



Netra™ CT 900 服务器管理和 参考手册

Sun Microsystems, Inc.
www.sun.com

文件号码 820-0575-10
2007 年 1 月, 修订版 A

请将有关本文档的意见和建议提交至: <http://www.sun.com/hwdocs/feedback>

版权所有 2006 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. 保留所有权利。

对于本文档中介绍的产品，Sun Microsystems, Inc. 对其所涉及的技术拥有相关的知识产权。需特别指出的是（但不局限于此），这些知识产权可能包含在 <http://www.sun.com/patents> 中列出的一项或多项美国专利，以及在美国和其他国家/地区申请的一项或多项其他专利或待批专利。

本文档及其相关产品的使用、复制、分发和反编译均受许可证限制。未经 Sun 及其许可方（如果有）的事先书面许可，不得以任何形式、任何手段复制本产品或文档的任何部分。

第三方软件，包括字体技术，均已从 Sun 供应商处获得版权和使用许可。

本产品的某些部分可能是从 Berkeley BSD 系统衍生出来的，并获得了加利福尼亚大学的许可。UNIX 是 X/Open Company, Ltd. 在美国和其他国家/地区独家许可的注册商标。

Sun、Sun Microsystems、Sun 徽标、Java、AnswerBook2、docs.sun.com、Netra 和 Solaris 是 Sun Microsystems, Inc. 在美国和其他国家/地区的商标或注册商标。

所有 SPARC 商标的使用均已获得许可，它们是 SPARC International, Inc. 在美国和其他国家/地区的商标或注册商标。标有 SPARC 商标的产品均基于由 Sun Microsystems, Inc. 开发的体系结构。

PICMG 和 PICMG 徽标以及 AdvancedTCA 和 AdvancedTCA 徽标都是 PCI 工业计算机制造商协会 (PCI Industrial Computers Manufacturers Group) 的注册商标。

OPEN LOOK 和 Sun™ 图形用户界面是 Sun Microsystems, Inc. 为其用户和许可证持有者开发的。Sun 感谢 Xerox 在研究和开发可视或图形用户界面的概念方面为计算机行业所做的开拓性贡献。Sun 已从 Xerox 获得了对 Xerox 图形用户界面的非独占性许可证，该许可证还适用于实现 OPEN LOOK GUI 和在其他方面遵守 Sun 书面许可协议的 Sun 许可证持有者。

美国政府权利 - 商业用途。政府用户应遵循 Sun Microsystems, Inc. 的标准许可协议，以及 FAR（Federal Acquisition Regulations，即“联邦政府采购法规”）的适用条款及其补充条款。

本文档按“原样”提供，对于所有明示或默示的条件、陈述和担保，包括对适销性、适用性或非侵权性的默示保证，均不承担任何责任，除非此免责声明的适用范围在法律上无效。



请回收



Adobe PostScript

目录

前言 xvii

1. 简介 1

Netra CT 900 服务器软件 1

机框管理器简介 5

 ATCA 中的智能平台管理概述 5

 机框管理器和机框管理卡 7

 机框管理器特性 7

 机框管理器切换 8

 切换详细说明 9

System Administrator 接口选项 10

系统管理任务 11

物理地址与逻辑插槽的对应关系 11

2. 配置系统 13

访问机框管理卡 14

设置 U-Boot 15

 U-Boot 接口 16

 U-Boot 环境变量 16

 为环境变量赋值 19

 为机框管理器配置环境变量 19

配置机框管理卡以太网端口	20
使用第一个以太网接口	20
为第一个网络接口指定其他的 IP 地址	20
RMCP 地址传播	21
使用第二个以太网接口	21
将双 USB 网络接口用于冗余通信	21
更改默认的 ShMM 网络参数	22
▼ 更改默认的 ShMM 网络参数	23
设置机框管理器配置文件	26
详细级别说明	40
设置日期和时间	41
从时间服务器获取日期和时间	42
在机框管理卡上设置用户帐户	43
▼ 添加进行 RMCP 访问的用户帐户	43
用户名限制	44
密码	44
在机框管理器上配置 OpenHPI	44
/etc/openhpi.conf 文件	44
▼ 修改 /etc/openhpi.conf 文件	45
/etc/snmpd.conf 文件	45
访问控制	46
SNMPv3 配置	48
设置陷阱并通知目标	48
▼ 更新 /etc/snmpd.conf 文件	49

- 3. 管理系统 51
 - IPMI LAN 接口 51
 - IPMI 命令 52
 - 机框管理器命令行界面 52
 - 启动命令行界面 53
 - CLI 命令 54
 - 监视系统 58
 - 显示板信息和 IPMC 信息 58
 - 显示 FRU 信息 63
 - IPMI FRU 信息布局 63
 - 环境 FRU 64
 - 刀片 FRU 64
 - 示例 64
 - 显示机框信息 71
 - 示例 71
 - 重新初始化机框管理器 77
 - 重新初始化 U-Boot 环境 77
 - ▼ 重新初始化 U-Boot 环境 77
 - 重新初始化文件系统 78
 - 重新设置登录密码 79
 - 对机框管理卡进行重新编程 80
 - 固件可靠升级过程 80
 - 闪存分区 81
 - /var/upgrade 文件系统 82
 - 可靠升级过程状态文件 82
 - 可靠升级实用程序 83
 - 可靠升级实用程序使用方案 87
 - 可靠升级示例 88

对 CPLD 进行编程	99
▼ 对 ShMM 的 CPLD 映像重新编程	99
连接到节点板控制台	99
在机框管理器和节点板之间建立控制台会话	100
▼ 从机框管理器启动控制台会话	100
▼ 结束控制台会话	101
手动正常关闭节点板	101
▼ 关闭节点板	102
A. 机框管理器 CLI 命令	105
activate	106
alarm	107
board	108
boardreset	110
busres	111
显示总线型电子钥控 (E-Keying) 管理资源的状态	112
释放指定的资源	112
锁定/解除锁定指定的资源	113
发送 Bused Resource Control (Query) 命令	113
设置资源的所有者	114
发送 Bused Resource Control (Bus Free) 命令	115
console	116
deactivate	117
debuglevel	117
exit quit	118
fans	119
flashupdate	120
fru	121
frucontrol	123

frudata 124
frudatar 126
frudataw 127
fruinfo 128
getfanlevel 129
getfruledstate 130
gethysteresis 132
getipmbstate 133
getlanconfig 134
 auth_support 137
 auth_enables 138
 ip 139
 ip_source 139
 mac 140
 subnet_mask 140
 ipv4_hdr_param 141
 pri_rmcp_port 141
 sec_rmcp_port 142
 arp_control 142
 arp_interval 143
 dft_gw_ip 143
 dft_gw_mac 144
 backup_gw_ip 144
 backup_gw_mac 145
 community 145
 destination_count 146
 destination_type 146
 destination_address 147

getpefconfig 148
 control 150
 action_control 150
 startup_delay 151
 alert_startup_delay 151
 event_filter_count 152
 event_filter 152
 event_filter_data1 153
 alert_policy_count 154
 alert_policy 155
 system_guid 155
 alert_string_count 156
 alert_string_key 156
 alert_string 157
 oem_filter_count 157
 oem_filter 158
getsensoreventenable 159
getthreshold | threshold 160
help 162
ipmc 164
localaddress 166
minfanlevel 166
sel 167
sensor 171
sensordata 175
sensorread 177
session 179
setextracted 180

- setfanlevel 180
- setfruledstate 181
- sethysteresis 183
- setipmbstate 183
- setlanconfig 184
 - auth_enables 186
 - ip 186
 - subnet_mask 187
 - ipv4_hdr_param 187
 - arp_control 188
 - arp_interval 188
 - dft_gw_ip 189
 - backup_gw_ip 189
 - community 190
 - destination_type 190
 - destination_address 191
- setlocked 191
- setpefconfig 192
 - control 194
 - action_control 194
 - startup_delay 195
 - alert_startup_delay 195
 - event_filter 196
 - event_filter_data1 197
 - alert_policy 198
 - system_guid 199
 - alert_string_key 200
 - alert_string 200
 - oem_filter 201

setsensoreventenable 202

setthreshold 203

shelf 205

- 显示机框 FRU 信息 206
- 修改最大外部可用电流 211
- 修改最小预期工作电压 213
- 修改“机框管理器控制的激活”标志 215
- 修改“机框管理器控制的取消激活”标志 220
- 修改 FRU 最大功率容量 221
- 修改下次打开电源前的延迟 223
- 修改 FRU 激活准备就绪允许时间 224
- 重新排列 FRU 激活和电源描述符的顺序 225
- 刷新机框 FRU 信息 227
- 更新机框 FRU 信息存储设备 229

shelfaddress 229

shmstatus 230

showhost 231

showunhealthy 232

switchover 232

terminate 233

user 233

- 显示用户信息 234
- 添加新用户 235
- 删除用户 236
- 启用和禁用用户 236
- 修改用户名 238
- 修改用户密码 239
- 修改指定用户和指定通道的通道访问设置 240

version 241

B. Sun OEM IPMI 命令 243

Get Version 244

Set Boot Page 245

Get Boot Page 246

Set Front Panel Reset Button State 247

Get Front Panel Reset Button State 248

Set Ethernet Force Front Bit 249

Get Ethernet Force Front Bit 250

Get RTM Status 251

图

- 图 1-1 Netra CT 服务器中软件与硬件接口的逻辑表示 4
- 图 1-2 ATCA 机框示例 6
- 图 1-3 机框管理器切换信号 8
- 图 2-1 机框报警面板连接器 14
- 图 3-1 IPMI FRU 信息布局 63

表

表 1-1	适用于系统管理员的 Netra CT 服务器软件	2
表 1-2	Netra CT 900 系统板接入方法	3
表 1-3	支持切换的硬件信号和接口	9
表 1-4	物理地址与逻辑插槽的对应关系	11
表 2-1	默认的 U-Boot 环境变量	16
表 2-2	机框管理器配置参数	27
表 3-1	Sun OEM IPMI 命令	52
表 3-2	机框管理器 CLI 命令汇总	54
表 3-3	16 MB 的闪存分区 (reliable_upgrade=y)	81
表 3-4	与机框管理器 CLI 控制台相关的命令	100
表 A-1	getlanconfig 的 LAN 配置参数	135
表 A-2	PEF 配置参数	148
表 A-3	setlanconfig 的 LAN 配置参数	184
表 A-4	setpefconf 的 PEF 配置参数	192
表 A-5	shelf 命令的参数	206
表 B-1	Sun OEM IPMI 命令	243

前言

《Netra CT 900 服务器管理和参考手册》包含适用于 Netra™ CT 900 服务器系统管理员的配置和管理信息，还提供了机框管理器和 IPMI 命令参考信息。

本手册假设您熟悉 UNIX® 命令和网络、PICMG® 3.x AdvancedTCA® 基本规范以及智能平台管理接口 (Intelligent Platform Management Interface, IPMI)。

本书的结构

第 1 章对 Netra CT 900 服务器软件进行了介绍。

第 2 章包含有关如何配置系统的信息。

第 3 章介绍如何管理系统。

附录 A 提供机框管理器各个命令行界面 (command-line interface, CLI) 命令的语法和用途。

附录 B 介绍特定于 Sun 的 OEM 定义智能平台管理接口 (Intelligent Platform Management Interface, IPMI) 命令。

使用 UNIX 命令

本文档不会介绍基本的 UNIX 命令和操作过程，如关闭系统、启动系统和配置设备等。欲获知此类信息，请参阅以下文档：

- 系统附带的软件文档
- Solaris™ 操作系统 (Solaris™ Operating System, Solaris OS) 的有关文档，其 URL 如下：
<http://docs.sun.com>

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	\$
Bourne shell 和 Korn shell 超级用户	#

印刷约定

字体*	含义	示例
AaBbCc123	命令、文件和目录的名称；计算机屏幕输出	编辑 <code>.login</code> 文件。 使用 <code>ls -a</code> 列出所有文件。 % You have mail.
AaBbCc123	用户键入的内容，与计算机屏幕输出的显示不同	% su Password:
<i>AaBbCc123</i>	保留未译的新词或术语以及要强调的词。要使用实名或值替换的命令行变量。	这些称为 <i>class</i> 选项。 要删除文件，请键入 rm filename 。
新词术语强调	新词或术语以及要强调的词。	您 必须 成为超级用户才能执行此操作。
《书名》	书名	阅读《用户指南》的第 6 章。

* 浏览器的设置可能会与这些设置有所不同。

相关文档

下表列出了 Netra CT 900 服务器文档。除《Important Safety Information for Sun Hardware Systems》外，您可以从以下位置获得所列出的所有联机文档：

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

书名	文件号码
《Netra CT 900 Server Product Notes》	819-1180
《Netra CT 900 Server Overview》	819-1174
《Netra CT 900 Server Installation Guide》	819-1175
《Netra CT 900 Server Service Manual》	819-1176
《Netra CT 900 Server Switch Software Reference Manual》	819-3774
《Netra CT 900 Server Safety and Compliance Manual》	819-1179
《Important Safety Information for Sun Hardware Systems》（仅提供印刷版本）	816-7190-10

您可能需要参考以下产品的文档以获取其他信息：Solaris OS、OpenBoot™ PROM 固件、Netra CP3010 板、Netra CP3020 板和 Netra CP3060 板。

文档、支持和培训

Sun 提供的服务	URL
文档	http://www.sun.com/documentation/
支持	http://www.sun.com/support/
培训	http://www.sun.com/training/

联系 Sun 技术支持

如果您遇到通过本文档无法解决的技术问题，请访问以下网址：

<http://www.sun.com/service/contacting>

Sun 欢迎您提出意见

Sun 致力于提高其文档的质量，并十分乐意收到您的意见和建议。您可以通过以下网址提交您的意见和建议：

<http://www.sun.com/hwdocs/feedback>

请在您的反馈信息中包含文档的书名和文件号码：

《Netra CT 900 服务器管理和参考手册》，文件号码 820-0575-10。

第1章

简介

本章包括以下各节：

- 第 1 页的 “Netra CT 900 服务器软件”
- 第 5 页的 “机框管理器简介”
- 第 11 页的 “系统管理任务”

Netra CT 900 服务器软件

Netra CT 900 服务器软件包括：

- 机框管理器
- 操作系统和应用程序
- 固件

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。

表 1-1 对软件进行了说明，图 1-1 则是软件与硬件配合使用的逻辑示意图。

表 1-1 适用于系统管理员的 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, 出厂时就已经安装。此软件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	Solaris 操作系统 (Solaris Operating System, 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 Netra CP3010、Netra CP3020 和 CP3060 节点板。可以选择在 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	Netra CP3020 也可以运行 Monta Vista Carrier Grade Linux OS。
固件	OpenBoot PROM 固件	即 Sun 支持的节点板 (如 Netra CP3010 板) 上的固件, 用于控制引导。它还有诊断功能。
	U-Boot	即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)	即系统管理控制器固件, 用于通过 Sun 支持的节点板 (如 Netra CP3010 板) 上的 IPMI 控制器进行通信。

Netra CT 900 服务器有两个机框管理卡 (ShMM), 并提供因特定硬件和软件事件而引发的从活动机框管理卡到备用机框管理卡的机框管理卡故障转移。活动机框管理卡用于对大多数连接到中间背板的部件进行系统级配置和管理。备用机框管理卡则为活动机框管理卡提供冗余和故障转移功能。

交换结构板在内部连接了机框管理卡和节点板, 其后部具有供外部连接使用的以太网端口。

Netra CP3010 节点板接受并拥有外围设备, 如磁盘。节点板还运行用户应用程序。在 Netra CT 900 服务器中, 每块节点板运行各自的操作系统副本, 因此, 每块节点板都被视为一个服务器。机框管理卡、节点板、交换结构板和其他系统现场可更换单元 (field-replaceable unit, FRU) 组成一个系统。

注 - 在本手册中, 术语节点板指 Sun 支持的 ATCA CPU 板 (如 Netra CP3010 板), 除非另行指定。

符合 PICMG® 3.x 标准的第三方 ATCA 节点板可用于 Netra CT 900 服务器。这些板不一定运行 Solaris OS, 而且不运行 Netra CT 900 服务器系统管理软件。因此, 对这些板的管理, 在程度上无法与对 Netra 节点板的管理相同。

表 1-2 概括说明了如何接入各种板。机框管理卡一次可支持 22 个会话（1 个 Tip 连接和 21 个 Telnet 连接）。

表 1-2 Netra CT 900 系统板接入方法

板	接入方法
机框报警面板 (Shelf Alarm Panel, SAP)	前面板具有下列端口： <ul style="list-style-type: none">• 两个带有 RJ-45 DTE 连接器的串行 (RS-232) 端口。串行端口 1 供上部机框管理卡 (ShMM1)（默认的活动卡）进行控制台连接。串行端口 2 供下部机框管理卡 (ShMM2)（默认的备用卡）进行控制台连接。• Telco 报警连接器 (DB-15)
交换机板（插槽 7 和插槽 8）	前面板上用于 Telnet 连接的多个以太网端口。 注 - 要从后部接入这些端口，需要后部转换模块 (rear transition module, RTM)。可以使用后部端口连接器或前部端口连接器，但不能同时使用二者。如果将电缆连接到这两个端口，则只有前部端口处于活动状态。
节点板（Sun 支持的 CPU 板） （插槽 1 到 6 和插槽 9 到 14）	对于 Netra CP3010 板，前面板具有下列端口： <ul style="list-style-type: none">• 两个用于 Tip 或 ASCII 终端连接的串行端口（控制台）• 两个用于 Telnet 连接的以太网端口• 一个 4X 串行连接 SCSI (Serial Attached SCSI, SAS) 端口连接器 注 - 要从后部接入这些端口，需要 RTM。可以使用后部端口连接器或前部端口连接器，但不能同时使用二者。如果将电缆连接到这两个端口，则只有前部端口处于活动状态。 对于 Netra CP3020 和 CP3060 板，前面板具有下列端口： <ul style="list-style-type: none">• 一个用于 Tip 或 ASCII 终端连接的串行端口（控制台）• 两个用于 Telnet 连接的以太网端口 有关更多信息，请参阅特定节点板的 Netra ATCA CPU 板文档。
第三方节点板（插槽 1 到 6 和插槽 9 到 14）	视第三方板而定。

硬件接口包括智能平台管理接口 (Intelligent Platform Management Interface, IPMI)、基接口和扩展接口，以及机框管理卡、节点板和交换结构板上的网络接口。

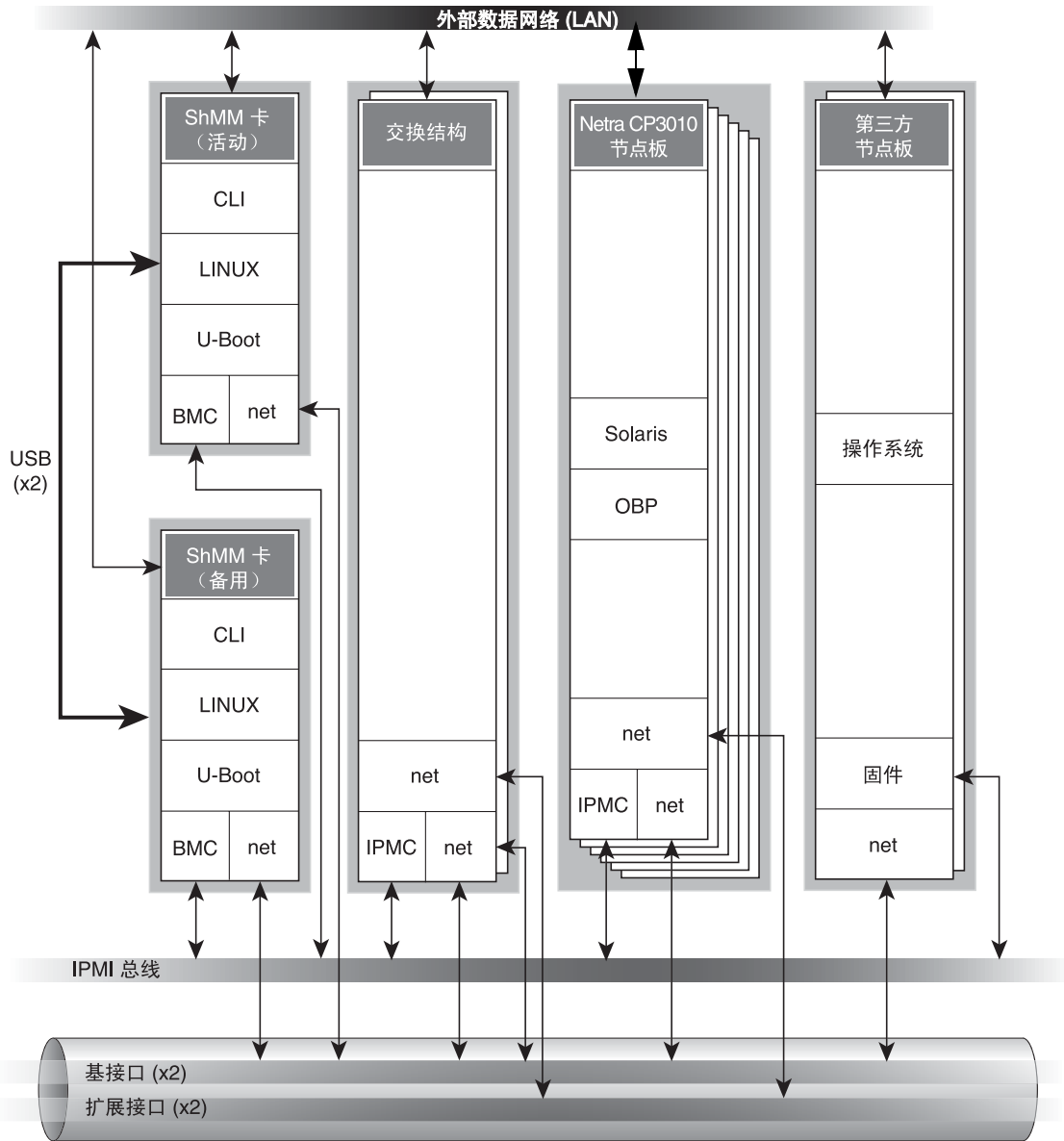


图 1-1 Netra CT 服务器中软件与硬件接口的逻辑表示

机框管理器简介

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机框管理器以及对 ATCA 环境下使用的任一机框管理载体均通用的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) 构造块，其重点在于实现 FRU 动态填充和最大程度地提供服务可用性。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。

图 1-1 显示了根据 ATCA 规范确定的 ATCA 机框示例的逻辑元素。

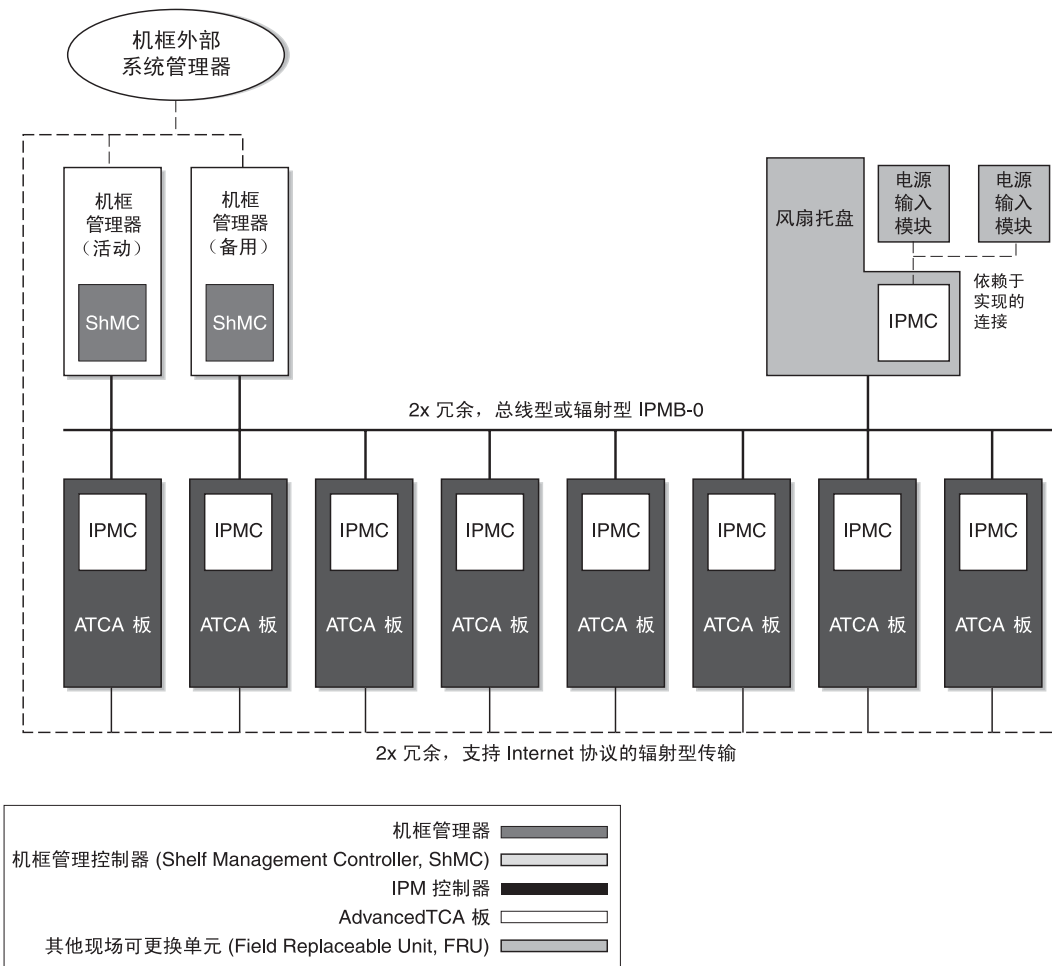


图 1-2 ATCA 机框示例

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接口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 FRU 填充及通用机框基础结构，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) 在机框管理器与 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。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如下：

- 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。

机框管理器切换

可为机框管理器配置活动/备用实例，以获得最大可用性。图 1-3 显示了在给定时间内只有活动实例进行交互的情况下系统管理器如何访问这两个实例。同样，在机框内只有活动实例通过 IPMB-0 与 IPM 控制器填充进行通信。这两个实例通过 TCP/IP 相互通信，并由活动实例向备用实例发布状态增量更新。其结果是，备用角色会在必要时快速进入活动角色。

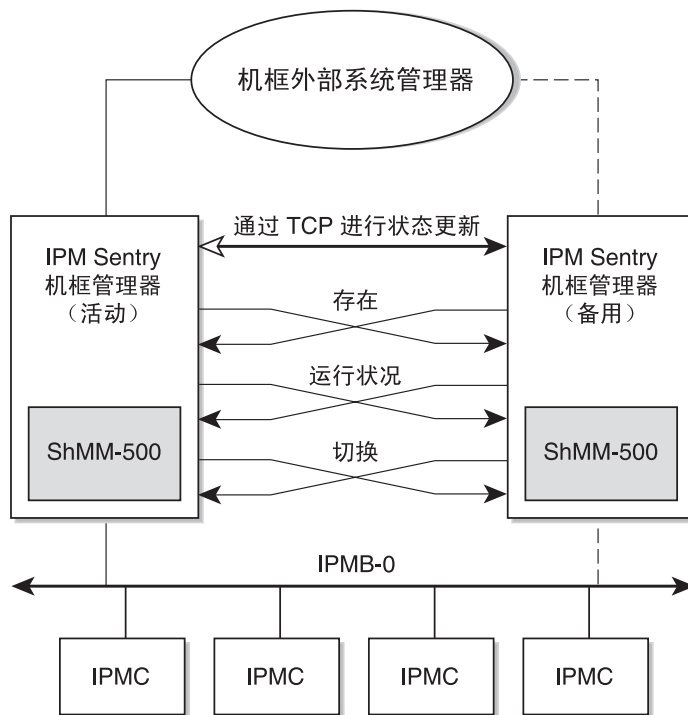


图 1-3 机框管理器切换信号

表 1-3 列出了信号和说明。

表 1-3 支持切换的硬件信号和接口

硬件	说明
USB 接口	机框管理卡之间的接口；它用于发送心跳和状态同步信息。两个机框管理卡必须以相同的状态（如，均已打开电源）查看同一 FRU（如特定风扇托盘或特定插槽中的节点板）。
#SWITCHOVER	备用实例可在必要时强制进行切换。
#PRSNT	此信号表示存在机框管理卡。
#HEALTHY	此信号表示机框管理卡（包括硬件和软件）的总体运行状况。

切换详细说明

活动机框管理器对 IPMB 公开 ShMC 设备（地址 20h），管理 IPMB 和 IPM 控制器，并通过 RMCP 及其他机框外部接口与系统管理器进行交互。它与备用机框管理器之间保持开放式 TCP 连接。活动机框管理器还将管理对象状态的所有更改传送到备用机框管理器。

备用机框管理器不对 IPMB 公开 ShMC，不主动管理 IPMB 和 IPM 控制器，也不通过机框外部接口与系统管理器进行交互（下面有一个例外）。但是，它在自己的内存（易失性和非易失性）中保留管理对象的状态，并按照活动机框管理器的指示对状态进行更新。

切换的结果就是备用机框管理器成为活动机框管理器。下面定义了两种类型的切换：

- 协同切换 – 活动和备用机框管理器协商将活动机框管理器的职责转到备用机框管理器；通过在活动或备用机框管理器上执行的 CLI `switchover` 命令可支持这种模式。
- 强制切换 – 当备用机框管理器确定活动机框管理器不再处于活动状态或运行不正常时，就会强制接管活动机框管理器的职责。

当 Remote Healthy 或 Remote Presence 底层信号变为非活动状态时，备用机框管理器就认为活动机框管理器已卸任。Remote Presence 信号可监视对等机框管理器的存在；此信号变为非活动状态意味着已从机框中拆除托管对等机框管理器的板。Remote Healthy 信号由对等机框管理器在初始化期间设置；此信号变为非活动状态意味着远程机框管理器运行不正常（通常是已关闭电源或复位）。

需要备用机框管理器采取某些措施的另一种情况是当机框管理器之间的 TCP 连接关闭的时候。当两个机框管理器之间的通信链路中断时、当活动机框管理器上的机框管理进程终止（无论是自动终止还是非自动终止）时，或当软件出现异常时，均需要备用机框管理器采取相应措施。对连接启用 TCP `keepalive` 选项后，该连接会在活动机框管理卡关闭或复位后马上关闭。

如果机框管理器终止，TCP 连接可能会在 Remote Healthy 信号变为非活动状态之前关闭。为确定 TCP 连接关闭的原因，备用机框管理器会立即对 Remote Healthy 信号的状态采样；如果该信号仍处于活动状态，则在一定延迟后再次采样。当 Remote Healthy 信号最终变为非活动状态时，备用机框管理器就断定活动机框管理器已停用，并启动切换。

如果 Remote Healthy 信号保持活动状态，则备用机框管理器就断定机框管理器之间的通信链路已中断。这种情况下，不会启动任何切换；相反，备用机框管理器会反复重新初始化自身并尝试建立与活动机框管理器的连接，直到通信链路恢复为止。重新引导机框管理卡，然后自动重新启动机框管理器，即可完成重新初始化。机框管理器中的特殊逻辑可确保在启动时，如果对等机框管理器已处于活动状态，则它不会尝试变为活动状态。

机框管理器使用监视程序计时器来防止因死循环或其他软件错误而变得无法响应。如果触发了活动机框管理器上的监视程序计时器，则该机框管理卡将复位，从而使备用机框管理卡上的 Remote Healthy 信号变为非活动状态，因此触发切换。

执行切换后，当前活动机框管理器将重新初始化，激活缓存的状态信息，并从 IPMB 上的 IPM 控制器收集必要信息。此活动机框管理器随后会对 IPMB 公开 ShMC 设备（地址 20h），并采用 RMCP 曾使用的 IP 地址以及原先活动机框管理器与系统管理器之间的其他机框外交互。由于 RMCP 会话信息是从活动机框管理器向备用机框管理器传播，因此 RMCP 会话在切换后仍然存在。对于使用 RMCP 的系统管理器来说，切换是透明的。

执行切换后，原先的活动机框管理器可以不复存在，也可以将自身重新初始化为备用机框管理器。如果重新初始化为备用机框管理器，则需要重新引导先前的活动机框管理卡上的操作系统。

System Administrator 接口选项

机框管理器的另一个主要子系统可实现 System Administrator 接口。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 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 System Administrator 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 System Administrator 使用机框管理器作为代理向机框中的 IPM 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。

系统管理任务

Netra CT 900 服务器系统管理通常包括安装、配置和管理任务。

Netra CT 900 服务器上的 Solaris OS 管理（包括添加 Solaris 用户帐户）通过登录到节点板来执行。Netra CT 900 服务器管理则通过登录到机框管理卡和使用机框管理卡 CLI 来执行。出于配置和管理目的，可以在 Netra CT 900 服务器中将机框管理卡用作单一登录点。

在随后几章中对系统管理任务进行了说明。

物理地址与逻辑插槽的对应关系

如果从前面看 Netra CT 900 服务器，那么物理插槽是从左到右按顺序编号的。[表 1-4](#) 给出了物理与逻辑插槽对应关系及地址。

表 1-4 物理地址与逻辑插槽的对应关系

物理插槽	1	2	3	4	5	6	7	8	9	10	11	12	13	14	ShMM #1	ShMM #2
逻辑插槽	13	11	9	7	5	3	1	2	4	6	8	10	12	14	N/A	N/A
HW 地址 (十六进制)	4D	4B	49	47	45	43	41	42	44	46	48	4A	4C	4E	8	9
IPMB 地址 (十六进制)	9A	96	92	8E	8A	86	82	84	88	8C	90	94	98	9C	10	12

第2章

配置系统

本章假定您在 Netra CT 900 服务器和 Netra ATCA 节点板上已经安装了 Solaris 操作系统及所需的修补程序。

Netra CT 900 系统的配置工作主要是通过活动机框管理卡命令行界面 (command-line interface, CLI) 来进行的。通过活动机框管理卡 CLI，可以对节点板、交换结构板、机框管理卡、电源输入模块 (power entry module, PEM) 和风扇托盘等部件进行系统级配置和管理。既可以在本地使用机框管理卡 CLI 接口，也可以远程使用它。

本章包括以下各节：

- 第 14 页的 “访问机框管理卡”
- 第 15 页的 “设置 U-Boot”
- 第 20 页的 “配置机框管理卡以太网端口”
- 第 22 页的 “更改默认的 ShMM 网络参数”
- 第 26 页的 “设置机框管理器配置文件”
- 第 41 页的 “设置日期和时间”
- 第 43 页的 “在机框管理卡上设置用户帐户”

访问机框管理卡

机框管理器运行于专门实现的 Linux 之上。Linux 的最低层是 U-Boot 固件监视器。当您最初访问任一机框管理卡 (ShMM) 时, 必须使用 ASCII 终端或 Tip 程序通过串行端口 (控制台) 进行访问。每个机框管理卡可同时支持多个会话 (Tip 连接和 Telnet 连接)。活动机框管理卡的默认 TCP/IP 地址是 192.168.0.2。

通过串行端口连接到机框管理卡时, 请将串行终端或仿真器连接到机框报警面板 (shelf alarm panel, SAP) 前面的两个串行端口之一。使用串行端口 1 连接到上部机框管理卡 (ShMM1), 它是默认的活动机框管理卡。使用串行端口 2 连接到默认的备用卡 (ShMM2)。图 2-1 显示了上部和下部机框管理卡的串行端口的位置。终端或调制解调器设置应设置为 115200, N, 8, 1。

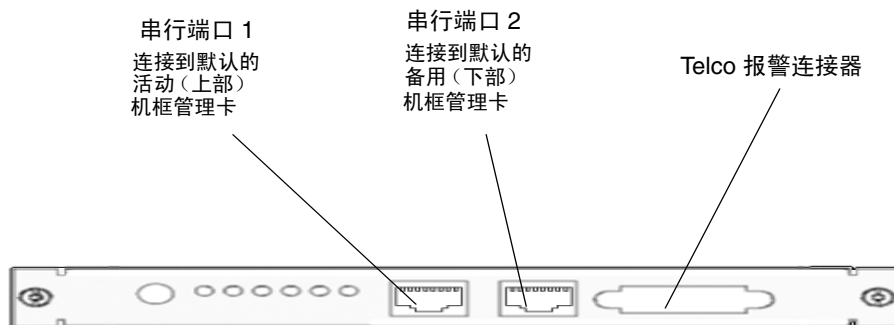


图 2-1 机框报警面板连接器

首次访问机框管理卡时, 请以 root 身份登录, 并使用默认密码 sunct900。此帐户设置为完全授权 (权限)。无法删除此帐户。不过, 在 Netra CT 900 服务器可操作之前, 为安全起见, 您应当更改此帐户的密码。

使用 Linux `passwd` 命令更改 root 密码，如下所示：

```
# passwd

Changing password for root
Enter the new password (minimum of 5, maximum of 8 characters)
Please use a combination of upper and lower case letters and
numbers.
Enter new password: xxxxxxxx
Re-enter new password: xxxxxxxx
Password changed.
#
```

登录后，使用 `clia shmstatus` 命令检验您是否已登录到活动机框管理卡，然后再继续操作。如果您登录到的是备用机框管理卡，可以使用 `clia switchover` 命令将此机框管理卡的状态更改为 Active，或者退出并登录到活动机框管理卡。（有关更多信息，请参见第 230 页的 "[shmstatus](#)" 和第 232 页的 "[switchover](#)"。）

以下各节介绍有关使用机框管理卡 CLI 配置机框管理卡以太网端口以及设置用户帐户和密码的信息。有关使用机框管理卡 CLI 的更多信息，请参见第 3 章。

注 – 除非另行指定，否则本手册中使用的术语**机框管理卡**是指活动机框管理卡或备用机框管理卡。在本手册中，这两个卡的提示符简化为 `ShMM #`。

设置 U-Boot

在机框管理卡 (ShMM) 加电和重新引导时，硬件开始执行闪存中的 U-Boot 固件。该固件执行 ShMM 的基本初始化，除非用户明确禁用自动引导功能（禁用的结果是强制固件切换到维护用户命令界面），否则将开始引导 Linux 内核。从内核及驻留在闪存中的 root 文件系统映像引导 Linux。U-Boot 将内核映像重新定位到 RAM，并设置内核参数，然后将控制权传递给内核入口点。

U-Boot 接口

通过 ShMM 的串行端口即可访问 U-Boot，U-Boot 要求特定于所需操作环境的配置。ShMM 加电后，控制台会显示以下信息：

```
U-Boot 1.1.2 (Nov 11 2005 - 11:32:08)

CPU: Au1550 324 MHz, id: 0x02, rev: 0x00
Board: ShMM-500
S/N: 08004610
DRAM: 128 MB
Flash: 64 MB
In: serial
Out: serial
Err: serial
Net: Au1X00 ETHERNET
Hit any key to stop autoboot: 0
#
```

是允许输入用户命令的提示符。

U-Boot 环境变量

U-Boot 包含一组环境变量，使用前应先配置这些变量。[表 2-1](#) 介绍了一组默认提供的变量。

表 2-1 默认的 U-Boot 环境变量

环境变量	说明
addmisc	将 quiet、reliable_upgrade 和 console 设置附加到 bootargs 中。通常不对此变量进行修改。
baudrate	串行端口波特率，默认值为 115200。
bootargs	将被传递到 Linux 内核的命令行。可包含对其他 U-Boot 环境变量的引用，这些引用在运行时解析。默认值为： root=/dev/ram rw console=ttyS0,115200 reliable_upgrade=y
bootcmd	为完成自动引导而执行的 U-Boot 命令。
bootdelay	自动引导延迟值，以秒为单位。
bootfile	指定 net 和 nfs 引导选项应当使用的内核映像的参数。

表 2-1 默认的 U-Boot 环境变量（续）

环境变量	说明
console	内核、init 脚本控制台端口和波特率的设置。默认值为 console=ttyS0,115200。
ethaddr	主单片以太网控制器的 MAC 地址。此变量的值由 U-Boot 自动设置。此地址将传递到内核以太网驱动程序。
ethladdr	辅助以太网控制器的 MAC 地址。此变量的值由 U-Boot 自动设置。此地址将传递到内核以太网驱动程序。
flash_reset	指示 Linux 清除闪存文件系统（/etc 和 /var），从而恢复为出厂默认设置（y/n）。清除闪存后，系统启动脚本将此值设置回 n。默认值为 n。
gateway	默认网关 IP 地址。可以将此变量作为内核命令行的一部分传递，以便自动配置网络接口路由。
hostname	网络主机名，默认值为 sentry。
io_config	确定是否为 PSC 控制器配置了双从地址配置（y/n）。默认设置：y。
ipaddr	主单片以太网接口所使用的 IP 地址。如果 rc_ifconfig 变量设置为 y，则 ipaddr 变量用于配置由 ipdevice 自动指定的网络接口。请注意，系统启动脚本将此变量的最低有效位设置为 ShMM 载体的硬件地址的最低有效位；即，如果硬件地址是一个偶数值，则 IP 地址的最后一位设置为 0，否则设置为 1。为了支持冗余 ShMM 的配套 IP 地址配置，上述设置在启动脚本 /etc/netconfig 中执行。要禁用此功能，只需删除 /etc/readhwaddr 文件。
ipladdr	辅助以太网接口所使用的 IP 地址。可以将此变量作为内核命令行的一部分传递，以便自动配置相应的内核网络接口。
ipdevice	与 ipaddr 对应的设备，默认值为 eth0。
ipldevice	与 ipladdr 对应的设备，默认值为 eth1。
kernel_start	闪存中内核映像的绝对起始地址。此变量在引导过程中由 U-Boot 自动设置。
logging	指定是在 ram 中还是在 flash 中维护消息日志文件。默认值为 ram，这也是推荐选项。
net	此变量可用于替换 bootcmd，作为从 TFTP 服务器引导内核和 .rfs 映像的一种方法。
netmask	网络掩码，默认值为 255.255.255.0。
password_reset	指示 Linux 恢复出厂默认密码（对于用户 root 来说是空密码）。默认值为 n。
post_normal	确定每次引导时执行的一系列 POST 测试。如果未设置，则使用编译时默认设置。以此变量值的形式列出的测试名称用空格字符分隔。
post_poweron	确定仅在加电复位后（与每次引导时相对）执行的一系列 POST 测试。如果未设置，则使用编译时默认设置。以此变量值的形式列出的测试名称用空格字符分隔。

表 2-1 默认的 U-Boot 环境变量 (续)

环境变量	说明
quiet	指示内核在引导时不向串行控制台输出进度消息。默认值为 quiet=quiet。
ramargs	根据要从 RAM 磁盘 (ramdisk) 挂载的 root 文件系统, 在 bootargs 变量中设置内核命令行。
ramdisk	指定 net 和 nfs 引导选项应当使用的 .rfs 映像。
ramsize	系统内存大小, 以字节为单位。默认设置: 根据生成时配置块中的 SDRAM 配置编码进行计算。
rc_ifconfig	允许 /etc/rc 脚本设置 IP 地址, 而不是从 shelfman 文件获得地址。默认值为 n (允许 shelfman 设置 IP 地址)。
rc2	指定要调用的辅助 RC 脚本。这是特定于载体的启动脚本。默认值为 /etc/rc.acb3 或指定目标平台的其他适当脚本。
reliable_upgrade	确定是否对 ShMM-500 启用可靠软件升级过程 (y/n)。默认设置: y。目前不支持将此变量设置为 n。如果将此变量设置为 n, 则 ShMM 下次引导时会发出错误消息并挂起。
rfs_start	闪存中 root 文件系统映像的绝对起始地址。此变量在引导过程中由 U-Boot 自动设置。
rmcpaddr	RMCP 服务的默认 IP 地址。
serverip	TFTP 服务器的 IP 地址。
start_rc2_daemons	指示在引导之后辅助启动脚本是启动还是不启动 snmpd/boa 和 shelfman 守护进程。默认值为 y。
time_server	时间服务器, 供运行时同步所用。如果未指定此变量, 则系统启动时从硬件时钟提取时间。 注: 指定此变量后, 必须将 ip1device 变量设置为 usb0 以便进行正确同步。
timezone	格式为 CCCn 的本地时区, 其中 n 是格林尼治标准时间 (Greenwich Mean Time, GMT) 偏移, 可以为负值, 而 CCC 标识时区。默认值为 UTC0。

为环境变量赋值

要为环境变量赋值，请采用以下格式：

```
setenv variable_name new_value
```

例如：

```
# setenv bootdelay 1
```

正确设置了所有环境变量后，必须将它们存回到闪存中，以便这些变量在关闭 ShMM 电源后仍保持设置状态。可以使用 `saveenv` 命令来执行此操作。

```
# saveenv
```

也可以将以 Linux 实用程序形式提供的 `setenv` 功能用于上述用途。要在 shell 提示符下显示 U-Boot 变量，请使用附加的 `getenv` 实用程序。

为机框管理器配置环境变量

首次启动 U-Boot 时，会定义以下默认环境变量：

```
bootcmd=run setup_bootargs; bootm BFB00000 BFC40000
bootdelay=3
baudrate=115200
ethaddr= 00:00:1a:18:xx:yy
ethladdr= 00:00:1a:18:xx:zz
serverip=192.168.0.7
netmask=255.255.0.0
hostname=sentry
gateway=192.168.0.1
ipdevice=eth0
ipladdr=192.168.1.3
ipldevice=eth1
rc2=/etc/rc.acb3
ipaddr=192.168.0.2
start_rc2_daemons=y
flash_reset=n
password_reset=n
logging=ram
rc_ifconfig=n
bootfile=sentry.mips.kernel
ramdisk=sentry.mips.rfsnet=tftp 80400000 $(bootfile); tftp
80800000 $(ramdisk); bootm 80400000 80800000
rmcpaddr=192.168.1.15
timezone=EST
bootargs=root=/dev/ram rw console=ttyS0,115200 reliable_upgrade=y
```

必须用适合于使用 ShMM 的网络环境的值，来重新配置上述环境变量中的几个变量。

配置机框管理卡以太网端口

每个机框管理卡使用两个连接到冗余交换机卡的以太网端口。由于 RMCP 是 ATCA 要求的唯一机框外部接口，因此机框外部以太网端口将作为 RMCP 端口引用，不过通过此端口也可以访问其他机框外部接口 (Telnet)。

连接好后，您必须使用具有完全权限的用户帐户登录到机框管理卡。先使用 CLI 命令配置端口，然后重新引导机框管理卡使所做更改生效。

使用第一个以太网接口

由于 RMCP 以太网端口直接连接到站点网络，因此 IP 地址的设置应当适合于该网络。例如，如果站点使用的 IP 地址范围是 192.168.0.x，则应当将 RMCP 以太网端口设置为该范围内的唯一 IP 地址，如 192.168.0.2。在冗余 ShMM 设置中，只有一个 ShMM（即活动 ShMM）会启用 RMCP 以太网端口的 RMCP IP 地址。备用 ShMM 为 RMCP 以太网端口指定相同的 IP 地址，但只有当备用 ShMM 承担活动角色时才启用该地址。这样的话，RMCP IP 地址便可在故障转移情况下保持可用性。

为第一个网络接口指定其他的 IP 地址

在默认配置中，只有在机框管理器启动并指定了 RMCP IP 地址后，才会为第一个网络接口指定 IP 地址（从而通过网络访问 ShMM）。但有时需要以下配置：为 RMCP 网络接口指定了 IP 地址，而且引导操作系统后可立即通过网络访问 ShMM。此时，还需要 RMCP IP 地址与原先指定的 IP 地址共存，而不是在机框管理器启动时取代它。

