 将关闭所有电源并关闭前面板上的所有 LED 指示灯。RTM 上的蓝色 LED 指示灯将保持处于亮起状态。
5	CPU 温度 2	阈值 (0x01), “温度” (0x01)	CPU (Niagra 2) 的内部管芯温度	如果此温度超过 112 摄氏度, 将关闭所有电源并关闭前面板上的所有 LED 指示灯。RTM 上的蓝色 LED 指示灯将保持处于亮起状态。
6	板温度	阈值 (0x01), “温度” (0x01)	刀片服务器温度: ADM1026 的环境温度	如果此温度超过 88 摄氏度, 将关闭所有电源并关闭前面板上的所有 LED 指示灯。RTM 上的蓝色 LED 指示灯将保持处于亮起状态。
7	12.0V	阈值 (0x01), “电压” (0x02)	12.0V 电源干线的电压测量值	如果此电压不符合规格或为零, 则所有其他电源干线将出现故障 (STBY 除外)。CP3260 和 RTM 将无法工作。如果 3.3V STBY 处于打开状态, IPMC 将可以正常工作。供应: 所有 DC/DC 转换器

表 J-1 Sun Netra CP3260 刀片服务器传感器列表 (续)

传感器编号	传感器名称	传感器类型	传感器说明	传感器超出限制时的故障状况
8	5.0V	阈值 (0x01), “电压” (0x02)	5.0V 电源干线的电压测量值	如果此电压不符合规格或为零, 则大部分其他电源干线将出现故障。CP3260 将无法工作。如果 3.3V STBY 处于打开状态, IPMC 将可以正常工作。供应: 大多数 DC/DC 转换器
9	3.3V	阈值 (0x01), “电压” (0x02)	3.3V 电源干线的电压测量值	如果此电压不符合规格或为零, CP3260 上的许多组件将无法工作。因此, CP3260 将无法工作。如果 3.3V STBY 处于打开状态, IPMC 将可以正常工作。供应: 多个支持设备
10	3.3V STBY	阈值 (0x01), “电压” (0x02)	3.3V STBY 电源干线的电压测量值	如果此电压不符合规格或为零, CP3260 和 RTM 将无法工作。供应: 所有 I2C 设备、IPMC
11	3.0 VBAT/STBY	阈值 (0x01), “电压” (0x02)	3.0 VBAT/STBY 电源干线的电压测量值	如果此电压不符合规格或为零, 则说明刀片服务器上的电池已损坏或丢失。电池并不是 CP3260 或 RTM 正常工作所必需的。供应: 3.3V STBY
12	1.0V VDD	阈值 (0x01), “电压” (0x02)	1.0V VDD 电源干线的电压测量值	如果此电压不符合规格或为零, CP3260 上的许多组件将无法工作。因此, CP3260 将无法工作。如果 3.3V STBY 处于打开状态, IPMC 将可以正常工作。供应: 多个支持设备、服务处理器
13	1.1V CPU	阈值 (0x01), “电压” (0x02)	1.1V CPU 电源干线的电压测量值	如果此电压不符合规格或为零, CPU 或主机将无法工作。如果所有其他电源干线均已打开, 服务处理器和 IPMC 仍将继续工作。供应: CPU 核心
14	VDD 1.1V	阈值 (0x01), “电压” (0x02)	VDD 1.1V 电源干线的电压测量值	如果此电压不符合规格或为零, CP3260 上的许多组件将无法工作。因此, CP3260 将无法工作。如果 3.3V STBY 处于打开状态, IPMC 将可以正常工作。供应: 多个支持设备、服务处理器

表 J-1 Sun Netra CP3260 刀片服务器传感器列表 (续)

传感器编号	传感器名称	传感器类型	传感器说明	传感器超出限制时的故障状况
15	1.5V	阈值 (0x01), “电压” (0x02)	1.5V 电源干线的电压测量值	如果此电压不符合规格或为零, CP3260 上的许多组件将无法工作。因此, CP3260 将无法工作。如果 3.3V STBY 处于打开状态, IPMC 将可以正常工作。供应: CPU I/O、FBDIMM
16	VDD 1.8V FBDIMM	阈值 (0x01), “电压” (0x02)	VDD 1.8V 电源干线的电压测量值	如果此电压不符合规格或为零, CPU 或主机将无法工作。如果所有其他电源干线均已打开, 服务处理器和 IPMC 仍将继续工作。供应: FBDIMM
17	VDD 2.5V	阈值 (0x01), “电压” (0x02)	VDD 2.5V 电源干线的电压测量值	如果此电压不符合规格或为零, CP3260 上的许多组件将无法工作。因此, CP3260 将无法工作。如果 3.3V STBY 处于打开状态, IPMC 将可以正常工作。供应: 多个支持设备、服务处理器
18	VDD_IO 1.2V	阈值 (0x01), “电压” (0x02)	VDD_IO 1.2V 电源干线的电压测量值	如果此电压不符合规格或为零, CPU 或主机将无法工作。如果所有其他电源干线均已打开, 服务处理器和 IPMC 仍将继续工作。供应: CPU I/O
19	版本更改	离散 (0x6f), “已保留” (0x2b)	固件更新事件	IPMC 在固件更新/冷复位后报告事件。
20	P48V 报警	离散 (0x70), “OEM 已保留” (0xc0)	48V 电源的电压测量值	检测电源模块的 48V 电源输入 A 或 48V 电源输入 B 是否转换为 12V。

表 J-1 Sun Netra CP3260 刀片服务器传感器列表（续）

传感器编号	传感器名称	传感器类型	传感器说明	传感器超出限制时的故障状况
21	VDD 1.8V M0	阈值 (0x01), “电压” (0x02)	VDD 1.8V 电源干线的电压测量值。	如果此电压不符合规格或为零, CPU 或主机将无法工作。如果所有其他电源干线均已打开, 服务处理器和 IPMC 仍将继续工作。 供应: FBDIMM
22	系统固件进度	离散 (0x6f), “系统固件进度” (0x0f),	监视系统固件进度状态。	此传感器将监视固件进度。系统固件通过 IPMC 将固件进度事件发送到系统事件日志。
23	正常重新引导	离散 (0x6f), “OEM 已保留” (0xc0)	监视正常重新引导的状态。	当正常重新引导计时器启动、停止或过期时, 此传感器会将事件记录到系统事件日志中。

图 J-1 Sun Netra CP3260 电压分配和 IPMC 传感器对应关系

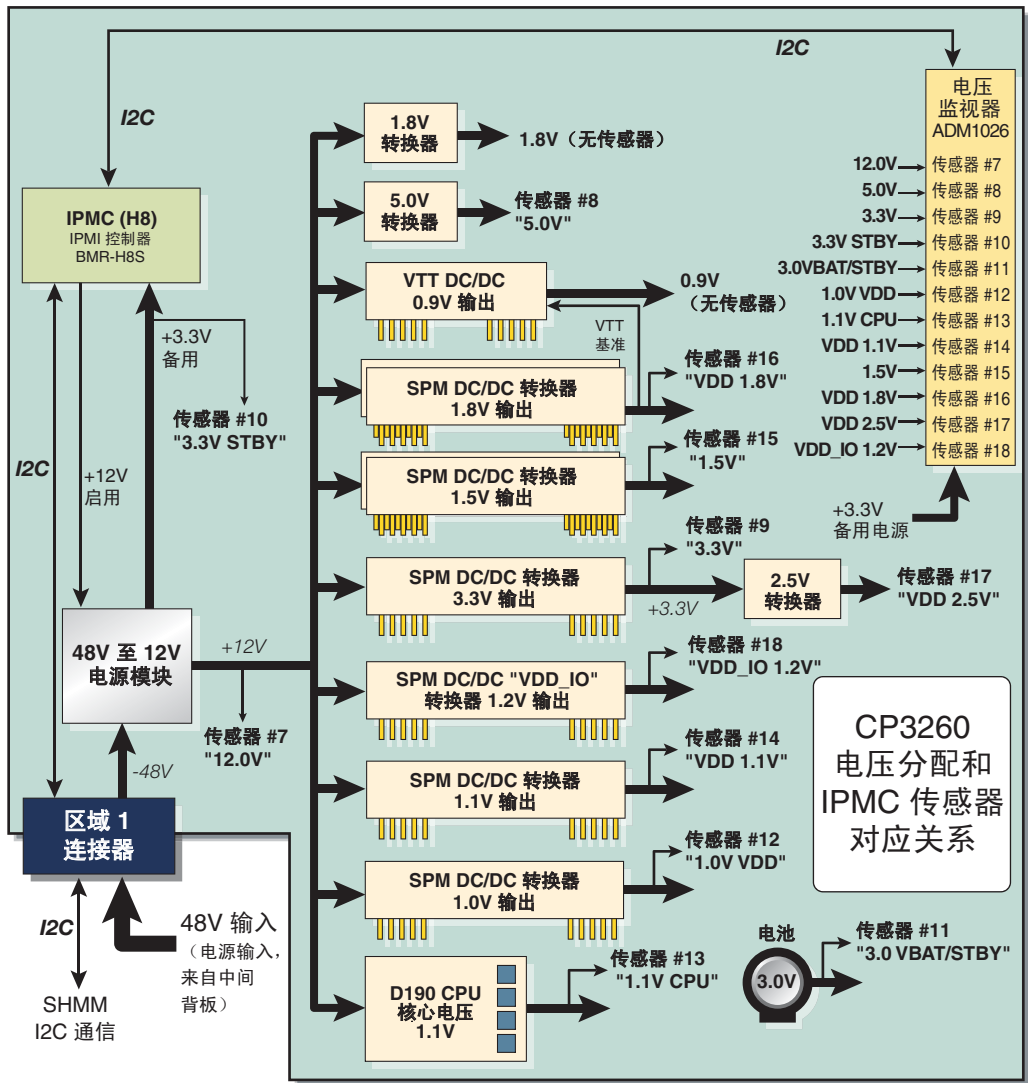
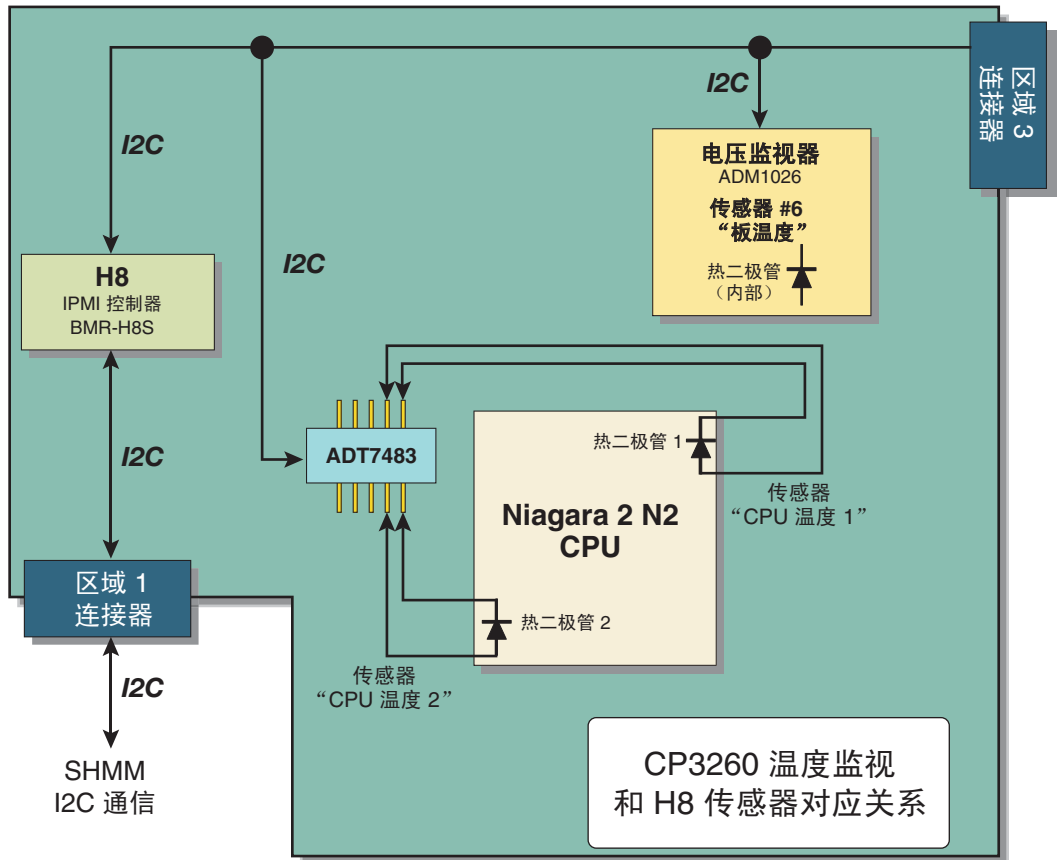