要实现这种配置，必须指示机框管理器不要将 RMCP IP 地址指定给第一个网络适配器本身 (eth0)，而要指定给其第一个别名 (eth0:1)。这种情况下，在操作系统启动期间初始 IP 地址将被指定给网络适配器本身 (eth0)。这种初始指定是在初始化脚本 /etc/rc 中进行的；具体实现方法如下：

1. 按如下方式启用 U-Boot 变量 rc_ifconfig:

```
setenv rc_ifconfig y
```

2. 为 U-Boot 变量 ipaddr 指定初始 IP 地址。例如:

```
setenv ipaddr 192.168.1.240
```


3. 将机框管理器配置文件 `/etc/shelfman.conf` 中 `RMCP_NET_ADAPTER` 的值更改为 `eth0:1`。例如：

```
RMCP_NET_ADAPTER = eth0:1
```

在冗余配置中，对于两个 ShMM，允许将 U-Boot 变量 `ipaddr` 设置为相同的值。指定给两个冗余 ShMM 中每一个 ShMM 的实际初始 IP 地址基于 `ipaddr` 的值，但是视 ShMM 的硬件地址进行修改。IP 地址的最低有效位将被设置为硬件地址的最低有效位。在上例中，硬件地址为偶数的 ShMM 的 IP 地址为 `192.168.1.240`，硬件地址为奇数的 ShMM 的 IP 地址为 `192.168.1.241`。通过删除文件 `/etc/readhwaddr` 可以关闭这种 IP 地址修改功能。

RMCP 地址传播

机框管理器的一个可选功能允许对外部网络公开备用 ShMM，但所用的 IP 地址仅在最低有效位上与 RMCP IP 地址不同。备用 ShMM 的网络掩码和默认网关与活动 ShMM 的网络掩码和默认网关相同。例如，如果 RMCP IP 地址为 `192.168.0.2`，则备用 ShMM 将有对应的 IP 地址 `192.168.0.3`，而且网络掩码和默认网关相同。要启用此功能，必须在机框管理器配置文件 (`/etc/shelfman.conf`) 中将机框管理器配置参数 `PROPAGATE_RMCP_ADDRESS` 定义为 `TRUE`。

使用第二个以太网接口

通过第二个网络接口可将机框管理器与其中一个 ATCA 网络集线器板相连接。基于双 USB 的网络接口用于冗余机框管理器之间的通信。

将双 USB 网络接口用于冗余通信

在 ShMM 上，通过两个 USB 连接实现了另外两个网络接口。在此配置中，它们始终与两个冗余机框管理器相连接。这些接口名为 `usb0` 和 `usb1`。接口 `usb0` 始终存在，而接口 `usb1` 仅当接口 `usb0` 在对等机框管理器上处于活动状态时（这意味着对等机框管理器已实际安装并在运行）才存在。此外，接口采用交叉连接，即第一个机框管理器上的 `usb0` 连接到第二个机框管理器上的 `usb1`，反之亦然。

机框管理器支持将 USB 网络接口用于冗余机框管理器之间的通信。要使用此功能，必须在机框管理器配置文件 `/etc/shelfman.conf` 中定义两个冗余网络适配器，如下所示：

```
REDUNDANCY_NET_ADAPTER = "usb0"  
REDUNDANCY_NET_ADAPTER2 = "usb1"
```

定义冗余网络接口的子网掩码时，还有一个注意事项。传统情况下，仅当使用了一个冗余网络适配器时，才会从 `/etc/shelfman.conf` 中指定的冗余 IP 地址派生出两个不同的 IP 地址。它们被指定给冗余连接的两个端点，而且仅在最低有效位上有区别。

但是，如果使用了两个冗余网络适配器，则会用到四个不同的 IP 地址，每个端点一个地址（两个冗余机框管理器中的每个机框管理器有两个端点）。为确保正常运行，同一机框管理器上的两个端点（`usb0` 和 `usb1`）必须属于不同的逻辑网络，而一个机框管理器上的 `usb0` 和另一个机框管理器上的 `usb1` 必须属于相同的逻辑网络。基于以上考虑，另外两个 IP 地址是通过切换 `/etc/shelfman.conf` 中指定的冗余 IP 地址中子网掩码的最低有效位派生的。因此，子网掩码必须比指定的冗余 IP 地址类的默认值更为精确。如果未指定子网掩码，默认情况下会将其设置为 `255.255.255.128`；如果是为了获得冗余而使用了 USB 网络接口，它也是 `/etc/shelfman.conf` 中此参数的推荐值。

下面是为 USB 网络接口派生 IP 地址的示例。

假定在 `/etc/shelfman.conf` 中进行了以下定义：

```
REDUNDANCY_IP_ADDRESS = 192.168.1.2
REDUNDANCY_NETMASK    = 255.255.255.128
```

在硬件地址为偶数的 ShMM 上，IP 地址的指定如下所示：

```
usb0:192.168.1.2（不变）
usb1:192.168.1.130（切换网络掩码的最低有效位）
```

在硬件地址为奇数的 ShMM 上，IP 地址的指定如下所示：

```
usb0:192.168.1.131（切换 IP 地址和网络掩码的最低有效位）
usb1:192.168.1.3（切换 IP 地址的最低有效位）
```

更改默认的 ShMM 网络参数

如果将 ShMM 配置为在特定网络环境下运行，需要更改以下网络参数：

- RMCP IP 地址
- RMCP 网关地址
- RMCP 网络掩码

更改 RMCP 网络参数是一个多步过程。必须先更新 U-Boot 网络环境变量，然后必须通过机框管理器 CLI 更新已引导的活动 ShMM 模块网络设置。

▼ 更改默认的 ShMM 网络参数

1. 将串行端口控制台连接到 ShMM 模块。

控制台配置通常是 115200 波特，N/8/1。重新引导 ShMM 载体并按空格键中断自动引导进程。此时会出现以下内容：

```
U-Boot 1.1.2 (Nov 11 2005 - 11:32:08)

CPU: Au1550 324 MHz, id: 0x02, rev: 0x00
Board: ShMM-500
S/N: 00 00 00 00 00 00 00 00 03 03 03
DRAM: 128 MB
Flash: 64 MB
In:    serial
Out:   serial
Err:   serial
Net:   Au1X00 ETHERNET
Hit any key to stop autoboot:  0
ShMM #
```

2. 回显当前的网络设置。

```
ShMM # printenv rmcpaddr netmask gateway
rmcpaddr=192.168.0.44
netmask=255.255.255.0
gateway=192.168.0.1
ShMM #
```

3. 更改设置并提交给非易失性存储。

```
ShMM # setenv rmcpaddr 10.1.1.10
ShMM # setenv netmask 255.255.0.0
ShMM # setenv gateway 10.1.1.1
ShMM # saveenv
Un-Protected 1 sectors
Erasing sector 0 ...Erasing sector at 0x 800000
ok.
Saving Environment to EEPROM...done.
ShMM #
```

4. 将 ShMM 引导至完全运行状态，并以用户 root 身份登录。

```
ShMM # reset

U-Boot 1.1.2 (Nov 11 2005 - 11:32:08)

CPU: Aul550 324 MHz, id: 0x02, rev: 0x00
Board: ShMM-500
S/N: 00 00 00 00 00 00 00 00 00 03 03 03
DRAM: 128 MB
Flash: 64 MB
In: serial
Out: serial
Err: serial
Net: AulX00 ETHERNET
Hit any key to stop autoboot: 0
## Booting image at bfb00000 ...
   Image Name: MIPS Linux-2.4.26
   Created:    2005-05-07 17:35:21 UTC
   Image Type: MIPS Linux Kernel Image (gzip compressed)
   Data Size:  843144 Bytes = 823.4 kB
   Load Address: 80100000
   Entry Point: 802bc040
   Verifying Checksum ...OK
   Uncompressing Kernel Image ...OK
## Loading Ramdisk Image at bfc40000 ...
   Image Name: sentry RFS Ramdisk Image
...
...
sentry login: root

BusyBox v0.60.5 (2005.05.07-17:27+0000) Built-in shell (msh)
#
```

5. 让 ShMM 启动。

注 – 在 U-Boot 固件中更改的设置不一定传播到 Linux 环境。这是因为机框管理器必须保留自己的网络配置数据副本，才能对故障转移情形进行管理。

如果这是首次引导机框管理器，或者已将闪存设备复位到引导之前的出厂默认设置，则机框管理器使用 U-Boot 提供的网络设置来设置该网络环境（这样您在 U-Boot 中所做的更改将得以传播）。

否则，需要执行以下步骤在机框管理器环境下配置网络设置。

6. 确认您已登录到活动机框管理器。

您只需使用 `cp1d` 命令在活动机框管理卡上进行更改，该命令同时也会通过冗余接口使用网络配置更改来更新备用机框管理卡。如果您未登录到活动 ShMM，请连接到另一个 ShMM 设备并继续执行步骤 7。

```
# cp1d
CPLD word: E806
    0002h - Local Healthy
    0004h - Switchover Request Local
    0800h - Hot Swap Latch Open
    2000h - Active
    4000h - Interrupt Status
    8000h - Reboot Was Caused By Watchdog
#
```

7. 获取当前的 IP 设置。

```
# clia getlanconfig 1

IPM Sentry Shelf Manager Command Line Interpreter

Authentication Type Support: 0x15 ( None MD5 Straight Password/Key )
Authentication Type Enables:
    Callback level: 0x00
    User level: 0x15 ( "None" "MD5" "Straight Password/Key" )
    Operator level: 0x15 ( "None" "MD5" "Straight Password/Key" )
    Administrator level: 0x15 ( "None" "MD5" "Straight Password/Key" )
    OEM level: 0x00
IP Address: 206.25.139.28
IP Address Source: Static Address (Manually Configured) (0x01)
MAC Address: 00:50:c2:22:50:30
Subnet Mask: 0.0.0.0
IPv4 Header Parameters: 0x40:0x40:0x10
Primary RMCP Port Number: 0x026f
Secondary RMCP Port Number: 0x0298
BMC-generated ARP Control: 0x02
    Enable BMC-generated ARP Response
Gratuitous ARP Interval: 2.0 seconds
Default Gateway Address: 206.25.139.3
Default Gateway MAC Address: 00:00:00:00:00:00
Backup Gateway Address: 0.0.0.0
Backup Gateway MAC Address: N/A
Community String: "public"
Number of Destinations: 16
Destination Type:
    N/A
Destination Address:
    N/A
#
```

8. 更改 IP 设置，如下所示。

```
# clia setlanconfig 1 ip 10.1.1.10

IPM Sentry Shelf Manager Command Line Interpreter
IP set successfully

# clia setlanconfig 1 subnet_mask 255.255.0.0

IPM Sentry Shelf Manager Command Line Interpreter
Subnet Mask set successfully

# clia setlanconfig 1 dft_gw_ip 10.1.1.1

IPM Sentry Shelf Manager Command Line Interpreter
Default Gateway Address set successfully
#
```

设置机框管理器配置文件

机框管理器配置文件 (`shelfman.conf`) 位于 `/etc` 目录中。该文件中的每一行要么是注释行（以 `#` 开头），要么是 `name = value` 对，代表配置参数的赋值。`name` 和 `value` 用等号 (=) 分隔。

配置参数名称不区分大小写。每个配置参数都属于以下类型之一：布尔型、数值、字符串或 IP 地址。

值的格式与配置参数类型相符，如下所示：

布尔型	布尔型可以用字符串 <code>FALSE</code> 或 <code>TRUE</code> 来表示，也可以分别用其数值形式 <code>0</code> 或 <code>1</code> 来表示。
数值	整（可能带符号的）数值；也支持十六进制表示法 <code>"0x..."</code> 。
字符串	一个带引号或不带引号（采用双引号 <code>"</code> ）的字符串。带引号的字符串可以包含空白字符；不带引号的字符串以第一个空白字符结尾。为每个面向字符串的配置参数单独指定最大字符串大小。
IP 地址	采用点分十进制 (<code>xxx.xxx.xxx.xxx</code>) 表示法的 IP 地址。

使用 `$envvar` 表示法可以将环境变量值指定为配置参数值；在这种情况下，读取配置文件时将替换变量 `envvar` 的值。例如：

```
DEFAULT_RMCP_IP_ADDRESS = $IPADDR
```

初启机框管理器之后，IP 地址与 IPMI LAN 配置参数一起存储。可以通过任一 RMCP 接口或 CLI 机框外部接口来访问或修改 LAN 配置参数，在机框管理器重新启动后，将优先考虑 LAN 配置参数，再考虑 `shelfman` 配置文件。这是为了确保通过这些接口对 LAN IP 地址和网关所做的任何修改的一致性。但是，如果机框 FRU 信息中的机框管理器 IP 连接记录包含 IP 地址，则它优先于机框外部的所有其他设置或 RMCP IP 地址。建议机框 FRU 信息不要指定此地址或将其设置为 0.0.0.0，以便确保可以通过机框管理器配置文件和 IPMI LAN 配置参数来控制此地址。

当前支持以下配置参数：

表 2-2 机框管理器配置参数

名称	类型	默认值	说明
<code>2_X_SYSTEM</code>	布尔型	None	如果指定了此参数，它会将当前系统明确指定为 AdvancedTCA（如果为 FALSE）。如果未指定此参数 (TRUE)，将自动选择系统类型。建议不要指定此参数，除非需要覆盖用于系统类型的错误硬件检测算法。
<code>ALARM_CUTOFF_TIMEOUT</code>	数值	600 秒 (5 分钟)	报警切断超时（超过该时间后将取消激活报警切断），以秒为单位。
<code>ALLOW_CLEARING_CRITICAL_ALARM</code>	布尔型	FALSE	如果设置为 TRUE，则可以通过 CLI 命令 <code>clia alarm clear</code> 来清除紧急报警状态。
<code>ALTERNATE_CONTROLLER</code>	布尔型	TRUE	使用 <code>address = ShMM hardware address</code> 的机框管理器上的备用控制器。
<code>AUTO_SEND_MESSAGE</code>	布尔型	TRUE	自动将发送到非机框管理器 IPMB 地址的 RMCP 请求转换为定向到该地址的 Send Message 请求。
<code>CARRIER</code>	字符串 (16)	PPS	安装了 ShMM 的特定载体板的名称。
<code>CARRIER_OPTIONS</code>	字符串 (256)	""	特定于载体的选项；针对每个支持的载体分别定义。
<code>CONSOLE_LOGGING_ENABLED</code>	布尔型	FALSE	向启动了机框管理器的控制台输出日志消息。
<code>COOLING_IGNORE_LOCAL_CONTROL</code>	布尔型	FALSE	不对风扇设备使用本地控制功能；机框管理器会明确管理风扇级别。
<code>COOLING_POLL_TIMEOUT</code>	数值	30 秒	连续调用冷却监视和管理线程之间的最长时间（以秒为单位）。
<code>CTCA_FRU_RESET_TIMEOUT</code>	数值	N/A	N/A
<code>CTCA_HEALTHY_TIMEOUT</code>	数值	N/A	N/A
<code>CTCA_INITIAL_FAN_LEVEL</code>	数值	N/A	N/A

表 2-2 机框管理器配置参数 (续)

名称	类型	默认值	说明
DEFAULT_GATEWAY_IP_ADDRESS	IP 地址	None	如果在 IPMI LAN 配置参数中将通道 1 的相应参数设置为 0.0.0.0, 则为用于机框外部 (基于 RMCP 的) 通信的网关所用的默认 IP 地址。如果 LAN 配置参数中提供了非零网关 IP 地址, 则忽略机框管理器配置文件中提供的值。
DEFAULT_RMCP_IP_ADDRESS	IP 地址	None	用于机框外部 (基于 RMCP 的) 通信的默认 IP 地址; 该地址在机框管理器冗余实例之间切换。仅当 IPMI LAN 配置参数中将通道 1 的相应参数设置为 0.0.0.0 并且机框 FRU 信息中的机框管理器 IP 连接记录所含 IP 地址设置为 0.0.0.0 时, 才使用此 IP 地址。如果 LAN 配置参数和/或机框 FRU 信息中提供了非零 IP 地址, 则忽略机框管理器配置文件中提供的值。
DEVICE_POLL_TIMEOUT	数值	10 秒	机框管理器使用 Get Device ID 命令连续轮询 IPMB 设备之间的时间间隔 (以秒为单位)。
EXIT_IF_NO_SHELF_FRU	布尔型	FALSE	如果为 TRUE, 则在找不到机框 FRU 时机框管理器退出 (有可能使 ShMM 复位)。
INITIAL_FAN_LEVEL	数值	5	机框管理器应用于风扇托盘的初始风扇级别。通常风扇级别值的范围是 0 - 15, 其中 0 是最慢的风扇速度, 15 是最快的风扇速度。
IPMB_ADDRESS	数值	0	机框管理器的 IPMB 地址, 可覆盖硬件地址。值为 0 会使机框管理器从硬件读取硬件地址, 并将 IPMB 地址设置为硬件地址 * 2。
IPMB_RETRIES	数值	3	如果没有收到针对 IPMB 请求的响应, 则在最终放弃之前尝试重新发送 IPMB 请求的次数。
IPMB_RETRY_TIMEOUT	数值	4 秒	在发送 IPMB 请求之后且在重试该请求之前, 机框管理器等待响应的的时间。
LOCAL_SHELF_FRU	布尔型	TRUE	在公开机框 FRU 信息 (从 /var/nvdata/shelf_fru_info 文件获得) 的机框管理器上创建本地 FRU 1。
M7_TIMEOUT	数值	-1 (秒)	FRU 处于 M7 状态的最长时间 (以秒为单位); 超过此时间后, FRU 将自动转到 M0 状态。-1 (默认值) 代表永久。将此参数设置为 0 可完全阻止 FRU 进入 M7 状态。
MAX_ALERT_POLICIES	数值	64	可用的 PEF 警报策略的最大数目。
MAX_ALERT_STRINGS	数值	64	可用的 PEF 警报字符串的最大数目。
MAX_DEFERRED_ALERTS	数值	32	未解决的 PEF 警报的最大数目。
MAX_EVENT_FILTERS	数值	64	可用的 PEF 事件过滤器的最大数目。

表 2-2 机框管理器配置参数 (续)

名称	类型	默认值	说明
MAX_EVENT_SUBSCRIBERS	数值	64	可同时订阅以便从机框管理器接收事件通知的实体的最大数目。
MAX_EVENT_SUBSCRIBER_IDLE_TIME	数值	60 秒	事件订阅者的最大超时时间 (以秒为单位), 介于事件到达的时刻与事件订阅者从机框管理器检索该事件的时刻之间。如果超过此超时时间, 将视订阅者无效并自动注销该订阅者。
MAX_PENDING_EVENT_NOTIFICATIONS	数值	1024	每个活动订阅者未解决的事件通知的最大数目。
MAX_PENDING_IPMB_REQUESTS	数值	64	等待响应的待定 IPMB 请求的最大数目。
MAX_SEL_ENTRIES	数值	1024	系统事件日志 (system event log, SEL) 中的最大条目数。
MAX_SESSIONS	数值	32	同时出现的 IPMI 会话的最大数目。
MAX_USERS	数值	32	最大 IPMI 用户数。
MIN_FAN_LEVEL	数值	1	最低风扇级别; 自动控制风扇级别时, 冷却管理代码不能将任何风扇的风扇级别降低到此值以下。
MIN_SHELF_FRUS	数值	2	机框中机框 FRU 的最小数目, 机框管理器必须检测到这些 FRU 才能成功启动。
PHYSICAL_SENSORS	布尔型	TRUE	基于 ADM1026 和 LM75 芯片上的物理传感器创建 IPMI 传感器。
POWER_UNLISTED_FRUS	布尔型	TRUE	允许未列在机框 FRU 信息的电源管理表中的 FRU 激活并加电。
PROPAGATE_RMCP_ADDRESS	布尔型	FALSE	如果为 TRUE, 则活动机框管理器将 RMCP IP 地址传播到备用机框管理器, 备用机框管理器使用该 IP 地址配置由 RMCP_NET_ADAPTER 变量指定的网络接口, 但是会转换最低有效位。
REDUNDANCY_ENABLED	布尔型	TRUE	以冗余模式运行机框管理器。
REDUNDANCY_NET_ADAPTER	字符串 (16)	usb0	用于在机框管理器冗余实例之间进行通信的网络适配器的名称。
REDUNDANCY_NET_ADAPTER2	字符串 (16)	usb1	用于在机框管理器冗余实例之间进行通信的第二个网络适配器的名称 (如果将双 USB 网络接口用于此通信目的)。
REDUNDANCY_NETMASK	数值	0	指定给冗余 IP 地址的网络掩码; 默认情况下 (如果为 0), 网络掩码由 IP 地址类自动确定。
REDUNDANCY_PORT	数值	1040	用于在机框管理器冗余实例之间进行交互的 TCP 端口。

表 2-2 机框管理器配置参数 (续)

名称	类型	默认值	说明
REDUNDANT_IP_ADDRESS	IP 地址	None	用于冗余通信的 IP 地址。此地址实际上指定一对 IP 地址, 这对 IP 地址仅在最低有效位上有区别。根据硬件地址将这对 IP 地址指定给冗余机框管理器。
RESERVATION_RETRIES	数值	10	机框管理器重试 Reserve Device SDR 命令的最大次数。
RMCP_NET_ADAPTER	字符串 (16)	eth0	用于基于 RMCP 的通信的网络适配器的名称。
RMCP_NET_ADAPTER2	字符串 (16)	None	如果硬件支持交叉连接链路, 则为用于基于 RMCP 的通信的备用网络适配器的名称。
SDR_READ_RETRIES	数值	3	机框管理器重试 Read Device SDR 命令的最大次数。
SEL_HIGH_WATERMARK	数值	0	控制 SEL 自动清除的算法的高水位标志 (high watermark); 如果 SEL 中可用条目的实际百分比低于此值, 或者 SEL 溢出, 则机框管理器会启动清除 SEL 旧记录的线程 (按存在时间由长至短的顺序清除)。
SEL_LOW_WATERMARK	数值	0	控制 SEL 自动清除的算法的低水位标志 (low watermark); 如果启动了从 SEL 清除旧记录的线程, 则会一直清除记录, 直到 SEL 中所用条目的百分比降到此值以下。
SHELF_FRU_IN_EEPROM	布尔型	TRUE	如果为 TRUE, 则按特定于载体的方式从底板上的 EEPROM 检索机框 FRU 信息; 如果为 FALSE, 则从闪存文件系统上的某个文件获取机框 FRU 信息。
SHELF_FRU_TIMEOUT	数值	5 秒	初始化期间机框管理器等待检测到机框 FRU 信息设备的时间间隔。
SHORT_SEND_MSG_RESPONSE	布尔型	TRUE	确定由机框管理器提供的 Send Message 响应的类型: PICMG 3.0 ECR 所要求的类型 (如果为 TRUE) 或者与早期版本的机框管理器兼容的类型 (如果为 FALSE)。
SWITCHOVER_TIMEOUT_ON_BROKEN_LINK	数值	10 秒	此参数影响当机框管理器与系统管理器之间的物理网络链路 (RMCP 链路) 中断时, 机框管理器何时启动切换或者是否启动切换。如果链路在超过此参数指定的秒数后仍然处于中断状态, 则进行切换; 如果链路在此超时时间内恢复, 则不进行切换。如果此参数的值为 -1, 则 RMCP 链路中断时不进行自动切换。
SYSLOG_LOGGING_ENABLED	布尔型	TRUE	向系统日志输出日志消息。
TASKLET_RETRIES	数值	3	在最终放弃之前, 每个机框管理器任务队列 (激活、取消激活、获取信息) 重试的次数。

表 2-2 机框管理器配置参数 (续)

名称	类型	默认值	说明
VERBOSITY	数值	7	机框管理器详细级别。
VERIFY_SHELF_FRU_CHECKSUM	布尔型	TRUE	启用对机框 FRU 信息记录中校验和的验证；如果设置为 FALSE，则机框管理器忽略校验和。
WATCHDOG_ENABLED	布尔型	TRUE	使用 CPLD 支持的硬件监视程序计时器。

默认情况下，初启 ShMM 时会自动使用配置文件变量。默认的配置文件的变量将导入以下几个由 U-Boot 设置的环境变量：

\$CARRIER_OPTIONS	默认的特定于载体的选项
\$IPADDR	默认的 RMCP IP 地址
\$IPDEVICE	默认的 RMCP 网络适配器
\$IP1ADDR	默认的冗余 IP 地址
\$IP1DEVICE	默认的冗余网络适配器
\$GATEWAY	用于 RMCP 通信的默认网关

如果需要，可以将机框管理器重置为出厂默认参数值。[代码示例 2-1](#) 显示了默认配置文件副本：

代码示例 2-1 默认的 shelfman.conf 文件

```
# /etc/shelfman.conf
#
# This is the PPS Shelf Manager configuration file.
# Copyright (c) 2005 Pigeon Point Systems.
# All rights reserved.
#
# CARRIER: This parameter is the name of the carrier-specific module to use.
# Default is PPS.
CARRIER = $CARRIER
#
# CARRIER_OPTIONS: This parameter specifies the carrier-specific options.
# Default is an empty string.
CARRIER_OPTIONS = $CARRIER_OPTIONS
#
# ALTERNATE_CONTROLLER: This parameter of boolean type specifies whether to
# use the alternate controller on the Shelf Manager with the address
# equal to the ShM hardware address. Default is TRUE.
#
ALTERNATE_CONTROLLER = TRUE
```

```
# ALLOW_CLEARING_CRITICAL_ALARM: This parameter of boolean type enables the
# ability to clear the critical alarm condition without the alarm cutoff
# button.Default is FALSE.
#
ALLOW_CLEARING_CRITICAL_ALARM = FALSE

# ALARM_CUTOFF_TIMEOUT: This parameter specifies the time interval in
# seconds for the Shelf Manager to hold the Alarm Cutoff state. Default
# interval is 600 seconds.
#
ALARM_CUTOFF_TIMEOUT = 600

# COOLING_IGNORE_LOCAL_CONTROL: This parameter of boolean type specifies
# whether the Shelf Manager should use local control capabilities on fan
# devices i.e. whether the Shelf Manager should explicitly manage fan
# levels or not. Default is FALSE.
#
COOLING_IGNORE_LOCAL_CONTROL = FALSE

# COOLING_POLL_TIMEOUT: This parameter specifies the maximum time (in
# interval is between subsequent invocations of the cooling monitoring and
# management facility. Default is 30 seconds.
#
COOLING_POLL_TIMEOUT = 30

# DEVICE_POLL_TIMEOUT: This parameter specifies the time (in seconds)
# between subsequent polls of the IPMB-0 devices by the Shelf Manager via
# sending the "Get Device ID" command to them.Default is 10 seconds.
#
DEVICE_POLL_TIMEOUT = 10

# IPMB_ADDRESS: This parameter defines the IPMB address of the Shelf
# Manager's slot. This parameter overrides the hardware address. The default
# value of 0 forces the Shelf Manager to use the hardware address and set its
# IPMB address to hardware address * 2.
#
# IPMB_ADDRESS = 0

# IPMB_RETRIES: This parameter is the number of attempts to re-send an IPMB
# request before finally giving up, if no response is received to this
# request. Default is 3.
#
IPMB_RETRIES = 3
```

代码示例 2-1 默认的 shelfman.conf 文件 (续)

```
# IPMB_RETRY_TIMEOUT: This parameter is the amount of time (in seconds) the
# Shelf Manager waits for a response after sending an IPMB request, before
# retrying it. Default is 4 seconds.
#
IPMB_RETRY_TIMEOUT = 4

# M7_TIMEOUT: This parameter specifies the maximum time interval (in
# seconds for a FRU to stay in M7 state. After the expiration of this time
# the FRU automatically transitions into the M0 state. Default is -1 which
# means "forever". Setting this parameter to 0 completely prevents FRUs from
# going into the M7 state.
#
M7_TIMEOUT = -1

# MAX_ALERT_POLICIES: This parameter specifies the number of available
entries
# in the PEF Alert Policy table. Default is 64.
#
MAX_ALERT_POLICIES = 64

# MAX_ALERT_STRINGS: This parameter specifies the number of available
entries
# in the PEF Alert String table. Default is 64.
#
MAX_ALERT_STRINGS = 64

# MAX_DEFERRED_ALERTS: This parameter sets the maximum number of outstanding
# PEF alerts. Default is 32.
#
MAX_DEFERRED_ALERTS = 32

# MAX_EVENT_FILTERS: This parameter specifies the number of available
entries
# in the PEF Event Filter table.
#
MAX_EVENT_FILTERS = 64

# MAX_OEM_FILTERS: This parameter specifies the number of available entries
# in the PEF OEM Event Filter table. Default is 16.
#
MAX_OEM_FILTERS = 16

# MAX_PENDING_IPMB_REQUESTS: The parameter sets the maximum number of
# pending IPMB requests awaiting response. Default is 192.
#
```

代码示例 2-1 默认的 shelfman.conf 文件 (续)

```
MAX_PENDING_IPMB_REQUESTS = 192

# MAX_SEL_ENTRIES: The parameter defines the SEL capacity in records.
# Default is 1024.
#
MAX_SEL_ENTRIES = 1024

# SEL_HIGH_WATERMARK: This parameter is the "high watermark" for the
algorithm
# algorithm that controls automatic SEL purging. The purging process will
# start when the actual percentage of free entries in SEL falls below this
# value or the SEL is full. During the purge the oldest SEL records are
# removed according their timestamp. Default is 10 percent i.e. start
# purging when SEL is full.
#
SEL_HIGH_WATERMARK = 10

# SEL_LOW_WATERMARK: This parameter is the "low watermark" for the algorithm
# that controls automatic SEL purging. When the SEL purging thread starts
# it removes records one by one until the percentage of remaining occupied
# entries in the SEL falls below this value. Default is 50 percent.
#
SEL_LOW_WATERMARK = 50

# MAX_SESSIONS: This parameter specifies the maximum number of simultaneous
# IPMI sessions.Default 32.
#
MAX_SESSIONS = 32

# MAX_USERS: This parameter specifies the maximum number of IPMI users.
# Default is 32.
#
MAX_USERS = 32

# INITIAL_FAN_LEVEL: This parameter specifies the initial fan level that the
# Shelf Manager applies to fan trays. Usually fan levels values are in
# 0..15 range where 0 is the slowest, and 15 is the fastest possible fan
# speed. This parameter has an alias CTCA_INITIAL_FAN_LEVEL for CompactPCI
# systems. Default is 5.
#
INITIAL_FAN_LEVEL = 5

# MIN_FAN_LEVEL: This parameter specifies the minimal fan level that can be
# set by the Cooling Management. Default is 0.
#
```

代码示例 2-1 默认的 shelfman.conf 文件 (续)

```
MIN_FAN_LEVEL = 1

# PHYSICAL_SENSORS: This parameter of boolean type specifies whether the
# Shelf Manager should create IPMI sensors based on physical sensors hosted
# by ADM1026 and LM75. Default is TRUE.
#
PHYSICAL_SENSORS = TRUE

# POWER_UNLISTED_FRUS: This parameter of boolean type specifies whether the
# Shelf Manager should power up and activate FRU devices that are not listed
# in the Power Management table of the Shelf FRU Information. Default is
# TRUE.
#
POWER_UNLISTED_FRUS = TRUE

# AUTO_SEND_MESSAGE: This parameter of boolean type specifies whether to
# auto-convert RMCP requests targeting a non-ShM IPMB address into "Send
# Message" requests directed to that address. Default is TRUE.
#
AUTO_SEND_MESSAGE = TRUE

# SHORT_SEND_MSG_RESPONSE: This parameter of boolean type determines the
# type of response on the Send Message command provided by the Shelf
# Manager:required by the PICMG 3.0 R1.0 ECN-001 if TRUE or compatible with
# previous versions of the Shelf Manager if FALSE. Default is TRUE.
#
SHORT_SEND_MSG_RESPONSE = TRUE

# SDR_READ_RETRIES: This parameter sets the number of times the Shelf
# Manager retries the "Read Device SDR" command. Default is 3.
#
SDR_READ_RETRIES = 3

# RESERVATION_RETRIES: This parameter specifies the number of times the
# Shelf Manager retries the "Reserve Device SDR" command. Default is 10.
#
RESERVATION_RETRIES = 10

# TASKLET_RETRIES: This parameter specifies the number of times each Shelf
# Manager tasklet (activation, deactivation, getting information) is
# retried before finally giving up. The default is 3.
#
TASKLET_RETRIES = 3

# SHELF_FRU_IN_EEPROM: This parameter of boolean type tells the Shelf
```

代码示例 2-1 默认的 shelfman.conf 文件 (续)

```
# Manager if it should use SEEPROMs as the Shelf FRU Info storage. If set
# to FALSE the "/var/nvdata/shelf_fru_info" file contents are used. Default
# is TRUE.
#
SHELF_FRU_IN_EEPROM = TRUE

# LOCAL_SHELF_FRU: This parameter of boolean type specifies whether the
# Shelf Manager should create a local FRU#1 that will expose the Shelf FRU
# Info (obtained from the "/var/nvdata/shelf_fru_info" file). If the Shelf
# FRU Info is acquired from EEPROM as a result of the SHELF_FRU_IN_EEPROM
# set to TRUE then this parameter ignored. Default is TRUE.
#
LOCAL_SHELF_FRU = TRUE

# SHELF_FRU_TIMEOUT: This parameter specifies the time interval (in seconds)
# during which the Shelf Manager detects and reads the Shelf FRU Information
# source devices at initial startup. Default is 15 seconds.
#
SHELF_FRU_TIMEOUT = 15

# MIN_SHELF_FRUS: This parameter specifies the minimum number of valid and
# equal Shelf FRU Information instances that must be found to determine the
# true Shelf FRU Information. Default is 2.
#
MIN_SHELF_FRUS = 2

# EXIT_IF_NO_SHELF_FRU: This parameter of boolean type tells the Shelf
# Manager if it should exit if no valid Shelf FRU Information data is found.
# Default is FALSE.
#
EXIT_IF_NO_SHELF_FRU = FALSE

# VERIFY_SHELF_FRU_CHECKSUM: This parameter boolean type specifies whether
# the Shelf FRU Information record checksums should be validated. The
# default is TRUE.
#
VERIFY_SHELF_FRU_CHECKSUM = TRUE

# WATCHDOG_ENABLED: This parameter of boolean type tells the Shelf Manager
# whether it should use the hardware watchdog timer supported by the CPLD or
# not. The default is TRUE.
#
WATCHDOG_ENABLED = TRUE

# REDUNDANCY_ENABLED: This parameter of boolean type tells Shelf Manager if
```


代码示例 2-1 默认的 shelfman.conf 文件（续）

```
# it should run in redundant mode or not. Default is TRUE.
#
REDUNDANCY_ENABLED = TRUE

# REDUNDANCY_PORT: The parameter specifies the TCP port number used for
# inter-host communications by redundant instances of the Shelf Manager.
# Default is 1040.
#
REDUNDANCY_PORT = 1040

# REDUNDANCY_NET_ADAPTER: This parameter specifies the name of network
# adapter used for communication between redundant ShMMs. Default is eth0 if
# it does not conflict with RMCP_NET_ADAPTER.
#
REDUNDANCY_NET_ADAPTER = $IP1DEVICE

# REDUNDANCY_NET_ADAPTER2: This parameter specifies the name of the second
# network adapter used for communication between redundant ShMMs (if USB
# interface is used for redundancy). By default, this parameter is not
# defined.
#REDUNDANCY_NET_ADAPTER2 = "usb1"

# REDUNDANT_IP_ADDRESS: This parameter specifies the IP address for network
# adapter used for redundant communications. This address actually provides
# a pair of IP addresses that differ in the least significant bit. They are
# assigned to redundant ShMs according to their hardware addresses, so they
# are equal on both ShMs. This parameter has no default value and must
# always be set.
#
REDUNDANT_IP_ADDRESS = $IP1ADDR

# REDUNDANCY_NETMASK: This parameter sets the network mask for the network
# adapter used for redundancy communications. Default is 255.255.255.0
#
# REDUNDANCY_NETMASK = 255.255.255.0

# RMCP_NET_ADAPTER: This parameter specifies the name of network adapter
# used for RMCP-based communications. Default is eth0:1 if it does not
# conflict with REDUNDANCY_NET_ADAPTER.
#
RMCP_NET_ADAPTER = $IPDEVICE

# RMCP_NET_ADAPTER2: This parameter specifies the alternate name of network
# adapter used for RMCP-based communications, if cross-connect links are
# supported by hardware. Undefined by default.
```

代码示例 2-1 默认的 shelfman.conf 文件 (续)

```
#
#RMCP_NET_ADAPTER2 = "eth1"

# DEFAULT_RMCP_IP_ADDRESS: This parameter specifies the default IP address
# for network adapter used for RMCP communications. It is switched over
# between redundant instances of the Shelf Manager. This address is only
# used if no IP address is set in the LAN Configuration Parameters for
# channel # 1. Default is the REDUNDANT_IP_ADDRESS parameter value.
#
DEFAULT_RMCP_IP_ADDRESS = $RMCPADDR

# PROPAGATE_RMCP_ADDRESS: This parameter specifies whether the RMCP IP
# address should be propagated to the backup Shelf Manager. If set, the
# backup Shelf Manager configures its network interface specified by
# RMCP_NET_ADAPTER using given IP address with the least significant bit
# inverted. Default is FALSE.
#
PROPAGATE_RMCP_ADDRESS = FALSE

# DEFAULT_RMCP_NETMASK: This parameter specifies the network mask for
# network adapter used for RMCP communications. Default is 255.255.255.0
#
# DEFAULT_RMCP_NETMASK = 255.255.255.0

# DEFAULT_GATEWAY_IP_ADDRESS: This parameter specifies the default gateway
# IP address used for RMCP-based communications. It should be equal for the
# redundant instances of the Shelf Manager. This address is only used if no
# gateway address is set in the LAN Configuration Parameters for channel 1.
# Default is no gateway.
#
DEFAULT_GATEWAY_IP_ADDRESS = $GATEWAY

# SWITCHOVER_TIMEOUT_ON_BROKEN_LINK: This parameter sets the number of
# seconds to wait before switchover if the RMCP link is down, i.e. system
# manager is inaccessible from the shelf manager. A zero value of this
# parameter leads to an immediate switchover on RMCP link fault detection.
# With a -1 value, no automatic switchovers on RMCP link faults will occur.#
# The default value is 10 second.
#
SWITCHOVER_TIMEOUT_ON_BROKEN_LINK = 10

# CONSOLE_LOGGING_ENABLED: This parameter of boolean type enables or
# disables log messages output to the console from which the Shelf Manager
# was started. Default is FALSE.
#
```

代码示例 2-1 默认的 shelfman.conf 文件 (续)

```
CONSOLE_LOGGING_ENABLED = FALSE

# SYSLOG_LOGGING_ENABLED: This parameter of boolean type enables or disables
# logging messages to the syslog facility. Default is TRUE.
#
SYSLOG_LOGGING_ENABLED = TRUE

# VERBOSITY: This parameter sets the Shelf Manager verbosity level. This
# value is actually a bitmask with each bit enabling a corresponding class
# of output messages. The current bit layout has 8 classes:
#
# Errors: 0x01
# Warnings: 0x02
# Information: 0x04
# Verbose Info: 0x08
# Debug Trace Messages: 0x10 (not recommended)
# Verbose Debug Trace: 0x20 (not recommended)
# Demo Messages: 0x40 (not recommended)
# Locks Information: 0x80 (not recommended)
# The default verbosity level is 7 i.e. errors, warnings and information.
#
VERBOSITY = 7

### PICMG 2.x specific settings

# 2_X_SYSTEM: If configured, this parameter explicitly specifies the current
# system as CompactPCI (if TRUE) or AdvancedTCA (if FALSE). If not specified
# the choice of the system type is made automatically. It is not recommended
# to specify this parameter, unless it is necessary to override an incorrect
# hardware detection algorithm for the system type. Default is FALSE.
#
# 2_X_SYSTEM = FALSE

# CTCA_FRU_RESET_TIMEOUT: This parameter specifies the time interval in
# msec which is used to hold the BD_SEL# line low in order to reset a
# CompactPCI board. Default is 500 milliseconds.
#
# CTCA_FRU_RESET_TIMEOUT = 500

# CTCA_HEALTHY_TIMEOUT: This parameter specifies the time interval in
# seconds during which the Shelf Manager waits for the HEALTHY# signal to
# appear after powering on a CompactPCI board. If the board HEALTHY# signal
# is not detected within the specified time, the Shelf Manager will
# deactivate this board. Default is 0 which means endless waiting.
#
```

```
# CTCA_HEALTHY_TIMEOUT = 0
#
### Notification settings

# MAX_EVENT_SUBSCRIBERS: The parameter defines the maximum number of
# entities
# that can simultaneously subscribe to receive event notifications
# from the Shelf Manager.
#
MAX_EVENT_SUBSCRIBERS = 64

# MAX_PENDING_EVENT_NOTIFICATIONS: The parameter defines the maximum number
# of outstanding event notifications for each active subscriber.
#
MAX_PENDING_EVENT_NOTIFICATIONS = 1024

# MAX_EVENT_SUBSCRIBER_IDLE_TIME: This parameter defines the maximum timeout
# for an event subscriber, in seconds, between the moment when an event
# arrives and the moment when the subscriber retrieves this event from the
# Shelf Manager. If this timeout is exceed, the subscriber is considered
# dead and is automatically unregistered.
#
MAX_EVENT_SUBSCRIBER_IDLE_TIME = 60
```

详细级别说明

使用详细级别可以向控制台或系统日志发送更多输出，具体取决于配置参数 `CONSOLE_LOGGING_ENABLED` 和 `SYSLOG_LOGGING_ENABLED` 的设置方式。`VERBOSITY` 配置参数是一个十六进制位掩码，每一位启用一种特定类型消息的输出：

0x01	错误消息
0x02	警告消息
0x04	提示性消息
0x08	详细的提示性消息
0x10	跟踪消息
0x20	详细的跟踪消息
0x40	在 IPM 控制器初始化期间针对发送到 IPM 控制器的重要命令所显示的消息
0x80	有关获得和释放内部锁的详细消息

默认调试级别是 7，此调试级别允许输出错误消息、警告消息和提示性消息。

设置日期和时间

初启系统时，时钟并未设置，因此必须初始化。最初将时钟设置为 1970 年 1 月 1 日。可以通过串行控制台访问日期。

```
# date
Thu Jan  1 03:16:30 UTC 1970
```

要更改日期，请使用 `date` 应用程序键入正确的日期。`date` 命令的格式为 `MMDDHHMMSSYYYY`，其中：

MM	月
DD	日
HH	小时（采用 24 小时制）
MM	分钟
SS	秒
YYYY	年

例如：

```
# date 04291628002003
Tue Apr 29 16:28:00 UTC 2003
```

要使日期永久保留，必须使用 `hwclock` 应用程序存储该日期。

```
# hwclock -systohc
```

在某些情况下，可能会出现错误消息：

```
mktime: cannot convert RTC time to UNIX time
```

可以忽略此错误。它是由于初始日期处于未初始化状态而导致的。

从时间服务器获取日期和时间

如果机框管理卡没有实时时钟 (real-time clock, RTC) 电池，则可以在系统启动期间从时间服务器获取系统日期和时间，之后定期对日期和时间进行同步。所选择的时间服务器必须按照 `rdate` 实用程序的要求通过 TCP 支持 RFC 868。要启用此功能，必须定义 U-Boot 变量 `time_server` 及附加变量 `timezone`（可选）。

`time_server` 变量包含启动后机框管理器向其查询系统时间的服务器的 IP 地址。此变量将作为环境变量 `TIMESERVER` 传播到 Linux 层。如果设置了此变量，则启动脚本 `/etc/netconfig` 将 `/etc/timesync` 脚本作为守护进程启动，后者以无限循环的方式运行并以 300 秒的默认间隔查询时间服务器。要更改此间隔，请编辑 `/etc/timesync` 脚本并更改 `INTERVAL` 变量的值。

注 – 指定 `time_server` 变量后，必须将 `ipldevice` 变量设置为 `usb0` 以便进行正确同步。

变量 `timezone` 包含当前时区的名称，后跟格林尼治标准时间 (Greenwich Mean Time, GMT) 偏移。对于格林尼治以西的时区，偏移为正；对于格林尼治以东的时区，偏移为负。此变量将作为环境变量 `TZ` 传播到 Linux 层。此变量的默认值为 `UTC0`；即国际协调时间 (Universal Coordinated Time, UTC)，与格林尼治时间相匹配。

时间服务器发送的时间为 GMT；如果未设置机框管理器的时区或设置不正确，则从时间服务器获取的时间将被错误地解释。机框管理器不使用时区的由三个字母组成的名称，但是会传播该名称以便设置 Linux 时区。（例如，如果使用时区名称 `XXX0`，则 `date` 命令会生成类似 `Thu Sep 9 21:24:24 XXX 2004` 的输出。）不支持夏令时。

下面是美国东部时间 `timezone` 定义的示例：

```
timezone = EST5
```

其中数字 5 指定时区位于 GMT 以西五小时。可以用任意三个字母替换 `EST`；这些字母用于标识 Linux `date` 命令输出（举例来说）中的时区。

在机框管理卡上设置用户帐户

使用机框管理器的 CLI 即可设置进行 RMCP 访问的用户帐户。在活动机框管理卡上输入用户信息，这些信息随即就可以在备用机框管理卡上镜像或共享。机框管理卡支持 32 个带密码的帐户。

▼ 添加进行 RMCP 访问的用户帐户

1. 登录到活动机框管理卡。
2. 添加用户：

```
# clia user add userid user-name channel-access-flags privilege-level password
```

其中变量参数具有以下含义：

userid - 有效的用户 ID

user-name - 用户名（最多 16 个字符）

channel-access-flag - SetUserInfo 命令的第一个字节（只有第 4、5 和 6 位是有意义的）

- 第 6 位 - 已启用 IPMI 消息传送
- 第 5 位 - 已启用链路验证
- 第 4 位 - 限于回调

privilege-level - 用户权限级别

password - 用户密码（会在不通知的情况下将密码截断为 16 个字符）

下面的示例说明如何添加名称为 root、有管理员权限级别、密码为 PICMG guru 的用户 9。

```
# clia user add 9 "root" 0x40 4 "PICMG guru"
IPM Sentry Shelf Manager Command Line Interpreter
User 9 added successfully
#
# clia user
IPM Sentry Shelf Manager Command Line Interpreter
1: ""
    Channels 0-15 Privilege level: "Administrator"
    Flags: "IPMI Messaging"
9: "root"
    Channels 0-15 Privilege level: "Administrator"
    Flags: "IPMI Messaging"
#
```

有关权限和 `clia user` 命令的更多信息，请参见第 233 页的 "user"。

用户名限制

用户名字段最多可包含 16 个字符。必须至少包含一个小写字母字符，而且第一个字符必须是字母。

username 的有效字符包括：

- 字母字符
- 数字字符
- 句点 (.)
- 下划线 (_)
- 连字符 (-)

密码

密码最多可包含 16 个字符，超出 16 个字符的部分将被截断。

在机框管理器上配置 OpenHPI

机框管理器包括对 OpenHPI 的支持，OpenHPI 是 SA 论坛的硬件平台接口 (Hardware Platform Interface, HPI) 的开放源代码实现。HPI 提供管理计算机硬件的接口，通常面向基于机箱和机架的服务器。使用 SNMP MIB 通过 OpenHPI SNMP 子代理即可访问 HPI。

有两个配置文件需要系统管理员注意：

- `/etc/openhpi.conf` - OpenHPI 配置文件
- `/etc/snmpd.conf` - SNMP 子代理配置文件

`/etc/openhpi.conf` 文件

必须更新 OpenHPI 配置文件 (`/etc/openhpi.conf`) 以便为 ShMM 提供正确的 IP 地址。更新该配置文件后，必须使 ShMM 复位以使更改生效。

▼ 修改 /etc/openhpi.conf 文件

1. 编辑 /etc/openhpi.conf 文件，并将 libipmidirect addr 参数的值更改为 ShMM 的 IP 地址。

/etc/openhpi.conf 文件样例如下所示。

```
OPENHPI_THREADED = "YES"
OPENHPI_UID_MAP = "/var/bin/uip_map"
plugin libipmidirect
handler libipmidirect {
    entity_root = "{SYSTEM_CHASSIS, 1}"
    name = "lan"
    addr = "_____ "
    port = "623"
    auth_type = "{none"
    auth_level = "admin"
    username = "openhpi"
    password = "openhpi"
    MaxOutstanding = "1"
    ActConnectionTimeout = "5000"
    logflags = " "
    logfile = "log"
    logfile_max = "10"
}
```

2. 在提示符下执行 reboot 命令，重新引导 ShMM。

例如：

```
# reboot
```

/etc/snmpd.conf 文件

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和设置陷阱指令。以下几节提供了有关访问控制、SNMPv3 配置和设置陷阱的信息。

访问控制

SNMP 子代理支持 RFC 2575 中定义的基于视图的访问控制模型 (View-Based Access Control Model, VACM)。因此，它识别配置文件中的以下关键字：

- com2sec
- group
- access
- view

此外，它还识别一些易于使用的包装指令：

- rocommunity
- rwcommunity
- rouser
- rwuser

本部分定义如何配置 snmpd 程序以接受各种类型和级别的访问。

```
rouser user [noauth|auth|priv] [OID]
```

```
rwuser user [noauth|auth|priv] [OID]
```

在 VACM 访问配置表中创建 SNMPv3 USM 用户。使用 group、access 和 view 的组合指令更为有效（且功能强大），但上述包装指令简单得多。

用户必须使用的验证和保密最低级别由第一个标记（默认为 auth）来指定。OID 参数可限制用户访问低于指定 OID 的所有内容。

```
rocommunity community [source] [OID]
```

```
rwcommunity community [source] [OID]
```

创建可用于访问代理的只读团体和读写团体。它们是更加复杂且功能更加强大的 com2sec、group、access 和 view 指令行的快速包装指令。它们不像这些指令那样有效，由于没有创建组，因此表可能更大。建议在复杂环境中不要使用这些指令。如果环境相对简单或者可以承受很小的负面性能影响，则可以使用这些指令。

下面的 com2sec 指令部分中介绍了 source 标记的格式。OID 标记可限制团体访问低于指定 OID 的所有内容。

```
com2sec name source community
```

指定从 source/community 对到安全 name 的映射。source 可以是主机名、子网或 default 一词。可以用 IP/掩码或 IP/位的形式指定子网。将选择与传入数据包匹配的 first source/community 组合。

```
group name model security
```

定义从 securitymodel/securityname 到 group 的映射。model 是 v1、v2c 或 usm 之一。

access name context model level prefix read write notify

从 *group/security* 和 *model/security* 级别映射到一个视图。*model* 是 any、v1、v2c 或 usm 之一。*level* 是 noauth、auth 或 priv 之一。*prefix* 指定 *context* 如何与传入 PDU 的环境相匹配, 值为 exact 或 prefix。*read*、*write* 和 *notify* 指定用于相应访问的视图。对于 v1 或 v2c 访问, *level* 是 noauth, *context* 为空。

view name type subtree [mask]

定义有名称的视图。*type* 为 included 或 excluded。*mask* 是十六进制八位字节的列表, 由句点 (.) 或冒号 (:) 分隔。如果没有指定 *mask*, 则默认为 ff。通过使用 *mask*, 能够以相对简单的方式控制对表中一行的访问。例如, 作为 ISP, 您可能考虑赋予每个客户对其自己接口的访问权限:

```
view cust1 included interfaces.ifTable.ifEntry.ifIndex.1 ff.a0
view cust2 included interfaces.ifTable.ifEntry.ifIndex.2 ff.a0

# interfaces.ifTable.ifEntry.ifIndex.1 == .1.3.6.1.2.1.2.2.1.1.1
# ff.a0 == 11111111.10100000
```

这些条目屏蔽并包括行索引, 但仍允许用户变更行的字段。

下面是 VACM 示例:

```
# sec.name source community
com2sec local localhost private
com2sec mynet 10.10.10.0/24 public
com2sec public default public

# sec.model sec.name
group mygroup v1 mynet
group mygroup v2c mynet
group mygroup usm mynet
group local v1 local
group local v2c local
group local usm local
group public v1 public
group public v2c public
group public usm public

# incl/excl subtree mask
view all included .1 80
view system included system fe
view mib2 included .iso.org.dod.internet.mgmt.mib-2 fc

# context sec.model sec.level prefix read write notify
access mygroup "" any noauth exact mib2 none none
access public "" any noauth exact system none none
access local "" any noauth exact all all all
```

SNMPv3 配置

`engineID` *string*

需要使用 `engineID` 配置子代理，以便能够对 SNMPv3 消息做出响应。通过此配置文件行，利用 *string* 来配置 `engineID`。`engineID` 的默认值是使用为计算机主机名找到的第一个 IP 地址来配置的。

`createUser` *username* (MD5 | SHA) *authpassphrase* [DES] [*privpassphrase*]

MD5 和 SHA 是要使用的验证类型，但是您必须在安装了 OpenSSL 的情况下构建软件包，才能使用 SHA。当前支持的唯一保密性协议是 DES。如果未指定 *privpassphrase*，则假定它与 *authpassphrase* 相同。

注 – 如果没有将所创建的用户同时添加到如上所述的 VACM 访问控制表中，那么这些用户是无用的。

注 – 密码短语最少包含 8 个字符。

设置陷阱并通知目标

`trapcommunity` *string*

定义发送陷阱时要使用的默认团体 *string*。请注意，必须在以下任一命令（紧跟在下面列出的三个命令）之前使用此命令，这三个命令专门与此团体字符串一起使用。

`trapsink` *host* [*community* [*port*]]
`trap2sink` *host* [*community* [*port*]]
`informsink` *host* [*community* [*port*]]

定义接收陷阱或通知的主机（`trapsink` 和 `trap2sink` 用于定义接收陷阱的主机，`informsink` 用于定义接收通知的主机）。守护进程启动时，便会发送 Cold Start 陷阱。它还会针对验证失败发送陷阱（如果启用了此项的话）。您可以指定多个 `trapsink`、`trap2sink` 和 `informsink` 行来指定多个目标。使用 `trap2sink` 来发送 SNMPv2 陷阱，并使用 `informsink` 来发送通知。如果未指定 *community*，则使用上述 `trapcommunity` 指令中的字符串。如果未指定 *port*，则使用已知的 SNMP 陷阱端口 (162)。

`trapsess` [*snmpcmdargs*] *host*

较为普通的陷阱配置标记，允许用任一版本的 SNMP 来指定任何类型的陷阱目标。这要求您同时指定 v2c 或 v3 版本号。

▼ 更新 `/etc/snmpd.conf` 文件

1. 编辑 `/etc/snmpd.conf` 文件，根据需要添加、更改或删除指令。
2. 在提示符下执行 `reboot` 命令，重新引导 **ShMM**。

第3章

管理系统

您可以使用机框管理卡命令行界面 (command-line interface, CLI) 来管理系统，也可以使用 RMCP 接口通过以太网来管理系统。

本章包含以下各节：

- 第 52 页的 “机框管理器命令行界面”
- 第 58 页的 “监视系统”
- 第 77 页的 “重新初始化机框管理器”
- 第 80 页的 “对机框管理卡进行重新编程”

IPMI LAN 接口

ATCA 规范要求使用 IPMI LAN 接口，该接口支持通过远程管理控制协议 (Remote Management Control Protocol, RMCP) 在机框管理器中进行 IPMI 消息传送。使用 RMCP 与机框进行通信的系统管理员可以与任何符合 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 IPM 控制器发出 IPMI 命令的能力。

IPMI 命令

PICMG 3.0 (ATCA 规范) 中介绍了标准的 IPMI 命令。该规范还提供了一些自定义 OEM IPMI 命令。Sun Microsystems 为自行设计的 ATCA 板提供了一组独特的 IPMI 命令。表 3-1 列出了这些命令，附录 B 对这些命令进行了介绍。

表 3-1 Sun OEM IPMI 命令

命令	操作码	语法
Get Version	0x80	#GET_VERSION
Set Boot Page [†]	0x81	#SET_BOOT_PAGE
Get Boot Page [†]	0x82	#GET_BOOT_PAGE
Set Front panel reset button state	0x83	#SET_FP_RESET_BUTTON
Get Front panel reset button state	0x84	#GET_FP_RESET_BUTTON_STATE
Set Ethernet Force Front bit	0x85	#SET_ETH_FORCE_FRONT
Get Ethernet Force Front bit	0x86	#GET_ETH_FORCE_FRONT
Get RTM status	0x88	#GET_RTM_PRESENCE

[†] 仅对 Sun Netra™ CP3010 节点板有效

机框管理器命令行界面

可使用机框管理器命令行界面 (command-line interface, CLI) 与机框的智能管理控制器、板和机框管理器本身进行通信 (通过键入一些文本命令)。CLI 是一组基于 IPMI 的命令，可以直接访问这些命令，也可通过更高级别的管理应用程序或脚本来对其进行访问。管理员可以通过 Telnet 连接或机框管理卡串行端口访问 CLI。使用 CLI，操作员可以访问有关机框当前状态的信息，包括当前 FRU 的填充情况、当前传感器值、阈值设置、最新事件和机框总体运行状况。

启动命令行界面

要使用 CLI，请首先登录机框管理卡上的 Linux 操作系统。登录后，从命令行运行带有特定参数的可执行 `clia`。第一个参数是命令动词。可执行 `clia` 位于虚拟 `root` 文件系统中，该文件系统由机框管理卡上运行的 Linux 维护。该可执行 `clia` 可连接到机框管理器软件的主进程，将命令信息传递给它，然后检索结果。启动 CLI 之前，机框管理器必须处于运行状态。

例如：

```
# clia ipmc

IPM Sentry Shelf Manager Command Line Interpreter

20: Entity: (d0, 0) Maximum FRU device ID: 20
    PICMG Version 2.0
    Hot Swap State: M4, Previous: M3, Last State Change Cause: Normal State
    Change (0)
#
```

如果启动时不带有参数，则 `clia` 将进入交互模式。在该模式下，程序将反复向终端发出提示，将用户输入作为下一条带有参数的命令接受，执行该命令，并在终端显示结果，直到用户键入命令 `exit` 或 `quit`。例如：

```
# clia

IPM Sentry Shelf Manager Command Line Interpreter

CLI> ipmc 20

20: Entity: (d0, 0) Maximum FRU device ID: 20
    PICMG Version 2.0
    Hot Swap State: M4, Previous: M3, Last State Change Cause: Normal State
    Change (0)

CLI> exit
#
```

CLI 命令

CLI 可实现以下命令。第 105 页的附录 A “机框管理器 CLI 命令” 中按命令名称的字母顺序对这些命令进行了详细介绍（将每个命令作为一个小节进行介绍）。

表 3-2 机框管理器 CLI 命令汇总

命令	参数	说明
activate	IPMB 地址 FRU 设备 ID	激活指定的 FRU。
alarm	报警类型	激活或清除 Telco 报警。
board	插槽编号（可选）	显示有关板的信息。
boardreset	插槽编号	复位指定的 ATCA 板。
busres	子命令，以及其参数	针对总线型电子钥控 (E-Keying) 管理资源执行指定操作。
console	插槽编号	打开指定插槽中节点板的控制台会话。
deactivate	IPMB 地址 FRU 设备 ID	取消激活指定的 FRU。
debuglevel	新调试级别（可选）	获取机框管理器的当前调试级别或设置新的调试级别。
exit/quit		在交互模式下从解释程序中退出。
fans	IPMB 地址（可选） FRU 设备 ID（可选）	显示有关风扇的信息。
flashupdate	服务器 IP 地址 固件映像的路径名	从指定服务器和路径名下载和更新有关 Netra CP3060 节点板的系统固件。
fru	IPMB 地址（可选） FRU 设备 ID（可选） FRU 类型（可选）	显示有关机框中一个或一组 FRU 的信息；FRU 是按类型或父 IPM 控制器选择的。
frucontrol	IPMB 地址 FRU 设备 ID 选项	向特定 FRU 发送 FRU 控制命令。
frudata	IPMB 地址（可选） FRU 设备 ID（可选） 数据块/字节偏移量（可选） 数据（可选）	提供对指定 FRU 中的 FRU 信息的原始访问权限。

表 3-2 机框管理器 CLI 命令汇总 (续)

命令	参数	说明
frudatar	IPMB 地址 FRU 设备 ID 文件名	读取指定 FRU 的 FRU 数据区, 并将该数据存储在指定文件中。
frudataw	IPMB 地址 FRU 设备 ID 文件名	将指定文件中的 FRU 数据写入指定 FRU 的 FRU 数据区。
fruinfo	IPMB 地址 FRU 设备 ID	提供用户友好的 FRU 信息输出。
getfanlevel	IPMB 地址 (可选) FRU 设备 ID (可选)	显示指定 FRU 所控制风扇的当前级别。
getfruledstate	IPMB 地址 (可选) FRU 设备 ID (可选) LED 指示灯 ID 或 ALL (可选)	显示 FRU LED 指示灯状态。
gethysteresis	IPMB 地址 (可选) 传感器名称 (可选) 传感器编号 (可选)	显示指定传感器的正、负滞后值。
getipmbstate	IPMB 地址 IPMB 链路编号 (可选)	显示目标地址处 IPMB-0 的当前状态。如果指定了链路编号且目标 IPMC 为 IPMB 集线器, 则将显示有关特定链路的信息。
getlanconfig	通道号 参数名或编号 (可选) 已设置的选择器 (可选)	获取和显示某个特定通道的 LAN 配置参数。
getpefconfig	参数名或编号 (可选) 已设置的选择器 (可选)	获取和显示某个 PEF 配置参数。
getsensoreventenable	IPMB 地址 (可选) 传感器名称 (可选) 传感器编号 (可选)	显示指定传感器所支持事件的当前传感器事件掩码值。
getthreshold、 threshold	IPMB 地址 (可选) 传感器名称 (可选) 传感器编号 (可选)	显示有关特定传感器的阈值信息。
help		显示所支持命令的列表。
ipmc	IPMB 地址 (可选)	显示有关机框中一个或全部 IPM 控制器的信息。
localaddress		检索当前机框管理器的 IPMB 地址。
minfanlevel	风扇级别 (可选)	显示或设置最低风扇级别。

表 3-2 机框管理器 CLI 命令汇总 (续)

命令	参数	说明
sel	IPMB 地址 (可选) 项目数量 (可选)	显示目标 IPM 控制器维护的系统事件日志中的一些最新项。
sensor	IPMB 地址 (可选) 传感器名称 (可选) 传感器编号 (可选)	显示有关一个或一组传感器的信息；传感器是按 IPM 控制器地址、编号或名称选择的。
sensordata	IPMB 地址 (可选) 传感器名称 (可选) 传感器编号 (可选)	显示特定传感器的值信息。
sensorread	IPMB 地址 传感器编号	显示特定传感器的原始值信息 (忽略任何描述传感器的传感器数据记录)。
session		显示有关活动 RMCP 会话的信息。
setextracted	IPMB 地址 FRU 设备 ID	通知机框管理器，已从机框中实际取出指定 FRU。
setfanlevel	IPMB 地址 FRU 设备 ID 级别	设置指定 FRU 所控制风扇的新级别。
setfruledstate	IPMB 地址 FRU 设备 ID LED 指示灯 ID 或 ALL LED 指示灯操作 LED 指示灯颜色 (可选)	针对给定 FRU，设置特定 LED 指示灯的状态或全部 LED 指示灯的状态。
sethysteresis	IPMB 地址 传感器名称或传感器编号 要设置的滞后 (pos 或 neg) 滞后值	为指定传感器设置新滞后值。
setipmbstate	IPMB 地址 IPMB 总线名称 (A 或 B) IPMB 链路编号 (可选) 要采取的操作	禁用/启用目标 IPM 控制器上的 IPMB-A 或 IPMB-B (或特定 IPMB 链路)。
setlanconfig	通道 参数名或编号 其他参数	设置指定通道的 LAN 配置参数值。
setlocked	IPMB 地址 FRU 设备 ID 状态	将指定 FRU 的锁定位设置为指定状态 (0 - 解除锁定, 1 - 锁定)。

表 3-2 机框管理器 CLI 命令汇总 (续)

命令	参数	说明
setpefconfig	参数名或编号 已设置的选择器 (可选) 参数值	为 PEF 配置参数设置新值。
setpowerlevel	IPMB 地址 FRU 设备 ID 功率级别或 OFF Copy	设置指定 FRU 的功率级别, 关闭 FRU 的电源, 将所需级别复制到当前级别。
setsensoreventenable	IPMB 地址 传感器名称 传感器编号 全局标志 声明事件掩码 (可选) 取消声明事件掩码 (可选)	更改特定传感器的事件启用掩码。
setthreshold	IPMB 地址 传感器名称 传感器编号 阈值类型 阈值	更改特定传感器的特定阈值 (上限/下限、临界/非临界/不可恢复)。
shelf	子命令, 以及其参数	显示有关机框的常规信息; 可通过几个子命令来设置机框属性和获取有关特定区域的其他信息。
shelfaddress	机框地址字符串 (可选)	获取或设置机框 FRU 信息中地址表的“机框地址”字段。
shmstatus		显示机框管理器的活动/备用状态。
showhost	插槽编号	显示有关 Netra CP3060 板上固件的版本信息。
showunhealthy		显示机框中运行状况不正常的部件。
switchover		切换至备用机框管理器。
terminate		终止机框管理器, 而不重新引导机框管理卡。
user	子命令, 以及其参数	显示有关机框管理器中 RMCP 用户帐户的信息, 并提供一种添加、删除和修改用户帐户的简单方式。
version		显示机框管理器版本信息。

大多数信息命令都支持两种执行模式：简短模式和详细模式，其差别在于所提供的信息量不同。简短模式为默认模式（标准模式）；详细模式是在命令行中使用选项 `-v` 选择的，该选项紧跟命令之后，位于位置参数之前。

监视系统

机框管理器 CLI 提供了许多用于监视系统和显示系统状态的命令。本节介绍用于监视系统的各种方法。有关更多信息，请参见第 54 页的“CLI 命令”或第 105 页的附录 A “机框管理器 CLI 命令”。

显示板信息和 IPMC 信息

板信息包括有关分配给 ATCA 插槽的 IPMB 地址范围内的各个 IPM 控制器的信息，以及有关这些控制器控制的其他各个 FRU 的信息。对于板上带有 IPM 控制器的 PICMG 3.0 系统，IPMB 的地址范围为 82h-A0h。

下文针对以下任务提供了相应示例，其中显示了所用命令及其输出。

- 显示有关服务器中所有板的标准信息
 - 显示有关某个板的详细信息
 - 列出板上的传感器
 - 显示板上传感器的数据
 - 列出服务器中的所有 IPMC
 - 显示有关特定 IPM 控制器的信息
 - 显示有关 IPM 控制器的详细信息
-
- 显示有关服务器中所有板的标准信息

在本示例中，仅提供了物理插槽 1 和 14 中的板。

```
# clia board
IPM Sentry Shelf Manager Command Line Interpreter
Physical Slot # 1
82: Entity: (0xd0, 0x0) Maximum FRU device ID: 0x08
    PICMG Version 2.0
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
82: FRU # 0
    Entity: (0xd0, 0x0)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
```

```

Device ID String: "IPM Sentry 6"
Physical Slot # 14
9c: Entity: (0xd0, 0x0) Maximum FRU device ID: 0x08
    PICMG Version 2.0
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
9c: FRU # 0
    Entity: (0xd0, 0x0)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: "IPM Sentry 6"
#

```

- 显示有关某个板的详细信息

本示例显示了有关物理插槽 14 中的板的详细信息。

```

# clia board -v 14
Physical Slot # 14
9c: Entity: (0xd0, 0x0) Maximum FRU device ID: 0x08
    PICMG Version 2.0
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID: 0x00, Revision: 0, Firmware: 1.01, IPMI ver 1.5
    Manufacturer ID: 00315a (PICMG), Product ID: 0000, Auxiliary
Rev: 01ac1014
    Device ID String: "IPM Sentry 6"
    Global Initialization: 0x0, Power State Notification: 0x0,
Device Capabilities: 0x29
    Controller provides Device SDRs
    Supported features: 0x29
    "Sensor Device" "FRU Inventory Device" "IPMB Event
Generator"
9c: FRU # 0
    Entity: (0xd0, 0x0)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: "IPM Sentry 6"
    Site Type: 0x00, Site Number: 14
    Current Power Level: 0x01, Maximum Power Level: 0x01, Current
Power Consumption: 20.0 Watts
#

```

● 列出板上的传感器

在本示例中，显示了 IPMB 地址为 92 的板上传感器的列表。

```
# clia sensor 92
IPM Sentry Shelf Manager Command Line Interpreter

92: LUN: 0, Sensor # 0 ("Hot Swap")
    Type: Discrete (0x6f), "Hot Swap" (0xf0)
    Belongs to entity: (0xa0, 96) [FRU # 0]
92: LUN: 0, Sensor # 1 ("IPMB Physical")
    Type: Discrete (0x6f), "IPMB Link" (0xf1)
    Belongs to entity: (0xa0, 96) [FRU # 0]

92: LUN: 0, Sensor # 255 ("BMC Watchdog")
    Type: Discrete (0x6f), "Watchdog 2" (0x23)
    Belongs to entity: (0x3, 96) [FRU # 0]

92: LUN: 0, Sensor # 255 ("+12.0V")
    Type: Threshold (0x01), "Voltage" (0x02)
    Belongs to entity: (0x14, 96) [FRU # 0]

92: LUN: 0, Sensor # 255 ("+5.0V")
    Type: Threshold (0x01), "Voltage" (0x02)
    Belongs to entity: (0x14, 96) [FRU # 0]

92: LUN: 0, Sensor # 255 ("+3.3V")
    Type: Threshold (0x01), "Voltage" (0x02)
    Belongs to entity: (0x14, 96) [FRU # 0]

92: LUN: 0, Sensor # 255 ("+2.5V")
    Type: Threshold (0x01), "Voltage" (0x02)
    Belongs to entity: (0x14, 96) [FRU # 0]

92: LUN: 0, Sensor # 2 ("CPU1 Temp")
    Type: Threshold (0x01), "Temperature" (0x01)
    Belongs to entity: (0x3, 96) [FRU # 0]

92: LUN: 0, Sensor # 3 ("CPU2 Temp")
    Type: Threshold (0x01), "Temperature" (0x01)
    Belongs to entity: (0x3, 96) [FRU # 0]

92: LUN: 0, Sensor # 4 ("Inlet Temp")
    Type: Threshold (0x01), "Temperature" (0x01)
    Belongs to entity: (0x3, 96) [FRU # 0]
    Type: Threshold (0x01), "Temperature" (0x01)
    Belongs to entity: (0x3, 96) [FRU # 0]

#
```


- 显示板上传感器的数据

在本示例中，显示了有关 IPMB 地址为 92 的板上 3 号的传感器（用于测量 CPU2 的温度）的信息。

```
# clia sensor 92 3
IPM Sentry Shelf Manager Command Line Interpreter

92: LUN: 0, Sensor # 3 ("CPU2 Temp")
    Type: Threshold (0x01), "Temperature" (0x01)
    Belongs to entity: (0x3, 96) [FRU # 0]
```

- 列出服务器中的所有 IPMC

本示例显示了 ipmc 命令的典型输出。

```
# clia ipmc
IPM Sentry Shelf Manager Command Line Interpreter
10: Entity: (0xf0, 0x60) Maximum FRU device ID: 0x08
    PICMG Version 2.1
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)

20: Entity: (0xf0, 0x1) Maximum FRU device ID: 0x08
    PICMG Version 2.1
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)

82: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x00
    PICMG Version 2.1
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)

88: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x00
    Hot Swap State: M7 (Communication Lost), Previous: M4 (Active),
Last State Change Cause: Communication Lost (0x4)

92: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x00
    PICMG Version 2.1
    Hot Swap State: M4 (Active), Previous: M7 (Communication Lost),
Last State Change Cause: Communication Lost (0x4)

96: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x00
    Hot Swap State: M7 (Communication Lost), Previous: M6
(Deactivation In Progress), Last State Change Cause: Communication
Lost (0x4)
```

```
20: Entity: (0xf0, 0x1) Maximum FRU device ID: 0x08
    PICMG Version 2.1
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
```

- 显示有关特定 IPM 控制器的信息

在本示例中，显示了有关地址为 9C 的 IPM 控制器的基本信息。

```
# clia ipmc 9c
IPM Sentry Shelf Manager Command Line Interpreter
9c: Entity: (0xd0, 0x0) Maximum FRU device ID: 0x08
    PICMG Version 2.0
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
#
```

- 显示有关 IPM 控制器的详细信息

在本示例中，显示了有关地址为 9C 的 IPM 控制器的详细信息。

```
# clia ipmc -v 9c
IPM Sentry Shelf Manager Command Line Interpreter
9c: Entity: (0xd0, 0x0) Maximum FRU device ID: 0x08
    PICMG Version 2.0
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID: 0x00, Revision: 0, Firmware: 1.01, IPMI ver 1.5
    Manufacturer ID: 00315a (PICMG), Product ID: 0000, Auxiliary
Rev: 01ac10ac
    Device ID String: "IPM Sentry 6"
    Global Initialization: 0x0, Power State Notification: 0x0,
Device Capabilities: 0x29
    Controller provides Device SDRs
    Supported features: 0x29
    "Sensor Device" "FRU Inventory Device" "IPMB Event
Generator"
#
```

显示 FRU 信息

可通过发出不带任何参数的 `clia fru` 命令显示有关系统中全部 FRU 的信息，也可通过提供 FRU 的地址和 FRU ID（可选）显示特定 FRU 的信息。有关机箱插槽编号与物理地址和 IPMB 地址的对应关系，请参见第 11 页的“物理地址与逻辑插槽的对应关系”。

在以下示例中，“20: FRU # 1”行中的 20 为中间背板的 IPMB 地址，1 为 FRU 设备 ID。

```
# clia fru
20: FRU # 1
  Entity: (0xf2, 0x60)
  Hot Swap State: M4 (Active), Previous: M3 (Activation In Process),
  Last State Change Cause: Normal State Change (0x0)
  Device ID String: "Shelf EEPROM 1"
```

IPMI FRU 信息布局

图 3-1 显示了 IPMI FRU 信息的组织方式。每个分区提供特定类型的数据。

- 通用标头包含区域偏移量。
- 内部使用区用于存储专有数据。
- 机箱信息区域包含机箱类型、部件号和序列号。
- 板信息区域包含制造商时间戳、制造商、产品名称、部件/序列号。
- 产品信息区域包含制造商、产品名称、部件/序列号、版本。
- 多记录区域包含动态数据。



图 3-1 IPMI FRU 信息布局

环境 FRU

环境 FRU 包括中间背板、机框管理卡、风扇、电源输入模块 (power entry module, PEM) 和机框报警面板 (shelf alarm panel, SAP)。所有环境 FRU 都只具有 IPMI FRU 信息，这些信息由第三方供应商提供和编程。

中间背板 FRU 信息包括 Sun 部件号和第三方部件号。Sun 还会在中间背板 FRU 信息的多记录区域中添加其他系统信息，如插槽、vlan、vtag 和其他数据。中间背板 FRU 信息存储在两个相同的 EEPROM 中。对一个 EEPROM 进行任何更改后会自动在另一个 EEPROM 中进行相同的更改。

刀片 FRU

Netra CT 900 服务器集线器板（位于插槽 7 和 8 中）只具有 IPMI FRU 信息。Sun 节点板带有两个不同的 EEPROM：一个包含 IPMI FRU 信息，另一个包含 Sun FRU 信息。

示例

下文针对以下任务提供了相应示例，其中显示了所用命令及其输出。

- 显示有关机框中所有 FRU 的标准信息
 - 显示有关地址为 9C 的所有 FRU 的标准信息
 - 显示有关地址为 20 的 FRU 1 的详细信息
 - 按原始格式显示 FRU 信息
 - 按用户友好格式显示 FRU 信息
- 显示有关机框中所有 FRU 的标准信息

本示例显示了不带任何参数的 fru 命令所显示的标准 FRU 信息。

```
# clia fru
IPM Sentry Shelf Manager Command Line Interpreter

10: FRU # 0
    Entity: (0xf0, 0x60)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: "ShMM-500"

12: FRU # 0
    Entity: (0xf0, 0x60)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: "ShMM-500"
```

```
20: FRU # 0
    Entity: (0xf0, 0x1)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: "IPM Sentry BMC"

20: FRU # 1
    Entity: (0xf2, 0x60)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: "Shelf EEPROM 1"

20: FRU # 2
    Entity: (0xf2, 0x61)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: "Shelf EEPROM 2"

20: FRU # 3
    Entity: (0x7, 0x6f)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: "SAP Board"

20: FRU # 4
    Entity: (0x1e, 0x0)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: "Fan Tray 0"

20: FRU # 5
    Entity: (0x1e, 0x1)
    Device ID String: "Fan Tray 1" Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)

20: FRU # 6
    Entity: (0x1e, 0x2)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: "Fan Tray 2"

20: FRU # 7
    Entity: (0xa, 0x60)
```

```

Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
Device ID String: "PEM A"

20: FRU # 8
Entity: (0xa, 0x61)
Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
Device ID String: "PEM B"

82: FRU # 0
Entity: (0xa0, 0x60)
Hot Swap State: M7 (Communication Lost), Previous: M4 (Active),
Last State Change Cause: Unknown (0xf)
Device ID String: "ATS1460"

9a: FRU # 0
Entity: (0xa0, 0x60)
Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
Device ID String: "NetraCP-3010"

```

- 显示有关地址为 9C 的所有 FRU 的标准信息

在本示例中，只显示了物理地址为 9c 的所有 FRU 的 FRU 信息。

```

# clia fru 9c
IPM Sentry Shelf Manager Command Line Interpreter
9c: FRU # 0
Entity: (0xd0, 0x0)
Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
Device ID String: "IPM Sentry 6"
#

```

● 显示有关地址为 20 的 FRU 1 的详细信息

在本示例中，显示了物理地址为 20 的 FRU 设备 ID 1 的详细 FRU 信息。

```
# clia fruinfo 20 1
IPM Sentry Shelf Manager Command Line Interpreter
20: FRU # 1, FRU Info
Common Header:      Format Version = 1

Internal Use Area:
  Version = 1
Chassis Info Area:
  Version      = 1
  Chassis Type      = (23)
  Chassis Part Number = 11592-450
  Chassis Serial Number =

Board Info Area:
  Version      = 1
  Language Code      = 25
  Mfg Date/Time      = Jun 16 00:00:00 2005 (4973760 minutes
                        since 1996)
  Board Manufacturer = Schroff
  Board Product Name = ShMM-ACB-III Shelf Manager (Radial
                        IPMB)
  Board Serial Number = 0000001
  Board Part Number   = 21593-251
  FRU Programmer File ID = Schroff_11592450_AA.inf

Product Info Area:
  Version      = 1
  Language Code      = 25
  Manufacturer Name = Schroff
  Product Name      = 12U 14-Slot ATCA Chassis
  Product Part / Model# = 11592-450
  Product Version    = Dual Star (Radial IPMB)
  Product Serial Number = 0000001
  Asset Tag          =
  FRU Programmer File ID = Schroff_11592450_AA.inf

Multi Record Area:
  PICMG Shelf Manager IP Connection Record (ID=0x13)
  Version = 1

  Record Type          = Management Access Record
  Version = 2
```

```

Sub-Record Type: Component Name (0x05)

PICMG Address Table Record (ID=0x10)
    Version = 0

PICMG Backplane Point-to-Point Connectivity Record (ID=0x04)
    Version = 0

PICMG Backplane Point-to-Point Connectivity Record (ID=0x04)
    Version = 0

PICMG Shelf Activation And Power Management Record (ID=0x12)
    Version = 0

PICMG Shelf Power Distribution Record (ID=0x11)
    Version = 0

PICMG Radial IPMB-0 Link Mapping Record (ID=0x15)
    Version = 0

Record Type                = 0xf0 OEM Record
    Version = 2

```

- 按原始格式显示 FRU 信息

在本示例中，按原始格式显示了全部 FRU 和特定 FRU 的 FRU 信息。

```

# clia frudata
IPM Sentry Shelf Manager Command Line Interpreter
10: FRU # 0 Raw FRU Info Data
    FRU Info size: 435
12: FRU # 0 Raw FRU Info Data
    FRU Info size: 435
20: FRU # 0 Raw FRU Info Data
    FRU Info size: 152
20: FRU # 1 Raw FRU Info Data
    FRU Info size: 8192
20: FRU # 2 Raw FRU Info Data
    FRU Info size: 8192
20: FRU # 3 Raw FRU Info Data
    FRU Info size: 2048
20: FRU # 4 Raw FRU Info Data
    FRU Info size: 2048
20: FRU # 5 Raw FRU Info Data
    FRU Info size: 2048

```



```

20: FRU # 6 Raw FRU Info Data
    FRU Info size: 2048
20: FRU # 7 Raw FRU Info Data
    FRU Info size: 2048
20: FRU # 8 Raw FRU Info Data
    FRU Info size: 2048
20: FRU # 254 Raw FRU Info Data
    FRU Info size: 3068
9a: FRU # 0 Raw FRU Info Data
    FRU Info size: 512
#
# clia frudata 20 1 0
IPM Sentry Shelf Manager Command Line Interpreter
20: FRU # 1 Block # 0 Raw FRU Info Data
    FRU Info size: 8192
    01 01 22 24 31 3E 00 49 01 A0 A1 A2 A3 A4 A5 A6
    A7 A8 A9 AA AB AC AD E0 E1 E2 E3 E4 E5 E6 D0 D1

```

● 按用户友好格式显示 FRU 信息

本示例显示了用户友好版本的 FRU 信息。

```

# clia fruinfo 20 1
IPM Sentry Shelf Manager Command Line Interpreter
20: FRU # 1, FRU Info
Common Header:      Format Version = 1

Internal Use Area:
    Version = 1
Chassis Info Area:
    Version      = 1
    Chassis Type          = (23)
    Chassis Part Number   = 11592-450
    Chassis Serial Number =

Board Info Area:
    Version      = 1
    Language Code          = 25
    Mfg Date/Time         = Jun 16 00:00:00 2005 (4973760 minutes
                           since 1996)
    Board Manufacturer    = Schroff
    Board Product Name     = ShMM-ACB-III Shelf Manager (Radial
                           IPMB)
    Board Serial Number    = 0000001
    Board Part Number      = 21593-251
    FRU Programmer File ID = Schroff_11592450_AA.inf

```

Product Info Area:

Version = 1
Language Code = 25
Manufacturer Name = Schroff
Product Name = 12U 14-Slot ATCA Chassis
Product Part / Model# = 11592-450
Product Version = Dual Star (Radial IPMB)
Product Serial Number = 0000001
Asset Tag =
FRU Programmer File ID = Schroff_11592450_AA.inf

Multi Record Area:

PICMG Shelf Manager IP Connection Record (ID=0x13)
Version = 1

Record Type = Management Access Record
Version = 2

Sub-Record Type: Component Name (0x05)

PICMG Address Table Record (ID=0x10)
Version = 0

PICMG Backplane Point-to-Point Connectivity Record (ID=0x04)
Version = 0

PICMG Backplane Point-to-Point Connectivity Record (ID=0x04)
Version = 0

PICMG Shelf Activation And Power Management Record (ID=0x12)
Version = 0

PICMG Shelf Power Distribution Record (ID=0x11)
Version = 0

PICMG Radial IPMB-0 Link Mapping Record (ID=0x15)
Version = 0

Record Type = 0xf0 OEM Record
Version = 2

UNKNOWN Manufacturer ID = 0x303833

显示机框信息

您可以使用带有其有效参数之一的 `clia shelf` 命令显示关键机框 FRU 的 FRU 信息。您还可以获取机框的当前操作数据并修改机框 FRU 信息中的某些字段。有效的 `shelf` 参数包括：

- `cooling_state` 或 `cs`
- `fans_state` 或 `fs`
- `address_table` 或 `at`
- `power_distribution` 或 `pd`
- `power_management` 或 `pm`
- `pci_connectivity` 或 `pcic`
- `ha_connectivity` 或 `ha`
- `h110_connectivity` 或 `h110c`
- `point-to-point_connectivity` 或 `ppc`

有关更多信息，请参见第 206 页的“显示机框 FRU 信息”。

示例

下文针对以下任务提供了相应示例，其中显示了所用命令及其输出。

- [显示机框冷却状态](#)
- [显示机框风扇状态](#)
- [显示地址表](#)
- [显示电源管理信息](#)
- [显示配电信息](#)

● 显示机框冷却状态

本示例显示了用于显示机框冷却状态的命令及输出。

```
# clia shelf cooling_state
IPM Sentry Shelf Manager Command Line Interpreter
  Cooling state: "Normal"

# clia shelf -v cooling_state
IPM Sentry Shelf Manager Command Line Interpreter

  Cooling state: "Normal"
  Sensor(s) at this state: (0x9a,4,0) (0x9a,5,0) (0x10,2,0) (0x9a,3,0)
                           (0x20,120,0) (0x20,121,0) (0x20,122,0) (0x20,123,0)
                           (0x20,200,0) (0x20,201,0) (0x20,240,0) (0x20,241,0)
                           (0x20,242,0)
```

- 显示机框风扇状态

本示例显示了用于显示风扇状态的命令及输出。

```
# clia shelf fans_state
IPM Sentry Shelf Manager Command Line Interpreter
  Fans state: "Normal"

# clia shelf -v fans_state
IPM Sentry Shelf Manager Command Line Interpreter
  Fans state: "Normal"
  Sensor(s) at this state: (0x10,7,0) (0x10,8,0) (0x10,9,0) (0x10,10,0)
                          (0x10,11,0) (0x10,12,0)

#
```

- 显示地址表

本示例显示了用于显示机框地址表的命令及输出。

```
# clia shelf address_table

IPM Sentry Shelf Manager Command Line Interpreter

PICMG Address Table Record (ID=0x10)
  Version = 0
  Shelf Address = 1
  Address Table Entries# = 16
  Hw Addr: 41, Site # 7, Type: "AdvancedTCA Board" 00
  Hw Addr: 42, Site # 8, Type: "AdvancedTCA Board" 00
  Hw Addr: 43, Site # 6, Type: "AdvancedTCA Board" 00
  Hw Addr: 44, Site # 9, Type: "AdvancedTCA Board" 00
  Hw Addr: 45, Site # 5, Type: "AdvancedTCA Board" 00
  Hw Addr: 46, Site # 10, Type: "AdvancedTCA Board" 00
  Hw Addr: 47, Site # 4, Type: "AdvancedTCA Board" 00
  Hw Addr: 48, Site # 11, Type: "AdvancedTCA Board" 00
  Hw Addr: 49, Site # 3, Type: "AdvancedTCA Board" 00
  Hw Addr: 4a, Site # 12, Type: "AdvancedTCA Board" 00
  Hw Addr: 4b, Site # 2, Type: "AdvancedTCA Board" 00
  Hw Addr: 4c, Site # 13, Type: "AdvancedTCA Board" 00
  Hw Addr: 4d, Site # 1, Type: "AdvancedTCA Board" 00
  Hw Addr: 4e, Site # 14, Type: "AdvancedTCA Board" 00
  Hw Addr: 08, Site # 1, Type: "Dedicated ShMC" 03
  Hw Addr: 09, Site # 2, Type: "Dedicated ShMC" 03
```

● 显示电源管理信息

在本示例中，显示了相应命令及其输出。

```
# clia shelf pm
IPM Sentry Shelf Manager Command Line Interpreter
PICMG Shelf Activation And Power Management Record (ID=0x12)
  Version = 0
  Allowance for FRU Activation Readiness: 20 seconds
  FRU Activation and Power Description Count: 19
  Hw Address: 41, FRU ID: 0xfe, Maximum FRU Power Capabilities: 200 Watts
  Shelf Manager Controlled Activation: Enabled
  Delay Before Next Power On: 0.0 seconds

  Hw Address: 42, FRU ID: 0xfe, Maximum FRU Power Capabilities: 200 Watts
  Shelf Manager Controlled Activation: Enabled
  Delay Before Next Power On: 0.0 seconds

  Hw Address: 43, FRU ID: 0xfe, Maximum FRU Power Capabilities: 200 Watts
  Shelf Manager Controlled Activation: Enabled
  Delay Before Next Power On: 0.0 seconds

  Hw Address: 44, FRU ID: 0xfe, Maximum FRU Power Capabilities: 200 Watts
  Shelf Manager Controlled Activation: Enabled
  Delay Before Next Power On: 0.0 seconds

  Hw Address: 45, FRU ID: 0xfe, Maximum FRU Power Capabilities: 200 Wattss
  Shelf Manager Controlled Activation: Enabled
  Delay Before Next Power On: 0.0 seconds

  Hw Address: 46, FRU ID: 0xfe, Maximum FRU Power Capabilities: 200 Watts
  Shelf Manager Controlled Activation: Enabled
  Delay Before Next Power On: 0.0 seconds

  Hw Address: 47, FRU ID: 0xfe, Maximum FRU Power Capabilities: 200 Watts
  Shelf Manager Controlled Activation: Enabled
  Delay Before Next Power On: 0.0 seconds

  Hw Address: 48, FRU ID: 0xfe, Maximum FRU Power Capabilities: 200 Watts
  Shelf Manager Controlled Activation: Enabled
  Delay Before Next Power On: 0.0 seconds

  Hw Address: 49, FRU ID: 0xfe, Maximum FRU Power Capabilities: 200 Watts
  Shelf Manager Controlled Activation: Enabled
  Delay Before Next Power On: 0.0 seconds

  Hw Address: 4a, FRU ID: 0xfe, Maximum FRU Power Capabilities: 200 Watts
  Shelf Manager Controlled Activation: Enabled
  Delay Before Next Power On: 0.0 seconds
```

Hw Address: 4b, FRU ID: 0xfe, Maximum FRU Power Capabilities: 200 Watts
Shelf Manager Controlled Activation: Enabled
Delay Before Next Power On: 0.0 seconds

Hw Address: 4c, FRU ID: 0xfe, Maximum FRU Power Capabilities: 200 Watts
Shelf Manager Controlled Activation: Enabled
Delay Before Next Power On: 0.0 seconds

Hw Address: 4d, FRU ID: 0xfe, Maximum FRU Power Capabilities: 200 Watts
Shelf Manager Controlled Activation: Enabled
Delay Before Next Power On: 0.0 seconds

Hw Address: 4e, FRU ID: 0xfe, Maximum FRU Power Capabilities: 200 Watts
Shelf Manager Controlled Activation: Enabled
Delay Before Next Power On: 0.0 seconds

Hw Address: 44, FRU ID: 0xfe, Maximum FRU Power Capabilities: 200 Watts
Shelf Manager Controlled Activation: Enabled
Delay Before Next Power On: 0.0 seconds

Hw Address: 45, FRU ID: 0xfe, Maximum FRU Power Capabilities: 200 Watts
Shelf Manager Controlled Activation: Enabled
Delay Before Next Power On: 0.0 seconds

Hw Address: 08, FRU ID: 0xfe, Maximum FRU Power Capabilities: 24 Watts
Shelf Manager Controlled Activation: Enabled
Delay Before Next Power On: 0.0 seconds

Hw Address: 09, FRU ID: 0xfe, Maximum FRU Power Capabilities: 24 Watts
Shelf Manager Controlled Activation: Enabled
Delay Before Next Power On: 0.0 seconds

Hw Address: 20, FRU ID: 0xfe, Maximum FRU Power Capabilities: 100 Watts
Shelf Manager Controlled Activation: Enabled
Delay Before Next Power On: 0.0 seconds

#

● 显示配电信息

本示例显示了用于获取机框配电信息的命令及输出。

```
# clia shelf pd
IPM Sentry Shelf Manager Command Line Interpreter
PICMG Shelf Power Distribution Record (ID=0x11)
    Version = 0
    Feed count: 8
    Feed 00:
        Maximum External Available Current: 25.0 Amps
        Maximum Internal Current: 25.0 Amps
        Minimum Expected Operating Voltage: -40.5 Volts
        Actual Power Available: 1012.500 Watts
        Currently Used Power: 30.000 Watts
        Feed-to-FRU Mapping entries count: 3
            FRU Addr: 49, FRU ID: 0xfe
            FRU Addr: 4b, FRU ID: 0xfe
            FRU Addr: 4d, FRU ID: 0xfe
    Feed 01:
        Maximum External Available Current: 25.0 Amps
        Maximum Internal Current: 25.0 Amps
        Minimum Expected Operating Voltage: -40.5 Volts
        Actual Power Available: 1012.500 Watts
        Currently Used Power: 30.000 Watts
        Feed-to-FRU Mapping entries count: 3
            FRU Addr: 49, FRU ID: 0xfee
            FRU Addr: 4b, FRU ID: 0xfe
            FRU Addr: 4d, FRU ID: 0xfe
    Feed 02:
        Maximum External Available Current: 25.0 Amps
        Maximum Internal Current: 25.0 Amps
        Minimum Expected Operating Voltage: -40.5 Volts
        Actual Power Available: 1012.500 Watts
        Currently Used Power: 40.000 Watts
        Feed-to-FRU Mapping entries count: 4
            FRU Addr: 41, FRU ID: 0xfe
            FRU Addr: 43, FRU ID: 0xfe
            FRU Addr: 45, FRU ID: 0xfe
            FRU Addr: 47, FRU ID: 0xfe
    Feed 03:
        Maximum External Available Current: 25.0 Amps
        Maximum Internal Current: 25.0 Amps
        Minimum Expected Operating Voltage: -40.5 Volts
        Actual Power Available: 1012.500 Watts
        Currently Used Power: 40.000 Watts
```

```
Feed-to-FRU Mapping entries count: 4
FRU Addr: 41, FRU ID: 0xfe
FRU Addr: 43, FRU ID: 0xfe
FRU Addr: 45, FRU ID: 0xfe
FRU Addr: 47, FRU ID: 0xfe
Maximum External Available Current: 25.0 Amps
Maximum Internal Current: 25.0 Amps
Minimum Expected Operating Voltage: -40.5 Volts
Actual Power Available: 1012.500 Watts
Currently Used Power: 40.000 Watts
Feed-to-FRU Mapping entries count: 4
FRU Addr: 42, FRU ID: 0xfe
FRU Addr: 44, FRU ID: 0xfe
FRU Addr: 46, FRU ID: 0xfe
FRU Addr: 48, FRU ID: 0xfe
```