图 J-2 Sun Netra CP3260 温度监视和 H8 传感器对应关系



附录 K

Sun Netra CP3270 刀片服务器传感器 对应表和故障隔离

本附录定义了 Sun Netra CP3270 ATCA 刀片服务器的传感器。

Sun Netra CP3270 刀片服务器传感器列表

传感器的编号和名称由刀片服务器 Sun Netra CP3270 上的 IPMC 处理器通过 ATCA 机箱中的 ShMM 进行报告。

表 K-1 Sun Netra CP3270 刀片服务器传感器列表

传感器编号	传感器名称	传感器类型	传感器说明	传感器超出限制时的故障状况
0	FRU 0 热交换	离散 (0x6f), “热交换” (0xf0)	CP3270 FRU 的热交换	N/A
1	热交换 AMC 5	离散 (0x6f), “热交换” (0xf0)	AMC FRU 的热交换	N/A
2	热交换 ARTM	离散 (0x6f), “热交换” (0xf0)	ARTM FRU 的热交换	N/A
3	IPMB 物理	离散 (0x6f), “热交换” (0xf0)	IPMB 的链路状态	未收到来自 IPMB (A 或 B) 的回复。IPMB A 或 B 总线的状态通过监视 IPMB 隔离器上的 READY 信号来报告。
4	BMC 监视程序	离散 (0x6f), “监视程序 2” (0x23)	BMC 的监视程序状态	N/A
5	CPU 0 温度	阈值 (0x01), “温度” (0x01)	CPU 结点温度、插槽 0、域 0	此传感器指示 CPU 是否超出温度范围。超过 94 摄氏度将导致刀片服务器关闭。
6	CPU 1 温度	阈值 (0x01), “温度” (0x01)	CPU 结点温度、插槽 0、域 1	此传感器指示 CPU 是否超出温度范围。超过 94 摄氏度将导致刀片服务器关闭。
7	Vbat	阈值, 电压	CMOS 电池电压	此传感器指示电池电量不足, 需要更换电池。
8	P3V3_STBY	阈值 (0x01), “电压” (0x02)	3.3V STBY 电源干线的电压测量值	如果此电压不符合规格, 刀片服务器和 H8 将无法工作。此干线是刀片服务器上的管理组件 (包括所有 I2C 设备和 H8) 的电源。
9	P12V	阈值 (0x01), “电压” (0x02)	12.0V 电源干线的电压测量值	如果此电压不符合规格或为零, 则所有其他电源干线将出现故障 (STBY 除外)。刀片服务器和 RTM 将无法工作。如果 3.3V STBY 处于打开状态, IPMC 将可以正常工作。供应: 所有 DC/DC 转换器

表 K-1 Sun Netra CP3270 刀片服务器传感器列表 (续)

传感器编号	传感器名称	传感器类型	传感器说明	传感器超出限制时的故障状况
10	P5V	阈值 (0x01), “电压” (0x02)	5.0V 电源干线的电压测量值	如果此电压不符合规格或为零, 则大部分其他电源干线将出现故障。刀片服务器将无法工作。如果 3.3V STBY 处于打开状态, IPMC 将可以正常工作。供应: 大多数 DC/DC 转换器
11	P3V3	阈值 (0x01), “电压” (0x02)	3.3V 电源干线的电压测量值	如果此电压不符合规格或为零, 刀片服务器上的许多组件将无法工作。因此, 刀片服务器将无法工作。如果 3.3V STBY 处于打开状态, IPMC 将可以正常工作。供应: 多个支持设备
12	P1V05_PCH	阈值 (0x01), “电压” (0x02)	VCC PCM 的电压测量值	如果此电压不符合规格或为零, 刀片服务器上的许多组件将无法工作。因此, 刀片服务器将无法工作。设备: PCM
13	P1V5_DDR3_CPU0	阈值 (0x01), “电压” (0x02)	VCC 内存的电压测量值	如果此电压不符合规格或为零, 刀片服务器上的许多组件将无法工作。因此, 刀片服务器将无法工作。设备: 内存
14	P1V5_DDR3_CPU1	阈值 (0x01), “电压” (0x02)	VCC 内存的电压测量值	如果此电压不符合规格或为零, 刀片服务器上的许多组件将无法工作。因此, 刀片服务器将无法工作。设备: 内存
15	P0V75_DDR3_CPU0	阈值 (0x01), “电压” (0x02)	VCC 内存的电压测量值	如果此电压不符合规格或为零, 刀片服务器上的许多组件将无法工作。因此, 刀片服务器将无法工作。设备: 内存
16	P0V75_DDR3_CPU1	阈值 (0x01), “电压” (0x02)	VCC 内存的电压测量值。	如果此电压不符合规格或为零, 刀片服务器上的许多组件将无法工作。因此, 刀片服务器将无法工作。设备: 内存

表 K-1 Sun Netra CP3270 刀片服务器传感器列表 (续)

传感器编号	传感器名称	传感器类型	传感器说明	传感器超出限制时的故障状况
17	VTT CPU0	阈值 (0x01), “电压” (0x02)	VTT 电源干线的电压测量值。	如果此电压为 1.260 (不符合规格) 或为零, 刀片服务器上的许多组件将无法工作。因此, 刀片服务器将无法工作。
18	VTT CPU1	阈值 (0x01), “电压” (0x02)	VTT 电源干线的电压测量值。	如果此电压为 1.260 (不符合规格) 或为零, 刀片服务器上的许多组件将无法工作。因此, 刀片服务器将无法工作。
19	VCCP_CPU0	阈值 (0x01), “电压” (0x02)	VCC CPU 的电压测量值。	如果此电压不符合规格或为零, 刀片服务器上的许多组件将无法工作。因此, 刀片服务器将无法工作。设备: CPU
20	VCCP_CPU1	阈值 (0x01), “电压” (0x02)	VCC CPU 的电压测量值。	如果此电压不符合规格或为零, 刀片服务器上的许多组件将无法工作。因此, 刀片服务器将无法工作。设备: CPU
21	版本更改	离散 (0x6f), “已保留” (0x2b)	固件更新事件	在固件更新/冷复位后报告事件
22	系统事件	离散 (0x6f), “系统事件” (0x12)	系统复位事件	此传感器向 ShMM 报告 IPMC 复位事件。此传感器让 NetConsole 应用程序知道 IPMC 已执行复位操作且 NetConsole 会话必须重新启动。
23	CPU 0 存在	离散 (0x6f), “实体存在” (0x25)		指示 CPU0 是否已安装。
24	CPU 1 存在	离散 (0x6f), “实体存在” (0x25)		指示 CPU1 是否已安装。
25	P48V 报警	离散 (0x70), “OEM 已保留” (0xc0)	48V 电源的电压测量值	检测电源模块的 48V 电源输入 A 或 48V 电源输入 B 是否转换为 12V。

表 K-1 Sun Netra CP3270 刀片服务器传感器列表（续）

传感器编号	传感器名称	传感器类型	传感器说明	传感器超出限制时的故障状况
26	系统固件进度	离散 (0x6f), “系统固件进度” (0x0f),	监视系统固件进度状态。	此传感器将监视固件进度。系统固件通过 IPMC 将固件进度事件发送到系统事件日志。
27	正常重新引导	离散 (0x6f) “OEM 已保留” (0xc0)	监视正常重新引导的状态。	当正常重新引导计时器启动、停止或过期时, 此传感器会将事件记录到系统事件日志中。
28	极限温度	离散 (0x6f) “OEM 已保留” (0xc0)	监视 CPU 的极限温度状态。	此传感器跟踪 CPLD 中的 Therm Trip (极限温度) 位。如果 CPU 产生极限温度, CPLD 将关闭电源并在 CPLD 中设置 Therm Tip (极限温度) 位 (偏移 2, 位 0)。IPMC 必须了解有关极限温度电源关闭的情况, 将刀片服务器置于 M1 状态, 并生成相应的事件。