Feed 05:

```
Maximum External Available Current: 25.0 Amps
Maximum Internal Current: 25.0 Amps
Minimum Expected Operating Voltage: -40.5 Volts
Actual Power Available: 1012.500 Watts
Currently Used Power: 40.000 Watts
Feed-to-FRU Mapping entries count: 4
FRU Addr: 42, FRU ID: 0xfe
FRU Addr: 44, FRU ID: 0xfe
FRU Addr: 46, FRU ID: 0xfe
FRU Addr: 48, FRU ID: 0xfe
```

Feed 06:

```
Maximum External Available Current: 25.0 Amps
Maximum Internal Current: 25.0 Amps
Minimum Expected Operating Voltage: -40.5 Volts
Actual Power Available: 1012.500 Watts
Currently Used Power: 100.000 Watts
Feed-to-FRU Mapping entries count: 6
FRU Addr: 08, FRU ID: 0xfe
FRU Addr: 09, FRU ID: 0xfe
FRU Addr: 20, FRU ID: 0xfe
FRU Addr: 4a, FRU ID: 0xfe
FRU Addr: 4c, FRU ID: 0xfe
FRU Addr: 4e, FRU ID: 0xfe
```

Feed 07:

```
Maximum External Available Current: 25.0 Amps
Maximum Internal Current: 25.0 Amps
Minimum Expected Operating Voltage: -40.5 Volts
Actual Power Available: 1012.500 Watts
Currently Used Power: 100.000 Watts
```



```
Feed-to-FRU Mapping entries count: 6
FRU Addr: 08, FRU ID: 0xfe
FRU Addr: 09, FRU ID: 0xfe
FRU Addr: 20, FRU ID: 0xfe
FRU Addr: 4a, FRU ID: 0xfe
FRU Addr: 4c, FRU ID: 0xfe
FRU Addr: 4e, FRU ID: 0xfe
```

```
#
```

重新初始化机框管理器

本节介绍如何重新初始化 U-Boot 环境变量、闪存中的文件系统以及机框管理卡中的登录密码。

重新初始化 U-Boot 环境

U-Boot 环境变量存储在机框管理卡的 EEPROM 中。要恢复 U-Boot 环境变量的出厂默认值，必须首先删除存储在 EEPROM 中的环境变量，然后复位机框管理卡（或关闭该卡的电源然后再打开其电源）。

▼ 重新初始化 U-Boot 环境

1. 在 U-Boot 提示符下输入以下命令清除 EEPROM:

```
ShMM # eeeprom write 80400000 0 1000

EEPROM @0x50 write: addr 80400000  off 0000  count 4096 ... done
ShMM #
```

2. 按如下所示复位机框管理卡:

```
ShMM # reset
U-Boot 1.1.2 (Nov 27 2005 - 19:17:09)

CPU: Aul550 324 MHz, id: 0x02, rev: 0x00
Board: ShMM-500
S/N: 8000041
DRAM: 128 MB
Flash: 64 MB
*** Warning - bad CRC, using default environment

In:    serial
Out:   serial
Err:   serial
Net:   AulX00 ETHERNET
Hit any key to stop autoboot:  0

ShMM #
```

3. 使用以下命令保存这些环境设置:

```
ShMM # saveenv
```

重新初始化文件系统

文件系统存储在闪存中, 可以很方便地将其重置为出厂默认值。U-Boot 具有一个称为 `flash_reset` 的环境变量。将该变量设置为 `y` 而后引导系统会将文件系统重新初始化为出厂默认值。

```
ShMM # setenv flash_reset y
ShMM # boot
```

重新初始化闪存后，`flash_reset` 变量会在系统启动时自动设置为 `n`。`bootcmd` 命令将开始引导 Linux 内核。文件系统会在此过程中得以重新初始化。控制台中将显示以下输出。

```
/etc/rc: Mounted /dev/pts
/etc/rc: Flash erase requested via U-BOOT var
/etc/rc: erasing mtdchar1 -> /etc
Erased 1024 Kibyte @ 0 -- 100% complete.
/etc/rc: erasing mtdchar0 -> /var
Erased 1536 Kibyte @ 0 -- 100% complete.
/etc/rc: Mounted /dev/mtdblock3 to /var
/etc/rc: /var/log mounted as FLASH disk
/etc/rc: Started syslogd and klogd
/etc/rc: /var/tmp mounted as RAM disk
/etc/rc: hostname demo
/etc/rc: /dev/mtdblock2 appears to be empty ... restoring from
factory /etc...
```

重新设置登录密码

登录机框管理卡的出厂默认设置为使用不带任何密码的用户 ID `root`。建议用户在配置机框管理器时更改密码。在忘记新密码时，可通过 `password_reset` U-Boot 变量将密码重置为出厂默认值。将该变量设置为 `y` 而后引导系统会删除 `root` 密码。

```
ShMM # setenv password_reset y
ShMM # boot
```

在引导过程中，控制台中将显示以下输出。

```
/etc/rc: hostname demo
```

对机框管理卡进行重新编程

对机框管理卡进行重新编程涉及对该卡中存储的几个固件映像进行重新编程。根据固件映像的重新编程方法，可从概念上把这组映像分成两组。这两组为：

- U-Boot、Linux 内核和 Linux 根文件系统 (root file system, RFS) 映像

通常会使用可靠的升级过程来对这些映像重新编程。也可以通过从 TFTP 服务器装入内核和 RFS 在 U-Boot 提示符下对内核和 RFS 进行重新编程。

- 复杂可编程逻辑器件 (Complex Programmable Logic Device, CPLD) 映像

可使用特殊的命令行实用程序 `cpldtool` 从机框管理卡命令行对这种映像重新编程。

下文提供了有关对这些映像进行重新编程的详细说明。

固件可靠升级过程

Monterey Linux 操作系统为正在运行并起作用的机框管理卡上的固件映像提供了可靠的升级过程。该过程支持 U-Boot 固件、Linux 内核和 Linux RFS（或这三种映像的任意组合）的升级。如果软件升级尝试失败（例如，因安装了有问题的 U-Boot 固件映像、不能引导机框管理卡或者机框管理器不能启动），则可靠升级过程将自动回退到持久性闪存中固件的以前版本。

机框管理卡的闪存分为两个区域。在其中一个区域中建立了一组稳定的固件后，该区域会被指定为持久性区域。安装新固件时，该新固件会进入另一个区域，该区域最初被指定为临时区域。一旦临时区域中的一组新固件得到验证后，该区域将指定为持久性区域并继续使用，直到将来的升级循环重新开始这一过程。

可靠的升级硬件机制可确保，无论在临时闪存中安装了什么样的固件，机框管理卡总是会设法从这样的软件副本进行引导，该软件副本要么可以完全正常工作，要么足够健全，可以确定升级会话中是否出现故障，进而采取适当的更正操作来恢复为持久性闪存中的安全软件副本。

从更高层面来看，可靠升级硬件机制由一个软件协议支持，该软件协议基于将升级会话的状态记录到非易失性文件 `/var/upgrade/status` 来实现（请参见第 82 页的“可靠升级过程状态文件”）。该软件协议可确保，只有在所有要求的操作（包括特定应用程序所需的自定义 `hook` 脚本所定义的操作）成功完成后可靠升级才可结束。

Monterey Linux 中实现的可靠升级过程对在机框管理卡中运行的嵌入式应用程序不产生影响。该过程提供一组足够的“hook”，使机框管理卡上运行的特定应用程序可确保在可靠升级过程的适当时间执行自定义操作。本节其余部分将重点介绍针对使用这些 `hook` 实现的 IPM Sentry 机框管理器固件的可靠升级的置备情况。

闪存分区

机框管理卡提供的硬件机制允许系统内存映射中闪存的上半部分和下半部分进行交换，该交换由 MIPS 处理器中运行的软件控制。实现这一功能是为了支持闪存中软件映像的可靠升级过程。可靠软件升级过程假设闪存设备中包含该软件的两个副本，分别位于闪存的上半部分和下半部分。所有机框管理卡在出厂时都进行了分区，其中闪存设备已分为两个大小均等的部分，每个部分将用于保存机框管理器软件的一个副本。

Linux 层使用 U-Boot 环境变量 `reliable_upgrade`（请参见第 16 页的“U-Boot 环境变量”）来确定是否启用可靠升级过程。必须将该变量设置为 `y`，并以 `bootargs` 内核参数字符串形式将其传递给 Linux 内核。Linux Flash MTD 层将在闪存分区初始化时检查 `reliable_upgrade` 参数，并根据参数值（以及机框管理卡中安装的闪存设备的大小）以适当的方式对闪存设备进行分区。

本节假设已将机框管理卡配置为支持可靠升级，并且其中包含两个独立的闪存区域。表 3-3 提供了采用这种配置 (`reliable_upgrade=y`) 的机框管理卡中所维护的闪存分区的摘要：

表 3-3 16 MB 的闪存分区 (`reliable_upgrade=y`)

闪存中的偏移量 (以 MB 为单位)	大小 (以 MB 为单位)	设备节点	挂载方式 (启动时)	内容
0	0.5	<code>/dev/mtdchar10,</code> <code>/dev/mtdblock10</code>	<code>/var/upgrade</code>	<code>/var/upgrade</code> JFFS2 文件系统的后半部分
$0.5 + (\text{FLASH_SIZE} - 16) / 2$	1.5	<code>/dev/mtdchar5,</code> <code>/dev/mtdblock5</code>	不挂载	另一个 <code>/var</code> JFFS2 文件系统
$\text{FLASH_SIZE} / 2 - 62$	1	<code>/dev/mtdchar6,</code> <code>/dev/mtdblock6</code>	不挂载	另一个 <code>/etc</code> JFFS2 文件系统
$\text{FLASH_SIZE} / 2 - 53$	1	<code>/dev/mtdchar7</code>	不挂载	另一个 Linux 内核映像
$\text{FLASH_SIZE} / 2 - 44$	0.25	<code>/dev/mtdchar8</code>	不挂载	另一个 U-Boot 固件映像
$\text{FLASH_SIZE} / 2 - 3.754.25$	3.75	<code>/dev/mtdchar9</code>	不挂载	另一个 Linux 根文件系统 (root file system, rfs) 映像
$\text{FLASH_SIZE} / 28$	0.5	<code>/dev/mtdchar10,</code> <code>/dev/mtdblock10</code>	<code>/var/upgrade</code>	<code>/var/upgrade</code> JFFS2 文件系统的前半部分
$\text{FLASH_SIZE} - 7.58.5$	1	<code>/dev/mtdchar0,</code> <code>/dev/mtdblock0</code>	<code>/var</code>	<code>/var</code> JFFS2 文件系统
$\text{FLASH_SIZE} - 610$	1	<code>/dev/mtdchar1,</code> <code>/dev/mtdblock1</code>	<code>/etc</code>	<code>/etc</code> JFFS2 文件系统
$\text{FLASH_SIZE} - 5.11$	1	<code>/dev/mtdchar2</code>	不挂载	Linux 内核映像
$\text{FLASH_SIZE} - 412$	0.25	<code>/dev/mtdchar3</code>	不挂载	U-Boot 固件映像
$\text{FLASH_SIZE} - 3.712.255$	3.75	<code>/dev/mtdchar4</code>	不挂载	Linux 根文件系统 (root file system, rfs) 映像

/var/upgrade 文件系统

如第 80 页的“固件可靠升级过程”中所述，如果 `reliable_upgrade` 为 `y`，则 Monterey Linux 会在 `/var/upgrade` 处将 1 MB 分区挂载为 JFFS2 文件系统。该文件系统用于托管可靠升级过程状态文件（请参见第 82 页的“可靠升级过程状态文件”）。

请务必注意，`/var/upgrade` JFFS2 分区由两个非连续闪存块（每个 0.5 MB）组成，一个位于闪存设备的下半部分，一个位于闪存设备的上半部分。Monterey Linux 可利用 Linux MTD 和 JFFS2 层的功能来支持非连续闪存扇区中的文件系统，以便以这种方式实现 `/var/upgrade`。

使 `/var/upgrade` 可用于可靠升级过程的 JFFS2 文件系统的另一个特性是，JFFS2 内部结构不基于闪存扇区数目或闪存中的绝对偏移量创建任何相关项（如链接表）。相反，在分区中挂载文件系统时，JFFS2 会扫描构成分区的所有闪存扇区，并按内部 in-RAM 表示法重新创建文件系统的逻辑内容。该特性可确保，无论从闪存的哪半部分引导 ShMM，Linux 都能将 `/var/upgrade` 挂载为 JFFS2 文件系统，并使用该文件系统以前的内容。

可靠升级过程状态文件

软件可靠升级过程在位于专用文件系统 (`/var/upgrade`) 中的文件 `/var/upgrade/status` 中维护最新升级过程会话的状态，无论 ShMM 从哪个闪存引导，Linux 都会挂载该专用文件系统。如果该文件存在，则它将包含目前正在进行的或最近完成的升级过程会话的状态。

`/var/upgrade/status` 是一个 ASCII 格式的文件，它包含一条或多条记录（每条记录占一行），每条记录用于描述升级过程中特定步骤的状态。记录行的格式如下：

```
<step>: <status>
```

其中，`step` 是 1 至 14（步骤 14 表示升级会话已完成）之间的整数，`status` 是一个人工可读的字符串，用于描述升级过程会话当前步骤的状态。

可靠升级实用程序（请参见第 83 页的“可靠升级实用程序”）使用状态文件来维护基于可靠升级过程硬件机制的软件协议，以便可靠地确定升级过程的状态并根据具体情况继续进行处理。

可靠升级实用程序

为了执行可靠升级过程和检查最新升级的状态，提供了一个特殊用户空间—可靠升级实用程序。

只能通过超级用户 (root) 帐户调用该实用程序。任何通过非超级用户帐户来运行该实用程序的尝试都将被拒绝。

作为实用程序执行的第一步，它会检查 `reliable_upgrade` U-Boot 环境变量（请参见第 81 页的“闪存分区”）是否已设置为 `y`（该 U-Boot 环境变量以内核参数字符串形式传递给 Linux 内核）。如果该检查失败，实用程序会立即终止并退出，同时显示一个相应的错误代码。

如果调用实用程序时带有 `-s`、`-c` 或 `-f` 任一选项，则可使用该实用程序来执行可靠升级过程。在升级过程中，实用程序会将它在升级过程的各个步骤中执行的每个操作的状态都记录到 `/var/upgrade/status` 中。如果实用程序检测到故障，则它会向 `/var/upgrade/status` 中添加一条指示升级过程未成功完成的记录，并退出，同时显示相应的错误代码，从而终止可靠升级过程。

该实用程序可列显输出到 `stdout` 的所有提示性消息。为支持 `-v` 说明符的任何选项提供该说明符可提高提示性消息的详细程度。该实用程序可列显输出到 `stderr` 的所有错误消息。

该实用程序具有以下语法：

- `rupgrade_tool -s {--dst=src}... [--proto=protocol] [-d] [--hook=args] [-v]`
- `rupgrade_tool -c [-v]`
- `rupgrade_tool -f [--hook=args] [-v]`
- `rupgrade_tool -w [-f]`
- `rupgrade_tool -S [-v]`
- `rupgrade_tool -u`
- `rupgrade_tool -h`

其中的参数定义如下：

```
-s {--dst=src}... [--proto=protocol] [--hook=args] [-v]
```

启动可靠升级过程。该步骤需要由机框管理器协同完成，其中包括以下操作：

- 在本地或通过网络获取要复制的映像
- 将映像复制到临时闪存
- 终止 ShMM 上运行的机框管理器实例（如果有）
- 将非易失性数据复制到临时闪存
- 复位机框管理卡并指示其从临时闪存引导

因为最后一个步骤，调用 `rupgrade_tool -s` 通常不返回，而是会复位机框管理卡。如果 `rupgrade_tool -s` 返回，则指示在着手复位机框管理卡以便从临时闪存引导之前，可靠升级过程已失败并已终止。

在实用程序启动升级过程的第一个步骤之前，它将删除 `/var/upgrade/status` 文件（请参见第 82 页的“可靠升级过程状态文件”）。换言之，一旦调用了 `rupgrade_tool -s`，以前升级过程会话的状态（如果有）将丢失，并被新的升级过程会话状态覆盖。

在对 `rupgrade_tool -s` 的调用中，可以有一个或多个 `--dst=src` 说明符。其中每个说明符都定义要安装的升级映像文件的名称，并确定该文件将安装在机框管理卡闪存的什么位置。

`dst` 定义新安装的升级映像的目标位置，可以为以下其中的一个：

- `u` - 升级临时 U-Boot 固件映像分区 (`/dev/mtdchar3`) 中的 U-Boot 映像。
- `k` - 升级临时 Linux 内核映像分区 (`/dev/mtdchar2`) 中的 Linux 内核映像。
- `r` - 升级临时根文件系统映像分区 (`/dev/mtdchar4`) 中的根文件系统映像。

`src` 指定要复制到由 `dst` 指定的临时闪存分区中的升级映像文件。

映像升级的工作方式如下。对于每个指定的 `src` 映像，都将使用指定的复制协议将映像复制到机框管理卡。如果没有提供 `-d` 说明符，则会首先将映像复制到机框管理卡的 RAM 文件系统中（具体而言，是复制到 `/tmp` 目录中），然后将其移到闪存中（即，复制到闪存的目标分区中，然后从 RAM 文件系统中移除）。如果在对 `rupgrade_tool -s` 的调用中提供了 `-d` 说明符，则会跳过复制到 `/tmp` 目录的中间复制过程，直接将映像复制到闪存中的目标位置。此说明符适用于机框管理卡中的运行时内存不足以执行复制到 RAM 文件系统的中间复制过程的情形。

如果没有提供 `-d` 说明符，则可靠升级过程将调用特殊脚本，其主要目的是在将映像复制到 `/tmp` 目录后对其进行验证。如果提供 `-d` 说明符，则不会执行这样的验证。

当前，随机框管理器提供的脚本 `/etc/upgrade/step4vsh` 不执行特定映像验证步骤，但它负责填充在对 `rupgrade_tool` 的当前调用中未提供映像的闪存分区（与在部分升级情形中发生的情况一样）。这些分区将会从当前持久性闪存复制到临时闪存中。例如，如果当前部分升级只提供新 RFS 映像，则该脚本会将 U-Boot 和内核分区从持久性闪存复制到临时闪存中。

一旦将第一个映像安装到其目标位置，实用程序就会继续安装第二个映像（如果有），依此类推，直到将所有提供的映像文件都成功安装到闪存中。如果未能成功安装某个映像，则会立即终止升级过程（而不是跳过失败的映像而继续安装下一个映像）。

这种方法使用户可以单独升级机框管理卡固件的三个组成部分（U-Boot、内核和 RFS 映像）。不过，请记住，会从持久性闪存复制那些未显式更新的部分。

建议使用以下升级方法之一：

- 显式升级所有三个分区。
- 在显式升级的分区少于三个时，省略 `-d` 说明符，在这种情况下，以上提到的特殊脚本会自动确保每个升级实际上都是一个涵盖所有三个分区的完整升级。

protocol 指定用于将每个指定的 *src* 文件拉至机框管理器的文件复制协议，它可为以下任一项：

- **no** - 不执行任何复制。该协议假设，在启动可靠升级过程之前，将所有指定的 *src* 文件都推送至 `/tmp` 目录。此协议选项不能与 `-d` 选项一起使用。
- **cp:dir** - 简单复制。该协议假设，将使用 `cp` 命令从机框管理器本地文件系统中的指定目录复制所有指定的 *src* 文件。该协议可能会很有用，例如，在从已挂载 NFS 的文件系统乃至从 JFFS2 文件系统安装升级映像时可以使用该协议。
- **ftp:server:dir:user[:pwd]** - 从远程 FTP 服务器复制。该协议假设，服务器会将所有指定的 *src* 文件从 FTP 服务器主机（该主机可以按主机名或 IP 地址形式指定）复制到机框管理卡。所有映像都必须驻留在远程 FTP 服务器中 *dir* 指定的目录中。使用用户参数指定的帐户来建立 FTP 连接，其密码由可选 `pwd` 参数指定。如果未提供 `pwd`，则实用程序会提示输入密码。

如果未能将某个映像复制到机框管理器，则会导致实用程序终止升级过程（而不是跳过失败的映像而继续复制下一个映像）。

对于每个通过 `-s` 选项升级的临时闪存分区，将会在检查完映像的有效性之后、将要 *把 src* 映像移至闪存之前，为要升级的分区授予写入权限。在将全部映像移至闪存后，会立即取消分区的写入权限。加之所有包含 U-Boot、Linux 内核和根文件系统映像的分区在引导机框管理器时都为只读这一事实，可以确保应用程序不会意外地删除关键引导映像。

在将所有指定的映像都安装到闪存中其各自的目标位置后，实用程序会调用一个 *hook* 脚本。该脚本可在升级映像已安装到闪存中但升级过程尚未启动可靠升级过程的硬件机制时，通过启用 ShMM 的升级监视程序计时器 (`watchdog timer, WDT`) 来启用应用程序所需的自定义操作。

hook 脚本 `/etc/upgrade/step4hshm` 随机框管理器一起提供。它可执行以下操作：

- 终止机框管理器，在不重新启动机框的情况下切换到备用 ShMM；停止 ATCA 监视程序计时器。
- 挂载临时 `/etc` 和 `/var` 闪存分区并删除其中的所有文件。
- 选择性地 *将 /etc 目录的当前内容复制到临时 /etc 闪存分区。*
- 选择性地 *将当前非易失性机框管理器信息从目录 /var/nvdata 复制到临时 /var 文件系统；或选择性地将整个 /var 目录复制到临时 /var 闪存分区中。*
- 临时（直到下次引导）将引导延迟设置为 0；这样做的目的是最大限度地减少下次引导的时间，并防止可靠升级监视程序计时器过早过期。

将该脚本作为一个子 `shell` 进行调用并为其指定一个参数，该参数可以是 *args* 指定的字符串，也可以不使用 *args*（空字符串）。指定的参数定义脚本的操作模式，该操作模式将确定诸如将非易失性信息从持久性闪存分区复制到临时闪存分区之类的操作。该脚本可采用以下 *args* 值，并执行相应操作：

- 不提供任何参数 - 该脚本会删除临时 `/etc` 目录和临时 `/var` 目录，然后将机框管理器非易失性信息从目录 `/var/nvdata` 复制到临时 `/var` 分区。这是默认的操作模式；在这种情况下，会保留非易失性数据，但将从新 RFS 映像中获取机框管理器配置文件。
- **erase** - 该脚本会删除临时 `/etc` 目录和临时 `/var` 目录；在下次引导过程中会根据 RFS 默认值恢复这两个目录；不会保留当前机框管理器非易失性数据和配置。

- `etc_copy` - 该脚本会删除临时 `/etc` 目录和临时 `/var` 目录，然后将 `/etc` 的内容和 `/var/nvdata` 目录中的非易失性信息复制到临时闪存分区。在这种情况下，非易失性数据和机框管理器配置文件都将得以保留。
- `copy` - 该脚本会删除临时 `/etc` 目录和临时 `/var` 目录，然后将 `/etc` 目录和 `/var` 目录的全部内容复制到临时分区中。在这种情况下，不仅会复制配置也会复制放在 `/var/bin` 中的可执行文件，并覆盖 RFS 映像中同名的可执行文件。如果 `/var/bin` 目录包含一些在升级过程中必须保留的特殊可执行文件（例如，机框管理器的某个特殊版本或其他实用程序），则此操作模式会很有用。

在成功时，该脚本会返回 0，在失败时会返回一个非零值。如果返回非零值，则会终止升级过程。

该实用程序启动升级 WDT 时会具有 12.8 秒的超时时限。该超时时限被认为足以供任何软件在复位后进行引导，以便进入到能够调用 `rupgrade_tool -c`（它可在升级 WDT 处于活动状态时选通升级 WDT）而不必在中间过程选通升级 WDT 这样的状态。该实用程序会在复位 ShMM 之前执行对升级 WDT 的选通操作。

- `-c [-v]`

在从临时闪存引导 ShMM 后继续执行可靠升级过程。从 `/etc/rc` 脚本执行 `rupgrade_tool -c` 调用。如下所述，`rupgrade_tool -c` 所呈现的某些情况指示升级过程出现故障并需要采取一些更正操作，其中包括将导致需要对 ShMM 进行软复位的一些操作。这意味着，调用 `rupgrade_tool -c` 可能不会返回，而是可能会导致复位 ShMM。如果发生复位，则会将 ShMM 恢复为持久性闪存中安装的软件。

如果升级 WDT 处于活动状态，并已在调用 `rupgrade -c` 之前的任一步骤触发，则意味着 ShMM 已恢复为持久性闪存中的软件。在这种情况下，实用程序将禁用升级 WDT，恢复为使用持久性闪存，并终止升级过程。

如果升级 WDT 处于活动状态但未触发，则意味着 ShMM 从临时闪存成功引导（到目前位置）。实用程序将选通升级 WDT，然后退出并显示返回码 0，指示正在进行某个升级过程会话。

如果升级 WDT 处于不活动状态，但 `/var/upgrade/status` 文件的内容指示升级过程仍在进行中，则意味着 ShMM 因为在升级过程的某个步骤关闭电源然后再打开电源而进行了重新引导。这种情况下，实用程序所执行的更正操作与升级 WDT 处于活动状态且已触发的情况所执行的更正操作相同。

最后，如果升级 WDT 处于不活动状态，且 `/var/upgrade/status` 不存在或指示升级过程已结束（可能成功，也可能不成功），则该实用程序将退出，并显示返回值 1，指示未在进行升级过程。

- `-f [--hook=args] [-v]`

完成升级过程。在机框管理器成功完成初始化后，从机框管理器内部执行 `rupgrade_tool -f` 调用。如果机框管理器未自动启动，则会在 `/etc/rc` 脚本结束时执行该调用。

一旦调用 `rupgrade_tool -f` 后，它就会选通升级 WDT，并继续建立新持久性闪存，然后禁用升级 WDT。

在完成之前，该实用程序会使用一条指示成功完成升级过程的记录来更新 `/var/upgrade/status`，然后退出，并显示值 0。

- `-w [-f]`

列显最新升级过程的当前状态。实质上，该选项会将 `/var/upgrade/status` 文件的内容转储到 `stdout` 中。

如果升级过程成功完成，则 `rupgrade_tool -w` 将返回值 0；如果升级过程未成功完成，则会返回值 1；如果不存在 `/var/upgrade/status`，则会返回错误代码。

如果提供了 `-f` 说明符，则 `rupgrade_tool -w` 会在退出之前删除 `/var/upgrade/status` 文件。

- `-S [-v]`

选通升级 WDT。`rupgrade_tool -S` 用作 `shell` 级界面，供新安装的软件使用以验证其健全性。

`rupgrade_tool -S` 将返回值 0。

- `-u`

撤消一个成功的升级会话，恢复到以前的持久性闪存设备。

`rupgrade_tool -u` 可导致 ShMM 重新引导。

- `-h`

列显输出到 `stdout` 的帮助。

可靠升级实用程序使用方案

请按照以下顺序使用可靠升级实用程序以便执行 ShMM 升级：

1. 用户调用 `rupgrade_tool -s` 以启动升级过程。可在本地通过 ShMM 串行控制台执行该调用，也可借助 `telnet`、`rsh`、`ssh` 或任何等效实用程序通过网络远程执行该调用。
2. 用户等待 `rupgrade_tool -s` 重新引导 ShMM。如果用户在本地连接到串行控制台，则可从串行控制台中列显的 U-Boot 固件和 Linux 消息中清楚地看到重新引导状态。如果与 ShMM 远程连接，则重新引导的状态就不那么显而易见。例如，Telnet 连接会在重新引导机框管理卡时超时。用户可假设升级过程已成功执行，也可等待一段时间，即升级会话完成所需的一段时间，然后调用 `rupgrade_tool -w`（也是通过上述的任一种远程 shell 工具执行远程调用）来查明升级会话的状态。等待的时间长短取决于升级映像的大小、用于将映像拉至机框管理卡的复制协议以及映像验证脚本所执行的操作。
3. 在机框管理卡中，启动脚本 `/etc/rc` 将无条件地调用 `rupgrade_tool -c`。如果调用返回值 1（指示没有正在进行升级），或返回错误代码值（指示升级会话已失败），则启动脚本会继续进行正常模式的引导序列。但是，如果返回值 0（指示正在进行一个升级会话），则启动脚本将继续进行新安装软件的健全性验证，并在操作期间调用 `rupgrade_tool -S` 来选通升级 WDT，以防验证所用时间超过升级 WDT 的超时时限，最后将启动机框管理器来执行最终验证。监视程序计时器间隔已设置为 12.8 秒，因此 `/etc/rc` 脚本中调用 `rupgrade_tool -c` 和选通 WDT 之间以及选通 WDT 和启动机框管理器之间的处理时间都不得超过 12.8 秒。

4. 在初始化期间，机框管理器将再次选通升级 WDT，然后再尝试建立与对等机框管理器的网络连接。建立网络连接可能会花费六秒钟时间。建立网络连接并成功完成初始化（这表明新配置有效）后，机框管理器将调用 `rupgrade_tool -f`，它会完成升级过程。
5. 用户可以选择性地调用 `rupgrade_tool -w` 来查明升级会话的状态。如上所述，该选项可能对于远程升级会话尤其有用，在远程升级会话中，不能从串行控制台列显的消息中直接观察到升级的进度，而在本地升级会话中则可以。

完成可靠升级后，如果新映像因某种原因不可接受，用户可将其恢复为原始映像。为此，用户需要调用 `rupgrade_tool -u`。

如有必要，可通过开发一个在远程网络主机上运行的简单脚本，以便轻松地自动执行上述序列。另外，操作员可以在本地通过串行控制台手动执行可靠升级，或通过网络远程手动执行可靠升级。

可靠升级示例

示例 1:

本示例显示了所有三个组成部分（U-Boot、内核和 RFS 映像）的可靠升级，其中会将 `/etc` 和 `/var/nvdata` 非易失性目录复制到临时闪存中。从本地 `/tmp` 获取所有映像（这意味着已用某种未明确指定的方式将映像复制到该处）。从 `/tmp/u-boot.bin` 中获取 U-boot 映像，从 `/tmp/sentry.kernel` 中获取内核映像，从 `/tmp/sentry.rfs` 中获取 RFS 映像。将会从串行控制台启动该升级过程。控制台日志中穿插了一些注释，用于提供有关升级过程的各个步骤的其他背景信息。

首先，从命令提示符下启动 `rupgrade_tool`。其中的参数显示所有三个闪存映像都将更新，保留的机框管理器非易失性数据和配置文件也将更新。

```
# rupgrade_tool -s --k=sentry.kernel --r=sentry.rfs
--u=u-boot.bin --hook=etc_copy -v
rupgrade_tool: PLB is 5
rupgrade_tool: EEPROM page saved
rupgrade_tool: persistent flash is 0
rupgrade_tool: provisional flash is 1
rupgrade_tool: copying image(s)
```

升级实用程序尝试调用验证脚本来检查 `/tmp` 中当前提供的映像。如果未在 `/tmp` 中找到任何指定的文件指示符，该实用程序将停止并生成类似如下的消息。

```
rupgrade_tool: cannot open /tmp/u-boot.bin for reading.
rupgrade_tool: failed to copy images to flash
```

该实用程序继续将映像复制到临时闪存中指定的目标位置。

```
rupgrade_tool: invoking scripts (step4v*) [--u=u-boot.bin --k=
sentry.kernel --r=sentry.rfs --hook=etc_copy]
rupgrade_tool: copying u-boot.bin from /tmp to /dev/mtdchar8 using
'cp' protocol
rupgrade_tool: copying sentry.kernel from /tmp to /dev/mtdchar7
using 'cp' protocol
rupgrade_tool: copying sentry.rfs from /tmp to /dev/mtdchar9 using
'cp' protocol
rupgrade_tool: invoking scripts (step4h*) [etc_copy]
```

此时，已调用 `step4hshm` hook 脚本，该脚本会停止机框管理器并将非易失性信息复制到临时闪存中。

```
/etc/upgrade/step4hshm: Stopping Shelf Manager...
/etc/upgrade/step4hshm: Erasing /var and /etc, copying
/var/nvdata...
/etc/upgrade/step4hshm: Operation: copy /etc and /var/nvdata.
/etc/upgrade/step4hshm: Copying completed.
rupgrade_tool: image(s) copy OK
rupgrade_tool: watchdog started
rupgrade_tool: selected provisional flash
rupgrade_tool: reboot
Restarting system.
```

此时，可靠升级过程将复位 ShMM。这会导致 U-boot 从临时闪存启动。

```
* Resetting Integrated Peripherals

U-Boot 1.1.2 (May 12 2005 - 21:27:13)

CPU: Au1550 324 MHz, id: 0x02, rev: 0x00
Board: ShMM-500
S/N: 8000044
DRAM: 128 MB
Flash: 64 MB
In: serial
Out: serial
Err: serial
Net: Au1X00 ETHERNET
Hit any key to stop autoboot: 0
## Booting image at bfb00000 ...
Image Name: MIPS Linux-2.4.26
Created: 2005-06-24 13:29:50 UTC
Image Type: MIPS Linux Kernel Image (gzip compressed)
```

```

Data Size:      844843 Bytes = 825 kB
Load Address:  80100000
Entry Point:   802bc040
Verifying Checksum ... OK
Uncompressing Kernel Image ... OK
## Loading Ramdisk Image at bfc40000 ...
Image Name:    sentry RFS Ramdisk Image
Created:       2005-04-22  9:10:41 UTC
Image Type:    MIPS Linux RAMDisk Image (gzip compressed)
Data Size:     2465924 Bytes =  2.4 MB
Load Address:  00000000
Entry Point:   00000000
Verifying Checksum ... OK

Starting kernel ...

init started: BusyBox v0.60.5 (2005.06.15-14:45+0000) multi-call
binary
/etc/rc: Mounted /proc
/etc/rc: Mounting filesystems...
/etc/rc: Mounted /dev/pts
/etc/rc: Mounted /dev/mtdblock0 to /var
/etc/rc: Mounted /dev/mtdblock10 to /var/upgrade

```

此时，rc 脚本将调用 `rupgrade_tool -c` 以检查是否正在进行可靠升级。该工具返回 0，确认正在进行一个升级。如果结果是这样，rc 脚本将继续执行启动过程。

```

/etc/rc: Checking the reliable upgrade watchdog timer
/etc/rc: Mounted ram disk to /var/log
/etc/rc: Started syslogd and klogd
/etc/rc: Mounted ram disk to /var/tmp
/etc/rc: Setting hostname shmm+193

```

因为正在进行一个可靠升级，所以将在 rc 脚本中再次选通监视程序计时器。

```

/etc/rc: Strobing the reliable upgrade watchdog timer
/etc/rc: Mounted /dev/mtdblock1 to /etc
/etc/rc: Calling /etc/rc.carrier3
Board Hardware Address: 0xFE
/etc/netconfig: /etc/hosts has valid 192.168.1.193 entry
/etc/netconfig: Updating /etc/profile.sentry with IP settings
/etc/netconfig: ifconfig eth0 192.168.1.193
/etc/netconfig: ifconfig eth1 192.168.0.193
/etc/netconfig: route add default gw 192.168.1.253
/etc/netconfig: Starting inetd...

```

```

/etc/rc.carrier3: Starting up IPMBs ...
/etc/rc.carrier3: Updating /etc/profile.sentry with specific settings
/etc/rc.carrier3: Starting snmpd...
/etc/rc.carrier3: Starting httpd...
/etc/rc.carrier3: Starting Shelf Manager ...
<I> 02:48:08.463 [171] IPM Sentry Shelf Manager ver. 2.0.0. Built
on Jun 27 2005 14:48:57
<*> 02:48:08.469 [171] Limits: code=(400000:506f0), end_data=
10062000, start_stack=7fff7e30, esp=7fff78a0, eip=2ab0d2e4
<*> 02:48:08.469 [171] Stack limits: curr=1ff000, max=7fffffff
<*> 02:48:08.470 [171] Data limits: curr=7fffffff, max=7fffffff
<*> 02:48:08.900 [171] *** Lock log print buffer at 1003b7f0 ***
<*> 02:48:08.900 [171] *** Pthread lock log print buffer at
1003f820 ***

```

机框管理器通过执行 `rupgrade_tool -f` 来启动并完成可靠升级。

```

eth0: link up
eth1: link up
eth1: going to full duplex

shmm+193 login:root

BusyBox v0.60.5 (2005.05.12-22:46+0000) Built-in shell (msh)

```

然后，用户通过发出 `rupgrade_tool -w` 命令来检查可靠升级的状态。

```

# rupgrade_tool -w
Recent upgrade status:
1:PLB is 5
1:EEPROM page saved
2:persistent flash is 1
3:provisional flash is 0
4:copying image(s)
4:invoking scripts (step4v*) [--u=u-boot.bin --k=sentry.kernel --
r=sentry.rfs --hook=etc_copy]
4:copying u-boot.bin from /tmp to /dev/mtdchar8 using 'cp'
protocol
4:copying sentry.kernel from /tmp to /dev/mtdchar7 using 'cp'
protocol
4:copying sentry.rfs from /tmp to /dev/mtdchar9 using 'cp'
protocol
4:invoking scripts (step4h*) [etc_copy]
4:image(s) copy OK
5:watchdog started

```

```
6:selected provisional flash
7:reboot
9:WDT not fired, upgrade in progress.
11:provisional flash 0, updating EEPROM
12:EEPROM updated
13:upgrade WDT disabled
13:invoking scripts (step13h*) []
14:upgrade completed successfully
#
```

示例 2:

本示例仅显示了 RFS 映像的可靠升级，其中会将 /etc 和 /var/nvdata 非易失性目录复制到临时闪存。从 IP 地址为 192.168.1.253 的 FTP 服务器获取 RFS 映像。FTP 服务器中 RFS 映像的路径为 /tftpboot/ru-mips/sentry.mips.rfs。将会从 Telnet 会话启动该升级过程。

注 – 因为只显式更新 RFS 映像，所以会自动将 U-Boot 映像和内核映像从持久性闪存分区复制到临时闪存分区。

本地系统必须可通过网络访问 FTP 服务器（即，其网络适配器必须运行并进行了配置，ShMM 与 FTP 服务器之间必须存在路由）。在本示例中，ShMM 的网络地址配置为 192.168.1.174（它与 FTP 服务器位于同一网络中）：

```
# telnet 192.168.1.174
Trying 192.168.1.174...
Connected to 192.168.1.174.
Escape character is '^]'.

BusyBox on shmm+174 login: root

BusyBox v0.60.5 (2005.05.07-17:27+0000) Built-in shell (msh)
```


rupgrade_tool -s 的参数指示: 仅升级 RFS, 复制协议为 FTP, 访问指定 IP 地址和文件, 使用用户 admin 且未提供密码。

```
# rupgrade_tool -s --r=sentry.mips.rfs
--proto=ftp:192.168.1.253:/tftpboot/ru-mips:admin --hook=etc_copy
-v
rupgrade_tool: PLB is 5
rupgrade_tool: EEPROM page saved
rupgrade_tool: persistent flash is 1
rupgrade_tool: provisional flash is 0
rupgrade_tool: copying image(s)
rupgrade_tool: copying sentry.rfs from
192.168.1.253:/tftpboot/ru-mips:admin to /tmp using 'ftp' protocol
220 hydra FTP server (Version wu-2.4.2-academ[BETA-17] (1) Tue Jun
9 10:43:14 EDT 1998) ready.
USER admin
```

此处要求用户提供 FTP 站点的密码; 手动输入该密码。

```
331 Password required for admin.
Password:xxxxx
PASS *****
230 User admin logged in.
TYPE I
200 Type set to I.
PASV
227 Entering Passive Mode (192,168,1,253,9,20)
RETR /tftpboot/ru-mips/sentry.mips.rfs
150 Opening BINARY mode data connection for
/tftpboot/ru-mips/sentry.mips.rfs (2465988 bytes).
226 Transfer complete.
QUIT
221 Goodbye.
```

在下一步骤中，将调用特殊脚本 `step4vshm`，该脚本会将 U-Boot 和内核映像从持久性闪存复制到临时闪存中，然后，升级实用程序会继续将 RFS 映像复制到它在临时闪存中的指定位置。

```
rupgrade_tool: invoking scripts (step4v*) [--r=sentry.mips.rfs --
proto=ftp:192.168.1.253:/tftpboot/ru-mips:admin --hook=etc_copy]
/etc/upgrade/step4vshm: Erasing /dev/mtdchar7...Done
etc/upgrade/step4vshm: Copying Kernel from /dev/mtdchar2 to
/dev/mtdchar7...Done
/etc/upgrade/step4vshm: Erasing /dev/mtdchar8...Done
/etc/upgrade/step4vshm: Copying U-Boot from /dev/mtdchar3 to
/dev/mtdchar8...Done
rupgrade_tool: copying sentry.mips.rfs from /tmp to /dev/mtdchar9
using 'cp' protocol
```

调用 `step4hshm hook` 脚本，这将停止机框管理器并保留非易失性数据。然后，该实用程序将启动升级 WDT 并重新引导。

```
rupgrade_tool: invoking scripts (step4h*) [etc_copy]
/etc/upgrade/step4hshm: Stopping Shelf Manager...
/etc/upgrade/step4hshm: Erasing /var and /etc, copying
/var/nvdata..
/etc/upgrade/step4hshm: Operation: copy /etc and /var/nvdata.
/etc/upgrade/step4hshm: Copying completed.
rupgrade_tool: image(s) copy OK
rupgrade_tool: watchdog started
rupgrade_tool: selected provisional flash
rupgrade_tool: reboot
Restarting system.
Connection closed by foreign host.
```

此时，Telnet 会话在经过一段时间的不活动后结束；几秒钟后，可以通过调用 `rupgrade_tool -w` 再次重新连接到目标并检查可靠升级的状态。

```
# telnet 192.168.1.174
Trying 192.168.1.174...
Connected to 192.168.1.174.
Escape character is '^]'.

BusyBox on shmm+174 login: root

BusyBox v0.60.5 (2005.05.07-17:27+0000) Built-in shell (msh)
#
# rupgrade_tool -w
Recent upgrade status:
1:PLB is 5
```

```

1:EEPROM page saved
2:persistent flash is 1
3:provisional flash is 0
4:copying image(s)
4:copying sentry.mips.rfs from 192.168.1.253:/tftpboot/ru-
mips:admin to /tmp using 'ftp' protocol
4:invoking scripts (step4v*) [--r=sentry.rfs --hook=etc_copy]
4:copying sentry.mips.rfs from /tmp to /dev/mtdchar9 using 'cp'
protocol
4:invoking scripts (step4h*) [etc_copy]
4:image(s) copy OK
5:watchdog started
6:selected provisional flash
7:reboot
9:WDT not fired, upgrade in progress.
11:provisional flash 0, updating EEPROM
12:EEPROM updated
13:upgrade WDT disabled
13:invoking scripts (step13h*) []
14:upgrade completed successfully
#

```

示例 3:

本示例显示了一个不成功的可靠升级。在从临时闪存引导后、可靠升级未完成之前电源被关闭。重新打开电源后，出现了回滚到持久性闪存的情况。将会从串行控制台启动此可靠升级。假设所有三个映像都已位于 /tmp 中。

```

# rupgrade_tool -s --k=sentry.kernel --r=sentry.rfs --u=u-boot.bin
--hook=etc_copy -v
rupgrade_tool: PLB is 5
rupgrade_tool: EEPROM page saved
rupgrade_tool: persistent flash is 0
rupgrade_tool: provisional flash is 1
rupgrade_tool: copying image(s)
rupgrade_tool: invoking scripts (step4v*) [--u=u-boot.bin --k=
sentry.kernel --r=sentry.rfs --hook=etc_copy]
rupgrade_tool: copying u-boot.bin from /tmp to /dev/mtdchar8 using
'cp' protocol
rupgrade_tool: copying sentry.kernel from /tmp to /dev/mtdchar7
using 'cp' protocol
rupgrade_tool: copying sentry.rfs from /tmp to /dev/mtdchar9 using
'cp' protocol
rupgrade_tool: invoking scripts (step4h*) [etc_copy]
Stopping Shelf Manager...

```

```
IPM Sentry Shelf Manager Command Line Interpreter
```

```
Terminating the
```

```
Erasing /var and /etc, copying /var/nvdata...
```

```
Operation: copy /etc and /var/nvdata.
```

```
Copying completed.
```

```
rupgrade_tool: image(s) copy OK
```

```
rupgrade_tool: watchdog started
```

```
rupgrade_tool: selected provisional flash
```

```
rupgrade_tool: reboot
```

```
Restarting system.
```

可靠升级过程将在此时复位 ShMM，并从临时闪存启动 U-boot。

```
** Resetting Integrated Peripherals
```

```
U-Boot 1.1.2 (Nov 11 2005 - 15:16:25)
```

```
CPU: Aul550 324 MHz, id: 0x02, rev: 0x00
```

```
Board: ShMM-500
```

```
S/N: 8000044
```

```
DRAM: 128 MB
```

```
Flash: 64 MB
```

```
In: serial
```

```
Out: serial
```

```
Err: serial
```

```
Net: AulX00 ETHERNET
```

```
Hit any key to stop autoboot: 0
```

```
## Booting image at bfb00000 ...
```

```
Image Name: MIPS Linux-2.4.26
```

```
Created: 2005-04-11 10:35:08 UTC
```

```
Image Type: MIPS Linux Kernel Image (gzip compressed)
```

```
Data Size: 843129 Bytes = 823.4 kB
```

```
Load Address: 80100000
```

```
Entry Point: 802bc040
```

```
Verifying Checksum ... OK
```

```
Uncompressing Kernel Image ... OK
```

```
## Loading Ramdisk Image at bfc40000 ...
```

```
Image Name: sentry RFS Ramdisk Image
```

```
Created: 2005-04-22 9:10:41 UTC
```

```
Image Type: MIPS Linux RAMDisk Image (gzip compressed)
```

```
Data Size: 2400736 Bytes = 2.3 MB
```

```
Load Address: 00000000
Entry Point: 00000000
Verifying Checksum ... OK
```

此时电源已关闭。过一段时间后，电源又重新打开。因为断电，临时闪存中的赋值将丢失，因此，系统会恢复到持久性闪存。

```
U-Boot 1.1.2 (Nov 11 2005 - 15:16:25)
```

```
CPU: Au1550 324 MHz, id: 0x02, rev: 0x00
```

```
Board: ShMM-500
```

```
S/N: 8000048
```

```
DRAM: 128 MB
```

```
Flash: 64 MB
```

```
In: serial
```

```
Out: serial
```

```
Err: serial
```

```
Net: Au1X00 ETHERNET
```

```
Hit any key to stop autoboot: 0
```

```
## Booting image at bfb00000 ...
```

```
Image Name: MIPS Linux-2.4.26
```

```
Created: 2005-04-11 10:35:08 UTC
```

```
Image Type: MIPS Linux Kernel Image (gzip compressed)
```

```
Data Size: 843129 Bytes = 823.4 kB
```

```
Load Address: 80100000
```

```
Entry Point: 802bc040
```

```
Verifying Checksum ... OK
```

```
Uncompressing Kernel Image ... OK
```

```
## Loading Ramdisk Image at bfc40000 ...
```

```
Image Name: sentry RFS Ramdisk Image
```

```
Created: 2005-04-11 18:27:17 UTC
```

```
Image Type: MIPS Linux RAMDisk Image (gzip compressed)
```

```
Data Size: 2372311 Bytes = 2.3 MB
```

```
Load Address: 00000000
```

```
Entry Point: 00000000
```

```
Verifying Checksum ... OK
```

```
Starting kernel ...
```

```
init started: BusyBox v0.60.5 (2005.02.07-16:45+0000) multi-call
binary
```

```
hub.c: new USB device AU1550-1, assigned address 2
```

```
usb0: ? speed config #1: Ethernet Gadget
```

```
usb1: register usbnet usb-AU1550-1, Linux Device
```

```
serial#=8000048: not found
```

```
/etc/rc: Mounted /proc
/etc/rc: Mounting filesystems...
/etc/rc: Mounted /dev/pts
/etc/rc: Mounted /dev/mtdblock0 to /var
/etc/rc: Mounted /dev/mtdblock10 to /var/upgrade
```

rc 脚本中的下一个步骤是调用 `rupgrade_tool -c`，以便检查是否正在进行一个可靠升级。该检查确定所尝试的可靠升级已失败。消息 `restoring ADM1060 EEPROM to RAM` 指的是 ShMM 系统监控设备 (ADM1060)，该设备监控 ShMM 引导过程并为可靠升级实现某些硬件方面的支持。该消息指示，影响引导过程的关键变量会被恢复到尝试进行可靠升级之前的状态。

```
/etc/rc: Checking the reliable upgrade watchdog timer
rupgrade_tool: Watchdog not active.
rupgrade_tool: restoring ADM1060 EEPROM to RAM
rupgrade_tool: upgrade failed
/etc/rc: Rupgrade -c Ret: 255
/etc/rc: Mounted ram disk to /var/log
/etc/rc: Started syslogd and klogd
/etc/rc: Mounted ram disk to /var/tmp
/etc/rc: Setting hostname shmm+173
/etc/rc: Mounted /dev/mtdblock1 to /etc
/etc/rc: Calling /etc/rc.carrier3
Board Hardware Address: 0xFE
/etc/netconfig: /etc/hosts has valid 192.168.1.173 entry
/etc/netconfig: Updating /etc/profile.sentry with IP settings
/etc/netconfig: Starting inetd...
/etc/rc.carrier3: Starting up IPMBs ...
/etc/rc.carrier3: Updating /etc/profile.sentry with specific
settings
/etc/rc.carrier3: RC2 daemons not started by request
```

对 CPLD 进行编程

ShMM 包括一个 CPLD 设备。此 CPLD 设备负责控制 ShMM 操作的几个关键方面，如硬件级别的冗余接口。

▼ 对 ShMM 的 CPLD 映像重新编程

1. 将以下文件上传到您的机框管理卡的 `/var/bin` 目录中：

- `cpld_tool`
- `shmm-cpld-erase.xsvf`
- `shmm-cpld-1x.xsvf`
- `shmm-cpld.xsvf`

2. 按如下所示清除 CPLD：

```
# cpld_tool -x shmm-cpld-erase.xsvf
```

3. 对 CPLD 的给定映像进行编程：

```
# cpld_tool -x shmm-cpld.xsvf
```

4. 验证 CPLD 映像的有效性：

用户 ID 应该为 `0x33623030`：

```
# cpld_tool -u  
0x33623030
```

连接到节点板控制台

机框管理器提供了连接到节点板，并从活动机框管理卡 (ShMM) 打开控制台会话的功能。首先应从串行端口或以太网端口登录到活动 ShMM。

注 – 主（即上部）机框管理卡必须是活动机框管理卡才能使用控制台功能。还必须在 Netra CT 900 服务器中间背板的插槽 7 中安装交换机卡。

一旦与节点板建立了控制台会话，就可以运行系统管理命令（如 `passwd`），读取状态消息和错误消息或停止该特定插槽中的板。

注 – 将控制台电缆或串行电缆连接到节点板的串行端口后，控制台输出将转到连接了电缆的控制台，而不是 ShMM 上的控制台会话，即使在接入电缆时 ShMM 的控制台会话处于活动状态也是如此。

在机框管理器和节点板之间建立控制台会话

针对控制台应用而配置了系统后，您可以登录到活动的 ShMM，并打开某个插槽的控制台。Netra CT 系统允许每个节点板插槽具有一个控制台会话。

表 3-4 显示了与机框管理器 CLI 控制台相关的命令，您可以从机框管理卡上的当前登录会话执行这些命令。

表 3-4 与机框管理器 CLI 控制台相关的命令

命令	说明
<code>clia console slot_no</code>	进入控制台模式并连接到 <code>slot_no</code> 中的节点板，此处的 <code>slot_no</code> 是节点板所在的插槽编号。
<code>~q</code> 或 <code>~.</code>	断开与当前控制台会话的连接。

▼ 从机框管理器启动控制台会话

1. 登录到主（即上部）ShMM。

可以通过连接到串行端口或连接到以太网端口的终端登录到主（即上部）ShMM。

2. 检查主 ShMM 是否为活动 ShMM。

登录后，请使用 `clia shmstatus` 命令验证是否登录到活动 ShMM，然后再继续进行其他操作。如果登录到备用 ShMM 上，请使用 `clia switchover` 命令将该 ShMM 更改为 Active。（有关更多信息，请参见第 105 页的“机框管理器 CLI 命令”中的 `shmstatus` 和 `switchover`。）

3. 打开与某节点板的控制台会话。

```
# clia console slot_no
```

其中，`slot_no` 为 1 至 6 和 9 至 14 之间的数字。例如，要打开插槽 4 中板的控制台，请输入以下命令：

```
# clia console 4
```

现在，您可以访问插槽 4 中的节点板。根据该特定插槽中板的状态以及上一个用户是否已注销该 shell，您会看到以下几种提示符之一：

- `console login%`（Solaris 级别）
- `#`（Solaris 级别，上一个用户以超级用户身份登录，且在断开与控制台之间的连接之前未注销）
- `ok`（OpenBoot PROM 级别）
- `#`（Monta Vista Linux）

▼ 结束控制台会话

1. (可选) 注销 **OS shell**。
2. 通过在提示符下输入转义序列 `~q` 或 `~.` (波浪号加句点), 断开与控制台的连接:

```
prompt ~q
```

与控制台断开连接并不会使您从远程主机自动注销。除非您从远程主机上注销, 否则, 连接到该板的下一个控制台用户会看到您上次会话的 **shell** 提示符。

注 - 完成控制台会话后总是要将其注销。

手动正常关闭节点板

在此发行版中, 提供了手动正常关闭的功能, 以防止意外移除节点板或 **FRU**。正常关闭是指以连贯的方式停止 (关闭) 在节点板的有效负荷中运行的所有应用程序和 **OS**, 以及有效负荷自身。

在移除某个 **Netra** 节点板或对其进行热交换之前, 应该正常关闭该节点板上运行的应用程序和操作系统。利用机框管理器的网络控制台功能, 系统管理员可以启动节点板的控制台会话, 并可按照该站点的关闭过程执行正常关闭操作。

完成关闭操作后, 系统管理员将启用 **FRU** (或节点板) 的取消激活功能, 并打开板上的热交换锁扣。替换或重新安装节点板后, 管理员将禁用 **FRU** (或节点板) 的取消激活功能, 以防止打开热交换锁扣后执行非正常关闭操作。

下面的过程详细介绍了此处理所包含的步骤。

▼ 关闭节点板

此过程要求节点板上的热交换锁扣处于关闭状态。

注 – 如果热交换锁扣处于打开状态，则蓝色热交换 LED 指示灯将不断闪烁，该板将不会变为热交换就绪状态（稳定亮起的蓝色 LED 指示灯指示处于就绪状态）。要更正这一状况，请在开始本过程之前关闭热交换锁扣。

1. 登录到活动 **ShMM**。

2. 启动与该节点板的控制台会话。

请参见第 100 页的“从机框管理器启动控制台会话”。

3. 关闭节点板中的应用程序和操作系统。

请按照站点的正常关闭过程操作。完成关闭操作后，结束控制台会话（请参见[结束控制台会话](#)）。

4. 使用以下机框管理器 CLI 命令针对节点板启用机框管理器控制的取消激活。

```
# clia shelf deactivate hardware-addr fru-id 0
```

例如：

```
# clia shelf deactivate 0x41 0xfe 0
IPM Sentry Shelf Manager Command Line Interpreter
IPM Sentry Shelf Manager Command Line Interpreter
Updating Shelf FRU Info, address: 0x41, FRU ID # 254
    Cached information updated
    Wrote Information to the Shelf FRU
#
```

5. 使用以下机框管理器 CLI 命令验证是否已启用机框管理器控制的取消激活。

```
# clia shelf pm
```

查找 Shelf Manager Controlled Deactivation: Enabled 消息。

例如:

```
# clia shelf pm
IPM Sentry Shelf Manager Command Line Interpreter
PICMG Shelf Activation And Power Management Record (ID=0x12)
    Version = 1
    Allowance for FRU Activation Readiness: 10 seconds
    FRU Activation and Power Description Count: 16
    Hw Address: 41, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
    Shelf Manager Controlled Activation: Enabled
    Shelf Manager Controlled Deactivation: Enabled
    Delay Before Next Power On: 0.0 seconds
```

6. 打开节点板上的热交换锁扣。
7. 当节点板蓝色热交换 LED 指示灯呈稳定亮起状态时, 移除或替换该节点板。



注意 - 对模块进行操作时, 请始终佩戴接地的防静电腕带。

8. 替换或重新安装节点板后, 使用以下机框管理器 CLI 命令针对节点板禁用机框管理器控制的取消激活。

```
# clia shelf deactivate hardware-addr fru-id 1
```

例如:

```
# clia shelf deactivate 0x41 0xfe 1
IPM Sentry Shelf Manager Command Line Interpreter
IPM Sentry Shelf Manager Command Line Interpreter
Updating Shelf FRU Info, address: 0x41, FRU ID # 254
    Cached information updated
    Wrote Information to the Shelf FRU
#
```


机框管理器 CLI 命令

本章介绍各个 CLI 命令并提供每个可用命令的语法和用途。CLI 支持 AdvancedTCA 机框环境。

为方便起见，可以按照以下方式引用关键类型的机框部件，来代替基于 IPMB 地址和数字 FRU 标识符的引用表示法：

- `board n | b n`
- `fan_tray n | ft n`
- `shm 1 | 2`

注 – 可以使用符号 `shm 1` 和 `shm 2` 访问机框 FRU 的地址表中描述的冗余机框管理器。在本手册中，`shm 1` 对应于硬件地址数值较小的机框管理器，`shm 2` 对应于硬件地址数值较大的机框管理器。

在冗余配置中，备用机框管理器并不支持下面列出的所有命令。备用机框管理器仅识别以下命令：

- `debuglevel`
- `localaddress`
- `shmstatus`
- `switchover`

大多数信息命令都支持两种执行模式：简短模式和详细模式，其差别在于所提供的信息量不同。简短模式为默认模式（标准模式）；详细模式是在命令行中使用选项 `-v` 选择的，该选项紧跟命令之后，位于位置参数之前。

在下面的命令语法中，可选元素括在方括号 [,] 中，命令行中的变量元素（例如，IPMB 地址和 FRU 设备 ID）以斜体列显。竖线 (|) 用来分隔参数选项。

activate

语法:

```
activate IPMB-address fru-id
activate board n
activate shm n
```

用途:

该命令用于将 IPMI 命令 Set FRU Activation (Activate FRU) 发送到指定的 FRU。FRU 通过使用所属 IPM 控制器的 IPMB 地址以及 FRU 设备 ID 进行指定。FRU 设备 ID 0 指定适合 PICMG 3.0 环境的 IPM 控制器。

在 PICMG 3.0 环境中，该命令主要适用于那些未在机框 FRU 信息中的电源管理表中列出的 FRU，或 Shelf Manager Controlled Activation 属性设置为 FALSE 的 FRU。机框管理器不会自动激活这些 FRU，这些 FRU 将保持在 M2（激活请求）状态。一旦其他 FRU 达到 M2 状态，机框管理器就会自动激活它们。尝试激活未处于 M2 状态的 FRU 将是徒劳。

示例:

激活地址为 9C 的 IPM 控制器。

```
# clia activate 9c 0
IPM Sentry Shelf Manager Command Line Interpreter
Command issued via IPMB, status = 0 (0x0)
Command executed successfully
#
```

alarm

语法:

```
alarm [clear|minor|major|critical]
```

用途:

该命令可用于访问 Telco 报警输出。使用参数 `minor`、`major` 和 `critical`，用户可以设置相应的报警输出。这些操作是累计的；也就是说，执行命令 `clia alarm minor` 和 `clia alarm major` 之后，将同时设置次要报警和重要报警。`clear` 操作可以清除次要报警输出和重要报警输出，不能清除紧急报警输出。

不带参数的命令调用将返回 Telco 报警输出的状态。

示例:

```
# clia alarm
IPM Sentry Shelf Manager Command Line Interpreter
  alarm mask: 0x00
# clia alarm major
IPM Sentry Shelf Manager Command Line Interpreter
Returned completion code: 0
# clia alarm
IPM Sentry Shelf Manager Command Line Interpreter
  alarm mask: 0x02
# clia alarm clear
IPM Sentry Shelf Manager Command Line Interpreter
Returned completion code: 0
# clia alarm
IPM Sentry Shelf Manager Command Line Interpreter
  alarm mask: 0x00
```

board

语法:

```
board [-v] [physical-slot-address]
```

用途:

该命令和 `boardreset` 命令与命令集的其余命令不同，区别在于这两个命令可用于 ATCA 板，而且它们使用物理插槽编号而不是 IPM 控制器地址和 FRU 设备 ID 作为参数。命令 `board` 用于显示有关已分配给 ATCA 插槽的 IPMB 地址范围内的每个 IPM 控制器的信息，以及有关由这些控制器控制的每个附加 FRU 的信息。第 121 页的 "`fru`" 一节和第 164 页的 "`ipmc`" 一节中提供了要显示的项的列表。对于板上带有 IPM 控制器的 PICMG 3.0 系统，IPMB 地址的范围是 82h-A0h

应以十进制数的形式指定物理地址。对于 PICMG 3.0 系统，物理地址和 IPMB 地址之间的对应关系在机框 FRU 信息中指定。如果机框 FRU 信息中不包含地址表，则使用以下映射表（逻辑插槽编号的映射）。

插槽编号	IPMB 地址
1	9A
2	96
3	92
4	8E
5	8A
6	86
7	82
8	84
9	88
10	8C
11	90
12	94
13	98
14	9C

示例:

获取有关系统中所有板（其中只有物理插槽 1 和 14 中插有板）的标准信息。

```
# clia board
IPM Sentry Shelf Manager Command Line Interpreter
Physical Slot # 1
82: Entity: (0xd0, 0x0) Maximum FRU device ID: 0x08
    PICMG Version 2.0
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
82: FRU # 0
    Entity: (0xd0, 0x0)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: "IPM Sentry 6"
Physical Slot # 14
9c: Entity: (0xd0, 0x0) Maximum FRU device ID: 0x08
    PICMG Version 2.0
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
9c: FRU # 0
    Entity: (0xd0, 0x0)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: "IPM Sentry 6"
#
```

获取有关物理插槽 14 中板的详细信息。

```
# clia board -v 14
Physical Slot # 14
9c: Entity: (0xd0, 0x0) Maximum FRU device ID: 0x08
    PICMG Version 2.0
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID: 0x00, Revision: 0, Firmware: 1.01, IPMI ver 1.5
    Manufacturer ID: 00315a (PICMG), Product ID: 0000, Auxiliary
Rev: 01ac1014
    Device ID String: "IPM Sentry 6"
    Global Initialization: 0x0, Power State Notification: 0x0,
Device Capabilities: 0x29
    Controller provides Device SDRs
    Supported features: 0x29
    "Sensor Device" "FRU Inventory Device" "IPMB Event
Generator"
```

```
9c: FRU # 0
    Entity: (0xd0, 0x0)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: "IPM Sentry 6"
    Site Type: 0x00, Site Number: 14
    Current Power Level: 0x01, Maximum Power Level: 0x01, Current
Power Consumption: 20.0 Watts
#
```

boardreset

语法:

```
boardreset physical-slot-address
```

用途:

该命令用于复位指定物理插槽中的板，方法是向该板发送 IPMI 命令 FRU Control (Cold Reset)。

应以十进制数的形式指定物理地址。对于 PICMG 3.0 系统，物理地址和 IPMB 地址之间的对应关系在机框 FRU 信息中指定。如果机框 FRU 信息未包含地址表，则使用以下映射表（逻辑插槽编号的映射）。FRU 设备 ID 为 0。

插槽编号	IPMB 地址
1	9A
2	96
3	92
4	8E
5	8A
6	86
7	82
8	84
9	88
10	8C
11	90
12	94
13	98
14	9C

示例:

复位物理插槽 14 (IPMB 地址为 9C, FRU ID 为 0) 中的板。

```
# clia boardreset 14
IPM Sentry Shelf Manager Command Line Interpreter
Board 14 reset, status returned 0
#
```

busres

语法:

`busres subcommand`

支持以下子命令:

- `info [resource]`
- `release resource`
- `force resource`
- `lock resource`
- `unlock resource`
- `query [-v] resource [target [noupdate]]`
- `setowner resource target`
- `sendbusfree resource target`

用途:

该命令用于显示有关总线型电子钥控 (E-Keying) 管理资源的当前状态的信息, 并允许更改此状态。

所有子命令中都包含一个相同的参数, 即资源 ID。资源 ID 是一个从 0 开始的资源编号或一个简短的资源名称。定义了以下资源名称和编号:

编号	简短名	说明
0	mtb1	金属测试总线对 1
1	mtb2	金属测试总线对 2
2	clk1	同步时钟组 1
3	clk2	同步时钟组 2
4	clk3	同步时钟组 3

以下各小节针对几个主要用途来介绍 busres 命令的语法。

显示总线型电子钥控 (E-Keying) 管理资源的状态

语法:

```
busres info [resource]
```

用途:

该命令用于显示有关指定资源或所有资源（如果未指定资源 ID）的当前状态的信息。

参数 *resource* 是资源 ID。第 111 页的 "busres" 中提供了受支持资源 ID 的列表。

示例:

获取有关金属测试总线对 2 的状态的信息

```
# clia busres info mtb2
IPM Sentry Shelf Manager Command Line Interpreter
Metalic Test Bus pair 2 (ID 1): Owned by IPMC 0x82, Locked
#
```

释放指定的资源

语法:

```
busres release | force resource
```

用途:

该命令用于将 Bused Resource Control 请求发送到资源的当前所有者，指示所有者释放该资源。如果该命令的语法为 `busres release resource`，则会发送 Bused Resource Control (Release) 命令。如果该命令的语法为 `busres force resource`，则会发送 Bused Resource Control (Force) 命令。有关这些 ATCA 命令的详细说明，请参阅 PICMG 3.0 R1.0 规范的第 3.7.3.4 节。

参数 *resource* 是资源 ID。第 111 页的 "busres" 中提供了受支持资源 ID 的列表。

示例:

强制当前所有者释放金属测试总线对 2。

```
# clia busres force mtb2
IPM Sentry Shelf Manager Command Line Interpreter
Force operation succeeded
#
```

锁定/解除锁定指定的资源

语法:

```
busres lock | unlock resource
```

用途:

该命令用于锁定 (`busres lock resource`) 或解除锁定 (`busres unlock resource`) 指定的资源。如果资源已锁定, 则当另一个 IPM 控制器将 Bused Resource Control (Request) 命令发送到机框管理器时, 机框管理器将以“拒绝”状态作为响应。如果资源已解除锁定, 则当另一个 IPM 控制器将 Bused Resource Control (Request) 命令发送到机框管理器时, 机框管理器将以“忙”状态作为响应, 并将 Bused Resource Control (Release) 发送给当前所有者。如果当前所有者释放该资源, 则机框管理器在接收到下一个请求时会将该资源授予请求者。

只能锁定由 IPM 控制器拥有的资源。一旦当前所有者释放了资源, 该资源的锁定也就被解除。

参数 `resource` 是资源 ID。第 111 页的“busres”中提供了受支持资源 ID 的列表。

示例:

锁定同步时钟组 3。

```
# clia busres lock clk3
IPM Sentry Shelf Manager Command Line Interpreter
Lock operation succeeded
#
```

发送 Bused Resource Control (Query) 命令

语法:

```
busres [-v] query resource [target [noupdate]]
```

用途:

该命令用于将 Bused Resource Control (Query) 请求发送到指定的 IPM 控制器。如果未在命令行中指定 IPM 控制器, 则会将请求发送到资源的当前所有者。一旦接收到响应, 就会在资源表中进行相应的更改 (例如, 如果被认为是当前所有者的 IPM 控制器以“无控制”状态作为响应, 则会对表进行修改以反映这一事实), 除非提供了 `noupdate` 标志。如果在命令行中传递了该标志, 则不会基于接收到的信息对资源表进行任何更改。

参数 *resource* 是资源 ID。第 111 页的 "busres" 中提供了受支持资源 ID 的列表。

参数 *target* 用于指定将要接收请求的 IPM 控制器的 IPMB 地址。

标志 *noupdate*（如果存在）指示不会使用接收到的查询请求响应信息来更新资源表。

在机框管理器的当前修订版中，如果未指定 *-v* 标志，则不会提供任何其他信息。

示例:

将对金属测试总线对 1 的查询发送到地址为 0x82 的 IPM 控制器。不根据响应更新资源表。

```
# clia busres query mtb1 0x82 noupdate
IPM Sentry Shelf Manager Command Line Interpreter
No Control: IPMC 0x82 is not the owner of resource 0
#
```

设置资源的所有者

语法:

```
busres setowner resource target
```

用途:



注意 – 该命令适用于有经验的用户，用户应自行判断是否使用。

该命令用于直接设置资源表中指定资源的所有者。该命令不发送任何总线型资源控制命令，即使在执行该命令之前资源具有不同所有者也是如此。这是一个底层命令，只应将其用于测试和恢复目的。

参数 *resource* 是资源 ID。第 111 页的 "busres" 中提供了受支持资源 ID 的列表。

参数 *target* 用于指定设置为资源所有者的 IPM 控制器的 IPMB 地址。将 0 用作 IPMB 地址可指定任何 IPM 控制器都不拥有该资源。

示例:

将板 1 设置为金属测试总线对 1 的新所有者。

```
# clia busres setowner mtb1 board 1
IPM Sentry Shelf Manager Command Line Interpreter
New owner is set successfully
#
```

发送 Bused Resource Control (Bus Free) 命令

语法:

```
busres sendbusfree resource target
```

用途:



注意 – 该命令适用于有经验的用户，用户应自行判断是否使用。

该命令用于将 Bused Resource Control (Bus Free) 请求发送到指定的 IPM 控制器。在发送该请求之前不会对该资源进行任何操作，即使其他 IPM 控制器拥有该资源也是如此。不过，将会基于对该请求的响应更新资源表。也就是说，如果 IPM 控制器接受对该资源的拥有权，则会在该表中将其设置为新的所有者。这是一个底层命令，只应将其用于测试和恢复目的。

参数 *resource* 是资源 ID。第 111 页的 "busres" 中提供了受支持资源 ID 的列表。

参数 *target* 用于指定将要接收请求的 IPM 控制器的 IPMB 地址。将 0 用作 IPMB 地址可指定任何 IPM 控制器都不拥有该资源。

示例:

将对金属测试总线对 1 的总线释放请求发送到地址为 0x82 的 IPM 控制器。

```
# clia busres sendbusfree mtb1 0x82
IPM Sentry Shelf Manager Command Line Interpreter
IPMC rejected ownership of the resource
#
```

console

语法:

```
console slot-number
```

用途:

该命令用于在指定物理插槽中的节点板上建立控制台终端会话。机框管理器允许在每个节点板上创建一个控制台会话。有效的插槽编号为 1-6 和 9-14。

注 - 主（即上部）机框管理卡必须是活动机框管理卡才能使用控制台功能。还必须在 Netra CT 900 服务器中间背板的插槽 7 中安装交换机卡。

一旦与节点板上建立了控制台会话，就可以运行系统管理命令（如 `passwd`），读取状态消息和错误消息或停止该特定插槽中的板。

注 - 将控制台电缆或串行电缆连接到节点板的串行端口后，控制台输出将转到连接了电缆的控制台，而不是 ShMM 上的控制台会话，即使在接入电缆时 ShMM 的控制台会话处于活动状态也是如此。

要断开与当前控制台会话的连接，请输入 `~q` 或 `~.`（波浪号加句点）。

示例:

启动物理插槽 4 中节点板上的控制台会话。

```
# clia console 4
prompt
```

deactivate

语法:

```
deactivate IPMB-address fru-id
deactivate board n
deactivate shm n
```

用途:

该命令用于将 IPMI 命令 Set FRU Activation (Deactivate FRU) 发送到指定的 FRU。FRU 通过使用所属 IPM 控制器的 IPMB 地址以及 FRU 设备 ID 进行指定。FRU 设备 ID 0 指定适合 PICMG 3.0 环境的 IPM 控制器。

示例:

取消激活地址为 9C 的 IPM 控制器。

```
# clia deactivate 9c 0
IPM Sentry Shelf Manager Command Line Interpreter
Command issued via IPMB, status = 0 (0x0)
    Command executed successfully
#
```

debuglevel

语法:

```
debuglevel [new-value]
```

用途:

该命令用于显示 IPM Sentry 机框管理器的当前调试级别，或将其设置为新值（如果指定了新值）。

调试级别是一个被视为位掩码的范围为 0x0000 至 0x00FF 的十六进制数。掩码中的每一位（如果设置）启用一种特定类型的调试输出：

- 0x0001 - 错误消息
- 0x0002 - 警告消息
- 0x0004 - 提示性消息

- 0x0008 - 详细提示性消息
- 0x0010 - 跟踪消息
- 0x0020 - 详细跟踪消息
- 0x0040 - 在 IPM 控制器初始化期间针对发送到 IPM 控制器的重要命令所显示的消息
- 0x0080 - 有关获取和释放内部锁的详细消息

机框管理器的默认调试级别是 0x0007，但是可以在机框管理器启动过程中在命令中使用 `-v` 选项覆盖该值。CLI 提供了在运行时更改调试级别的附加功能。

示例:

获取当前调试级别，然后将其设置为 0x001F。

```
# clia debuglevel
IPM Sentry Shelf Manager Command Line Interpreter
Debug Mask is 0x0007
# clia debuglevel 1f
IPM Sentry Shelf Manager Command Line Interpreter
# clia debuglevel
IPM Sentry Shelf Manager Command Line Interpreter
Debug Mask is 0x001f
```

exit | quit

语法:

```
exit
quit
```

用途:

命令 `exit` 或 `quit` 用于退出 CLI 交互模式（可以通过发出不带参数的 `clia` 进入该模式）。

示例:

```
# exit
#
```

fans

语法:

```
fans [-v] [IPMB-address [FRU-device-ID]]  
fans fan_tray n
```

用途:

该命令用于显示有关指定的风扇 FRU 的信息。如果省略 FRU 设备 ID，则该命令将显示有关由位于指定地址的 IPM 控制器控制的所有风扇 FRU 的信息。如果还省略 IPMB 地址，则该命令将显示有关机框管理器能识别的所有风扇 FRU 的信息。将显示以下信息:

- IPMB 地址和 FRU 设备 ID
- 最低速度级别
- 最高速度级别
- 最高持续速度级别
- 当前级别（覆盖级别和本地控制级别，若两者都存在）

示例:

获取有关 IPMB 地址为 20 的所有风扇 FRU 的风扇信息。

```
# clia fans 20  
IPM Sentry Shelf Manager Command Line Interpreter  
20: FRU # 3  
  Current Level: 6  
  Minimum Speed Level: 0, Maximum Speed Level: 15  
20: FRU # 4  
  Current Level: 255 "Automatic"  
  Minimum Speed Level: 0, Maximum Speed Level: 15  
20: FRU # 5  
  Current Level: 255 "Automatic"  
  Minimum Speed Level: 0, Maximum Speed Level: 15  
#
```

flashupdate

语法:

```
flashupdate slot-number -s server-ip-address -f fw-image-pathname
```

用途:

该命令用于使用指定位置的固件映像更新 Netra CP3060 板系统固件。该命令仅对安装在 Netra CT 900 服务器中的 Netra CP3060 板有效。更新的 Netra CP3060 板系统固件包括 ALOM-CMT、Hypervisor（系统管理程序）、OBP、Post（开机自检程序）和 VBSC 固件。

注 – 访问以下 URL，可以找到指向 Netra CP3060 固件下载站点的链接：
<http://www.sun.com/downloads/>

要使用该命令，需要知道以下信息：

- 要从中下载固件映像的 FTP 服务器的 IP 地址
- 该 FTP 服务器的用户名和密码（需要在提示符下输入）
- 存储映像的路径

slot-number 包含 Netra CP3060 板的插槽编号，*-s server-ip-addresss* 参数指定从中下载固件映像的服务器的 IP 地址，*-f fw-image-pathname* 指定固件映像所在路径的全路径名。

示例:

下载并更新 Netra CP3060 板上的系统固件。请注意，该过程可能要花几分钟才能完成。成功完成之后，使用 `boardreset` 命令将板复位。

```
# clia flashupdate 2 -s 123.45.67.89
  -f /sysfw/System_Firmware-6_2_5-Netra_CP3060.bin
Username: username
Password: *****
.....
.....
.....
Update complete. Reset device to use new software.

# clia boardreset slot-number
```

fru

语法:

```
fru [-v] [addr [id=fru_id | type=site_type]] | [type=site_type  
[/site_number]]  
fru board n  
fru shm n  
fru fan_tray n
```

用途:

该命令用于显示有关特定 FRU 的信息。如果省略 FRU 设备 ID，则该命令将显示有关由位于指定地址的 IPM 控制器控制的所有 FRU 的信息。如果还省略 IPMB 地址，则该命令将显示有关机框管理器能识别的所有 FRU 的信息。

此外，位置类型可以选择 FRU。应当在命令参数中以十六进制形式指定位置类型。FRU 及其位置类型之间的关联存储在机框 FRU 信息中。PICMG 3.0 规范中定义的位置类型如下:

- 00h - AdvancedTCA 板
- 01h - 电源输入模块
- 02h - 机框 FRU 信息
- 03h - 专用 ShMC
- 04h - 风扇托盘
- 05h - 风扇过滤器托盘
- 06h - 报警
- 07h - AdvancedTCA 模块 (夹层)
- 08h - PMC
- 09h - 后部转换模块
- C0h-CFh - 由 OEM 定义
- 所有其他保留值

在标准模式下将显示有关 FRU 的以下信息:

- IPMB 地址和 FRU 设备 ID
- 实体 ID、实体实例
- 位置类型和编号 (如果已知)

- 当前热交换状态、先前热交换状态和 FRU 上次状态更改的原因。PICMG 3.0 规范中对热交换状态 M0 至 M7 定义如下：
 - M0 - 未安装
 - M1 - 非活动
 - M2 - 激活请求
 - M3 - 正在激活
 - M4 - FRU 活动
 - M5 - 取消激活请求
 - M6 - 正在取消激活
 - M7 - 通信丢失

仅在详细模式下显示有关 FRU 的以下信息：

- FRU 设备类型、设备类型修饰符（仅限于 FRU-device-ID != 0）。该信息取自 FRU 传感器数据记录 (Sensor Data Record, SDR) 并符合 IPMI 规范的第 37.12 节。
- FRU SDR 中的设备 ID 字符串
- 当前 FRU 电源级别和最高 FRU 电源级别；当前指定的功耗（以瓦特为单位）

示例：

获取有关地址为 9C 的所有 FRU 的标准信息。

```
# clia fru 9c 0
IPM Sentry Shelf Manager Command Line Interpreter
9c: FRU # 0
    Entity: (0xd0, 0x0)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: "IPM Sentry 6"
#
```

获取有关地址为 9C 的所有 FRU 的详细信息。

```
# clia fru -v 9c 0
IPM Sentry Shelf Manager Command Line Interpreter
9c: FRU # 0
    Entity: (0xd0, 0x0)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: "IPM Sentry 6"
    Site Type: 0x00, Site Number: 14
    Current Power Level: 0x01, Maximum Power Level: 0x01, Current
Power Consumption: 20.0 Watts
#
```

获取有关地址为 20 的 FRU 1 的详细信息。

```
# clia fru -v 20 id=1
IPM Sentry Shelf Manager Command Line Interpreter
20: FRU # 1
    Entity: (0x1, 0x1)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device Type: "FRU Inventory Device behind management
controller" (0x10), Modifier 0x0
    Device ID String: "IPM Sentry 1.1"
    Current Power Level: UNKNOWN, Maximum Power Level: UNKNOWN,
Current Power Consumption: UNKNOWN
#
```

frucontrol

语法:

```
frucontrol IPMB-address fru-id option
```

```
frucontrol board n option
```

```
frucontrol shm n option
```

```
frucontrol fan_tray n option
```

用途:

该命令用于将 FRU 控制命令发送到指定的 FRU，针对 FRU 有效负荷执行指定的操作。FRU 通过使用所属 IPM 控制器的 IPMB 地址以及 FRU 设备 ID 进行指定。FRU 设备 ID 0 指定适合 PICMG 3.0 环境的 IPM 控制器。

参数 *option* 用于指定要使用的 FRU 控制命令的选项。可以将其指定为以下符号值之一:

- cold_reset (缩写为 cr) - 用于执行 FRU 有效负荷的冷复位
- warm_reset (缩写为 wr) - 用于执行 FRU 有效负荷的热复位
- graceful_reboot (缩写为 gr) - 用于执行 FRU 有效负荷的正常重新引导
- diagnostic_interrupt (缩写为 di) - 用于发出诊断中断

示例:

向 IPMB 地址为 9C 的 FRU 0 发出冷复位命令。

```
# clia frucontrol 9c 0 cr
IPM Sentry Shelf Manager Command Line Interpreter
  FRU Control: Controller 0x9c, FRU ID # 0, command 0x00, status
0 (0x0)
  Command executed successfully
#
```

frudata

语法:

```
frudata [addr [fru_id [block_offset]]]
frudata addr fru_id byte_offset byte 1 [byte2 ... [byte 16] ...]
```

可以使用以下项替换 *addr fru_id*:

```
board n
shm n
fan_tray n
```

用途:

该命令用于访问原始格式的 FRU 信息。可以使用该命令读取或写入 FRU 信息，具体取决于命令的格式。

在读取格式中，该命令采取可选的 32 字节块编号。

在写入格式中，该命令需要一个字节偏移参数。用户最多可以修改 65535 字节的 FRU 信息。

`frudataw` 和 `frudatar` 是 `frudata` 命令的变体。使用 `frudataw`，用户可以将 ShMM 闪存文件系统上的文件写入机框中特定 FRU 的 FRU 信息存储器（请参见第 126 页的 "`frudatar`"）。使用 `frudatar`，用户可以将特定 FRU 的 FRU 信息存储器的内容传输到 ShMM 闪存文件系统上的文件中（请参见第 126 页的 "`frudatar`"）。

示例:

显示所有 FRU 的标准 FRU 数据。

```
# clia frudata
IPM Sentry Shelf Manager Command Line Interpreter
20: FRU # 0   Failure status: 203 (0xcb)
    Requested data not present
20: FRU # 1 Raw FRU Info Data
    FRU Info size: 529
20: FRU # 2   Failure status: 203 (0xcb)
    Requested data not present
82: FRU # 0 Raw FRU Info Data
    FRU Info size: 160
9c: FRU # 0 Raw FRU Info Data
    FRU Info size: 160
fc: FRU # 0 Raw FRU Info Data
    Requested data not present
. . .
#
```

该示例说明了如何显示 FRU 数据以及将数据写入 FRU 的方式。

```
# clia frudata 20 1 0
IPM Sentry Shelf Manager Command Line Interpreter
20: FRU # 1 Block # 0 Raw FRU Info Data
    FRU Info size: 529
    01 00 01 05  0E 18 00 D3  01 04 01 02  55 AA 83 55
    AA 55 C1 00  00 00 00 00  00 00 00 00  00 00 00 00
#
# clia frudata 20 1 1 0xfc 0xfe
IPM Sentry Shelf Manager Command Line Interpreter
Writing 2 bytes to IPM 0x20, FRU # 1, offset: 1, status = 0(0x0)
#
# clia frudata 20 1 0
IPM Sentry Shelf Manager Command Line Interpreter
20: FRU # 1 Block # 0 Raw FRU Info Data
    FRU Info size: 529
    01 FC FE 05  0E 18 00 D3  01 04 01 02  55 AA 83 55
    AA 55 C1 00  00 00 00 00  00 00 00 00  00 00 00 00
#
# clia frudata 20 1 1 0 1
IPM Sentry Shelf Manager Command Line Interpreter
Writing 2 bytes to IPM 0x20, FRU # 1, offset: 1, status = 0(0x0)
#
```

```
# clia frudata 20 1 0
IPM Sentry Shelf Manager Command Line Interpreter
20: FRU # 1 Block # 0 Raw FRU Info Data
    FRU Info size: 529
    01 00 01 05  0E 18 00 D3  01 04 01 02  55 AA 83 55
    AA 55 C1 00  00 00 00 00  00 00 00 00  00 00 00 00
#
```

frudatar

语法:

```
frudatar addr fru_id file_name
frudatar addr fru_id byte_offset byte 1 [byte2 ... [byte 16] ...]
```

可以使用以下项替换 *addr fru_id*:

```
board n
shm n
fan_tray n
```

用途:

该命令用于从指定的 FRU 读取 FRU 信息并以原始格式将其存储在 ShMM 闪存文件系统上的文件中（换言之，将 FRU 信息从指定的 FRU 上载到闪存文件中）。参数 *file name* 用于指定目标文件的路径。从 FRU 读取并写入目标文件的字节数等于响应指定 FRU 的 IPMI 命令 Get FRU Inventory Area Info 所返回的字节数。

示例:

读取特定 FRU 的 FRU 数据并将其存储在指定的文件中。

```
# clia frudatar 20 2 /var/tmp/20.2.bin
IPM Sentry Shelf Manager Command Line Interpreter

20: FRU # 2 Raw FRU Info Data
    FRU Info size: 176
    01 00 00 01  09 00 00 F5  01 08 19 84  C0 42 C7 53
    63 68 72 6F  66 66 D9 53  68 4D 4D 2D  41 43 42 2D
    46 43 20 53  68 65 6C 66  20 4D 61 6E  61 67 65 72
    86 10 04 41  10 14 01 89  D2 04 65 58  13 51 17 00
    00 C0 C1 00  00 00 00 EA  01 0D 19 C7  53 63 68 72
    6F 66 66 DD  46 61 6E 20  43 6F 6E 74  72 6F 6C 6C
```

```
65 72 20 6F 6E 20 53 68 4D 4D 2D 41 43 42 2D 46
43 89 D2 04 65 58 13 51 17 00 00 C9 52 65 76 2E
20 31 2E 30 30 86 10 04 41 10 14 01 C0 DF 2F 76
61 72 2F 6E 76 64 61 74 61 2F 66 61 6E 2D 66 72
75 2D 69 6E 66 6F 72 6D 61 74 69 6F 6E C1 00 26
#
```

frudataw

语法:

```
frudataw addr fru_id file_name
frudataw addr fru_id byte_offset byte 1 [byte2 ... [byte 16] ...]
```

可以使用以下项替换 *addr fru_id*:

```
board n
shm n
fan_tray n
```

用途:

该命令用于将 FRU 信息从 ShMM 闪存文件系统上的文件下载到指定的 FRU。该文件包含 FRU 信息的原始二进制映像。参数 *file name* 用于指定源文件的路径。

示例:

将文件中的 FRU 数据写入指定的 FRU。

```
# cli frudataw 20 2 /var/tmp/20.2.bin
IPM Sentry Shelf Manager Command Line Interpreter

Writing 16 bytes to IPM 0x20, FRU # 2, offset: 0, status = 0 (0x0)
Writing 16 bytes to IPM 0x20, FRU # 2, offset: 16, status = 0 (0x0)
Writing 16 bytes to IPM 0x20, FRU # 2, offset: 32, status = 0 (0x0)
Writing 16 bytes to IPM 0x20, FRU # 2, offset: 48, status = 0 (0x0)
Writing 16 bytes to IPM 0x20, FRU # 2, offset: 64, status = 0 (0x0)
Writing 16 bytes to IPM 0x20, FRU # 2, offset: 80, status = 0 (0x0)
Writing 16 bytes to IPM 0x20, FRU # 2, offset: 96, status = 0 (0x0)
Writing 16 bytes to IPM 0x20, FRU # 2, offset: 112, status = 0 (0x0)
Writing 16 bytes to IPM 0x20, FRU # 2, offset: 128, status = 0 (0x0)
Writing 16 bytes to IPM 0x20, FRU # 2, offset: 144, status = 0 (0x0)
Writing 16 bytes to IPM 0x20, FRU # 2, offset: 160, status = 0 (0x0)
File "/var/tmp/20.2.orig.bin" has been written to the FRU 20#2
#
```

fruinfo

语法:

```
fruinfo [-v] [-x] addr fru_id
```

可以使用以下项替换 *addr fru_id*:

```
board n  
shm n  
fan_tray n
```

用途:

该命令用于按用户友好的格式显示 FRU 信息。

示例:

显示特定 FRU 的 FRU 信息。

```
# clia fruinfo 20 1  
IPM Sentry Shelf Manager Command Line Interpreter  
20: FRU # 1, FRU Info  
Common Header: Format Version = 1  
Chassis Info Area:  
  Version      = 1  
  Chassis Type          = (1)  
  Chassis Part Number   = 0x55 0xAA  
  Chassis Serial Number = 5I:5  
Board Info Area:  
  Version      = 1  
Mfg Date/Time      = Jun 16 15:37:00 2011 (8129737 minutes  
since 1996)  
Board Manufacturer  = Pigeon Point Systems  
Board Serial Number = PPS0000000  
  Board Part Number   = A  
  FRU Programmer File ID =  
Product Info Area:  
  Version      = 1  
  Language Code      = 25  
  Manufacturer Name   = Pigeon Point Systems  
  Product Name        = Shelf Manager  
  Product Part / Model# = 000000  
  Product Version     = Rev. 1.00
```

```
Product Serial Number      = PPS0000000
Asset Tag                  =
FRU Programmer File ID    =
Multi Record Area:
Record Type                = Management Access Record
    Version = 2
Sub-Record Type: Component Name (0x05)
PICMG Address Table Record (ID=0x10)
    Version = 1
PICMG Backplane Point-to-Point Connectivity Record (ID=0x04)
    Version = 0
PICMG Shelf Power Distribution Record (ID=0x11)
    Version = 0
PICMG Shelf Activation And Power Management Record (ID=0x12)
    Version = 0
#
```

getfanlevel

语法:

```
getfanlevel IPMB-address fru-id
getfanlevel fan_tray n
```

用途:

该命令用于显示命令参数中指定的 FRU 所控制风扇的当前级别。

示例:

获取 IPMB 地址为 0x20 的 FRU #2 上风扇的风扇级别。

```
# cli getfanlevel 20 2
IPM Sentry Shelf Manager Command Line Interpreter
20: FRU # 2 Override Fan Level: 1, Local Fan Level: 255
#
```

getfruledstate

语法:

```
getfruledstate [-v] [IPMB-addr state [fru_id [LED_ID | ALL]]]
```

用途:

该命令用于显示为 LED 指示灯启用的所有控制级别的当前 FRU LED 指示灯状态。在详细模式下，还显示有关 LED 指示灯支持的颜色的信息。

可以显示有关给定 FRU 的特定 LED 指示灯或所有 LED 指示灯的信息。也可省略目标 LED 指示灯的 IPMB 地址和 FRU ID。如果省略 FRU ID，则将显示有关给定 IPM 控制器的所有 FRU 上的所有 LED 指示灯的信息。如果还省略 IPMB 地址，则将显示该机框中所有已知 LED 指示灯的信息。

示例:

显示 IPMB 地址为 FCh 的 IPM 控制器上所有 LED 指示灯的 LED 指示灯状态。

```
# clia getfruledstate fc
IPM Sentry Shelf Manager Command Line Interpreter

fc: FRU # 0, Led # 0 ("BLUE LED"):
    Local Control LED State: LED OFF

fc: FRU # 0, Led # 1 ("LED 1"):
    Local Control LED State: LED OFF

fc: FRU # 0, Led # 2 ("LED 2"):
    Local Control LED State: LED OFF

fc: FRU # 0, Led # 3 ("LED 3"):
    Local Control LED State: LED OFF

fc: FRU # 0, Led # 4 ("Application Specific LED# 1"):
    Local Control LED State: LED ON, color: GREEN
```

显示 IPMB 地址为 FCh 的 IPM 控制器上所有 LED 指示灯的 LED 指示灯状态。

```
# clia getfruLEDstate -v FC
IPM Sentry Shelf Manager Command Line Interpreter

fc: FRU # 0, Led # 0 ("BLUE LED"):
  Local Control LED State: LED OFF
  LED's color capabilities:
    Colors supported(0x02): BLUE
    Default LED Color in Local Control State(0x01): BLUE
    Default LED Color in Override State(0x01): BLUE

fc: FRU # 0, Led # 1 ("LED 1"):
  Local Control LED State: LED OFF
  LED's color capabilities:
    Colors supported(0x0C): RED GREEN
    Default LED Color in Local Control State(0x03): GREEN
    Default LED Color in Override State(0x03): GREEN

fc: FRU # 0, Led # 2 ("LED 2"):
  Local Control LED State: LED OFF
  LED's color capabilities:
    Colors supported(0x0C): RED GREEN
    Default LED Color in Local Control State(0x03): GREEN
    Default LED Color in Override State(0x03): GREEN

fc: FRU # 0, Led # 3 ("LED 3"):
  Local Control LED State: LED OFF
  LED's color capabilities:
    Colors supported(0x0C): RED GREEN
    Default LED Color in Local Control State(0x02): RED
    Default LED Color in Override State(0x02): RED

fc: FRU # 0, Led # 4 ("Application Specific LED# 1"):
  Local Control LED State: LED ON, color: GREEN
  LED's color capabilities:
    Colors supported(0x0C): RED GREEN
    Default LED Color in Local Control State(0x02): RED
    Default LED Color in Override State(0x02): RED
```

显示 IPMB 地址为 20h 的 IPM 控制器的 FRU #0 的 LED 指示灯状态。

```
# clia getfruledstate 20 0
IPM Sentry Shelf Manager Command Line Interpreter

20: FRU # 0, Led # 0 ("BLUE LED"):
    Local Control LED State: LED ON, color: BLUE

20: FRU # 0, Led # 1 ("LED 1"):
    Local Control LED State: LED OFF
```

显示 IPMB 地址为 20h 的 IPM 控制器上 FRU #0 的 LED #1 的 LED 指示灯状态。

```
# clia getfruledstate -v 20 0 1
IPM Sentry Shelf Manager Command Line Interpreter

20: FRU # 0, Led # 1 ("LED 1"):
    Local Control LED State: LED OFF
    LED's color capabilities:
        Colors supported(0x04): RED
        Default LED Color in Local Control State(0x02): RED
        Default LED Color in Override State(0x02): RED
```

gethysteresis

语法:

```
gethysteresis [IPMB-address [[lun: ]sensor id | sensor name]]
```

用途:

该命令用于显示指定传感器的当前滞后值。传感器必须是基于阈值的传感器。同时显示原始值和处理过的值。

如果目标控制器支持多个 LUN 上的传感器，则此命令允许用户使用逻辑单元号 (Logical Unit Number, LUN) 来限定传感器编号。如果省略 LUN，则会显示具有指定传感器编号的所有传感器的当前滞后值。*lun* 的取值可以为 0、1 或 3。(LUN 2 为保留值。) 因为假定控制器中传感器名称一般是唯一的，所以不使用 LUN 编号来限定传感器名称。但是，如果控制器中有多个同名传感器，则显示有关所有这些传感器的信息。如果省略 *IPMB-address*，则将显示位于指定 IPMB 地址的所有传感器的当前滞后级别。

示例:

显示 IPMB 地址为 FCh 的 IPM 控制器上 2 号传感器 (sensor # 2) 的滞后值。

```
# clia gethysteresis FC 2
IPM Sentry Shelf Manager Command Line Interpreter

fc: LUN: 0, Sensor # 2 ("lm75 temp")
    Type: Threshold (0x01), "Temperature"(0x01)
        Positive hysteresis, Raw data: 0x00   Processed data:
0.00000 degrees C
        Negative hysteresis, Raw data: 0x00   Processed data:
0.00000 degrees C
```

getipmbstate

语法:

```
getipmbstate IPMB-address [link]   (在辐射型 IPMB-0 环境中)
getipmbstate IPMB-address           (在总线型 IPMB-0 环境中)
```

用途:

该命令用于显示目标 IPM 控制器上 IPMB 0 的当前状态。该状态取自目标 IPMC 上 IPMB 链路传感器 (传感器类型为 F1) 提供的传感器数据。同时列显有关总线 A 和总线 B 的信息。

在总线型环境和辐射型环境中, 该命令的用法有所不同。在总线型环境中, 或者在目标 IPMC 不是 IPMB 集线器的辐射型环境中, 不使用参数 *link*。显示有关目标 IPM 控制器上 IPMB A 和 IPMB B 状态的信息。

在辐射型环境中, 如果目标 IPM 控制器为 IPMB 集线器, 则该命令的用法如下:

- 如果省略 *link*, 则该命令将列显有关所有辐射型 IPMB 链路状态的信息。该状态取自 IPM 控制器上多个 IPMB 链路传感器的传感器数据。
- 如果使用 *link*, 则该命令将列显有关特定辐射型 IPMB 链路 (1 至 95) 的信息。该链路的状态取自 IPM 控制器上相应 IPMB 链路传感器的状态。

在这两种情况下, 都同时显示有关 IPMB A 和 IPMB B 状态的信息。

示例:

显示 IPMB 地址为 92h 的 IPMC 上的 IPMB 0 的当前状态。

```
# clia getipmbstate 92
IPM Sentry Shelf Manager Command Line Interpreter

92: LUN: 0, Sensor # 1 ("IPMB LINK")
  Bus Status: 0x8 (IPMB-A Enabled, IPMB-B Enabled)
  IPMB A State: 0x8 (LocalControl, No failure)
  IPMB B State: 0x8 (LocalControl, No failure)
```

显示辐射型环境中机框管理器的链路 8 的当前状态。

```
# clia getipmbstate 20 8
IPM Sentry Shelf Manager Command Line Interpreter

20: Link: 8, LUN: 0, Sensor # 12 ("IPMB LINK 8")
  Bus Status: 0x8 (IPMB-A Enabled, IPMB-B Enabled)
  IPMB A State: 0x8 (LocalControl, No failure)
  IPMB B State: 0x8 (LocalControl, No failure)
```

getlanconfig

语法:

```
getlanconfig channel [parameter-name [additional-parameters]]
getlanconfig channel [parameter-number [additional-parameters]]
```

用途:

该命令用于显示指定通道上指定 LAN 配置参数的值。如果未指定配置参数名称或编号,则会显示指定通道的所有配置参数。

表 A-1 列出了 getlanconfig 命令支持的 LAN 配置参数的名称和编号：

表 A-1 getlanconfig 的 LAN 配置参数

参数名称	编号	说明
auth_support	1	一个 8 位值，包含 LAN 通道的验证类型支持标志。
auth_enables	2	五个 8 位值，包含 LAN 通道的回调、用户、操作员、管理员和 OEM 权限级别的验证类型启用标志。
ip	3	一个字符串值，包含指定给 LAN 通道的 IP 地址，该地址以点分十进制记法表示（例如，192.168.0.15）。
ip_source	4	一个对指定 IP 地址的源进行编码的值。
mac	5	一个字符串值，包含指定给 LAN 通道的 MAC 地址，该地址以用：符号分隔的六个十六进制字节值表示（例如，00:A0:24:C6:18:2F）。
subnet_mask	6	一个字符串值，包含指定给 LAN 通道的子网掩码，该子网掩码以点分十进制记法表示（例如，255.255.255.0）。
ipv4_hdr_param	7	三个 8 位值，包含各种用于发送 RMCP 包的 IPv4 标头参数： <ul style="list-style-type: none"> • 生存时间 • IP 标头标志（位 [7:5]） • 优先级（位 [7:5]）和服务类型（位 [4:1]）
pri_rmcp_port	8	一个 16 位值，包含主 RMCP 端口号（用于常规 RMCP 通信的端口）。
sec_rmcp_port	9	一个 16 位值，包含辅助 RMCP 端口号（用于安全 RMCP 通信的端口）。
arp_control	10	两个用于控制 LAN 通道上地址解析协议 (address resolution protocol, ARP) 行为的标志： <ul style="list-style-type: none"> • 启用响应 ARP 请求 • 启用发送无故 ARP (Gratuitous ARP)
arp_interval	11	以定点格式表示的无故 ARP (Gratuitous ARP) 的时间间隔（以秒为单位，可能包含小数部分）。
dft_gw_ip	12	一个字符串值，包含以点分十进制记法表示的默认网关的 IP 地址。
dft_gw_mac	13	一个字符串值，包含以用冒号 (:) 分隔的六个十六进制字节值表示的默认网关的 MAC 地址。
backup_gw_ip	14	一个字符串值，包含以点分十进制记法表示的备用网关的 IP 地址。
backup_gw_mac	15	一个字符串值，包含以用冒号 (:) 分隔的六个十六进制字节值表示的备用网关的 MAC 地址。

表 A-1 getlanconfig 的 LAN 配置参数（续）

community	16	一个字符串值（最多 18 个符号），此值置于 PET 陷阱的 Community String 字段中。
destination_count	17	LAN 通道上支持的 LAN 警报目标的最大数目。
destination_type	18	由指定的已设置选择器标识的目标类型。如果未指定已设置的选择器，则会显示所有目标类型。每个目标类型条目包含以下字段： <ul style="list-style-type: none"> • 目标类型 (0-7) • 警报确认标志 • 警报确认超时/重试时间间隔（1-256，以秒为单位） • 重试次数 (0-7)
destination_address	19	与指定的已设置选择器关联的目标地址。如果未指定已设置的选择器，则会显示所有目标地址。每个目标地址条目包含以下字段： <ul style="list-style-type: none"> • 网关选择器：0 - 使用默认网关，1 - 使用备用网关 • IP 地址（以点分十进制格式表示的字符串） • MAC 地址（由六个以冒号[:] 分隔的十六进制字节值组成的字符串）

示例：

以下各小节提供了有关每个所支持参数的更多详细信息。

获取并显示通道 1 的 LAN 参数表。

```
# clia getlanconfig 1

Authentication Type Support: 0x15 (None MD5 Straight Password/Key)
Authentication Type Enables: 0x00
    User level: 0x15 ( None MD5 Straight Password/Key )
    Operator level: 0x15 ( None MD5 Straight Password/Key )
    Administrator level: 0x15 ( None MD5 Straight Password/Key )
    OEM level: 0x00
IP Address: 172.16.2.203
IP Address Source: Static Address (Manually Configured) (01)
MAC Address: 90:91:91:91:91:91
Subnet Mask: 255.255.255.0
IPv4 Header Parameters: 0x40:0x40:0x10
Primary RMCP Port Number: 0x026f
Secondary RMCP Port Number: 0x0298
BMC-generated ARP Control: 02
    Enable BMC-generated Gratuitous Response
Gratuitous ARP Interval: 2.0 seconds
Default Gateway Address: 0.0.0.0
```

```
Default Gateway MAC Address: N/A
Backup Gateway Address: 0.0.0.0
Backup MAC Address: N/A
Community String: "public"
Number of Destinations: 16
#
```

auth_support

语法:

```
getlanconfig channel auth_support
getlanconfig channel 1
```

用途:

该命令用于显示 LAN 参数 `auth_support` 的当前值。该参数指定机框管理器支持的验证类型，它以一个单字节表示，被视为位掩码，其中位的定义如下：

- 0x01 - 无
- 0x02 - MD2
- 0x04 - MD5
- 0x10 - 直接密码/密钥
- 0x20 - OEM 专用

其他位为保留位，应设置为 0。

除原始的十六进制值外，还显示所设置位的符号值。

示例:

```
# cli getlanconfig 1 auth_support
IPM Sentry Shelf Manager Command Line Interpreter
Authentication Type Support: 0x15 ( None MD5 Straight Password/Key
#
```

auth_enables

语法:

```
getlanconfig channel auth_enables  
getlanconfig channel 2
```

用途:

该命令用于显示 LAN 参数 `auth_enables` 的当前值。该参数指定机框管理器当前针对五种支持的权限级别（回调、用户、管理员、操作员和 OEM）中的每一种启用的验证类型，该参数以一个由五个字节组成的序列表示（每个字节对应各自的权限级别），被视为位掩码，其中位的定义如下：

- 0x01 - 无
- 0x02 - MD2
- 0x04 - MD5
- 0x10 - 直接密码/密钥
- 0x20 - OEM 专用

其他位为保留位，应设置为 0。

除原始的十六进制值外，还显示所设置位的符号值。

示例:

```
# clia getlanconfig 1 auth_enables  
IPM Sentry Shelf Manager Command Line Interpreter  
Authentication Type Enables:  
  Callback level: 0x00  
  User level: 0x15 ( None MD5 Straight Password/Key )  
  Operator level: 0x15 ( None MD5 Straight Password/Key )  
  Administrator level: 0x15 ( None MD5 Straight Password/Key )  
  OEM level: 0x00
```

ip

语法:

```
getlanconfig channel ip  
getlanconfig channel 3
```

用途:

该命令用于显示通道当前使用的 IP 地址（以点分十进制记法表示）。

示例:

```
# clia getlanconfig 1 ip  
IPM Sentry Shelf Manager Command Line Interpreter  
IP Address: 172.16.2.203
```

ip_source

语法:

```
getlanconfig channel ip_source  
getlanconfig channel 4
```

用途:

该命令用于显示 LAN 参数 `ip_source` 的当前值。该参数指定机框管理器使用的 IP 地址源，它以一个单字节表示，可以取以下值之一：

- 0 - 未指定
- 1 - 静态地址（手动配置）
- 2 - 运行 DHCP 的机框管理器获得的地址
- 3 - 由 BIOS 或系统软件装入的地址
- 4 - 运行其他地址分配协议的机框管理器获得的地址

其他值为保留值。

除原始十六进制值外，还显示符号值。

示例:

```
# clia getlanconfig 1 ip_source  
IPM Sentry Shelf Manager Command Line Interpreter  
IP Address Source: Static Address (Manually Configured) (0x01)  
#
```

mac

语法:

```
getlanconfig channel mac  
getlanconfig channel 5
```

用途:

该命令用于显示通道当前使用的 MAC 地址（以用冒号分隔的六个十六进制字节的形式表示）。

示例:

```
# clia getlanconfig 1 mac  
IPM Sentry Shelf Manager Command Line Interpreter  
MAC Address: 90:91:91:91:91:91  
#
```

subnet_mask

语法:

```
getlanconfig channel subnet_mask  
getlanconfig channel 6
```

用途:

该命令用于显示通道当前使用的 IP 子网掩码（以点分十进制记法表示）。

示例:

```
# clia getlanconfig 1 subnet_mask  
IPM Sentry Shelf Manager Command Line Interpreter  
Subnet Mask: 255.255.255.0  
#
```


ipv4_hdr_param

语法:

```
getlanconfig channel ipv4_hdr_param  
getlanconfig channel 7
```

用途:

该命令用于显示当前 IPv4 标头参数。这些参数表示为三个以十六进制记数法表示的单字节值（各值用冒号分隔）。这些字节的内容符合 IPMI 1.5 规范的第 19.2 节。

示例:

```
# clia getlanconfig 1 ipv4_hdr_param  
IPM Sentry Shelf Manager Command Line Interpreter  
IPv4 Header Parameters: 0x40:0x40:0x10  
#
```

pri_rmcp_port

语法:

```
getlanconfig channel pri_rmcp_port  
getlanconfig channel 8
```

用途:

该命令用于显示通道当前使用的 RMCP 主端口（以十六进制表示）。这是用于通过 RMCP 进行常规交互的端口。

示例:

```
# clia getlanconfig 1 pri_rmcp_port  
IPM Sentry Shelf Manager Command Line Interpreter  
Primary RMCP Port Number: 0x026f  
#
```

sec_rmcp_port

语法:

```
getlanconfig channel sec_rmcp_port  
getlanconfig channel 9
```

用途:

该命令用于显示通道当前使用的 RMCP 辅助端口（以十六进制表示）。这是用于通过 RMCP 进行安全交互的端口。

示例:

```
# clia getlanconfig 1 sec_rmcp_port  
IPM Sentry Shelf Manager Command Line Interpreter  
Primary RMCP Port Number: 0x0298  
#
```

arp_control

语法:

```
getlanconfig channel arp_control  
getlanconfig channel 10
```

用途:

该命令用于显示 LAN 参数 `arp_control` 的当前值。该参数指定机框管理器提供的附加 ARP 支持，它以一个单字节表示，被视作位掩码，其中位的定义如下：

- 1 - 启用机框管理器生成的无故 ARP (Gratuitous ARP)
- 2 - 启用机框管理器生成的 ARP 响应

其他位为保留位，应设置为 0。

除原始的十六进制值外，还显示所设置位的符号值。

示例:

```
# clia getlanconfig 1 arp_control  
IPM Sentry Shelf Manager Command Line Interpreter  
BMC-generated ARP Control: 02  
    Enable BMC-generated Gratuitous Response  
#
```

arp_interval

语法:

```
getlanconfig channel arp_interval  
getlanconfig channel 11
```

用途:

该命令用于显示通道当前使用的 ARP 时间间隔。该值为以定点数字格式显示的秒数。

示例:

```
# clia getlanconfig 1 arp_interval  
IPM Sentry Shelf Manager Command Line Interpreter  
Gratuitous ARP Interval: 2.0 seconds  
#
```

dft_gw_ip

语法:

```
getlanconfig channel dft_gw_ip  
getlanconfig channel 12
```

用途:

该命令用于显示通道使用的默认网关的 IP 地址（以点分十进制记法表示）。

示例:

```
# clia getlanconfig 1 dft_gw_ip  
IPM Sentry Shelf Manager Command Line Interpreter  
Default Gateway Address: 0.0.0.0  
#
```

dft_gw_mac

语法:

```
getlanconfig channel dft_gw_mac  
getlanconfig channel 13
```

用途:

该命令用于显示通道使用的默认网关的 MAC 地址（以用冒号分隔的六个十六进制字节的形式表示）。

示例:

```
# clia getlanconfig 1 dft_gw_mac  
IPM Sentry Shelf Manager Command Line Interpreter  
Default Gateway MAC Address: N/A  
#
```

backup_gw_ip

语法:

```
getlanconfig channel backup_gw_ip  
getlanconfig channel 14
```

用途:

该命令用于显示通道使用的备用网关的 IP 地址（以点分十进制记法表示）。

示例:

```
# clia getlanconfig 1 backup_gw_ip  
IPM Sentry Shelf Manager Command Line Interpreter  
Backup Gateway Address: 0.0.0.0  
#
```

backup_gw_mac

语法:

```
getlanconfig channel backup_gw_mac  
getlanconfig channel 15
```

用途:

该命令用于显示通道使用的备用网关的 MAC 地址（以用冒号分隔的六个十六进制字节的形式表示）。

示例:

```
# clia getlanconfig 1 backup_gw_mac  
IPM Sentry Shelf Manager Command Line Interpreter  
Backup Gateway MAC Address: N/A  
#
```

community

语法:

```
getlanconfig channel community  
getlanconfig channel 16
```

用途:

该命令用于显示 PET 陷阱中使用的团体字符串参数。

示例:

```
# clia getlanconfig 1 community  
IPM Sentry Shelf Manager Command Line Interpreter  
Community String: "public"  
#
```

destination_count

语法:

```
getlanconfig channel destination_count  
getlanconfig channel 17
```

用途:

该命令用于显示通道可用的警报目标的最大数目。这是 IPM Sentry 机框管理器的一个配置参数，只能通过 `shelfman` 配置文件进行更改。

示例:

```
# clia getlanconfig 1 destination_count  
IPM Sentry Shelf Manager Command Line Interpreter  
Number of Destinations: 16  
#
```

destination_type

语法:

```
getlanconfig channel destination_type [set-selector]  
getlanconfig channel 18 [set-selector]
```

用途:

该命令用于显示目标表中索引等于 `set-selector` 的元素。索引是从 0 开始的。选择器 0 用于对易失性目标进行寻址。将显示有关目标的以下信息:

- 目标选择器
- 警报目标类型 (PET 陷阱或 OEM 目标; 是否应对该警报进行确认)
- 警报确认超时
- 重试计数

如果省略已设置的选择器, 则会显示所有活动目标及其编号。

示例:

```
# clia getlanconfig 1 destination_type 2
IPM Sentry Shelf Manager Command Line Interpreter
DST Type # 2, Type: Acknowledged PET Trap Destination (0x80), ACK
Timeout / Retry Interval: 3 seconds, Retries: 5

# clia getlanconfig 1 destination_type
IPM Sentry Shelf Manager Command Line Interpreter
DST Type # 0, Type: Acknowledged reserved (0x81), ACK Timeout /
Retry Interval: 2 seconds, Retries: 6
DST Type # 1, Type: Unacknowledged reserved (0x02), ACK Timeout /
Retry Interval: 3 seconds, Retries: 4
DST Type # 2, Type: Acknowledged PET Trap Destination (0x80), ACK
Timeout / Retry Interval: 3 seconds, Retries: 5
#
```

destination_address

语法:

```
getlanconfig channel destination_address [set-selector] getlanconfig
channel 19 [set-selector]
```

用途:

该命令用于显示目标地址表中索引等于 *set-selector* 的元素。索引是从 0 开始的。选择器 0 用于对易失性目标进行寻址。将显示有关目标的以下信息:

- 目标选择器
- 地址格式 (默认为 IP+MAC)
- 目标 IP 地址
- 目标 MAC 地址
- 使用哪个网关 (默认与备用)

如果省略已设置的选择器, 则会显示所有活动目标地址及其编号。

示例:

```
# clia getlanconfig 1 destination_address 2
IPM Sentry Shelf Manager Command Line Interpreter
DST Addresses # 2, Address Format: IPv4 IP Address followed by DIX
ethernet / 802.3 MAC Address (0x00)
    Gateway: Default (0x00), Alerting IP: 172.16.2.100, Alerting
MAC: 90:93:93:93:93:93
#
```

getpefconfig

语法:

```
getpefconfig  
getpefconfig parameter-name [additional-parameters]  
getpefconfig parameter-number [additional-parameters]
```

用途:

该命令用于显示指定的 PEF 配置参数的值。如果既没指定配置参数名称，也没指定参数编号，则将显示所有 PEF 配置参数。

表 A-2 列出了 PEF 配置参数的名称和编号:

表 A-2 PEF 配置参数

参数名称	编号	说明
control	1	一个 8 位值，代表 PEF 的控制标志（启用 PEF、启用 PEF 启动延迟等）。
action_control	2	一个 8 位值，代表 PEF 操作全局控制标志（启用复位、启用关闭电源等）。
startup_delay	3	系统加电并复位后延迟 PEF 的时间（以秒为单位）。
alert_startup_delay	4	系统加电并复位后延迟警报的时间（以秒为单位）。
event_filter_count	5	事件过滤器的最大数目。
event_filter	6	由指定的已设置选择器标识的事件过滤器表项。如果未指定已设置的选择器，则会显示所有活动事件过滤器。
event_filter_data1	7	由指定的已设置选择器标识的事件过滤器表项的第一个字节。如果未指定已设置的选择器，则会显示所有活动事件过滤器。
alert_policy_count	8	警报策略的最大数目。
alert_policy	9	由指定的已设置选择器标识的警报策略表项。如果未指定已设置的选择器，则会显示所有活动警报策略。
system_guid	10	用于填充 PET 陷阱中 GUID 字段的 GUID。
alert_string_count	11	警报字符串的最大数目。
alert_string_key	12	由指定的已设置选择器标识的警报字符串键值。如果未指定已设置的选择器，则会显示所有警报字符串键值。

表 A-2 PEF 配置参数 (续)

alert_string	13	由指定的已设置选择器标识的警报字符串。如果未指定已设置的选择器，则会显示所有警报字符串。
oem_filter_count	96	OEM 过滤器的最大数目。
oem_filter	97	由指定的已设置选择器标识的 OEM 过滤器表项。如果未指定已设置的选择器，则会显示所有活动事件过滤器。

示例:

以下各小节提供了有关每个所支持参数的更多详细信息。

获取并显示整个 PEF 参数表。

```
# clia getpefconfig
IPM Sentry Shelf Manager Command Line Interpreter
PEF parameters:
  PEF control: 0x00
  PEF Action Global Control: 0x00
  PEF Startup Delay: 60 seconds
  PEF Alert Startup Delay: 60 seconds
  PEF Number of Event Filters: 64
  PEF Number of OEM Filters: 16
  Active Event Filters:
    None
  Active event filter data:
    None
  Alert Policies Count: 64
  Policy:
    None
  PEF GUID: Using the system GUID
Alert Strings Count: 64
Alert string key:
  None
Alert Strings:
  None
#
```

control

语法:

```
getpefconfig control  
getpefconfig 1
```

用途:

该命令用于显示 PEF 参数 control 的当前值。该参数为一个单字节，被视作位掩码，其中位的定义如下：

- 0x01 - 启用 PEF
- 0x02 - 启用为 PEF 操作生成事件消息
- 0x04 - 系统加电并复位时启用 PEF 启动延迟
- 0x08 - 启用 PEF 警报启动延迟

其他位为保留位，应设置为 0。

示例:

```
# clia getpefconfig control  
IPM Sentry Shelf Manager Command Line Interpreter  
PEF control: 0x07  
    Enable PEF  
    Enable Event Message for PEF Actions  
    Enable PEF Startup Delay  
#
```

action_control

语法:

```
getpefconfig action_control  
getpefconfig 2
```

用途:

该命令用于显示 PEF 参数 action_control 的当前值。该参数为一个单字节，被视作位掩码，其中位的定义如下：

- 0x01 - 启用警报操作
- 0x02 - 启用关闭电源操作
- 0x04 - 启用复位操作
- 0x08 - 启用关闭再打开电源操作
- 0x01 - 启用 OEM 操作
- 0x20 - 启用诊断中断

其他位为保留位，应设置为 0。

示例:

```
# clia getpefconfig action_control
IPM Sentry Shelf Manager Command Line Interpreter
PEF Action Global Control: 0x3f
    Enable Alert Action
    Enable Power Down Action
    Enable Reset Action
    Enable Power Cycle Action
    Enable OEM Action
    Enable Diagnostic Interrupt
#
```

startup_delay

语法:

```
getpefconfig startup_delay
getpefconfig 3
```

用途:

该命令用于显示 PEF 参数 `startup_delay` 的当前值。该参数为一个单字节，表示 PEF 设备在启动时延迟的秒数。

示例:

```
# clia getpefconfig startup_delay
IPM Sentry Shelf Manager Command Line Interpreter
    PEF Startup Delay: 60 seconds
#
```

alert_startup_delay

语法:

```
getpefconfig startup_delay
getpefconfig 4
```

用途:

该命令用于显示 PEF 参数 `alert_startup_delay` 的当前值。该参数为一个单字节，表示警报设备在启动时延迟的秒数。

示例:

```
# clia getpefconfig alert_startup_delay
IPM Sentry Shelf Manager Command Line Interpreter
PEF Alert Startup Delay: 60 seconds
#
```

event_filter_count

语法:

```
getpefconfig event_filter_count
getpefconfig 5
```

用途:

该命令用于显示 PEF 参数 `event_filter_count` 的当前值。该只读值为事件过滤器表的大小。该值是 IPM Sentry 机框管理器的一个配置参数，只能通过 `shelfman` 配置文件进行更改。

示例:

```
# clia getpefconfig event_filter_count
IPM Sentry Shelf Manager Command Line Interpreter
PEF Number of Event Filters: 64
#
```

event_filter

语法:

```
getpefconfig event_filter [set-selector]
getpefconfig 6 [set-selector]
```

用途:

该命令用于显示事件过滤器表中索引等于 `set-selector` 的元素。索引是从 1 开始的。将显示有关每个事件过滤器的以下信息:

- 过滤器配置: 过滤器由软件配置还是由制造商预先配置
- 事件过滤器操作掩码
- 警报策略编号

- 事件严重程度
- 匹配的事件源地址（255 表示任何地址）
- 匹配的源通道/LUN（255 表示匹配任何源通道/LUN）
- 匹配的传感器类型
- 匹配的传感器编号
- 匹配的事件触发器（事件/读取类型）
- 事件偏移掩码
- 事件数据第 1、2 和 3 个字节的 AND、比较 1 (CMP1) 和比较 2 (CMP2) 掩码。

如果省略已设置的选择器，则会显示所有活动事件过滤器表项及其编号。

示例：

```
# clia getpefconfig event_filter 2

IPM Sentry Shelf Manager Command Line Interpreter
Active Event Filters:
0x02: Software Configurable Filter
    Action Mask: 0x01
    Policy Number: 1, Severity: Critical Condition
    Source Address: 0x20, LUN: 3, Channel: 15
    Sensor Type: Hot Swap (0xf0), Sensor # 255 (ANY)
    Event Trigger: 0xff (ANY), Event Offset Mask: 0xffff
    0: AND: 0x0f, CMP1: 0xff, CMP2: 0x00
    1: AND: 0x00, CMP1: 0x00, CMP2: 0x00
    2: AND: 0xff, CMP1: 0xff, CMP2: 0x00

#
```

event_filter_data1

语法：

```
getpefconfig event_filter_data1 [set-selector]
getpefconfig 7 [set-selector]
```

用途：

该命令用于显示事件过滤器表中索引等于 *set-selector* 的元素的第一个字节。索引是从 1 开始的。该字节以十六进制格式显示。该字节中位的含义如下：

- 0x80 – 已启用该过滤器。
- 0x40 – 该过滤器由制造商进行预配置，不应通过软件进行更改。

其他位为保留位，应为 0。

如果省略已设置的选择器，则会显示每个活动事件过滤器表项的第一个字节及相应的过滤器编号。

示例：

```
# clia getpefconfig event_filter_data1 2
IPM Sentry Shelf Manager Command Line Interpreter

    Active event filter data:
        0x02: 0x80 Enabled 1, Configuration: 0 ("Software
Configurable Filter")
#
```

alert_policy_count

语法：

```
getpefconfig alert_policy_count
getpefconfig 8
```

用途：

该命令用于显示 PEF 参数 `alert_policy_count` 的当前值。该只读值为警报策略表的大小。该值是 IPM Sentry 机框管理器的一个配置参数，只能通过 `shelfman` 配置文件进行更改。

示例：

```
# clia getpefconfig alert_policy_count
IPM Sentry Shelf Manager Command Line Interpreter

    Alert Policies Count: 64
#
```

alert_policy

语法:

```
getpefconfig alert_policy [set-selector]
getpefconfig 9 [set-selector]
```

用途:

该命令用于显示警报策略表中索引等于 *set-selector* 的元素。索引是从 1 开始的。将显示有关每个警报策略的以下信息:

- 策略编号
- 策略类型 (相对于发送到上一个目标的警报)
- 目标通道号
- 目标选择器
- 警报字符串键值

如果省略已设置的选择器, 则会显示所有活动警报策略表项及其编号。

示例:

```
# clia getpefconfig alert_policy 2
IPM Sentry Shelf Manager Command Line Interpreter

Policy:
    0x02: Policy# 5, Policy Type: 0, Channel: 1, DST: 1, Alert
String Sel: 1
#
```

system_guid

语法:

```
getpefconfig system_guid
getpefconfig 10
```

用途:

该命令用于显示 PEF 参数 *system_guid* 的当前值。该参数表示通过 PET 陷阱 PDU 发送到警报目标的 GUID。可将该 GUID 定义为一个独立的 GUID 或定义为等同于系统 GUID (可以通过 Get System GUID IPMI 命令获得)。

示例:

```
# clia getpefconfig system_guid
IPM Sentry Shelf Manager Command Line Interpreter
PEF GUID: 23662f7f-ba1b-4b65-8808-94ca09c9bbb0
#
```

alert_string_count

语法:

```
getpefconfig alert_string_count  
getpefconfig 11
```

用途:

该命令用于显示 PEF 参数 `alert_string_count` 的当前值。该只读值为警报字符串表的大小，亦即同时使用的警报字符串的最大数目。该值是 IPM Sentry 机框管理器的配置参数，只能通过 `shelfman` 配置文件进行更改。

示例:

```
# clia getpefconfig alert_string_count  
  
IPM Sentry Shelf Manager Command Line Interpreter  
  
Alert Strings Count: 64  
  
#
```

alert_string_key

语法:

```
getpefconfig alert_string_key [set-selector]  
getpefconfig 12 [set-selector]
```

用途:

该命令用于显示警报字符串键值表中索引等于 `set-selector` 的元素。索引是从 1 开始的。可以使用索引 0 来指定易失性警报字符串。为了生成警报，每个键值会将事件过滤器与警报字符串相关联。将显示有关每个警报字符串键值的以下信息:

- 警报字符串键值编号
- 关联的事件过滤器编号
- 关联的警报字符串编号

如果省略已设置的选择器，则会显示所有活动警报字符串键值表项及其编号。

示例:

```
# clia getpefconfig alert_string_key 2  
IPM Sentry Shelf Manager Command Line Interpreter  
Alert string key: set selector 2, event_filter 0x10, string_set  
0x11  
  
#
```


alert_string

语法:

```
getpefconfig alert_string [set-selector]  
getpefconfig 13 [set-selector]
```

用途:

该命令用于显示警报字符串表中索引等于 *set-selector* 的元素。索引是从 1 开始的。可以使用索引 0 来指定易失性警报字符串。该命令可同时显示整个字符串。

如果省略已设置的选择器，则会显示所有已定义的警报字符串及其编号。

示例:

```
# clia getpefconfig alert_string 2  
IPM Sentry Shelf Manager Command Line Interpreter  
Alert Strings:  
    0x02: "This is the alert string"  
#
```

oem_filter_count

语法:

```
getpefconfig oem_filter_count  
getpefconfig 96
```

用途:

该命令用于显示 PEF 参数 *oem_filter_count* 的当前值。该只读值为 OEM 过滤器表的大小。该值是 IPM Sentry 机框管理器的一个配置参数，只能通过 *shelfman* 配置文件进行更改。

OEM 过滤器表是 Pigeon Point Systems 定义的对 IPMI 规范的 OEM 扩展。除平台事件外，它还允许将 PEF 应用到加 OEM 时间戳的和未加时间戳的 SEL 项（记录类型范围为 C0h-FFh）。

示例:

```
# clia getpefconfig oem_filter_count  
IPM Sentry Shelf Manager Command Line Interpreter  
PEF Number of OEM Filters: 16  
#
```

oem_filter

语法:

```
getpefconfig oem_filter [set-selector]  
getpefconfig 97 [set-selector]
```

用途:

OEM 过滤器表是 Pigeon Point Systems 定义的对 IPMI 规范的 OEM 扩展。除平台事件外，它还允许将 PEF 应用到加 OEM 时间戳的和未加时间戳的 SEL 项（记录类型范围为 C0h-FFh）。

OEM 过滤器表的各项都会定义应用该 OEM 过滤器的记录类型范围（在 OEM 记录类型的范围内）以及将记录类型相符的记录放置在 SEL 中时要调用的警报策略编号。

该命令用于显示 OEM 过滤器表中索引等于 *set-selector* 的元素。索引是从 1 开始的。将显示有关每个 OEM 过滤器的以下信息：

- 第 1 个字节：SEL 记录类型范围的下限
- 第 2 个字节：SEL 记录类型范围的上限
- 第 3 个字节：针对其记录类型符合第 1 个字节和第 2 个字节中所指定范围的 SEL 项所要调用的警报策略编号。

如果省略已设置的选择器，则会显示所有活动 OEM 过滤器表项及其编号。

示例:

```
# clia getpefconfig oem_filter  
IPM Sentry Shelf Manager Command Line Interpreter  
Active OEM Filters:  
0x01: OEM range boundary 0xff: 0xff, alert policy # 1  
#
```

getsensoreventenable

语法:

```
getsensoreventenable [IPMB-address [sensor-name | [lun:]sensor-number]]
getsensoreventenable board n [sensor-name | [lun:]sensor-number]]
getsensoreventenable shm n [sensor-name | [lun:]sensor-number]]
```

该命令用于显示指定传感器的当前事件启用掩码值。

如果目标控制器支持多个 LUN 上的传感器，则此命令允许用户使用逻辑单元号 (Logical Unit Number, LUN) 来限定传感器编号。如果省略 LUN，则会显示有关所有 LUN 上具有指定传感器编号的传感器的信息。*lun* 的取值可以为 0、1 或 3。(LUN 2 为保留值。)

因为假定控制器中传感器名称一般是唯一的，所以不使用 LUN 编号来限定传感器名称。但是，如果控制器中有多个同名传感器，则显示有关所有这些传感器的信息。

该命令会显示指定传感器所支持事件的当前传感器事件掩码值。另外，还会显示每个传感器的以下属性:

- 所属 IPM 控制器的 IPMB 地址
- 可按其访问传感器的传感器编号、传感器名称 (SDR 中的设备 ID 字符串) 和 LUN
- 传感器类型

示例:

获取 IPM 控制器 FE 上温度传感器 Local Temp 的事件启用值。

```
# clia getsensoreventenable -v fe "Local Temp"

IPM Sentry Shelf Manager Command Line Interpreter

fe: LUN: 0, Sensor # 3 ("Local Temp")
    Type: Threshold (0x01), "Temperature" (0x01)
    Assertion event mask: 0x0a80
        Assertion event for "Upper Non-Recoverable Going High"
enabled
        Assertion event for "Upper Critical Going High" enabled
        Assertion event for "Upper Non-Critical Going High" enabled
    Deassertion event mask: 0x0a80
        Deassertion event for "Upper Non-Recoverable Going High"
enabled
```

```
Deassertion event for "Upper Critical Going High" enabled
Deassertion event for "Upper Non-Critical Going High"
enabled
#
```

获取同一传感器的事件启用信息，但指定传感器 LUN 和编号。

```
# clia getsensoreventenable -v fe 0:3

IPM Sentry Shelf Manager Command Line Interpreter

fe: LUN: 0, Sensor # 3 ("Local Temp")
    Type: Threshold (0x01), "Temperature" (0x01)
    Assertion event mask: 0x0a80
        Assertion event for "Upper Non-Recoverable Going High"
enabled
        Assertion event for "Upper Critical Going High" enabled
        Assertion event for "Upper Non-Critical Going High" enabled
    Deassertion event mask: 0x0a80
        Deassertion event for "Upper Non-Recoverable Going High"
enabled
        Deassertion event for "Upper Critical Going High" enabled
        Deassertion event for "Upper Non-Critical Going High"
enabled
#
```

getthreshold | threshold

语法:

```
getthreshold [IPMB-address [sensor-name | [lun:]sensor-number]]
getthreshold board n [sensor-name | [lun:]sensor-number]]
getthreshold shm n [sensor-name | [lun:]sensor-number]]
```

也可以使用动词 `threshold` 来代替 `getthreshold`。

用途:

该命令用于显示指定传感器所支持阈值的当前阈值。传感器必须是基于阈值的传感器。同时显示原始值和处理过的值。另外，还会显示每个传感器的以下属性:

- 所属 IPM 控制器的 IPMB 地址
- 可按其访问传感器的传感器编号、传感器名称 (SDR 中的设备 ID 字符串) 和 LUN
- 传感器类型和事件/读取类型代码

如果目标控制器支持多个 LUN 上的传感器，则此命令允许用户使用逻辑单元号 (Logical Unit Number, LUN) 来限定传感器编号。如果省略 LUN，则会显示有关所有 LUN 上具有指定传感器编号的传感器的信息。*lun* 的取值可以为 0、1 或 3。(LUN 2 为保留值。)

因为假定控制器中传感器名称一般是唯一的，所以不使用 LUN 编号来限定传感器名称。但是，如果控制器中有多个同名传感器，则显示有关所有这些传感器的信息。

示例:

获取 IPM 控制器 FE 上温度传感器 Local Temp 的阈值。

```
# clia getthreshold -v fe "Local Temp"

IPM Sentry Shelf Manager Command Line Interpreter

fe: LUN: 0, Sensor # 3 ("Local Temp")
    Type: Threshold (0x01), "Temperature" (0x01)
        Lower Critical Threshold, Raw Data: 0x80, Processed Data:
-128.000000 degrees C
        Upper Non-Critical Threshold, Raw Data: 0x50, Processed
Data: 80.000000 degrees C
        Upper Critical Threshold, Raw Data: 0x50, Processed Data:
80.000000 degrees C
        Upper Non-Recoverable Threshold, Raw Data: 0x50, Processed
Data: 80.000000 degrees C
#
```

获取同一传感器的阈值信息，但指定传感器 LUN 和编号。

```
# clia getthreshold -v fe 0:3

IPM Sentry Shelf Manager Command Line Interpreter

fe: LUN: 0, Sensor # 3 ("Local Temp")
    Type: Threshold (0x01), "Temperature" (0x01)
        Lower Critical Threshold, Raw Data: 0x80, Processed Data:
-128.000000 degrees C
        Upper Non-Critical Threshold, Raw Data: 0x50, Processed
Data: 80.000000 degrees C
        Upper Critical Threshold, Raw Data: 0x50, Processed Data:
80.000000 degrees C
        Upper Non-Recoverable Threshold, Raw Data: 0x50, Processed
Data: 80.000000 degrees C
#
```

help

语法:

```
help [command [subcommand]]
```

用途:

该命令用于显示所支持命令及其语法的帮助信息。

示例:

获取命令及其语法的列表。

```
# clia help
IPM Sentry Shelf Manager Command Line Interpreter
Command Line Interface command set:
Parameters are case insensitive
In general:
    IPMB address is hexadecimal ALWAYS.
    All other numbers may be either decimal and hexadecimal (0x notation
    required for hexadecimal numbers)
    -v turns on verbose output

activate <addr> <fru_id>
alarm <alarm status/action>
board [slot_number]
boardreset <slot number>
busres force <res>
busres info [<res>]
busres lock <res>
busres query [-v] <res> [<target> [nouupdate]]
busres release <res>
busres sendbusfree <res> <target>
busres setowner <res> <target>
busres unlock <res>
console [slot_number]
deactivate <addr> <fru_id>
debuglevel [<mask>]
exit
fans <addr> <fru id>
fru [<addr> [id=<fru_id> | type=<site_type>]] | [type=<site_type>
    [/<site_number>]]
frucontrol <addr> <fru_id> <command>
frudata [<addr>] [<fru id>] [<block number>]
```

```

frudata shm <N> [<block number>]
frudata <addr> <fru id> <byte offset> <byte_1> [byte2 ..[byte_16]]
frudatar <addr> <fru id> <file name>
frudataw <addr> <fru id> <file name>
fruinfo <addr> <fru_id>
getfanlevel <addr> <fru_id>
getlanconfig <channel number> <parameter number> | <parameter name>
getpefconfig <parameter name> | <parameter number> [<set selector>]
getsensoreventenable [ <addr> [ [ lun: ]<sensor_id> | <sensor name> ] ]
getthreshold [ <addr> [ [ lun: ]<sensor id> | <sensor name> ] ]
help [<command>]
ipmc [<addr>]
localaddress
minfanlevel [<min fan level>]
poll
quit
sel [clear] [ <addr> [ <number of items> [<number of first item>] ] ]
sel info [<addr>]
sensor [ <addr> [ [ lun: ]<sensor id> | <sensor name> ] ]
sensordata [ <addr> [ [ lun: ]<sensor id> | <sensor name> ] ]
sensorread <addr> [ lun: ]<sensor id>
session
setextracted <addr> <fru_id>
setfanlevel <addr> <fru_id> <state>
setlanconfig <channel number> <parameter number> | parameter name
    <parameters ...>
setlocked <addr> <fru_id> <value>
setpefconfig <parameter name> | <parameter number> [<set selector>]
    <parameters ...>
setsensoreventenable <addr> [ lun: ]<sensor_id> | <sensor name> global
    [assertion_events [deassertion_events]]
setthreshold <addr> [ lun: ]<sensor_id> | <sensor name> unc | uc | unr
    | lnc | lc | lnr [-r] value
shelf <parameters>
shelfaddress ["<shelf address>"]
shmstatus
shownhealthy
switchover
threshold [ <addr> [ [ lun: ]<sensor id> | <sensor name> ] ]
user [<user id>]
user add <user id> <user name> <flags> <privilege level> <password>
    user channel <user id> <channel number> <flags> <privilege level>
    user delete <user id>
    user delete <user id>
    user enable <user id> 1|0
    user name <user id> <user name>
    user passwd <user id> <user password>
version

```

```
# clia help shelf pwrreorder
IPM Sentry Shelf Manager Command Line Interpreter
  Set the Power Order
  PwrReorder <addr1> <fru_id1> before/after <addr2> <fru_id2>
#
```

获取特定命令的帮助信息。

```
# clia help shelf pwrreorder
IPM Sentry Shelf Manager Command Line Interpreter
  Set the Power Order
  PwrReorder <addr1> <fru_id1> before/after <addr2> <fru_id2>
#
```

ipmc

语法:

```
ipmc [-v] [IPMB-address]
ipmc board n
ipmc fan_tray n
```

用途:

该命令用于显示有关位于指定地址的 IPM 控制器的信息，或显示有关机框管理器能识别的所有 IPM 控制器的信息（如果省略 *IPMB-address*）。

在标准模式下将显示有关 IPM 控制器的以下信息:

- 控制器的 IPMB 地址（以两个十六进制数字表示）
- IPM 控制器的实体 ID 和实体实例
- IPM 控制器的可能的最大 FRU 设备 ID
- PICMG 扩展版本。对于符合 PICMG 3.0 规范的 IPM 控制器而言，此版本应该是 2.0 版。

当前热交换状态、先前热交换状态和 IPM 控制器的 FRU 设备 0（代表 IPM 控制器自身）上次状态更改的原因。PICMG 3.0 规范中对热交换状态 M0 至 M7 定义如下:

- M0 - 未安装
- M1 - 非活动
- M2 - 激活请求
- M3 - 正在激活
- M4 - FRU 活动
- M5 - 取消激活请求
- M6 - 正在取消激活
- M7 - 通信丢失

在详细模式下还将显示有关 IPM 控制器的以下信息：

- Get Device ID IPMI 命令返回的信息，包括制造商 ID、产品 ID、设备 ID、设备固件修订版和支持的 IPMI 版本
- 控制器 SDR 中的设备 ID 字符串
- 控制器 SDR 中的电源状态通知属性（以十六进制数表示）
- 控制器 SDR 中的全局初始化属性（以十六进制数表示）
- 控制器 SDR 中的设备功能属性（以十六进制数表示）
- 控制器是否提供设备 SDR
- 支持的功能掩码以及每个位的文本解释
- 受电子钥控 (E-Keying) 控制的端口及其状态（启用/禁用）的列表

示例：

获取有关地址为 9C 的 IPM 控制器的信息。

```
# clia ipmc 9c
IPM Sentry Shelf Manager Command Line Interpreter
9c: Entity: (0xd0, 0x0) Maximum FRU device ID: 0x08
    PICMG Version 2.0
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
#
```

获取有关地址为 9C 的 IPM 控制器的详细信息。

```
# clia ipmc -v 9c
IPM Sentry Shelf Manager Command Line Interpreter
9c: Entity: (0xd0, 0x0) Maximum FRU device ID: 0x08
    PICMG Version 2.0
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID: 0x00, Revision: 0, Firmware: 1.01, IPMI ver 1.5
    Manufacturer ID: 00315a (PICMG), Product ID: 0000, Auxiliary
Rev: 01ac10ac
    Device ID String: "IPM Sentry 6"
    Global Initialization: 0x0, Power State Notification: 0x0,
Device Capabilities: 0x29
    Controller provides Device SDRs
    Supported features: 0x29
    "Sensor Device" "FRU Inventory Device" "IPMB Event
Generator"
#
```

localaddress

语法:

```
localaddress
```

用途:

该命令用于基于当前机框管理器的硬件地址（而不是其通用 BMC 地址 0x20）显示其 IPMB 地址。冗余机框管理器之间的 IPMB 地址会不同（但其 BMC 地址相同）。

示例:

```
# clia localaddress
IPM Sentry Shelf Manager Command Line Interpreter
Local IPMB Address = 0xFC
#
```

minfanlevel

语法:

```
minfanlevel [level]
```

用途:

该命令用于显示或设置最低风扇级别。在正常情况下，当温度条件保持正常时，冷却管理算法将逐渐降低系统中风扇的级别。但是，冷却管理算法不会尝试将风扇级别降低到配置参数 MIN_FAN_LEVEL 或该命令指定的最低级别。

最低风扇级别的默认值是 1。如果将最低风扇级别设置为较高的值，则可以通过命令 `clia setfanlevel` 或 ATCA 命令 `SetFanLevel`（通过 RMCP 提交）将风扇级别设置为低于该值。最低风扇级别仅影响冷却管理工具对风扇级别的自动管理。

如果使用该命令时不带参数，则会显示当前最低风扇级别。

如果使用该命令时带一个整数参数，则会将最低风扇级别设置为该参数值。

示例:

```
# clia minfanlevel 3
IPM Sentry Shelf Manager Command Line Interpreter
Minimal Fan Level is set to 3

# clia minfanlevel
IPM Sentry Shelf Manager Command Line Interpreter
Minimal Fan Level is 3
#
```

sel

语法:

```
sel [-v] [IPMB-address [record-count [starting-entry]]]
sel clear [IPMB-address]
sel info [IPMB-address]
```

可以使用 `board n` 或 `shm n` 缩写来代替 `IPMB-address`。

用途:

该命令用于显示指定 IPM 控制器（默认位于 IPMB 地址 20h）的系统事件日志 (System Event Log, SEL) 的内容。可以指定可选参数 `record-count`，该参数指示从 SEL 中记录编号为 `starting-entry` 的记录开始显示多少条记录。可选参数 `starting-entry` 是要列显的第一条 SEL 记录的条目编号（相对于 SEL 的起始位置）。`record-count` 和 `starting-entry` 的范围都必须介于 1 和该 SEL 中记录总数之间。可选参数 `starting-entry` 的默认值为 1。`starting-entry` 与 SEL 记录的 RecordID 字段无关。