附录 L

Netra SPARC T3-1BA 刀片服务器 传感器对应表和故障隔离

本附录定义了 Netra SPARC T3-1BA 刀片服务器的传感器，该刀片服务器是继 Sun Netra CP3260 ATCA 刀片服务器后的下一代刀片。

Netra SPARC T3-1BA 刀片服务器传感器列表

传感器的编号和名称由板载 IPMC 处理器通过 ATCA 机箱中的 ShMM 进行报告。

表 L-1 Netra SPARC T3-1BA 刀片服务器传感器列表

传感器编号	传感器名称	传感器类型	传感器说明	传感器超出限制时的故障状况
0	FRU 0 热交换	离散 (0x6f), “热交换” (0xf0)	刀片服务器 FRU 的热交换	N/A
1	ARTM 热交换	离散 (0x6f), “热交换” (0xf0)	RTM FRU 的热交换	N/A
2	IPMB 物理	离散 (0x6f), “IPMB 链路” (0xf1)	IPMB 的链路状态	未收到来自 IPMB (A 或 B) 的回复。IPMB A 或 B 总线的状态通过监视 IPMB 隔离器上的 READY 信号来报告。
3	BMC 监视程序	离散 (0x6f), “监视程序 2” (0x23)	BMC 的监视程序状态	N/A
4	CPU 0 温度	阈值 (0x01), “温度” (0x01)	CPU 的内部管芯温度	如果此温度超过 110 摄氏度, 将关闭所有电源并关闭前面板上的所有 LED 指示灯。RTM 上的蓝色 LED 指示灯将保持处于亮起状态。
5	CPU 1 温度	阈值 (0x01), “温度” (0x01)	CPU 的内部管芯温度	如果此温度超过 110 摄氏度, 将关闭所有电源并关闭前面板上的所有 LED 指示灯。RTM 上的蓝色 LED 指示灯将保持处于亮起状态。
6	Vbat	阈值 (0x01), “电压” (0x02)	3.0 VBAT/STBY 电源干线的电压测量值	如果此电压不符合规格或为零, 则说明刀片服务器上的电池已损坏或丢失。电池并不是刀片服务器或 ARTM 正常工作所必需的。供应: 3.3V STBY
7	3.3 V STBY	阈值 (0x01), “电压” (0x02)	3.3V STBY 电源干线的电压测量值	如果此电压不符合规格或为零, 刀片服务器和 RTM 将无法工作。供应: 所有 I2C 设备、IPMC

表 L-1 Netra SPARC T3-1BA 刀片服务器传感器列表（续）

传感器编号	传感器名称	传感器类型	传感器说明	传感器超出限制时的故障状况
8	12.0V	阈值 (0x01), “电压” (0x02)	12.0V 电源干线的电压测量值	如果此电压不符合规格或为零, 则所有其他电源干线将出现故障 (STBY 除外)。刀片服务器和 RTM 将无法工作。如果 3.3V STBY 处于打开状态, IPMC 将可以正常工作。供应: 所有 DC/DC 转换器
9	5.0V	阈值 (0x01), “电压” (0x02)	5.0V 电源干线的电压测量值	如果此电压不符合规格或为零, 则大部分其他电源干线将出现故障。刀片服务器将无法工作。如果 3.3V STBY 处于打开状态, IPMC 将可以正常工作。供应: 大多数 DC/DC 转换器
10	3.3V 主电源	阈值 (0x01), “电压” (0x02)	3.3V 电源干线的电压测量值	如果此电压不符合规格或为零, 刀片服务器上的许多组件将无法工作。因此, 刀片服务器将无法工作。如果 3.3V STBY 处于打开状态, IPMC 将可以正常工作。供应: 多个支持设备
11	B0 VDD 1.1V	阈值 (0x01), “电压” (0x02)	VDD 1.1V 电源干线的电压测量值	如果此电压不符合规格或为零, 刀片服务器上的许多组件将无法工作。因此, 刀片服务器将无法工作。如果 3.3V STBY 处于打开状态, IPMC 将可以正常工作。供应: 多个支持设备、服务处理器
12	B1 VDD 1.1V	阈值 (0x01), “电压” (0x02)	VDD 1.1V 电源干线的电压测量值	如果此电压不符合规格或为零, 刀片服务器上的许多组件将无法工作。因此, 刀片服务器将无法工作。如果 3.3V STBY 处于打开状态, IPMC 将可以正常工作。供应: 多个支持设备、服务处理器

表 L-1 Netra SPARC T3-1BA 刀片服务器传感器列表（续）

传感器编号	传感器名称	传感器类型	传感器说明	传感器超出限制时的故障状况
13	B2 VDD 1.1V	阈值 (0x01), “电压” (0x02)	VDD 1.1V 电源干线的电压测量值	如果此电压不符合规格或为零, 刀片服务器上的许多组件将无法工作。因此, 刀片服务器将无法工作。如果 3.3V STBY 处于打开状态, IPMC 将可以正常工作。供应: 多个支持设备、服务处理器
14	B3 VDD 1.1V	阈值 (0x01), “电压” (0x02)	VDD 1.1V 电源干线的电压测量值	如果此电压不符合规格或为零, 刀片服务器上的许多组件将无法工作。因此, 刀片服务器将无法工作。如果 3.3V STBY 处于打开状态, IPMC 将可以正常工作。供应: 多个支持设备、服务处理器
15	RF CPU VDD 1.5V	阈值 (0x01), “电压” (0x02)	VDD 1.5V 电源干线的电压测量值	如果此电压不符合规格或为零, 刀片服务器上的许多组件将无法工作。因此, 刀片服务器将无法工作。如果 3.3V STBY 处于打开状态, IPMC 将可以正常工作。供应: 多个支持设备、服务处理器
16	VTT MO 0.75V	阈值 (0x01), “电压” (0x02)	电压	
17	VDD MO 1.5V	阈值 (0x01), “电压” (0x02)	VDD 1.5V 电源干线的电压测量值	如果此电压不符合规格或为零, 刀片服务器上的许多组件将无法工作。因此, 刀片服务器将无法工作。如果 3.3V STBY 处于打开状态, IPMC 将可以正常工作。供应: 多个支持设备、服务处理器
18	VDD 1.5V	阈值 (0x01), “电压” (0x02)	VDD 1.5V 电源干线的电压测量值	如果此电压不符合规格或为零, 刀片服务器上的许多组件将无法工作。因此, 刀片服务器将无法工作。如果 3.3V STBY 处于打开状态, IPMC 将可以正常工作。供应: 多个支持设备、服务处理器

表 L-1 Netra SPARC T3-1BA 刀片服务器传感器列表（续）

传感器编号	传感器名称	传感器类型	传感器说明	传感器超出限制时的故障状况
19	VDD 2.5V	阈值 (0x01), “电压” (0x02)	VDD 2.5V 电源干线的电压测量值	如果此电压不符合规格或为零, 刀片服务器上的许多组件将无法工作。因此, 刀片服务器将无法工作。如果 3.3V STBY 处于打开状态, IPMC 将可以正常工作。供应: 多个支持设备、服务处理器
20	版本更改	离散 (0x6f), “已保留” (0x2b)	固件更新事件	IPMC 在固件更新/冷复位后报告事件。
21	系统事件	离散 (0x6f), “系统事件” (0x12)	跟踪 IPMC 复位	每次 IPMC 进行复位时, 都会向机框管理器发送一个事件。此传感器支持基于 SOL 的 NetConsole 使用。
22	P48V 报警	离散 (0x6f), “系统事件” (0x12)	监视 48V 干线	<p>监视 48V 干线并通过状态位报告状态。IPMC 将直接读取电源模块的电压寄存器值以确定电源干线是否存在。如果电压低于 38 伏, 则认为干线不存在。</p> <ul style="list-style-type: none"> • 0x1 (设置位 0), 两个干线均不存在。 • 0x2 (设置位 1), 仅干线 A 存在。 • 0x4 (设置位 2), 仅干线 B 存在。 • 0x8 (设置位 3), 干线 A 和 B 均存在。
23	系统固件进度	离散 (0x6f), “系统固件进度” (0x0f),	监视系统固件进度状态	此传感器将监视固件进度。系统固件通过 IPMC 将固件进度事件发送到系统事件日志。

表 L-1 Netra SPARC T3-1BA 刀片服务器传感器列表（续）

传感器编号	传感器名称	传感器类型	传感器说明	传感器超出限制时的故障状况
24	正常重新引导	离散 (0x6f), “OEM 已保留” (0xc0)	监视正常重新引导的状态	当正常重新引导计时器启动、停止或过期时, 此传感器会将事件记录到系统事件日志中。
25	插槽电源	阈值 (0x01), “电压” (0x02)	跟踪插槽正在使用的电源	此传感器仅供参考之用, 不生成任何事件。此传感器的额定值范围是 5 瓦至大约 300 瓦。该额定值设置为 150 瓦。这指示的并不是通常使用的值。电源使用情况将取决于刀片的状态和运行的 OS 的状态。
26	极限温度	离散 (0x6f), “系统事件” (0x12)	报告不可恢复 (UNC) 上限阈值违规事件	如果达到 UNC 阈值, 此传感器将发送事件通知。

附录 M

Sun Netra CP32x0 ARTM 传感器 对应表和故障隔离

本附录定义了 Sun Netra CP32x0 ARTM 的传感器。

注 – ARTM 传感器编号的变更取决于插入的节点板以及板的配置（安装的 AMC 卡的数量和类型）。

可以从以下位置获得相关文档：

<http://docs.sun.com/app/docs/prod/cp32x0.sas?l=en#hic>

Sun Netra CP32x0 ARTM 传感器列表

传感器的编号和名称由板载 Sun Netra CP32x0 IPMC 处理器通过 ATCA 机箱中的 ShMM 进行报告。

表 M-1 Sun Netra CP32x0 ARTM-HD 传感器列表

传感器名称	传感器类型	传感器说明	传感器超出限制时的故障状况
ARTM 3V3STBY	阈值 (0x01), “电压” (0x02)	RTM 上 3.3V STBY 电源干线的电压测量值	如果此电压不符合规格或为零, RTM 将无法工作且 RTM 上的所有 LED 指示灯将不会处于亮起状态。如果此电压为零且 CP3260 正在正常工作, 则表明 IPMC 尚未启用 RTM。
ARTM 3V3MAIN	阈值 (0x01), “电压” (0x02)	RTM 上 3.3V 电源干线的电压测量值	如果此电压不符合规格或为零, RTM 将无法工作。如果此电压为零且 CP3260 正在正常工作, 这可能表明 IPMC 尚未启用 RTM。
ARTM 12V	阈值 (0x01), “电压” (0x02)	RTM 上 12.0V 电源干线的电压测量值	如果此电压不符合规格或为零, RTM 将无法工作。如果此电压为零且 CP3260 正在正常工作, 这可能表明 IPMC 尚未启用 RTM。
ARTM 5V	阈值 (0x01), “电压” (0x02)	RTM 上 5.0V 电源干线的电压测量值	如果此电压不符合规格或为零, RTM 将无法工作。如果此电压为零且 CP3260 正在正常工作, 这可能表明 IPMC 尚未启用 RTM。
ARTM 1V2	阈值 (0x01), “电压” (0x02)	RTM 上 1.2V 电源干线的电压测量值	如果此电压不符合规格或为零, RTM 将无法工作。如果此电压为零且 CP3260 正在正常工作, 这可能表明 IPMC 尚未启用 RTM。
ARTM TEMP-AIR	阈值 (0x01), “温度” (0x01)	RTM 环境温度	未设置阈值, 因此未采取任何操作。仅显示报告的温度。
ARTM TEMP-LSI	阈值 (0x01), “温度” (0x01)	RTM 上 LSI 芯片的内部管芯温度	未设置阈值, 因此未采取任何操作。仅显示报告的温度。
ARTM TEMP-ADM	阈值 (0x01), “温度” (0x01)	RTM 板温度: ADM1026 的环境温度	未设置阈值, 因此未采取任何操作。仅显示报告的温度。