对于每条 SEL 记录，将显示以下信息字段:

- 记录 ID
- 记录类型（当前仅支持事件，所以会针对事件显示 Event 字样）
- 时间戳（对于加了时间戳的记录）
- 源地址参数：IPMB 地址、LUN 和通道号
- 生成该事件的传感器的类型和编号
- 事件/读取类型代码
- 三个字节的 事件数据（以原始格式和处理过的格式（如果可用）表示）

命令 `sel clear` 用于清除指定 IPM 控制器（默认位于 IPMB 地址 20h）上的 SEL。

使用 `-v` 选项可使 SEL 条目的输出更加用户友好。

示例:

读取机框管理器上的 SEL。

```
# clia sel info
IPM Sentry Shelf Manager Command Line Interpreter
20: SEL version: 1.5
    Number of log entries: 43
    Free space: 15680 bytes
    Last addition timestamp: Nov 19 17:12:47 2003
    Last erase timestamp: Oct 31 23:59:59 2003
    Supported operations: 0x0f

# clia sel 20 5
IPM Sentry Shelf Manager Command Line Interpreter
0x0027: Event: at Nov 19 17:12:42 2003; from:(0x9c,0,0);
sensor:(0xf0,0); event:0x6f(asserted): HotSwap: FRU 0 M4->M6,
Cause=0x1
0x0028: Event: at Nov 19 17:12:42 2003; from:(0x9c,0,0);
sensor:(0xf0,0); event:0x6f(asserted): HotSwap: FRU 0 M6->M1,
Cause=0x0
0x0029: Event: at Nov 19 17:12:46 2003; from:(0x9c,0,0);
sensor:(0xf0,0); event:0x6f(asserted): HotSwap: FRU 0 M1->M2,
Cause=0x2
0x002A: Event: at Nov 19 17:12:46 2003; from:(0x9c,0,0);
sensor:(0xf0,0); event:0x6f(asserted): HotSwap: FRU 0 M2->M3,
Cause=0x1
0x002B: Event: at Nov 19 17:12:47 2003; from:(0x9c,0,0);
sensor:(0xf0,0); event:0x6f(asserted): HotSwap: FRU 0 M3->M4,
Cause=0x0

# clia sel b4 5
IPM Sentry Shelf Manager Command Line Interpreter
0x00A4: Event: at Nov 19 01:24:25 2003; from:(0x20,0,0);
sensor:(0x02,4); event:0x1(asserted): "Lower Non-Critical",
Threshold: 0xb3, Reading: 0xb3
0x00B8: Event: at Nov 19 00:04:11 2003; from:(0x20,0,0);
sensor:(0x02,4); event:0x1(asserted): "Lower Non-Critical",
Threshold: 0xb3, Reading: 0xb3
0x00CC: Event: at Nov 19 00:36:32 2003; from:(0x20,0,0);
sensor:(0x02,7); event:0x1(asserted): "Lower Non-Critical",
Threshold: 0xae, Reading: 0x94
0x00E0: Event: at Nov 19 00:36:32 2003; from:(0x20,0,0);
sensor:(0x02,7); event:0x1(asserted): "Lower Critical",
Threshold: 0xac, Reading: 0x94
```

```
0x00F4: Event: at Nov 19 00:02:37 2003; from:(0x20,0,0);
sensor:(0x01,2); event:0x1(asserted): "Upper Critical",
Threshold: 0x13, Reading: 0x1c

# clia sel -v board 3 5
IPM Sentry Shelf Manager Command Line Interpreter
0x00A4: Event: at: Nov 19 01:24:25 2003; from IPM Controller: 0x20,
LUN: 0, Channel: 0
    "Voltage" (0x02) sensor # 4
    "Threshold" (0x01) event Asserted
    "Lower Non-Critical Going Low"
    Reading value: 0xb3
    Threshold value: 0xb3
0x00B8: Event: at: Nov 19 00:04:11 2003; from IPM Controller: 0x20,
LUN: 0, Channel: 0
    "Voltage" (0x02) sensor # 4
    "Threshold" (0x01) event Asserted
    "Lower Non-Critical Going Low"
    Reading value: 0xb3
    Threshold value: 0xb3
0x00CC: Event: at: Nov 19 00:36:32 2003; from IPM Controller: 0x20,
LUN: 0, Channel: 0
    "Voltage" (0x02) sensor # 7
    "Threshold" (0x01) event Asserted
    "Lower Non-Critical Going Low"
    Reading value: 0x94
    Threshold value: 0xae
0x00E0: Event: at: Nov 19 00:36:32 2003; from IPM Controller: 0x20,
LUN: 0, Channel: 0
    "Voltage" (0x02) sensor # 7
    "Threshold" (0x01) event Asserted
    "Lower Critical Going Low"
    Reading value: 0x94
    Threshold value: 0xac
0x00F4: Event: at: Nov 19 00:02:37 2003; from IPM Controller: 0x20,
LUN: 0, Channel: 0
    "Temperature" (0x01) sensor # 2
    "Threshold" (0x01) event Asserted
    "Upper Critical Going High"
    Reading value: 0x1c
    Threshold value: 0x13

#
```

从条目 # 15 (0x0f) 开始获取五个 SEL 条目。

```
# clia sel 20 5 15
IPM Sentry Shelf Manager Command Line Interpreter
0x000F: Event: at Nov 19 16:49:21 2003; from:(0x20,0,0);
sensor:(0xf0,3); event:0x6f(asserted): HotSwap: FRU 2 M2->M3,
Cause=0x1
0x0010: Event: at Nov 19 16:49:22 2003; from:(0x20,0,0);
sensor:(0xf0,2); event:0x6f(asserted): HotSwap: FRU 1 M2->M3,
Cause=0x1
0x0011: Event: at Nov 19 16:49:22 2003; from:(0x20,0,0);
sensor:(0xf0,2); event:0x6f(asserted): HotSwap: FRU 1 M3->M4,
Cause=0x0
0x0012: Event: at Nov 19 16:49:22 2003; from:(0xfc,0,0);
sensor:(0xf0,0); event:0x6f(asserted): HotSwap: FRU 0 M3->M4,
Cause=0x0
0x0013: Event: at Nov 19 16:49:22 2003; from:(0x20,0,0);
sensor:(0xf0,3); event:0x6f(asserted): HotSwap: FRU 2 M3->M4,
Cause=0x0
#
```

清除 SEL。

```
# clia sel clear
IPM Sentry Shelf Manager Command Line Interpreter
SEL clear: issued successfully
      SEL clearing completed
# clia sel
IPM Sentry Shelf Manager Command Line Interpreter
SEL is empty
#
```

sensor

语法:

```
sensor [-v] [IPMB-address [sensor-name | [lun:]sensor-number]]
sensor [-v] board n [sensor-name | [lun:]sensor-number]]
sensor [-v] shm n [sensor-name | [lun:]sensor-number]]
```

用途:

该命令用于显示有关特定传感器的信息。目标传感器按其 IPM 控制器的 IPMB 地址和传感器编号或传感器名称（传感器 SDR 中的设备 ID 字符串，用双引号括住）进行选择。如果既没有指定传感器名称也没有指定传感器编号，则显示有关指定 IPM 控制器上所有传感器的信息。如果没有指定任何参数，则显示有关所有已知传感器的信息。

如果目标控制器支持多个 LUN 上的传感器，则此命令允许用户使用逻辑单元号 (Logical Unit Number, LUN) 来限定传感器编号。如果省略 LUN，则显示有关所有 LUN 上具有指定传感器编号的传感器的信息。*lun* 的取值可以为 0、1 或 3 (LUN 2 为保留值)。

因为假定控制器中传感器名称一般是唯一的，所以不使用 LUN 编号来限定传感器名称。但是，如果控制器中有多个同名传感器，则显示有关所有这些传感器的信息。

在标准模式下，会显示有关各个传感器的以下信息：

- 所属 IPM 控制器的 IPMB 地址
- 可按其访问传感器的传感器编号、传感器名称 (SDR 中的设备 ID 字符串) 和 LUN
- 传感器类型和事件/读取类型代码
- 相关实体的实体 ID 和实体实例 (如果传感器与 FRU 相关联，则为 FRU 设备 ID)

仅在详细模式下显示传感器的以下信息 (有关这些属性的信息，请参阅 IPMI 规范)：

- 确认掩码
- 取消确认掩码
- 传感器状态 (如果是离散传感器) 或阈值 (如果是基于阈值的传感器) 的可设置/可读掩码

在详细模式下仅显示基于阈值的传感器的以下信息：

- 传感器单位：基本单位和修正单位
- 单位百分比、修饰符和比率
- 模拟格式和标志
- 线性参数，M、B、K1、K2 系数
- 公差和精度系数
- 额定值、最大额定值、最小额定值、最大值和最小值
- 阈值上限：非临界、临界和不可恢复
- 阈值下限：非临界、临界和不可恢复
- 滞后值：正数和负数

示例：

获取有关 IPM 控制器 FE 上传感器 FAN 4 的标准信息。

```
# clia sensor fe "FAN 4"
IPM Sentry Shelf Manager Command Line Interpreter
fe: LUN: 0, Sensor # 14 ("FAN 4")
    Type: Threshold (0x01), "Fan" (0x04)
    Belongs to entity: (0xd0, 0) [FRU # 0]
#
```

获取有关 IPM 控制器 9C 上传感器 2 的详细信息。

```
# clia sensor -v 9c 2
IPM Sentry Shelf Manager Command Line Interpreter
9c: LUN: 0, Sensor # 2 ("emulated temp")
    Type: Threshold (0x01), "Temperature" (0x01)
    Belongs to entity: (0xd0, 0) [FRU # 0]
    Assertion Mask: 0x7a95
        Lower Non-Critical Going Low
        Lower Critical Going Low
        Lower Non-Recoverable Going Low
        Upper Non-Critical Going High
        Upper Critical Going High
        Upper Non-Recoverable Going High
        Upper non-critical threshold is comparison returned
        Upper critical threshold is comparison returned
        Upper non-recoverable threshold comparison is returned
    Deassertion Mask: 0x7a95
        Lower Non-Critical Going Low
        Lower Critical Going Low
        Lower Non-Recoverable Going Low
        Upper Non-Critical Going High
        Upper Critical Going High
```



```

Upper Non-Recoverable Going High
Upper non-critical threshold is comparison returned
Upper critical threshold is comparison returned
Upper non-recoverable threshold comparison is returned
Settable / Readable Mask: 0x3f3f
Lower Non-Critical Threshold is Readable
Lower Critical Threshold is Readable
Lower Non-Recoverable Threshold is Readable
Upper Non-Critical Threshold is Readable
Upper Critical Threshold is Readable
Upper Non-Recoverable Threshold is Readable
Lower Non-Critical Threshold is Settable
Lower Critical Threshold is Settable
Lower Non-Recoverable Threshold is Settable
Upper Non-Critical Threshold is Settable
Upper Critical Threshold is Settable
Upper Non-Recoverable Threshold is Settable
Unit Percentage: OFF (0), Unit Modifier: none (0), Unit Rate:
none (0)
Analog Format: 2's complement (signed) (2)
Base Unit: degrees C (1), Modifier Unit: unspecified (0)
Linearization: linear (0), M = 1, B = 0, K1 = 0, K2 = 0
Tolerance = 0, Accuracy = 0, Accuracy EXP = 0
Analog Flags: 0x0
Nominal: 0 (0x00), Normal max: 0 (0x00), Normal min: 0 (0x00)
Sensor max: 127 (0x7f), Sensor min: 128 (0x80)
Upper Thresholds:
Non-Critical: 70 (0x46) Critical: 80 (0x50) Non-Recoverable:
90 (0x5a)
Lower Thresholds:
Non-Critical: 3 (0x03) Critical: 0 (0x00) Non-Recoverable:
251 (0xfb)
Hysteresis:
Positive: 2 (0x02), Negative 2 (0x02)
#

```

同上，但明确指定传感器的 LUN。

```
# clia sensor -v 9c 0:2
IPM Sentry Shelf Manager Command Line Interpreter
9c: LUN: 0, Sensor # 2 ("emulated temp")
Type: Threshold (0x01), "Temperature" (0x01)
Belongs to entity: (0xd0, 0) [FRU # 0]
Assertion Mask: 0x7a95
    Lower Non-Critical Going Low
    Lower Critical Going Low
    Lower Non-Recoverable Going Low
    Upper Non-Critical Going High
    Upper Critical Going High
    Upper Non-Recoverable Going High
    Upper non-critical threshold is comparison returned
    Upper critical threshold is comparison returned
    Upper non-recoverable threshold comparison is returned
Deassertion Mask: 0x7a95
    Lower Non-Critical Going Low
    Lower Critical Going Low
    Lower Non-Recoverable Going Low
    Upper Non-Critical Going High
    Upper Critical Going High
    Upper Non-Recoverable Going High
    Upper non-critical threshold is comparison returned
    Upper critical threshold is comparison returned
    Upper non-recoverable threshold comparison is returned
Settable / Readable Mask: 0x3f3f
    Lower Non-Critical Threshold is Readable
    Lower Critical Threshold is Readable
    Lower Non-Recoverable Threshold is Readable
    Upper Non-Critical Threshold is Readable
    Upper Critical Threshold is Readable
    Upper Non-Recoverable Threshold is Readable
    Lower Non-Critical Threshold is Settable
    Lower Critical Threshold is Settable
    Lower Non-Recoverable Threshold is Settable
    Upper Non-Critical Threshold is Settable
    Upper Critical Threshold is Settable
    Upper Non-Recoverable Threshold is Settable
Unit Percentage: OFF (0), Unit Modifier: none (0), Unit Rate:
none (0)
Analog Format: 2's complement (signed) (2)
Base Unit: degrees C (1), Modifier Unit: unspecified (0)
Linearization: linear (0), M = 1, B = 0, K1 = 0, K2 = 0
```

```
Tolerance = 0, Accuracy = 0, Accuracy EXP = 0
Analog Flags: 0x0
Nominal: 0 (0x00), Normal max: 0 (0x00), Normal min: 0 (0x00)
Sensor max: 127 (0x7f), Sensor min: 128 (0x80)
Upper Thresholds:
    Non-Critical: 70 (0x46) Critical: 80 (0x50) Non-Recoverable:
90 (0x5a)
Lower Thresholds:
    Non-Critical: 3 (0x03) Critical: 0 (0x00) Non-Recoverable:
251 (0xfb)
Hysteresis:
    Positive: 2 (0x02), Negative 2 (0x02)
#
```

sensordata

语法:

```
sensordata [IPMB-address [sensor-name | [lun:]sensor-number]]
sensordata [-v] board n [sensor-name | [lun:]sensor-number]]
sensordata [-v] shm n [sensor-name | [lun:]sensor-number]]
```

用途:

该命令用于显示指定传感器的实际值（对于基于阈值的传感器）或当前确认状态（对于离散传感器）。目标传感器按其 IPM 控制器的 IPMB 地址和传感器编号或传感器名称（传感器 SDR 中的设备 ID 字符串，用双引号括住）进行选择。如果既没有指定传感器名称也没有指定传感器编号，则显示指定 IPM 控制器上所有传感器的值。如果没有指定任何参数，则显示所有已知传感器的值。

如果目标控制器支持多个 LUN 上的传感器，则此命令允许用户使用逻辑单元号 (Logical Unit Number, LUN) 来限定传感器编号。如果省略 LUN，则显示有关所有 LUN 上具有指定传感器编号的传感器的信息。*lun* 的取值可以为 0、1 或 3（LUN 2 为保留值）。

因为假定控制器中传感器名称一般是唯一的，所以不使用 LUN 编号来限定传感器名称。但是，如果控制器中有多个同名传感器，则显示有关所有这些传感器的信息。

将显示各个传感器的以下信息：

- 所属 IPM 控制器的 IPMB 地址
- 可按其访问传感器的传感器编号、传感器名称（SDR 中的设备 ID 字符串）和 LUN
- 传感器类型和事件/读取类型代码
- 原始格式的传感器值（对于基于阈值的传感器）或当前确认状态的掩码（对于离散传感器）
- 越限状态（十六进制解码格式）

值/确认状态均以原始格式和处理过的格式显示。采用处理过的格式时，模拟值按照 M、B 和 R 进行转换，并与单位名称一起显示（例如，27 度）。按照事件/读取代码类型对离散值进行注释（例如，对于事件/读取代码 2，确认状态 0 显示为 Transition to Idle）。

示例：

获取 IPM 控制器 FE 上温度传感器 Local Temp 的传感器数据值。

```
# clia sensordata FE "Local Temp"
IPM Sentry Shelf Manager Command Line Interpreter
fe: LUN: 0, Sensor # 3 ("Local Temp")
   Type: Threshold (0x01), "Temperature" (0x01)
   Status: 0xc0
       All event messages enabled from this sensor
       Sensor scanning enabled
       Initial update completed
   Raw data: 22 (0x16)
   Processed data: 22.000000 degrees C
   Status: 0x00
```

获取 IPM 控制器 9C 上离散（热交换）传感器 (#0) 的传感器数据值。

```
# clia sensordata 9c 0
IPM Sentry Shelf Manager Command Line Interpreter
9c: LUN: 0, Sensor # 0 ("FRU 0 HOT_SWAP")
   Type: Discrete (0x6f), "Hot Swap" (0xf0)
   Status: 0xc0
       All event messages enabled from this sensor
       Sensor scanning enabled
       Initial update completed
   Sensor reading: 0x00
   Current State Mask 0x0010
```

获取相同传感器的传感器数据值，但使用 LUN 明确限定该传感器。

```
# clia sensordata 9c 0:0
IPM Sentry Shelf Manager Command Line Interpreter
9c: LUN: 0, Sensor # 0 ("FRU 0 HOT_SWAP")
Type: Discrete (0x6f), "Hot Swap" (0xf0)
Status: 0xc0
    All event messages enabled from this sensor
    Sensor scanning enabled
    Initial update completed
Sensor reading: 0x00
Current State Mask 0x0010
```

sensorread

语法:

```
sensorread IPMB-address [lun:]sensor-number
```

用途:

该命令用于显示指定传感器的原始值。sensorread 命令和 sensordata 命令的唯一区别在于: sensorread 命令不检查目标 IPM 控制器是否存在或传感器编号是否有效, 而只是通过 IPMB 直接发送 Get Sensor Reading 请求。由于该命令不检索传感器的 SDR, 因此它不能处理获得的数据。

如果目标控制器支持多个 LUN 上的传感器, 则此命令允许用户使用逻辑单元号 (Logical Unit Number, LUN) 来限定传感器编号。如果省略 LUN, 则使用 LUN 0。lun 的取值可以为 0、1 或 3。(LUN 2 为保留值。)

将显示各个传感器的以下信息:

- 所属 IPM 控制器的 IPMB 地址
- 可按其访问传感器的传感器编号、传感器名称 (SDR 中的设备 ID 字符串) 和 LUN
- 传感器类型和事件/读取类型代码
- 原始格式的传感器值 (对于基于阈值的传感器) 或当前确认状态的掩码 (对于离散传感器)

示例:

获取 IPM 控制器 FC 上传感器 4 的传感器数据值。请注意，`sensorread` 命令仅提供未处理过的传感器值。另请注意，该命令示例中，明确指定了 LUN。

```
# clia sensordata fc 4
IPM Sentry Shelf Manager Command Line Interpreter
fc: LUN: 0, Sensor # 4 ("3.3STBY voltage")
  Type: Threshold (0x01), "Voltage" (0x02)
  Status: 0xc0
    All event messages enabled from this sensor
    Sensor scanning enabled
    Initial update completed
  Raw data: 193 (0xc1)
  Processed data: 3.396800 Volts
  Status: 0x00

# clia sensorread fc 4
IPM Sentry Shelf Manager Command Line Interpreter
fc: LUN: 0, Sensor # 4
  Raw data: 193 (0xc1)
  Status: 0xc0
    All event messages enabled from this sensor
    Sensor scanning enabled
    Initial update completed
  Threshold Sensor Status: 0x00
  Discrete Sensor Current State Mask 0x0000

# clia sensorread fc 0:4
IPM Sentry Shelf Manager Command Line Interpreter
fc: LUN: 0, Sensor # 4
  Raw data: 193 (0xc1)
  Status: 0xc0
    All event messages enabled from this sensor
    Sensor scanning enabled
    Initial update completed
  Threshold Sensor Status: 0x00
  Discrete Sensor Current State Mask 0x0000

#
```

session

语法:

```
session
```

用途:

该命令用于显示有关活动 RMCP 会话的信息。信息中包括以下各项:

- 可能的最大会话数和当前活动会话数
- 对于每个当前活动会话:
 - 会话句柄
 - 会话激活期间使用的用户 ID 和用户名
 - 最高的会话权限级别
 - IPMI 通道号和类型
 - 对于 LAN 会话, 则为对等 IP 地址和端口号。

示例:

```
# clia session
IPM Sentry Shelf Manager Command Line Interpreter
32 sessions possible, 2 sessions currently active
Session: 1
  User: ID 1, Name: ""; Privilege Level: "Administrator"
  Channel: 1 ("LAN_802_3"); Peer IP address: 172.16.2.203, Port:
1764
Session: 2
  User: ID 1, Name: ""; Privilege Level: "Administrator"
  Channel: 1 ("LAN_802_3"); Peer IP address: 172.16.2.203, Port:
1765
#
```

setextracted

语法:

```
setextracted IPMB-address fru-id
```

用途:

该命令用于向机框管理器通知已从机框中实际取出指定 FRU。如果指定的 FRU 处于状态 M7，则机框管理器会将其置于状态 M0（FRU 实际上不存在）。

示例:

```
# clia setextracted 9c 0
IPM Sentry Shelf Manager Command Line Interpreter
Set FRU extracted state successfully
#
```

setfanlevel

语法:

```
setfanlevel IPMB-address fru-id level
setfanlevel fan_tray n level
setfanlevel all level
```

用途:

该命令用于为由命令参数中指定的 FRU 所控制的风扇设置新级别。最小值为 1，最大值为 15。

如果在该命令中使用 `all` 限定符，则该命令会尝试为机框中所有已知风扇设置相同级别。

示例:

将位于 IPMB 地址为 0x20 的 FRU #2 上的风扇的风扇级别设置为 5。

```
# clia setfanlevel 20 2 5

IPM Sentry Shelf Manager Command Line Interpreter

20: FRU # 2 Set Fan Level to: 5
#
```

将机框中所有已知风扇的风扇级别设置为 4:

```
# clia setfanlevel all 4

IPM Sentry Shelf Manager Command Line Interpreter

72: FRU # 0 Set Fan Level to: 4
76: FRU # 0 Set Fan Level to: 4
#
```

setfruledstate

语法:

```
setfruledstate IPMB-address fru-id LedId [ALL LedOp [LedColor]]

LedOp = ON | OFF | LOCAL | BLINK <onTime> <offTime> | TEST <onTime>
LedColor = BLUE | RED | GREEN | AMBER | ORANGE | WHITE | NONE | number
```

用途:

用户可以使用该命令设置给定 FRU 的特定 LED 指示灯的状态或所有 LED 指示灯的状态。

第一个参数 *IPMB-address* 是 IPM 控制器的 IPMB 地址。第二个参数 *fru-id* 是 FRU 设备 ID。第三个参数可以是 LED ID (一个数值), 也可以是 ALL。在后一种情况下, 指定的操作应用于所有 LED 指示灯。

参数 *LedOp* 用于根据 PICMG 3.0 规范指定应用于 FRU 的操作。操作定义如下:

- ON - 开启 LED 指示灯
- OFF - 关闭 LED 指示灯
- LOCAL - 恢复为本地控制 LED 指示灯

- BLINK - 反复地使 LED 指示灯开启 *onTime* 毫秒，然后再关闭 *offTime* 毫秒，使 LED 指示灯呈闪烁状态
- TEST - 运行指示灯测试（持续 *onTime* 毫秒）

对于 TEST 操作，*onTime* 必须小于 12800 毫秒（12.8 秒）；对于 BLINK 操作，*onTime* 和 *offTime* 值必须介于 10-2500 毫秒之间。

可选参数 *LedColor* 用于通过符号名称或十进制值指定颜色。颜色的符号名称与十进制值的对应关系遵循 PICMG 3.0 规范，如下所列。（如果没有指定参数，则使用默认 LED 指示灯颜色。）

- BLUE = 1
- RED = 2
- GREEN = 3
- AMBER = 4
- ORANGE = 6
- NONE = 14（不改变颜色）

示例：

关闭 IPMB 地址为 20h 的 IPM 控制器的 FRU #0 的 LED #1。

```
# clia setfruledstate 20 0 1 OFF
IPM Sentry Shelf Manager Command Line Interpreter

    Setting FRU's led state completed successfully, status = 0x0
#
```

实现对 IPMB 地址为 20h 的 IPM 控制器的 FRU #0 的 LED #1 进行本地控制。

```
# clia setfruledstate 20 0 1 LOCAL
IPM Sentry Shelf Manager Command Line Interpreter

    Setting FRU's led state completed successfully, status = 0x0
#
```

使 IPMB 地址为 20h 的 IPM 控制器的 FRU #0 的 LED #1 闪烁。闪烁时的颜色为默认颜色。亮起的持续时间为 100 毫秒，熄灭的持续时间为 200 毫秒。

```
# clia setfruledstate 20 0 0 BLINK 100 200
IPM Sentry Shelf Manager Command Line Interpreter

    Setting FRU's led state completed successfully, status = 0x0
#
```

sethysteresis

语法:

```
sethysteresis IPMB-address [lun:]sensor_id | sensor_name pos | neg [-r] value
```

用途:

该命令用于设置指定传感器的指定滞后值。传感器必须是基于阈值的传感器。它必须支持指定的阈值滞后值，且该滞后值必须是可设置的。

如果目标控制器支持多个 LUN 上的传感器，则此命令允许用户使用逻辑单元号 (Logical Unit Number, LUN) 来限定传感器编号。在该命令中，如果使用 `pos` 参数，则设置正的滞后值，如果使用 `neg` 参数，则设置负的滞后值。

示例:

针对 IPMB 地址为 0xFC 的 IPM 控制器的传感器 #2 设置正滞后值。

```
# clia sethysteresis FC 2 pos 10
IPM Sentry Shelf Manager Command Line Interpreter

    Positive hysteresis set successfully to 0xA, previous: 0x0
#
```

setipmbstate

语法:

```
setipmbstate IPMB-address A|B [link] 1|0 (在辐射型 IPMB-0 环境中)
setipmbstate IPMB-address A|B 1|0 (在总线型 IPMB-0 环境中)
```

用途:

该命令用于启用或禁用目标 IPM 控制器上的 IPMB 链路。第二个参数定义要启用或禁用的总线 (IPMB-A 或 IPMB-B)。最后一个参数定义要执行的操作: 1 - 启用链路, 0 - 禁用链路。

在总线型环境和辐射型环境中, 该命令的用法有所不同。在总线型环境中, 以及在目标 IPM 控制器不为 IPMB 集线器的辐射型环境中, 不使用参数 `link`。对于辐射型环境中的 IPMB 集线器控制器, 参数 `link` 是可选的。

如果使用 *link*，该命令将启用或禁用特定辐射型 IPMB 链路（1 到 95）。如果省略 *link*，则该命令将启用或禁用辐射型系统中 IPMB 集线器上的所有链路。

示例：

禁用 IPMB 地址为 92h 的 IPM 控制器上的 IPMB-A 链路。

```
# clia setipmbstate 92 A 0
IPM Sentry Shelf Manager Command Line Interpreter

Command executed successfully
```

启用机框管理器上的辐射型 IPMB 链路 3、总线 B（目标 IPM 控制器是 IPMB 集线器）。

```
# clia setipmbstate 20 B 3 1
IPM Sentry Shelf Manager Command Line Interpreter

Command executed successfully
```

setlanconfig

语法：

```
setlanconfig channel parameter-name additional-parameters
setlanconfig channel parameter-number additional-parameters
```

用途：

该命令用于设置指定通道上指定 LAN 配置参数的值。应该明确指定通道号、配置参数名称或编号以及参数值。

表 A-3 列出了 `setlanconfig` 命令支持的 LAN 配置参数的名称和编号：

表 A-3 `setlanconfig` 的 LAN 配置参数

参数名称	编号	说明
auth_enables	2	五个 8 位值，包含 LAN 通道的回调、用户、操作员、管理员和 OEM 权限级别的验证类型启用标志。
ip	3	一个字符串值，包含指定给 LAN 通道的 IP 地址，该地址以点分十进制记法表示。
subnet_mask	6	一个字符串值，包含指定给 LAN 通道的子网掩码，该子网掩码以点分十进制记法表示。

表 A-3 setlanconfig 的 LAN 配置参数 (续)

ipv4_hdr_param	7	三个 8 位值, 包含各种用于发送 RMCP 包的 IPv4 标头参数: <ul style="list-style-type: none"> • 生存时间 • IP 标头标志 (位 [7:5]) • 优先级 (位 [7:5]) 和服务类型 (位 [4:1])
arp_control	10	两个用于控制 LAN 通道上 ARP 行为的标志: <ul style="list-style-type: none"> • 启用响应 ARP 请求 • 启用发送无故 ARP (Gratuitous ARP)
arp_interval	11	以定点格式表示的无故 ARP (Gratuitous ARP) 时间间隔 (其中整数部分表示秒数, 小数部分表示毫秒数)。
dft_gw_ip	12	一个字符串值, 包含以点分十进制记法表示的默认网关的 IP 地址。
backup_gw_ip	14	一个字符串值, 包含以点分十进制记法表示的备用网关的 IP 地址。
community	16	一个字符串值 (最多 18 个符号), 此值置于 PET 陷阱的 Community String 字段中。
destination_type	18	由指定的已设置选择器标识的目标类型。必须为该参数指定已设置的选择器。每个目标类型条目包含以下字段: <ul style="list-style-type: none"> • 目标类型 (0-7) • 警报确认标志 • 警报确认超时/重试时间间隔 (1-256, 以秒为单位) • 重试次数 (0-7)
destination_address	19	与指定的已设置选择器关联的目标地址。必须为该参数指定已设置的选择器。每个目标地址条目包含以下字段: <ul style="list-style-type: none"> • 网关选择器: 0 - 使用默认网关, 1 - 使用备用网关 • IP 地址 (以点分十进制格式表示的字符串) • MAC 地址 (由六个以冒号 [:] 分隔的十六进制字节值组成的字符串)

auth_enables

语法:

```
setlanconfig channel auth_enables value1 value2 value3 value4 value5
setlanconfig channel 2 value1 value2 value3 value4 value5
```

用途:

该命令用于设置 LAN 参数 `auth_enables` 的当前值。该参数指定机框管理器当前针对五种支持的权限级别（回调、用户、管理员、操作员和 OEM）中的每一种启用的验证类型，该参数以一个由五个字节组成的序列表示（每个字节对应各自的权限级别），被视为位掩码，其中位的定义如下：

- 0x01 无
- 0x02 MD2
- 0x04 MD
- 5 0x10 直接密码/密钥
- 0x20 OEM 专用

参数 `value1` 到 `value5` 应该代表以十六进制表示的这些字节的值。机框管理器当前不支持“回调”和“OEM”权限级别。因此，与这些权限级别对应的 `value1` 和 `value5` 参数应指定为 0。

示例:

```
# clia setlanconfig 1 auth_enables 0 1 1 1 0
IPM Sentry Shelf Manager Command Line Interpreter
Authentication Type Enables set successfully
#
```

ip

语法:

```
setlanconfig channel ip value
setlanconfig channel 3 value
```

用途:

该命令用于设置通道当前使用的 IP 地址。该值应该代表以点分十进制记法表示的 IP 地址。

示例:

```
# clia setlanconfig 1 ip 172.16.2.203
IPM Sentry Shelf Manager Command Line Interpreter
IP Address set successfully
#
```

subnet_mask

语法:

```
setlanconfig channel subnet_mask value
setlanconfig channel 6 value
```

用途:

该命令用于设置通道当前使用的 IP 子网掩码。该值应该代表以点分十进制记法表示的 IP 子网掩码。

示例:

```
# clia setlanconfig 1 subnet_mask 255.255.255.0
IPM Sentry Shelf Manager Command Line Interpreter
Subnet Mask set successfully
#
```

ipv4_hdr_param

语法:

```
setlanconfig channel ipv4_hdr_param value1 value2 value3
setlanconfig channel 7 value1 value2 value3
```

用途:

该命令用于设置机框管理器的 IP 4 标头参数。这些参数表示为三个以十六进制记数法表示的单字节值: *value1*、*value2* 和 *value3*。这些字节的内容符合 IPMI 1.5 规范的第 19.2 节, 并包括以下属性:

- 第 1 个字节中的生存时间
- 第 2 个字节中的 IP 标头标志 (位 [7:5])
- 第 3 个字节中的优先级 (位 [7:5]) 和服务类型 (位 [4:1])

示例:

```
# clia setlanconfig 1 ipv4_hdr_param 37 E0 11
IPM Sentry Shelf Manager Command Line Interpreter
IPv4 Header Parameters set successfully
#
```

arp_control

语法:

```
setlanconfig channel arp_control value
setlanconfig channel 10 value
```

用途:

该命令用于设置 LAN 参数 `arp_control` 的当前值。该参数指定机框管理器提供的附加 ARP 支持，它以一个单字节表示，被视作位掩码，其中位的定义如下：

- 0x01 启用机框管理器生成的无故 ARP (Gratuitous ARP)
- 0x02 启用机框管理器生成的 ARP 响应

其他位为保留位，应设置为 0。

示例:

```
# clia setlanconfig 1 arp_control 3
IPM Sentry Shelf Manager Command Line Interpreter
BMC-generated ARP control set successfully
#
```

arp_interval

语法:

```
setlanconfig channel arp_interval value
setlanconfig channel 11 value
```

用途:

该命令用于设置通道当前使用的 ARP 时间间隔。该值应该代表以定点数字格式表示的秒数/毫秒数（可能带有小数部分）。根据 IPMI 中关于该参数的定义，它会被截断为能被 500 毫秒除尽的最大时间间隔。

示例:

```
# clia setlanconfig 1 arp_interval 3.5
IPM Sentry Shelf Manager Command Line Interpreter
Gratuitous ARP interval set successfully
#
```

dft_gw_ip

语法:

```
setlanconfig channel dft_gw_ip value
setlanconfig channel 12 value
```

用途:

该命令用于设置通道使用的默认网关的 IP 地址。该值应该代表以点分十进制记法表示的 IP 地址。

示例:

```
# clia setlanconfig 1 dft_gw_ip 172.16.2.100
IPM Sentry Shelf Manager Command Line Interpreter
Default Gateway Address set successfully
#
```

backup_gw_ip

语法:

```
setlanconfig channel backup_gw_ip value
setlanconfig channel 14 value
```

用途:

该命令用于设置通道使用的备用网关的 IP 地址。该值应该代表以点分十进制记法表示的 IP 地址。

示例:

```
# clia setlanconfig 1 backup_gw_ip 172.16.2.100
IPM Sentry Shelf Manager Command Line Interpreter
Backup Gateway Address set successfully
#
```

community

语法:

```
setlanconfig channel community value  
setlanconfig channel 16 value
```

用途:

该命令用于设置 PET 陷阱中使用的团体字符串参数。该值应该用双引号括住的字符串。

示例:

```
# clia setlanconfig 1 community "Community"  
IPM Sentry Shelf Manager Command Line Interpreter  
Community string set successfully  
#
```

destination_type

语法:

```
setlanconfig channel destination_type set-selector value1 value2 value3  
setlanconfig channel 18 set-selector value1 value2 value3
```

用途:

该命令用于设置目标表中索引等于 *set-selector* 的元素。索引是从 0 开始的。选择器 0 用于对易失性目标进行寻址。值 *value1*、*value2* 和 *value3* 依据 IPMI 规范的第 19.2 节提供有关新目标的信息。提供的信息如下:

- 警报目标类型 (PET 陷阱或 OEM 目标; 是否应对该警报进行确认)
- 警报确认超时
- 重试计数

示例:

```
# clia setlanconfig 1 destination_type 2 80 3 5  
IPM Sentry Shelf Manager Command Line Interpreter  
Destination Type set successfully  
#
```

destination_address

语法:

```
setlanconfig channel destination_address set-selector gateway-sel  
IP-address MAC-address  
setlanconfig channel 19 set-selector gateway-sel IP-address MAC-address
```

用途:

该命令用于设置目标地址表中索引等于 *set-selector* 的元素。索引是从 0 开始的。选择器 0 用于对易失性目标进行寻址。该命令参数提供以下必要的信息:

- *gateway-sel* - 要使用的网关: 0 表示使用默认网关, 1 表示使用备用网关
- *IP-address* - 以点分十进制记法表示的目标 IP 地址
- *MAC-address* - 以冒号分隔的六个十六进制字节表示的目标 MAC 地址

示例:

```
# clia setlanconfig 1 destination_address 2 0 172.16.2.100  
90:93:93:93:93:93  
IPM Sentry Shelf Manager Command Line Interpreter  
Destination Addresses set successfully  
#
```

setlocked

语法:

```
setlocked IPMB-address fru-id 0 |  
setlocked IPMB-address fru-id 1  
setlocked board n 0 | 1  
setlocked shm n 0 | 1  
setlocked fan_tray n 0 | 1
```

用途:

该命令用于将指定 FRU 的锁定位设置为指定状态（0 表示解除锁定，1 表示锁定）。FRU 通过使用所属 IPM 控制器的 IPMB 地址以及 FRU 设备 ID 进行指定。FRU 设备 ID 0 指定适合 PICMG 3.0 环境的 IPM 控制器。

根据 PICMG 3.0 规范，锁定位控制是否允许 FRU 自发地从状态 M1（非活动）转为状态 M2（激活请求）。如果设置了锁定位，则不允许进行此转换。机框管理器向 FRU 发送 Deactivate 命令后，FRU 会转为状态 M1，并设置锁定位，从而阻止后续状态转换。

该命令可用于重新激活以前手动取消激活的 FRU，方法是清除它的锁定位。

示例:

清除地址为 9C 的 IPM 控制器的锁定位，从而允许其重新激活。

```
# clia setlocked 9c 0 0
IPM Sentry Shelf Manager Command Line Interpreter
Lock set successfully to 0x0
#
```

setpefconfig

语法:

```
setpefconfig parameter-name additional-parameters
setpefconfig parameter-number additional-parameters
```

用途:

该命令用于为指定的 PEF 配置参数设置新值。表 A-4 列出了可通过该命令设置的 PEF 配置参数的名称和编号。

表 A-4 setpefconf 的 PEF 配置参数

参数名称	编号	说明
control	1	一个 8 位值，代表 PEF 的控制标志（启用 PEF、启用 PEF 启动延迟等）。
action_control	2	一个 8 位值，代表 PEF 的操作全局控制标志（启用复位、启用关闭电源等）。
startup_delay	3	系统加电并复位后延迟 PEF 的时间（以秒为单位）。
alert_startup_delay	4	系统加电并复位后延迟警报的时间（以秒为单位）。

表 A-4 setpefconf 的 PEF 配置参数 (续)

event_filter	6	<p>由指定的已设置选择器标识的事件过滤器表项。由以下 19 个十六进制的数值组成, 这些数值按照 IPMI 规范 1.5 版的表 15-2 中的定义进行编码:</p> <ul style="list-style-type: none"> • 过滤器配置 • 事件过滤器操作 • 警报策略编号 • 事件严重程度 • 生成器 ID 第 1 个字节 • 生成器 ID 第 2 个字节 • 传感器类型 • 传感器编号 • 事件触发器 (事件/读取类型) • 事件数据 1 事件偏移掩码 • 事件数据 1 AND 掩码 • 事件数据 1 比较 1 • 事件数据 1 比较 2 • 事件数据 2 AND 掩码 • 事件数据 2 比较 1 • 事件数据 2 比较 2 • 事件数据 3 AND 掩码 • 事件数据 3 比较 1 • 事件数据 3 比较 2
event_filter_data1	7	由指定的已设置选择器标识的事件过滤器表项的第一个字节。
alert_policy	9	<p>由指定的已设置选择器标识的警报策略表项。由以下五个十六进制的数值组成, 这些数值按照 IPMI 1.5 的表 15-4 中的定义进行编码:</p> <ul style="list-style-type: none"> • 策略编号 (4 位值) • 策略 (4 位值); 包括启用/禁用位 • 通道号 (4 位值) • 目标选择器 (4 位值) • 警报字符串集/选择器
system-guid	10	用于填充 PET 陷阱中 GUID 字段的 GUID。
alert_string_key	12	由指定的已设置选择器标识的警报字符串键值。由两个 8 位值组成: 事件过滤器编号和警报字符串集。
alert_string	13	由指定的已设置选择器标识的警报字符串。
oem_filter	97	<p>由指定的已设置选择器标识的 OEM 过滤器表项。该项由以下三个数值组成:</p> <ul style="list-style-type: none"> • 第 1 个字节: SEL 记录类型范围的下限 • 第 2 个字节: SEL 记录类型范围的上限 • 第 3 个字节: 针对其记录类型符合上述范围的 SEL 项所要调用的警报策略编号。

control

语法:

```
setpefconfig control value  
setpefconfig 1 value
```

用途:

该命令用于为 PEF 参数 `control` 设置新值。该参数为一个单字节，被视作位掩码，其中位的定义如下：

- 0x01 - 启用 PEF
- 0x02 - 启用为 PEF 操作生成事件消息
- 0x04 - 系统加电并复位时启用 PEF 启动延迟
- 0x08 - 启用 PEF 警报启动延迟

其他位为保留位，应设置为 0。应该按十六进制格式输入该值。

示例:

```
# clia setpefconfig control 7  
IPM Sentry Shelf Manager Command Line Interpreter  
PEF control set successfully  
#
```

action_control

语法:

```
setpefconfig action_control value  
setpefconfig 2 value
```

用途:

该命令用于为 PEF 参数 `action_control` 设置新值。该参数为一个单字节，被视作位掩码，其中位的定义如下：

- 0x01 - 启用警报操作
- 0x02 - 启用关闭电源操作
- 0x04 - 启用复位操作
- 0x08 - 启用关闭再打开电源操作
- 0x01 - 启用 OEM 操作
- 0x20 - 启用诊断中断

其他位为保留位，应设置为 0。应该按十六进制格式输入该值。

示例:

```
# clia setpefconfig action_control 3f
IPM Sentry Shelf Manager Command Line Interpreter
PEF action control set successfully
#
```

startup_delay

语法:

```
setpefconfig startup_delay value
setpefconfig 3 value
```

用途:

该命令用于为 PEF 参数 `startup_delay` 设置新值。该参数为一个单字节，表示 PEF 设备在启动时延迟的秒数。该秒数值以十进制数指定。

示例:

```
# clia setpefconfig startup_delay 45
IPM Sentry Shelf Manager Command Line Interpreter
PEF startup delay set successfully
#
```

alert_startup_delay

语法:

```
setpefconfig alert_startup_delay value
setpefconfig 4 value
```

用途:

该命令用于设置 PEF 参数 `alert_startup_delay` 的当前值。该参数为一个单字节，表示警报设备在启动时延迟的秒数。该秒数值以十进制数指定。

示例:

```
# clia setpefconfig alert_startup_delay 45
IPM Sentry Shelf Manager Command Line Interpreter
Alert startup delay set successfully
#
```

event_filter

语法:

```
setpefconfig event_filter set-selector value1 ... <value19>  
setpefconfig 6 set-selector value1 ... <value19>
```

用途:

该命令用于设置事件过滤器表中索引等于 *set-selector* 的元素。索引是从 1 开始的。新元素的内容通过 19 个十六进制数值（从 *value1* 到 *<value19>*）指定，这些数值按照 IPMI 规范 1.5 版的表 15-2 中的定义进行编码：

- 过滤器配置
- 事件过滤器操作
- 警报策略编号
- 事件严重程度
- 生成器 ID 第 1 个字节
- 生成器 ID 第 2 个字节
- 传感器类型
- 传感器编号
- 事件触发器（事件/读取类型）
- 事件数据 1 事件偏移掩码
- 事件数据 1 AND 掩码
- 事件数据 1 比较 1
- 事件数据 1 比较 2
- 事件数据 2 AND 掩码
- 事件数据 2 比较 1
- 事件数据 2 比较 2
- 事件数据 3 AND 掩码
- 事件数据 3 比较 1
- 事件数据 3 比较 2

示例:

设置事件过滤器 2 以在地址为 9C 的 IPM 控制器 FRU 0 到达状态 M0 时触发警报操作（将按照警报策略 #1 发送警报）：

```
# clia setpefconfig event_filter 2 80 1 1 10 9C FF F0 FF FF FF FF  
0F FF 0 0 0 0 FF FF 0  
IPM Sentry Shelf Manager Command Line Interpreter  
Event filter set successfully  
#
```


event_filter_data1

语法:

```
setpefconfig event_filter_data1 set-selector value
setpefconfig 7 set-selector value
```

用途:

该命令用于设置事件过滤器表中索引等于 *set-selector* 的元素的第一个字节。索引是从 1 开始的。应该按十六进制格式指定该字节。该字节中位的含义如下:

- 0x80 - 已启用该过滤器
- 0x40 - 该过滤器由制造商进行预配置，不应通过软件进行更改

其他位为保留位，应为 0。

可以使用该命令在事件过滤器的启用/禁用状态之间快速切换；即，在不重写整个表项的情况下打开和关闭事件过滤器。

示例:

打开事件过滤器 2。

```
# clia setpefconfig event_filter_data1 2 80
IPM Sentry Shelf Manager Command Line Interpreter
Event filter data1 set successfully
#
```

关闭事件过滤器 2。

```
# clia setpefconfig event_filter_data1 2 0
IPM Sentry Shelf Manager Command Line Interpreter
Event filter data1 set successfully
#
```

alert_policy

语法:

```
setpefconfig alert_policy set-selector value1 value2 value3 value4 value5  
setpefconfig 9 set-selector value1 value2 value3 value4 value5
```

用途:

该命令用于设置由指定的已设置选择器标识的警报策略表项。新元素的内容通过以下五个十六进制数值（从 *value1* 到 *value5*）指定，这些数值按照 IPMI 1.5 的表 15-4 中的定义进行编码：

- 策略编号（4 位值）
- 策略（4 位值）；包括启用/禁用位
- 通道号（4 位值）
- 目标选择器（4 位值）
- 警报字符串集/选择器

示例:

下面的示例使用以下属性设置警报策略表项 2:

- 策略编号 = 5
- 已启用
- 策略 = 始终向该目标发送警报
- 目标通道 = 1
- 目标选择器 = 1
- 警报字符串选择器 = 对所有事件使用字符串 1

```
# clia setpefconfig alert_policy 2 5 8 1 1 1  
IPM Sentry Shelf Manager Command Line Interpreter  
Policy set successfully  
#
```

system_guid

语法:

```
setpefconfig system_guid guid-value
setpefconfig 10 guid-value
setpefconfig system_guid none
setpefconfig 10 none
```

用途:

该命令用于设置 PEF 参数 `system_guid` 的当前值。该参数表示通过 PET 陷阱 PDU 发送到警报目标的 GUID。可将该 GUID 定义为一个独立的 GUID 或定义为等同于系统 GUID。

可将 `guid-value` 指定为一个符合 GUID 标准格式 `xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx` 的实际 GUID，也可将其指定为符号值 `none`。在第一种情况下，PEF 设备使用 PET 陷阱中指定的 GUID。在第二种情况下，PEF 设备默认使用 PET 陷阱的系统 GUID（执行 IPMI Get System GUID 命令所得结果）。