注 - 下图中 ARTM 传感器编号的变更取决于插入的节点板以及板的配置 (安装的 AMC 卡的数量和类型)。

图 M-1 Sun Netra CP32x0 ARTM-HD 电压分配和 IPMC 传感器对应关系

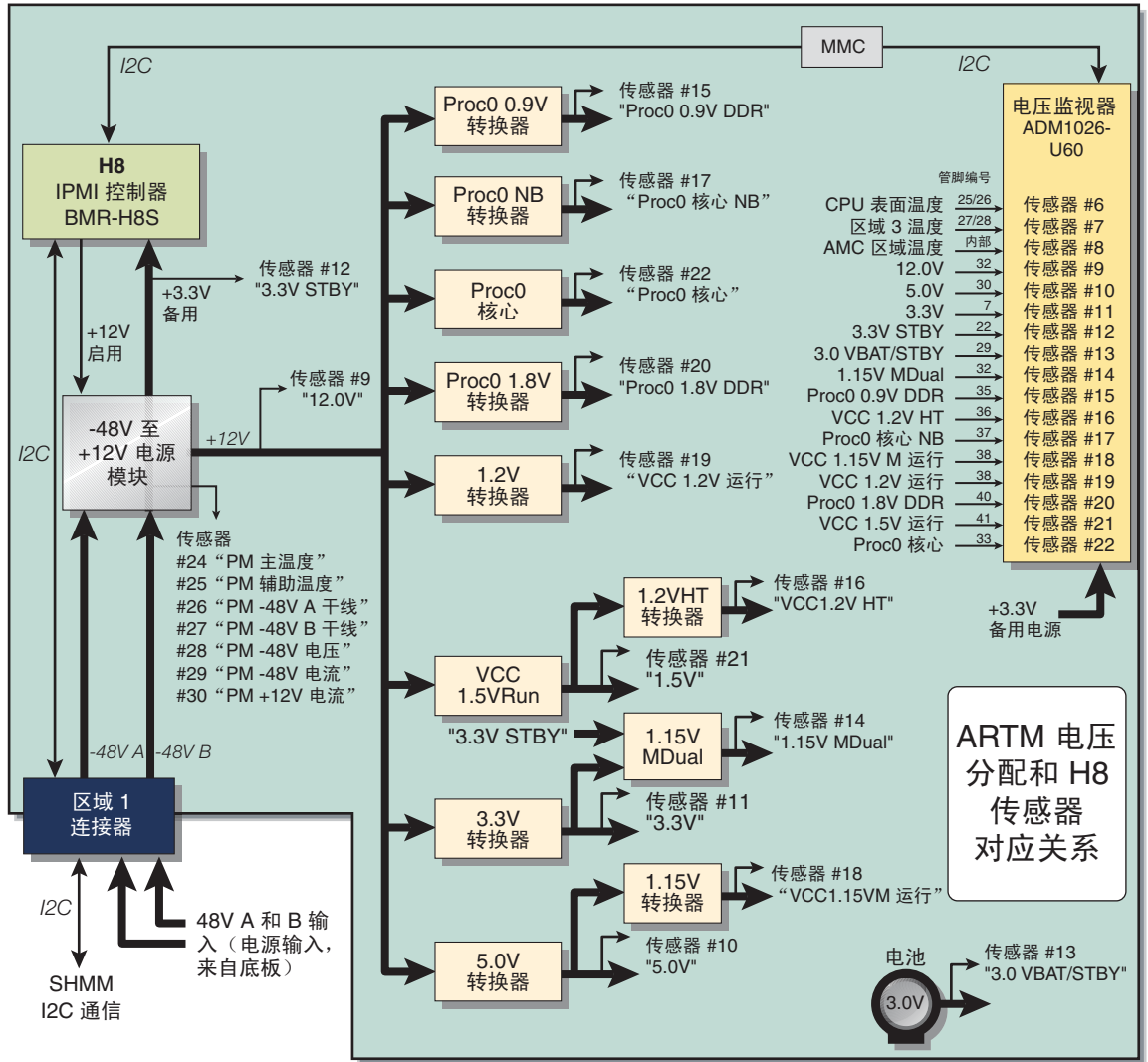


表 M-2 板的传感器编号转换

节点板	传感器编号
Sun Netra CP3220 刀片服务器, 未配置 AMC	N = 32
Sun Netra CP3220 刀片服务器, 配置有 AMC	N = 32 + AMC 传感器数量 (总数量因供应商而异)
Sun Netra CP3260 刀片服务器	N = 18

图 M-2 Sun Netra CP32x0 ARTM-HD 温度监视和 H8 传感器对应关系

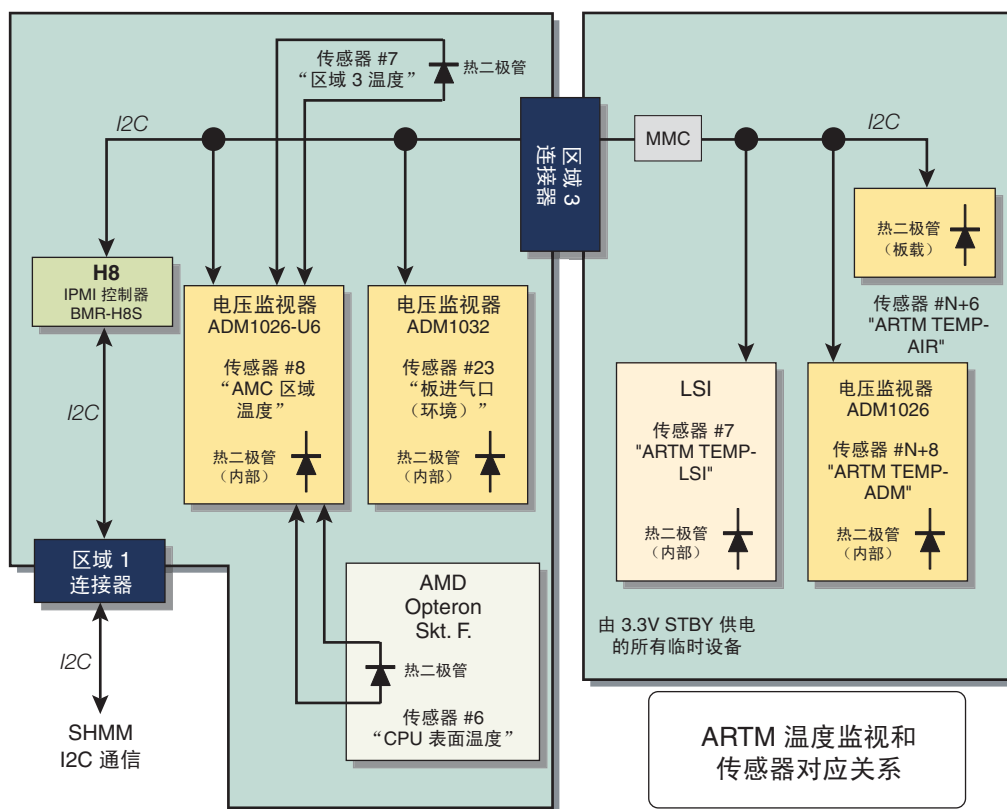


表 M-3 板的传感器编号转换

节点板	传感器编号
Sun Netra CP3220 刀片服务器， 未配置 AMC	$N = 32$
Sun Netra CP3220 刀片服务器， 配置有 AMC	$N = 32 + \text{AMC 传感器数量}$ （总数量因供应商而异）
Sun Netra CP3260 刀片服务器	$N = 18$

词汇表

熟悉以下术语和首字母缩略词对于管理 Oracle 的 Sun Netra CT900 服务器非常有用。

A

ATCA (Advanced Telecom Computing Architecture, 高级电信计算体系结构) 也称为 AdvancedTCA。适用于下一代电信级通信设备的一系列行业标准规范。AdvancedTCA 融合了高速互连技术、下一代处理器以及改进的可靠性、可管理性和可维护性等方面的最新趋势, 通过实行标准化, 刀片 (板) 和机箱 (机框) 的外形规格得以优化, 从而可实现以最低的成本进行通信。

B

**backup shelf
management card**

(备用机框管理卡) 任何能够为机框管理器功能提供支持的机框管理卡。

Base channel

(基本通道) 基接口内的物理连接, 最多由四个差分信号对组成。每个基本通道都是基接口内插槽对插槽连接的端点。

Base switch

(基本交换机) 一种支持基接口的交换机。基本交换机可为机框中安装的所有节点板提供 10/100/1000BASE-T 包交换服务。在 Sun Netra CT900 服务器中, 基本交换机位于机框的物理插槽 7 和 8 (逻辑插槽 1 和 2) 中, 支持与所有节点插槽和板的连接。支持交换接口和基接口的板也称为“交换机”。

Base interface

(基接口) 一种接口，用于支持机框中节点板和交换机之间的 10/100BASE-T 或 1000BASE-T 连接。需要具备中间背板以支持基接口，中间背板为所有节点板插槽与每个交换机插槽之间的四个差分信号对提供路由（在 Sun Netra CT900 服务器中，基本交换机插槽为物理插槽 7 和 8，逻辑插槽 1 和 2）。

D

data transport interface

(数据传输接口) 点对点接口和在总线上传输的信号的集合，用于在交换机和节点板上的有效载荷间提供互连。

Dual Star topology

(双星型拓扑) 一种互连光纤网络拓扑结构，在这种拓扑结构中，两种交换机资源为网络中的所有端点提供冗余连接。一对交换机在节点板之间提供冗余互连。

E

Electronic Keying or E-Keying

(电子钥控或 E 钥控) 一种协议，用于描述基接口、交换接口、更新通道接口与前板的同步时钟连接之间的兼容性。

ETSI European Telecommunications Standards Institute（欧洲电信标准协会）。

F

Fabric channel

(光纤网络通道) 一条光纤网络通道由两排信号对组成（每条通道共有八个信号对）。因此，每个连接器最多支持五个通道用于板与板之间的连接。还可以将通道看作由四个双端口组成。

Fabric interface

(交换接口) 一种区域 2 接口，可以为每个板或插槽提供 15 个连接，每个连接最多包含 8 个差分信号对（通道），这些信号对支持与最多 15 个其他插槽或板进行连接。中间背板可以支持多种配置（包括全网格拓扑和双星型拓扑）中的交换接口。可以将支持交换接口的板配置为光纤网络节点板、光纤网络交换机或启用了网格的板。交换接口的板实现由 PICMG 3.x 补充规范进行定义。

field-replaceable unit, FRU

(现场可更换单元) 从维修的角度来看，它是服务器中最小的不可拆分的单元。例如，磁盘驱动器、I/O 卡和电源输入模块就是 FRU。请注意，安装有各种卡和其他组件的服务器不是 FRU。但是，一台空服务器则属于 FRU。

frame (框架) 可以包含一个或多个机框的物理或逻辑实体。也称为机架，或者机箱（如果是封闭式的）。

front board (前板) 一种板，符合 PICMG 3.0 机械规格（8U x 280 毫米），包括一个 PCB 和一个面板。前板与区域 1 和区域 2 中间背板连接器相连接。前板可以（可选）与区域 3 中间背板连接器相连接，或直接与后部转换模块连接器相连接。前板安装在机框的前面。

Full channel

(全通道) 一种光纤网络通道连接，进行连接时会使用端点间的全部八个差分信号对。

Full Mesh topology

(全网格拓扑) 一种全网格配置，交换接口中可以支持这种配置，以便在机框中的每对插槽之间提供一个专用连接通道。配置了全网格的中间背板能够支持启用了网格的板或采用双星型配置进行安装的交换机和节点板。

H

hot-swap (热交换) 在不中断系统运行的情况下对外设部件或其他组件进行连接和断开连接。此功能的设计既考虑到了硬件，也考虑到了软件。

I

I²C 内部集成电路总线。用作当前 IPMB 基础的多主控、两线串行总线。

IPMB (Intelligent Platform Management Bus, 智能平台管理总线) 智能平台管理总线通信协议规范中描述的最低级别的硬件管理总线。

IPMB-0 hub

(IPMB-0 集线器) 一种集线器设备，可以为系统中的各种 FRU 提供多个辐射型 IPMB-0 链路。例如，IPMB-0 集线器可存在于具有辐射型 IPMB-0 链路的 ShMC 中。

IPMB-0 link

(IPMB-0 链路) 在辐射型拓扑中，为 IPMB-0 集线器和一个 FRU 之间的物理 IPMB-0 段。IPMB-0 集线器上的每个 IPMB-0 链路通常都与一个单独的 IPMB-0 传感器相关联。IPMB-0 链路也可以与总线型拓扑中的多个 FRU 相连。

IPM controller, IPMC

(IPM 控制器) FRU 中与 ATCA IPMB-0 进行接口的部分, 显示该 FRU 以及附属于该 FRU 的任何设备。

IPMI (Intelligent Platform Management Interface, 智能平台管理接口) 一种规范和机制, 用于对计算机系统的元素进行库存管理、监视、日志记录和控制。如智能平台管理接口规范中所定义。

L

logic ground

(逻辑接地) 一种在板和中间背板上使用的机框范围的电气网络, 作为板间传输的逻辑电平信号的基准和回路。

M

Mesh Enabled board

(启用了网格的板) 一种板, 用于提供与中间背板中所有其他板的连接。启用了网格的板支持交换接口, 也可以支持基接口。启用了网格的板可以使用 2 至 15 个交换接口通道 (通常使用所有 15 个通道) 来支持与机框中所有其他板进行直接连接。所支持的通道数将控制可在机框中连接的板的最大数目。可以将不使用基接口的启用了网格的板安装在编号最小的可用逻辑插槽中。支持基接口的启用了网格的板可以是基本交换机, 在这种情况下, 它们可以支持基本通道 1 和 2, 并可以安装到逻辑插槽 3 到 16 中。支持基接口的板仅使用基本通道 1 和 2 来支持 10/100/1000BASE-T 以太网。

midplane

(中间背板) 在功能上等同于底板。中间背板固定在服务器的背面。CPU 卡、I/O 卡和存储设备从前部连接至中间背板, 后部转换模块从后部连接至中间背板。

N

NEBS (Network Equipment/Building System, 网络设备/构建系统) 一组针对安装在美国电信控制室的设备的要求。这些要求包含人身安全、财产保护以及运行持续性。NEBS 测试包括使设备接受各种振动应力、防火以及其他环境和质量等衡量标准方面的测试。有三个级别的 NEBS 法规遵从性, 每一个级别都是前一级别的扩展。NEBS 级别 3 是最高级别, 证明某个设备可以安全地部署在“极端环境”中。电信中心局被认为是一个极端环境。

NEBS 标准由 Telcordia Technologies, Inc. (以前称为 Bellcore) 来维护。

- node board**（节点板） 一种板，专用于在星型拓扑中间背板中与中间背板内的交换机建立连接。节点板可以支持基接口和/或交换接口。支持交换接口的板使用光纤网络通道 1 和 2。支持基接口的板仅使用基本通道 1 和 2 来支持 10/100/1000BASE-T 以太网。
- node slot**（节点插槽） 中间背板中仅支持节点板的插槽。节点插槽不能支持交换机，因此节点板永远不能占用逻辑插槽 1 和 2。节点插槽仅适用于旨在支持星型拓扑的中间背板。节点插槽既支持基接口也支持交换接口。通常，一个节点插槽支持两个或四个光纤网络通道以及基本通道 1 和 2。每两个通道节点插槽分别与逻辑插槽 1 和 2 建立连接。四个通道节点插槽分别与逻辑插槽 1、2、3 和 4 建立连接。

P

PCI（Peripheral Component Interconnect，外设部件互连）一种将外设部件连接至计算机的标准。它的运行频率为 0 到 33 MHz，通过 124 管脚连接器一次传输 32 位数据，或者运行频率为 0 到 66 MHz，通过 188 管脚连接器传输 64 位数据。地址是以一个循环后跟一个数据字（在突发模式中为多个数据字）的方式发送的。

从技术上而言，PCI 是一种同步总线。其中包含的缓冲区可以解除 CPU 与相对较慢的外设部件之间的耦合，并允许它们以异步方式运行。您可以在板上拥有本地 PCI 总线，或者插入符合 PCI 规格的 PCI 卡。这不是异步的，因为所有设备均基于一个通用时钟运行。

physical address

（物理地址） 定义 FRU 物理插槽位置的地址。物理地址由位置类型和位置编号组成。

PICMG（PCI Industrial Computer Manufacturers Group，PCI 工业计算机制造商协会）由若干公司组成的协会，该协会负责制定电信和工业计算应用方面的开放式规范，其中包括 CompactPCI 标准。

R

rear-access（后部接入） Sun Netra CT900 服务器的一种配置选项，在这种配置中，所有电缆都从机框背部引出。

rear transition module

（后部转换模块） 一种仅在后部接入式 Sun Netra CT900 服务器中使用的卡，可以将连接器扩展到机框的背面。

Reliability, Availability, Serviceability, RAS

（可靠性、可用性、可维护性）

一种硬件和软件特性，用于实现或改进服务器的可靠性、可用性和可维护性。

S

- shelf** (机框) 若干组件的集合, 包括中间背板、前板、冷却设备、后部转换模块和电源输入模块。机框以前称为机箱。
- shelf address**
(机框地址) 一个长度和格式均可变的描述符, 最大长度为 20 个字节, 用于为管理域中的每个机框提供唯一标识符。
- shelf ground**
(机框接地) 一个连接至框架的安全接地回路, 可供所有板使用。
- Shelf Manager**
(机框管理器) 系统中的实体, 负责管理 AdvancedTCA 机框中的电源、冷却和互连 (通过电子钥控)。机框管理器还会在系统管理器接口与 IPMB-0 之间传递消息, 为系统信息库提供接口, 并对事件消息做出响应。可以将机框管理器部分或整体部署到 ShMC 或系统管理器硬件上。
- ShMC** (Shelf Management Controller, 机框管理控制器) 一种 IPMC, 也能够支持那些需要机框管理器的功能。
- SNMP** Simple Network Management Protocol (简单网络管理协议)。
- star topology**
(星型拓扑) 具有一个或多个集线器插槽 (用于在支持的节点插槽间建立连接) 的中间背板拓扑。
- switch** (交换机) 一种板, 专用于在星型拓扑中间背板中为中间背板内的许多节点板提供连接。交换机可以支持基接口和/或交换接口。使用交换接口的板通常会向所有 15 个可用光纤网络通道提供交换资源。支持基接口的交换机安装到逻辑插槽 1 和 2 中, 并使用全部 16 个基本通道来为最多 14 个节点板和另一个交换机提供 10/100/1000BASE-T 以太网交换资源。会指定一个基本通道, 以支持与机框管理卡的连接。
- switch slot**
(交换机插槽) 在星型拓扑中间背板中, 交换机插槽必须位于逻辑插槽 1 和 2 中。交换机插槽既支持基接口, 也支持交换接口。位于逻辑插槽 1 和 2 中的交换机插槽能够支持基接口交换机和交换接口交换机。不管采用哪种光纤网络拓扑, 逻辑插槽 1 和 2 始终都是交换机插槽。这些插槽分别支持多达 16 个基本通道和 15 个光纤网络通道。
- system** (系统) 可以包含一个或多个以下组件的管理实体: 节点和交换机、机框以及框架。

索引

A

AES 加密, 67
AMC 载体, 6
AMC (Advanced Mezzanine Card, 高级夹层卡), 6
ATCA 对应关系规范, 62
ATCA (Advanced Telecommunications Computing Architecture, 高级电信计算体系结构), 1

B

报警器, 105
保密性协议, 25

C

CDR, 49
CP3140 交换机刀片, 17
操作系统, 4
传感器, 32, 105
 表, 19
 类型信息, 33
串行接口, 7
次要阈值上限, 36
次要阈值下限, 36

D

DES, 25
destroy 值, 51
第 2 层交换, 3
第 3 层路由, 3
电压阈值, 51

E

engineID, 24

F

FASTPATH, 3, 125
访问控制, 24, 125
辐射状连接, 3

G

GPIO, 3
功率, 8
 故障, 51
 输入模块, 16
固件, 4
 Open Boot PROM, 4
管理
 工具, 105
 信息库, 18
 域, 18

H

HPI, 10
 检测, 18
 检查间隔, 18
 模型, 18
 用户, 10
 子代理, 17
 子代理配置文件, 18
互连资源, 8
会话, 11

I

- I2C 总线设备, 2
- IDR, 43
- IETF, 15
- IPM Sentry 机框管理器, 4
- IPMB, 6
- IPMB-0, 8
- IPMC, 4
- IPMI, 12
 - 概述, 6
 - 控制器, 51, 68
 - LAN 接口, 9
 - 平台事件过滤器, 8
 - 驱动程序, 68

J

- 机框, 2
- 机框管理器, 4, 6, 8
 - 功能, 8
 - IPMI, 51
 - 简介, 6
 - 接口选项, 9
 - 命令行界面 (command-line interface, CLI), 4
 - 软件, 4
- 监视程序, 51
 - 表, 19
 - 计时器, 8, 105
- 交换机说明, 3

K

- 开机自检 (POST), 4
- 控件
 - 信息, 38
- 控制, 105
 - 表, 19
 - 类型, 38
 - 平面, 3
- 库存
 - 控制表, 19
 - 数据系统信息库, 43, 105

L

- LED 指示灯, 2, 68

- Linux OpenIPMI 驱动程序, 69
- 冷启动陷阱, 62
- 冷却, 8
- 临界阈值上限, 36
- 临界阈值下限, 36

M

- MD5, 25
- MIB, 18
- MIB2, 18
- MMC (Module Management Controller, 模块管理控制器), 6
- Monta Vista Carrier Grade Linux OS, 4
- 命令行界面, 4
- 模拟控制, 39

N

- NMS, 16

O

- OEM
 - 区域, 49
 - 事件通知, 62
- OpenBoot PROM 固件, 4
- OpenHPI, 10
 - 规范, 10
 - 库, 11
 - 守护进程, 11
- OpenIPMI
 - 驱动程序, 13
- OpenSSL, 25

P

- PEM, 16
- PICMG, 68
 - 规范, 1
- PPS, 18
- 平台管理基础结构, 12

Q

- 嵌入式管理控制器, 67

R

RMCP, 4, 9
RPT, 11
热交换
 表, 19
 类型, 57
冗余基础结构, 2

S

SAF, 10
SAF-HPI MIB, 19
SAF-HPI-B.01.01 规范, 9
SAI-HPI-B.01.01 规范, 17
Serial-over-LAN, 67
SHA, 25
shelf (机框)
 报警面板, 2
 管理卡, 2, 4, 6, 8
 诊断, 2
ShMM, 2
SNMP, 15
 代理体系结构, 17
 对象, 18
 MIB 对象, 125
 守护进程配置文件, 18
 通知, 62
 陷阱, 125
 子代理, 17
 子代理配置, 24
SOL, 67
Solaris 操作系统, 4
设备 ID, 70
审计事件, 62
事件
 类别, 52
 类型, 52
 日志表, 58
 日志记录, 56
 时间戳, 52
 行, 18
 溢出操作, 18
事件消息, 51

实体, 10, 101
 表, 18, 19
 路径, 9, 101
 位置, 101
数据库结构, 15
数据平面, 3
双 ShMM 配置, 61

T

Telco 报警, 8
Telco 报警流量, 3
Telnet, 9
通信协议, 17
通知, 61

W

U-Boot, 4
网络管理器, 18
文档, xviii
温度
 异常, 8
 阈值, 51
物理包含分层结构, 101
物理实体, 18

X

系统
 分层结构, 101
 管理器, 7
 事件日志, 55
陷阱, 61, 125
消息传送协议, 67

Y

验证类型, 25
以太网信号, 2
硬件接口, 5
应用层协议, 15
应用程序, 4
域, 11
域事件日志, 55, 60
阈值传感器, 36

Z

支持状态, 125

重要阈值上限, 36

重要阈值下限, 36

主代理, 17

主事件表, 22

自定义数据记录, 49

字段, 47

字段类型, 48

资源, 10

表, 101

数据记录, 29, 105

资源存在表, 11