示例:

```
# clia setpefconfig system_guid 23662F7F-BA1B-4b65-8808-94CA09C9BBBB0
IPM Sentry Shelf Manager Command Line Interpreter
GUID set successfully
#
# clia setpefconfig system_guid none
IPM Sentry Shelf Manager Command Line Interpreter
Using the system GUID
#
```

alert_string_key

语法:

```
setpefconfig alert_string_key set-selector value1 value2  
setpefconfig 12 set-selector value1 value2
```

用途:

该命令用于设置警报字符串键值表中索引等于 *set-selector* 的元素。索引是从 1 开始的。可以使用 *Set-selector 0* 来指定易失性警报字符串。为了生成警报，每个键值会将事件过滤器与警报字符串相关联。每个键值由事件过滤器编号和警报字符串编号组成。这两个值都是 8 位值，分别由参数 *value1* 和 *value2* 以十六进制格式指定。

示例:

```
# clia setpefconfig alert_string_key 2 10 11  
IPM Sentry Shelf Manager Command Line Interpreter  
Alert string keys set successfully  
#
```

alert_string

语法:

```
setpefconfig alert_string set-selector <string-value>  
setpefconfig 13 set-selector <string-value>
```

用途:

该命令用于设置警报字符串表中索引等于 *set-selector* 的元素。索引是从 1 开始的。可以使用索引 0 来指定易失性警报字符串。字符串值应该用双引号 (") 括住，引号中可以包括特殊字符和换行符。

示例:

```
# clia setpefconfig alert_string 2 "This string has a line feed  
inside."  
IPM Sentry Shelf Manager Command Line Interpreter  
Alert string set successfully  
#
```

oem_filter

语法:

```
setpefconfig oem_filter set-selector value1 value2 value3
setpefconfig 97 set-selector value1 value2 value3
```

用途:

OEM 过滤器表是 Pigeon Point Systems 定义的对 IPMI 规范的 OEM 扩展。除平台事件外，它还允许将 PEF 应用到加 OEM 时间戳的和未加时间戳的 SEL 项（记录类型范围为 C0h-FFh）。

OEM 过滤器表的各项都会定义应用该 OEM 过滤器的记录类型范围（在 OEM 记录类型的范围内）以及将记录类型相符的记录放置在 SEL 中时要调用的警报策略编号。

该命令用于设置 OEM 过滤器表项，其编号由指定的已设置选择器标识。该项由以下三个数值组成：

- 第 1 个字节：SEL 记录类型范围的下限
- 第 2 个字节：SEL 记录类型范围的上限
- 第 3 个字节：针对其记录类型符合上述范围的 SEL 项所要调用的警报策略编号

示例:

```
# clia getpefconfig oem_filter
IPM Sentry Shelf Manager Command Line Interpreter
Active OEM Filters:
    0x01: OEM range boundary 0xff:0xff, alert policy # 1
#
# clia setpefconfig oem_filter 4 0xdc 0xf3 5
IPM Sentry Shelf Manager Command Line Interpreter
OEM filter set successfully
#
# clia getpefconfig oem_filter
IPM Sentry Shelf Manager Command Line Interpreter
Active OEM Filters:
    0x01: OEM range boundary 0xff: 0xff, alert policy # 1
    0x04: OEM range boundary 0xdc: 0xf3, alert policy # 5
#
```

setsensoreventenable

语法:

```
setsensoreventenable IPMB-address sensor-name global  
[assertion_events [deassertion_events]]
```

```
setsensoreventenable IPMB-address [lun:]sensor-number global  
[assertion_events [deassertion_events]]
```

用户可以使用以下项代替 *IPMB-address*:

```
board n  
shm n
```

用途:

该命令用于更改指定传感器的事件启用掩码。传感器按所属 IPM 控制器的 IPMB 地址和传感器名称或编号进行指定。另外，也可以使用板编号或专用机框管理器编号指定目标 IPM 控制器。

如果目标控制器支持多个 LUN 上的传感器，则此命令允许用户使用逻辑单元号 (Logical Unit Number, LUN) 来限定传感器编号。*lun* 的取值可以为 0、1 或 3。(LUN 2 为保留值。) 如果省略 LUN，则该命令将应用于编号最低的 LUN 上具有指定传感器编号的传感器。(例如，如果命令中指定了传感器 3 但没有明确使用 LUN 进行限定，而目标控制器上显示有 LUN 1 上的传感器 3 和 LUN 3 上的另一个传感器 3，则该命令将应用于 LUN 1 上的传感器 3。)

示例:

对 IPM 控制器 FE 上的温度传感器 Local Temp 启用 Lower Non-Critical Going Low 事件

```
# clia setsensoreventenable fe "Local Temp" 0x90 0x01 0x00  
IPM Sentry Shelf Manager Command Line Interpreter  
Event enable mask set successfully  
#  
# clia getsensoreventenable -v fe "Local Temp"  
IPM Sentry Shelf Manager Command Line Interpreter  
fe: LUN: 0, Sensor # 3 ("Local Temp")  
Type: Threshold (0x01), "Temperature" (0x01)  
Sensor scanning disabled  
Assertion event mask: 0x0001  
Assertion event for "Lower Non-Critical Going Low" enabled  
Deassertion event mask: 0x0000  
#
```

对同一传感器执行相同的操作，但使用 LUN 和传感器编号指定传感器：

```
# clia setsensoreventenable fe 0:3 0x90 0x01 0x00
IPM Sentry Shelf Manager Command Line Interpreter
Event enable mask set successfully
#
```

setthreshold

语法：

```
setthreshold IPMB-address sensor-name threshold-type [-r] value
setthreshold IPMB-address [lun:]sensor-number threshold-type [-r] value
```

用户可以使用以下项代替 *IPMB-address*：

```
board n
shm n
```

用途：

该命令用于将指定传感器的当前阈值更改为指定阈值。传感器按所属 IPM 控制器的 IPMB 地址和传感器名称或编号进行指定。目标传感器必须是基于阈值的传感器。可以将参数 *threshold-type* 指定为下列符号值之一：

- upper_non_recoverable（可以缩写为 unr）
- upper_critical（可以缩写为 uc）
- upper_non_critical（可以缩写为 unc）
- lower_non_recoverable（可以缩写为 lnr）
- lower_critical（可以缩写为 lc）
- lower_non_critical（可以缩写为 lnc）

默认情况下，目标值按处理过的形式指定（即，用“伏特”（Volts）表示电压传感器，用“摄氏度”（Celsius）表示温度传感器）。选项 *-r* 表示改用原始值（通常为以字节大小表示的数量，根据特定于传感器的规则进行转换）。

如果目标控制器支持多个 LUN 上的传感器，则此命令允许用户使用逻辑单元号（Logical Unit Number, LUN）来限定传感器编号。*lun* 的取值可以为 0、1 或 3。（LUN 2 为保留值。）如果省略 LUN，则该命令将应用于编号最低的 LUN 上具有指定传感器编号的传感器。（例如，如果命令中指定了传感器 3 但没有明确指定 LUN 进行限定，而目标控制器上显示有 LUN 1 上的传感器 3 和 LUN 3 上的另一个传感器 3，则该命令将应用于 LUN 1 上的传感器 3。）

示例:

将 IPM 控制器 9C 上的温度传感器 emulated temp 的非临界阈值上限值设置为 99 摄氏度。

```
# clia threshold 9c 2
IPM Sentry Shelf Manager Command Line Interpreter
9c: LUN: 0, Sensor # 2 ("emulated temp")
  Type: Threshold (0x01), "Temperature" (0x01)
    Lower Non-Critical Threshold, Raw Data: 0x03, Processed
Data: 3.000000 degrees C
    Lower Critical Threshold, Raw Data: 0x14, Processed Data:
20.000000 degrees C
    Lower Non-Recoverable Threshold, Raw Data: 0xfb, Processed
Data: -5.000000 degrees C
    Upper Non-Critical Threshold, Raw Data: 0x46, Processed
Data: 70.000000 degrees C
    Upper Critical Threshold, Raw Data: 0x50, Processed Data:
80.000000 degrees C
    Upper Non-Recoverable Threshold, Raw Data: 0x5a, Processed
Data: 90.000000 degrees C
#
# clia setthreshold 9c 0:2 unc 99
IPM Sentry Shelf Manager Command Line Interpreter
Threshold set successfully
#
# clia threshold 9c 0:2
IPM Sentry Shelf Manager Command Line Interpreter
9c: LUN: 0, Sensor # 2 ("emulated temp")
  Type: Threshold (0x01), "Temperature" (0x01)
    Lower Non-Critical Threshold, Raw Data: 0x03, Processed
Data: 3.000000 degrees C
    Lower Critical Threshold, Raw Data: 0x14, Processed Data:
20.000000 degrees C
    Lower Non-Recoverable Threshold, Raw Data: 0xfb, Processed
Data: -5.000000 degrees C
    Upper Non-Critical Threshold, Raw Data: 0x63, Processed
Data: 99.000000 degrees C
    Upper Critical Threshold, Raw Data: 0x50, Processed Data:
80.000000 degrees C
    Upper Non-Recoverable Threshold, Raw Data: 0x5a, Processed
Data: 90.000000 degrees C
#
```

shelf

语法:

`shelf subcommand`

支持以下子命令。

- `address_table`
- `cooling_state`
- `fans_state`
- `power_distribution`
- `power_management`
- `pci_connectivity`
- `ha_connectivity`
- `h110_connectivity`
- `point-to-point_connectivity`
- `MaxCurrent [feed] Amps`
- `MinVoltage [feed] Volts`
- `Activation addr fru_id 1|0`
- `Deactivation addr fru_id 1|0`
- `PwrCapability addr fru_id Watts`
- `PwrDelay addr fru_id 10ths_of_second`
- `Allowance seconds`
- `PwrReorder addr1 fru_id1 before|after addr2 fru_id2`
- `info_refresh`
- `info_force_update`

用途:

`shelf` 命令用于显示机框 FRU 的关键信息以及所选的有关机框的当前运行数据，并允许修改机框 FRU 信息中的某些字段。通过该命令显示或修改的信息类型在命令参数中进行指定。

以下各小节针对 `shelf` 命令的不同用途来介绍此命令的语法。

显示机框 FRU 信息

语法:

```
shelf [cooling_state | fans_state | address_table
      | power_distribution | power_management
      | pci_connectivity | ha_connectivity
      | h110_connectivity | point-to-point_connectivity]
```

用途:

该 Shelf 命令语法用于显示机框 FRU 的关键信息以及所选的有关机框的当前运行数据。该命令显示的信息类型在命令参数中进行指定。表 A-5 列出了 shelf 命令支持的参数:

表 A-5 shelf 命令的参数

命令参数	提供的信息
cooling_state (可以缩写为 cs)	显示机框的当前冷却状态: <ul style="list-style-type: none">• 正常 - 所有温度传感器都显示正常的运行温度。• 次要警报 - 至少有一个温度传感器处于次要警报状态。没有传感器处于重要或紧急警报状态。• 重要警报 - 至少有一个温度传感器处于重要警报状态。没有传感器处于紧急警报状态。• 紧急警报 - 至少有一个温度传感器处于紧急警报状态。
fans_state (可以缩写为 fs)	显示机框中风扇转速计的当前状态: <ul style="list-style-type: none">• 正常 - 所有风扇转速计传感器都显示正常的运行速度。• 次要警报 - 至少有一个风扇转速计传感器处于次要警报状态。没有传感器处于重要或紧急警报状态。• 重要警报 - 至少有一个风扇转速计传感器处于重要警报状态。没有传感器处于紧急警报状态。• 紧急警报 - 至少有一个风扇转速计传感器处于紧急警报状态。
address_table (可以缩写为 at)	显示机框 FRU 信息中的地址表记录。提供的信息如下: <ul style="list-style-type: none">• 机框地址 (根据其类型显示)• 地址表项列表, 其中显示每一项的硬件地址、位置编号以及位置类型。

表 A-5 shelf 命令的参数 (续)

<p><code>power_distribution</code> (可以缩写为 <code>pd</code>)</p>	<p>提供每路电源馈电的以下信息 (主要来自机框 FRU 信息的机框配电记录):</p> <ul style="list-style-type: none"> • 最大外部可用电流 • 最大内部电流 • 最小预期工作电压 • 实际可用功率 • 当前使用功率 • 与馈电线路连接的 FRU 的列表, 其中显示每个 FRU 的硬件地址和 FRU 设备 ID
<p><code>power_management</code> (可以缩写为 <code>pm</code>)</p>	<ul style="list-style-type: none"> • 机框 FRU 信息中的机框电源管理记录。该记录中包含 FRU 电源描述符的列表。对于每个描述符, 将提供以下信息: • 硬件地址 • FRU 设备 ID • FRU 最大功率容量 • 机框管理器控制的激活 • 下次打开电源前的延迟
<p><code>pci_connectivity</code> (可以缩写为 <code>pcic</code>)</p>	<p>机框 FRU 信息中的机框 PCI 连接记录。提供的信息如下:</p> <ul style="list-style-type: none"> • PCI 插槽描述符 • DSEL 连接 • 段 ID • 扩展 PCI 插槽描述符 • 地理地址 • 接口号 • 系统插槽容量
<p><code>ha_connectivity</code> (可以缩写为 <code>ha</code>)</p>	<p>机框 FRU 信息中的机框 HA 连接记录。提供的信息如下:</p> <ul style="list-style-type: none"> • 辐射型连接支持
<p><code>h110_connectivity</code> (可以缩写为 <code>h110c</code>)</p>	<p>机框 FRU 信息中的机框 H110 连接记录。提供的信息如下:</p> <ul style="list-style-type: none"> • 地理地址 • 段 ID
<p><code>point-to-point_connectivity</code> (可以缩写为 <code>ppc</code>)</p>	<p>机框 FRU 信息中的机框点对点连接记录。提供的信息如下:</p> <ul style="list-style-type: none"> • 通道类型 • 通道计数 • 插槽/硬件地址 • 通道描述符

对于命令参数 `cooling_state` 和 `fans_state`, 可以使用详细选项 `-v`。应该在命令参数之前输入该选项: `cli shelf -v cooling_state`。如果使用该选项, 则命令将显示与当前状态有关的传感器列表 (温度传感器或风扇转速计传感器)。每个传感器都以元组形式 (*IPMB-address*, *sensor_number*) 显示。

示例:

获取机框冷却状态。

```
# clia shelf cooling_state
IPM Sentry Shelf Manager Command Line Interpreter
Cooling state: "Normal"
#
```

获取机框风扇转速计状态（详细）。

```
# clia shelf -v fans_state
IPM Sentry Shelf Manager Command Line Interpreter
Fans state: "Major Alert"
Sensor(s) at this state: (0x7e,10) (0x7e,11) (0x7e,12) (0x7e,13)
                        (0x7e,14) (0x7e,15) (0x7e,16) (0x7e,17)
#
```

获取地址表。

```
# clia shelf address_table
IPM Sentry Shelf Manager Command Line Interpreter
PICMG Address Table Record (ID=0x10)
Version = 1
Shelf Address          =
Address Table Entries# = 16
Hw Addr: 41, Site # 1, Type: "AdvancedTCA Board" 00
Hw Addr: 42, Site # 2, Type: "AdvancedTCA Board" 00
Hw Addr: 43, Site # 3, Type: "AdvancedTCA Board" 00
Hw Addr: 44, Site # 4, Type: "AdvancedTCA Board" 00
Hw Addr: 45, Site # 5, Type: "AdvancedTCA Board" 00
Hw Addr: 46, Site # 6, Type: "AdvancedTCA Board" 00
Hw Addr: 47, Site # 7, Type: "AdvancedTCA Board" 00
Hw Addr: 48, Site # 8, Type: "AdvancedTCA Board" 00
Hw Addr: 49, Site # 9, Type: "AdvancedTCA Board" 00
Hw Addr: 4a, Site # 10, Type: "AdvancedTCA Board" 00
Hw Addr: 4b, Site # 11, Type: "AdvancedTCA Board" 00
Hw Addr: 4c, Site # 12, Type: "AdvancedTCA Board" 00
Hw Addr: 4d, Site # 13, Type: "AdvancedTCA Board" 00
Hw Addr: 4e, Site # 14, Type: "AdvancedTCA Board" 00
#
```

获取配电信息。

```
# clia shelf power_distribution
IPM Sentry Shelf Manager Command Line Interpreter
Power Distribution:
  Feed count: 1
  Feed 00:
    Maximum External Available Current: 50.0 Amps
    Maximum Internal Current: Not specified
    Minimum Expected Operating Voltage: -40.5 Volts
    Actual Power Available: 2025.000 Watts
    Currently Used Power: 160.000 Watts
    Feed-to-FRU Mapping entries count: 16
      FRU Addr: 41, FRU ID: fe
      FRU Addr: 42, FRU ID: fe
      FRU Addr: 43, FRU ID: fe
      FRU Addr: 44, FRU ID: fe
      FRU Addr: 45, FRU ID: fe
      FRU Addr: 46, FRU ID: fe
      FRU Addr: 47, FRU ID: fe
      FRU Addr: 48, FRU ID: fe
      FRU Addr: 49, FRU ID: fe
      FRU Addr: 4a, FRU ID: fe
      FRU Addr: 4b, FRU ID: fe
      FRU Addr: 4c, FRU ID: fe
      FRU Addr: 4d, FRU ID: fe
      FRU Addr: 4e, FRU ID: fe
      FRU Addr: 4f, FRU ID: fe
      FRU Addr: 50, FRU ID: fe
#
```

获取电源管理信息。

```
# clia shelf pm
IPM Sentry Shelf Manager Command Line Interpreter
PICMG Shelf Activation And Power Management Record (ID=0x12)
  Version = 0
  Allowance for FRU Activation Readiness: 10 seconds
  FRU Activation and Power Description Count: 16
  Hw Address: 41, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
  Shelf Manager Controlled Activation: Enabled
  Delay Before Next Power On: 0.0 seconds
  Hw Address: 42, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
```

```
Shelf Manager Controlled Activation: Enabled
Delay Before Next Power On: 0.0 seconds
Hw Address: 43, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
Shelf Manager Controlled Activation: Enabled
Delay Before Next Power On: 0.0 seconds
Hw Address: 44, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
Shelf Manager Controlled Activation: Enabled
Delay Before Next Power On: 0.0 seconds
Hw Address: 45, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
Shelf Manager Controlled Activation: Enabled
Delay Before Next Power On: 0.0 seconds
Hw Address: 46, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
Shelf Manager Controlled Activation: Enabled
Delay Before Next Power On: 0.0 seconds
Hw Address: 47, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
Shelf Manager Controlled Activation: Enabled
Delay Before Next Power On: 0.0 seconds
Hw Address: 48, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
Shelf Manager Controlled Activation: Enabled
Delay Before Next Power On: 0.0 seconds
Hw Address: 49, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
Shelf Manager Controlled Activation: Enabled
Delay Before Next Power On: 0.0 seconds
Hw Address: 4a, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
Shelf Manager Controlled Activation: Enabled
Delay Before Next Power On: 0.0 seconds
Hw Address: 4b, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
Shelf Manager Controlled Activation: Enabled
Delay Before Next Power On: 0.0 seconds
Hw Address: 4c, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
Shelf Manager Controlled Activation: Enabled
Delay Before Next Power On: 0.0 seconds
Hw Address: 4d, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
Shelf Manager Controlled Activation: Enabled
```

```

    Delay Before Next Power On: 0.0 seconds
    Hw Address: 4e, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
    Shelf Manager Controlled Activation: Enabled
    Delay Before Next Power On: 0.0 seconds
    Hw Address: 4f, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
    Shelf Manager Controlled Activation: Enabled
    Delay Before Next Power On: 0.0 seconds
    Hw Address: 50, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
    Shelf Manager Controlled Activation: Enabled
    Delay Before Next Power On: 0.0 seconds
#

```

修改最大外部可用电流

语法:

```
shelf maxcurrent [feed] current
```

用途:

该命令用于为指定的馈电线路编号设置最大外部可用电流，以及更新机框中所有已知的机框 FRU 信息实例。如果省略 *feed* 参数，则将为机框 FRU 信息中的第一个馈电线路（馈电线路 0）设置相应的值。

参数 *feed* 是机框 FRU 信息中的馈电线路编号，该编号从 0 开始并按照相应馈电线路的说明顺序排列。

参数 *current* 是所需的电流值（以安培为单位）。

示例:

将馈电线路 0 的最大外部可用电流从 44 安培改为 99 安培。

```

# clia shelf pd
IPM Sentry Shelf Manager Command Line Interpreter
PICMG Shelf Power Distribution Record (ID=0x11)
    Version = 0
    Feed count: 1
    Feed 00:
        Maximum External Available Current: 50.0 Amps
        Maximum Internal Current: Not specified
        Minimum Expected Operating Voltage: -40.5 Volts

```

```
Actual Power Available: 2025.000 Watts
Currently Used Power: 200.000 Watts
Feed-to-FRU Mapping entries count: 16
  FRU Addr: 41, FRU ID: 0xfe
  FRU Addr: 42, FRU ID: 0xfe
  FRU Addr: 43, FRU ID: 0xfe
  FRU Addr: 44, FRU ID: 0xfe
  FRU Addr: 45, FRU ID: 0xfe
  FRU Addr: 46, FRU ID: 0xfe
  FRU Addr: 47, FRU ID: 0xfe
  FRU Addr: 48, FRU ID: 0xfe
  FRU Addr: 49, FRU ID: 0xfe
  FRU Addr: 4a, FRU ID: 0xfe
  FRU Addr: 4b, FRU ID: 0xfe
  FRU Addr: 4c, FRU ID: 0xfe
  FRU Addr: 4d, FRU ID: 0xfe
  FRU Addr: 4e, FRU ID: 0xfe
  FRU Addr: 4f, FRU ID: 0xfe
  FRU Addr: 50, FRU ID: 0xfe
```

```
# clia shelf maxcurrent 0 99
```

```
IPM Sentry Shelf Manager Command Line Interpreter
Updating Shelf FRU Info
Cached information updated
```

```
# clia shelf pd
```

```
IPM Sentry Shelf Manager Command Line Interpreter
PICMG Shelf Power Distribution Record (ID=0x11)
  Version = 0
  Feed count: 1
  Feed 00:
    Maximum External Available Current: 99.0 Amps
    Maximum Internal Current: Not specified
    Minimum Expected Operating Voltage: -40.5 Volts
    Actual Power Available: 2025.000 Watts
    Currently Used Power: 200.000 Watts
    Feed-to-FRU Mapping entries count: 16
      FRU Addr: 41, FRU ID: 0xfe
      FRU Addr: 42, FRU ID: 0xfe
      FRU Addr: 43, FRU ID: 0xfe
      FRU Addr: 44, FRU ID: 0xfe
      FRU Addr: 45, FRU ID: 0xfe
      FRU Addr: 46, FRU ID: 0xfe
      FRU Addr: 47, FRU ID: 0xfe
      FRU Addr: 48, FRU ID: 0xfe
```



```
FRU Addr: 49, FRU ID: 0xfe
FRU Addr: 4a, FRU ID: 0xfe
FRU Addr: 4b, FRU ID: 0xfe
FRU Addr: 4c, FRU ID: 0xfe
FRU Addr: 4d, FRU ID: 0xfe
FRU Addr: 4e, FRU ID: 0xfe
FRU Addr: 4f, FRU ID: 0xfe
FRU Addr: 50, FRU ID: 0xfe
#
```

修改最小预期工作电压

语法:

```
shelf minvoltage [feed] voltage
```

用途:

该命令用于为指定的馈电路编号设置最小预期工作电压，以及更新机框中所有已知的机框 FRU 信息实例。如果省略 *feed* 参数，则将为机框 FRU 信息中的第一个馈电路（馈电路 0）设置相应的值。

参数 *feed* 是机框 FRU 信息中的馈电路编号，该编号从 0 开始并按照相应馈电路的说明顺序排列。

参数 *voltage* 是所需的电压值。

示例:

更改馈电路 0 的最小预期工作电压

```
# clia shelf pd
IPM Sentry Shelf Manager Command Line Interpreter
PICMG Shelf Power Distribution Record (ID=0x11)
  Version = 0
  Feed count: 1
  Feed 00:
    Maximum External Available Current: 99.0 Amps
    Maximum Internal Current: Not specified
    Minimum Expected Operating Voltage: -40.5 Volts
    Actual Power Available: 2025.000 Watts
    Currently Used Power: 200.000 Watts
    Feed-to-FRU Mapping entries count: 16
      FRU Addr: 41, FRU ID: 0xfe
      FRU Addr: 42, FRU ID: 0xfe
```

```
FRU Addr: 43, FRU ID: 0xfe
FRU Addr: 44, FRU ID: 0xfe
FRU Addr: 45, FRU ID: 0xfe
FRU Addr: 46, FRU ID: 0xfe
FRU Addr: 47, FRU ID: 0xfe
FRU Addr: 48, FRU ID: 0xfe
FRU Addr: 49, FRU ID: 0xfe
FRU Addr: 4a, FRU ID: 0xfe
FRU Addr: 4b, FRU ID: 0xfe
FRU Addr: 4c, FRU ID: 0xfe
FRU Addr: 4d, FRU ID: 0xfe
FRU Addr: 4e, FRU ID: 0xfe
FRU Addr: 4f, FRU ID: 0xfe
FRU Addr: 50, FRU ID: 0xfe
```

```
# clia shelf minvoltage 0 -59
```

```
IPM Sentry Shelf Manager Command Line Interpreter
Updating Shelf FRU Info
Cached information updated
```

```
# clia shelf pd
```

```
IPM Sentry Shelf Manager Command Line Interpreter
PICMG Shelf Power Distribution Record (ID=0x11)
Version = 0
Feed count: 1
Feed 00:
Maximum External Available Current: 99.0 Amps
Maximum Internal Current: Not specified
Minimum Expected Operating Voltage: -59.0 Volts
Actual Power Available: 2025.000 Watts
Currently Used Power: 200.000 Watts
Feed-to-FRU Mapping entries count: 16
FRU Addr: 41, FRU ID: 0xfe
FRU Addr: 42, FRU ID: 0xfe
FRU Addr: 43, FRU ID: 0xfe
FRU Addr: 44, FRU ID: 0xfe
FRU Addr: 45, FRU ID: 0xfe
FRU Addr: 46, FRU ID: 0xfe
FRU Addr: 47, FRU ID: 0xfe
FRU Addr: 48, FRU ID: 0xfe
FRU Addr: 49, FRU ID: 0xfe
FRU Addr: 4a, FRU ID: 0xfe
FRU Addr: 4b, FRU ID: 0xfe
FRU Addr: 4c, FRU ID: 0xfe
```

```
FRU Addr: 4d, FRU ID: 0xfe
FRU Addr: 4e, FRU ID: 0xfe
FRU Addr: 4f, FRU ID: 0xfe
FRU Addr: 50, FRU ID: 0xfe
#
```

修改“机框管理器控制的激活”标志

语法:

```
shelf activation hardware-addr fru-id [1/0]
shelf activation board n [1/0]
shelf activation board all [1/0]
shelf activation fan_tray n [1/0]
```

用途:

该命令用于针对指定 IPM 控制器的指定 FRU 显示或更改“机框管理器控制的激活”字段。该命令仅针对机框激活和电源管理记录中已存在的项修改“机框管理器控制的激活”标志。该命令还会更新机框管理器所用的机框 FRU 信息的缓存版本。因此，“机框管理器控制的激活”字段的新值将立即生效，而无需重新启动机框管理器。

参数 *hardware-addr* 是十六进制格式的 7 位硬件地址。

参数 *fru-id* 是十六进制格式的 FRU ID；0xFE 表示该硬件地址上的所有 FRU。

示例:

对硬件地址为 0x42（IPMB 地址为 0x84）的 IPM 控制器启用“机框管理器控制的激活”。

```
# clia shelf pm
IPM Sentry Shelf Manager Command Line Interpreter
PICMG Shelf Activation And Power Management Record (ID=0x12)
    Version = 0
    Allowance for FRU Activation Readiness: 10 seconds
    FRU Activation and Power Description Count: 16
    Hw Address: 41, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 42, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
```

Delay Before Next Power On: 0.0 seconds

Hw Address: 43, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts

Shelf Manager Controlled Activation: Enabled

Delay Before Next Power On: 0.0 seconds

Hw Address: 44, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts

Shelf Manager Controlled Activation: Enabled

Delay Before Next Power On: 0.0 seconds

Hw Address: 45, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts

Shelf Manager Controlled Activation: Enabled

Delay Before Next Power On: 0.0 seconds

Hw Address: 46, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts

Shelf Manager Controlled Activation: Enabled

Delay Before Next Power On: 0.0 seconds

Hw Address: 47, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts

Shelf Manager Controlled Activation: Enabled

Delay Before Next Power On: 0.0 seconds

Hw Address: 48, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts

Shelf Manager Controlled Activation: Enabled

Delay Before Next Power On: 0.0 seconds

Hw Address: 49, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts

Shelf Manager Controlled Activation: Enabled

Delay Before Next Power On: 0.0 seconds

Hw Address: 4a, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts

Shelf Manager Controlled Activation: Enabled

Delay Before Next Power On: 0.0 seconds

Hw Address: 4b, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts

Shelf Manager Controlled Activation: Enabled

```

    Delay Before Next Power On: 0.0 seconds

    Hw Address: 4c, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 4d, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 4e, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 4f, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 50, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

# clia shelf activation 42 0xfe 0
IPM Sentry Shelf Manager Command Line Interpreter
Updating Shelf FRU Info, address: 0x42, FRU ID # 254
    Cached information updated
    Wrote Information to the Shelf FRU

# clia shelf pm
IPM Sentry Shelf Manager Command Line Interpreter
PICMG Shelf Activation And Power Management Record (ID=0x12)
    Version = 0
    Allowance for FRU Activation Readiness: 10 seconds
    FRU Activation and Power Description Count: 16
    Hw Address: 41, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

```

Hw Address: 42, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts

Shelf Manager Controlled Activation: Disabled
Delay Before Next Power On: 0.0 seconds

Hw Address: 43, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts

Shelf Manager Controlled Activation: Enabled
Delay Before Next Power On: 0.0 seconds

Hw Address: 44, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts

Shelf Manager Controlled Activation: Enabled
Delay Before Next Power On: 0.0 seconds

Hw Address: 45, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts

Shelf Manager Controlled Activation: Enabled
Delay Before Next Power On: 0.0 seconds

Hw Address: 46, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts

Shelf Manager Controlled Activation: Enabled
Delay Before Next Power On: 0.0 seconds

Hw Address: 47, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts

Shelf Manager Controlled Activation: Enabled
Delay Before Next Power On: 0.0 seconds

Hw Address: 48, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts

Shelf Manager Controlled Activation: Enabled
Delay Before Next Power On: 0.0 seconds

Hw Address: 49, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts

Shelf Manager Controlled Activation: Enabled
Delay Before Next Power On: 0.0 seconds

Hw Address: 4a, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts

Shelf Manager Controlled Activation: Enabled
Delay Before Next Power On: 0.0 seconds

```
Hw Address: 4b, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
    Shelf Manager Controlled Activation: Enabled
    Delay Before Next Power On: 0.0 seconds

Hw Address: 4c, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
    Shelf Manager Controlled Activation: Enabled
    Delay Before Next Power On: 0.0 seconds

Hw Address: 4d, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
    Shelf Manager Controlled Activation: Enabled
    Delay Before Next Power On: 0.0 seconds

Hw Address: 4e, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
    Shelf Manager Controlled Activation: Enabled
    Delay Before Next Power On: 0.0 seconds

Hw Address: 4f, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
    Shelf Manager Controlled Activation: Enabled
    Delay Before Next Power On: 0.0 seconds

Hw Address: 50, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
    Shelf Manager Controlled Activation: Enabled
    Delay Before Next Power On: 0.0 seconds

#
```

修改“机框管理器控制的取消激活”标志

语法:

```
shelf deactivation hardware-addr fru-id [1/0]
shelf deactivation board n [1/0]
shelf deactivation board all [1/0]
shelf deactivation fan_tray n [1/0]
```

用途:

该命令用于针对指定 IPM 控制器的指定 FRU 显示或更改“机框管理器控制的取消激活”字段。该命令仅针对机框激活和电源管理记录中已存在的项修改“机框管理器控制的取消激活”标志。该命令还会更新机框管理器所用的机框 FRU 信息的缓存版本。因此，“机框管理器控制的取消激活”字段的新值将立即生效，而无需重新启动机框管理器。

参数 *hardware-addr* 是十六进制格式的 7 位硬件地址。

参数 *fru-id* 是十六进制格式的 FRU ID；0xFE 表示该硬件地址上的所有 FRU。

值为 0 表示启用“机框管理器控制的取消激活”，让机框管理器负责取消激活板。值为 1 表示禁用“机框管理器控制的取消激活”，阻止机框管理器自动取消激活板。如果禁用了“机框管理器控制的取消激活”，则由系统管理员负责取消激活板。

示例:

对硬件地址为 0x41（IPMB 地址为 0x84）的 IPM 控制器启用“机框管理器控制的取消激活”。

```
# clia shelf pm
IPM Sentry Shelf Manager Command Line Interpreter
PICMG Shelf Activation And Power Management Record (ID=0x12)
  Version = 1
  Allowance for FRU Activation Readiness: 10 seconds
  FRU Activation and Power Description Count: 16
  Hw Address: 41, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
  Shelf Manager Controlled Activation: Enabled
  Shelf Manager Controlled Deactivation: Disabled
  Delay Before Next Power On: 0.0 seconds

  Hw Address: 42, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
  Shelf Manager Controlled Activation: Enabled
  Shelf Manager Controlled Deactivation: Disabled
  Delay Before Next Power On: 0.0 seconds
```



```

# clia shelf deactivation 0x41 0xfe 0
IPM Sentry Shelf Manager Command Line Interpreter
Updating Shelf FRU Info, address: 0x41, FRU ID # 254
    Cached information updated
    Wrote Information to the Shelf FRU

# clia shelf pm
IPM Sentry Shelf Manager Command Line Interpreter
PICMG Shelf Activation And Power Management Record (ID=0x12)
    Version = 1
    Allowance for FRU Activation Readiness: 10 seconds
    FRU Activation and Power Description Count: 16
    Hw Address: 41, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
    Shelf Manager Controlled Activation: Enabled
    Shelf Manager Controlled Deactivation: Enabled
    Delay Before Next Power On: 0.0 seconds

    Hw Address: 42, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
    Shelf Manager Controlled Activation: Enabled
    Shelf Manager Controlled Deactivation: Disabled
    Delay Before Next Power On: 0.0 seconds

#

```

修改 FRU 最大功率容量

语法:

```

shelf pwrcapability hardware-addr fru-id value
shelf pwrcapability board n value
shelf pwrcapability fan_tray n value

```

用途:

该命令用于针对指定 IPM 控制器的指定 FRU 更改“FRU 最大功率容量”字段。

注 – 切勿将“FRU 最大功率容量”字段设置为超出您的机框环境安全范围的值。

该命令仅针对机框激活和电源管理记录中已存在的项修改此字段。该命令还会更新机框管理器所用的机框 FRU 信息的缓存版本。因此，“FRU 最大功率容量”字段的新值将立即生效，而无需重新启动机框管理器。

- 参数 *hardware-addr* 是十六进制格式的 7 位硬件地址。
- 参数 *fru-id* 是十六进制格式的 FRU ID；0xFE 表示该硬件地址上的所有 FRU。
- 参数 *value* 是该字段的新值（以瓦特为单位）。取值范围为 0 至 65535。

示例:

针对硬件地址为 0x42（IPMB 地址为 0x84）的 IPM 控制器，将“FRU 最大功率容量”设置为 150 瓦特。

```
# clia shelf pm
IPM Sentry Shelf Manager Command Line Interpreter
Power Management:
  Allowance for FRU Activation Readiness: 10 seconds
  FRU Activation and Power Description Count: 2
  Hw Address: 41, FRU ID: fe, Maximum FRU Power Capabilities:
200 Watts
  Shelf Manager Controlled Activation: Disabled
  Delay Before Next Power On: 2.2 seconds

  Hw Address: 42, FRU ID: fe, Maximum FRU Power Capabilities:
200 Watts
  Shelf Manager Controlled Activation: Disabled
  Delay Before Next Power On: 2.2 seconds
#
# clia shelf pwrcapability 42 0xfe 150
IPM Sentry Shelf Manager Command Line Interpreter
Updating Shelf FRU Info
  Cached information updated
#
# clia shelf pm
IPM Sentry Shelf Manager Command Line Interpreter
Power Management:
  Allowance for FRU Activation Readiness: 10 seconds
  FRU Activation and Power Description Count: 2
  Hw Address: 41, FRU ID: fe, Maximum FRU Power Capabilities:
200 Watts
  Shelf Manager Controlled Activation: Disabled
  Delay Before Next Power On: 2.2 seconds

  Hw Address: 42, FRU ID: fe, Maximum FRU Power Capabilities:
150 Watts
  Shelf Manager Controlled Activation: Disabled
  Delay Before Next Power On: 2.2 seconds
#
```

修改下次打开电源前的延迟

语法:

```
shelf pwrdelay hardware-addr fru-id value
shelf pwrdelay board n value
shelf pwrdelay fan_tray n value
```

用途:

该命令用于针对指定 IPM 控制器的指定 FRU 更改“下次打开电源前的延迟”字段。该命令仅针对机框激活和电源管理记录中已存在的项修改此字段。该命令还会更新机框管理器所用的机框 FRU 信息的缓存版本。因此，“下次打开电源前的延迟”字段的新值将立即生效，而无需重新启动机框管理器。

- 参数 *hardware-addr* 是十六进制格式的 7 位硬件地址。
- 参数 *fru-id* 是十六进制格式的 FRU ID；0xFE 表示该硬件地址上的所有 FRU。
- 参数 *value* 是该字段的新值（以十分之一秒为单位）。取值范围为 0 至 63。

示例:

针对硬件地址为 0x42（IPMB 地址为 0x84）的 IPM 控制器，将“下次打开电源前的延迟”设置为 5 秒。

```
# clia shelf pm
IPM Sentry Shelf Manager Command Line Interpreter
Power Management:
  Allowance for FRU Activation Readiness: 10 seconds
  FRU Activation and Power Description Count: 2
  Hw Address: 41, FRU ID: fe, Maximum FRU Power Capabilities:
200 Watts
  Shelf Manager Controlled Activation: Disabled
  Delay Before Next Power On: 2.2 seconds
  Hw Address: 42, FRU ID: fe, Maximum FRU Power Capabilities:
200 Watts
  Shelf Manager Controlled Activation: Disabled
  Delay Before Next Power On: 2.2 seconds
#
# clia shelf pwrdelay 42 0xfe 50
IPM Sentry Shelf Manager Command Line Interpreter
Updating Shelf FRU Info
  Cached information updated
#
# clia shelf pm
IPM Sentry Shelf Manager Command Line Interpreter
Power Management:
```

```

Allowance for FRU Activation Readiness: 10 seconds
FRU Activation and Power Description Count: 2
Hw Address: 41, FRU ID: fe, Maximum FRU Power Capabilities:
200 Watts
    Shelf Manager Controlled Activation: Disabled
    Delay Before Next Power On: 2.2 seconds
Hw Address: 42, FRU ID: fe, Maximum FRU Power Capabilities:
200 Watts
    Shelf Manager Controlled Activation: Disabled
    Delay Before Next Power On: 5.0 seconds
#

```

修改 FRU 激活准备就绪允许时间

语法:

```
shelf allowance value
```

用途:

该命令用于更改“FRU 激活准备就绪允许时间”参数。

参数 *value* 是该参数的新值（以秒为单位）。取值范围为 0 至 255。

示例:

将“FRU 激活准备就绪允许时间”设置为 5 秒。

```

# clia shelf pm
IPM Sentry Shelf Manager Command Line Interpreter
Power Management:
    Allowance for FRU Activation Readiness: 10 seconds
    FRU Activation and Power Description Count: 2
    Hw Address: 41, FRU ID: fe, Maximum FRU Power Capabilities:
200 Watts
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 2.2 seconds

    Hw Address: 42, FRU ID: fe, Maximum FRU Power Capabilities:
200 Watts
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 2.2 seconds
#

```

```

# clia shelf allowance 5

IPM Sentry Shelf Manager Command Line Interpreter
Updating Shelf FRU Info
#
# clia shelf pm
IPM Sentry Shelf Manager Command Line Interpreter
Power Management:
    Allowance for FRU Activation Readiness: 5 seconds
    FRU Activation and Power Description Count: 2
    Hw Address: 41, FRU ID: fe, Maximum FRU Power Capabilities:
200 Watts
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 2.2 seconds

    Hw Address: 42, FRU ID: fe, Maximum FRU Power Capabilities:
200 Watts
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 2.2 seconds

#

```

重新排列 FRU 激活和电源描述符的顺序

语法:

```
shelf pwrreorder hardware-addr-1 fru-id-1 before|after hardware-addr-2 fru-id-2
```

可以使用以下项替换 *hardware-address fru-id*:

```
board n
fan_tray n
```

用途:

该命令用于更改机框 FRU 信息中 FRU 激活和电源描述符的顺序。该命令只能重新排列已存在描述符的顺序。目前的实现形式还受限于只能在一个机框激活和电源管理记录内重新排列描述符的顺序。该命令还会更新机框管理器所用的机框 FRU 信息的缓存版本。因此，新的描述符顺序将立即生效，而无需重新启动机框管理器。

- 参数 *hardware-addr-1* 是需要移动的描述符的 7 位硬件地址（以十六进制格式表示）。
- 参数 *fru-id-1* 是需要移动的描述符的 FRU ID（以十六进制格式表示）；0xFE 表示该硬件地址上的所有 FRU。
- 参数 *hardware-addr-2* 是描述符的 7 位硬件地址（以十六进制格式表示），应该在其前或其后放置 *hardware-addr-1/fru-id-1* 描述符。
- 参数 *fru-id-2* 是描述符的 FRU ID（以十六进制格式表示），应该在其前/后放置 *hardware-addr-1/fru-id-1* 描述符。

示例:

将硬件地址为 0x42 (IPMB 地址为 0x84) 的 IPM 控制器的描述符放在硬件地址为 0x41 (IPMB 地址为 0x82) 的 IPM 控制器的描述符的前面。

```
# clia shelf pm
IPM Sentry Shelf Manager Command Line Interpreter
Power Management:
    Allowance for FRU Activation Readiness: 10 seconds
    FRU Activation and Power Description Count: 2
    Hw Address: 41, FRU ID: fe, Maximum FRU Power Capabilities:
200 Watts
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 2.2 seconds

    Hw Address: 42, FRU ID: fe, Maximum FRU Power Capabilities:
200 Watts
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 2.2 seconds
#
# clia shelf pwrreorder 42 0xfe before 41 0xfe
IPM Sentry Shelf Manager Command Line Interpreter
Updating Shelf FRU Info
    Cached information updated
#
# clia shelf pm
IPM Sentry Shelf Manager Command Line Interpreter
Power Management:
    Allowance for FRU Activation Readiness: 10 seconds
    FRU Activation and Power Description Count: 2
    Hw Address: 42, FRU ID: fe, Maximum FRU Power Capabilities:
200 Watts
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 2.2 seconds

    Hw Address: 41, FRU ID: fe, Maximum FRU Power Capabilities:
200 Watts
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 2.2 seconds
#
```

刷新机框 FRU 信息

语法:

```
shelf info_refresh
```

用途:

该命令用于使机框管理器重新读取以前在机框中找到的机框 FRU 信息源，并重新评估哪些源包含有效的机框 FRU 信息。假定已确认有效的机框 FRU 信息，则会使用新机框 FRU 信息的内容来更新所有机框 FRU 信息存储设备和机框 FRU 信息的缓存主副本。

按照 PICMG 3.0（第 3.6.4 节）中的指定，机框管理器会在初始化期间尝试查找可能的机框 FRU 信息存储设备。如果机框管理器找到至少两个包含有效机框 FRU 信息的 FRU 信息设备，则机框管理器将执行 election 以确定要使用哪些机框 FRU 信息源。这种选择是基于对存储设备所包含数据的验证以及对其内容的比较而进行的。成功完成选择后，机框管理器将创建机框 FRU 信息的缓存主副本（在易失性存储器中），该副本用于更新任何机框 FRU 信息源，并视为是机框 FRU 信息的唯一源。因此，所有与机框 FRU 信息相关的操作都使用此主副本，而且对主副本所做的更改会以增量更新方式自动传播到所有机框 FRU 信息源设备。

不过，并不支持动态重新配置。如果新的机框 FRU 信息与以前的机框 FRU 信息不同，则所做更改仅在重新引导机框管理器后才会完全生效。

示例:

成功刷新: 机框 FRU 信息的两个匹配源。

```
# clia shelf info_refresh

IPM Sentry Shelf Manager Command Line Interpreter

Read 0x20 # 2, size = 1024
Read 0x20 # 1, size = 1024
Found 2 Matching Shelf FRU Info

0x20 # 2, size = 1024 (data size = 775), "Valid" Shelf FRU,
"Matching"
0x20 # 1, size = 1024 (data size = 775), "Valid" Shelf FRU,
"Matching"
Shelf FRU Info was not changed

#

Unsuccessful refresh: both data sources contain non-matching or
invalid data.

# clia shelf info_refresh

IPM Sentry Shelf Manager Command Line Interpreter

Read 0x20 # 2, size = 1024
Read 0x20 # 1, size = 1024
No Matching Shelf FRU Info found

0x20 # 2, size = 1024 (data size = 293), "Invalid" Shelf FRU, "Non-
Matching"
0x20 # 1, size = 1024 (data size = 529), "Valid" Shelf FRU, "Non-
Matching"
Refresh was not done because system found only 1 (of 2) Matching
Shelf FRU info

#
```


更新机框 FRU 信息存储设备

语法:

```
shelf info_force_update
```

用途:

该命令用于检查机框 FRU 信息源设备，并将机框 FRU 信息主副本的内容复制到所有这些源设备。在机框 FRU 信息主副本和非易失性源设备之间存在冲突，且这种冲突无法自动解决的情况下（例如，EEPROM 和主副本两者彼此不同），该命令很有用。在这种情况下，操作员可以使用该命令强制使 EEPROM 与主副本的内容同步。此外，该命令还可以用于清除因最初的冲突而出现的错误状况；即，执行了该命令后，对机框 FRU 信息的后续更新将继续传播到 SEEPROM。

该命令将以异步方式启动机框 FRU 信息源设备的更新。

示例:

```
# clia shelf info_force_update

IPM Sentry Shelf Manager Command Line Interpreter

Starting the Shelf FRU Info source device update
#
```

shelfaddress

语法:

```
shelfaddress [up-to-30-characters-of-the-shelf-address]
```

用途:

该命令用于获取或设置机框 FRU 信息中地址表的“机框地址”字段。该命令使用 6 位压缩值，因此只允许使用大写字母和数字。

小写字母会自动转换为大写字母。

示例:

```
# clia shelfaddress
IPM Sentry Shelf Manager Command Line Interpreter
Shelf Address Info: "1234"
#
# clia shelfaddress "NEW SHELF ADDRESS"
IPM Sentry Shelf Manager Command Line Interpreter
Shelf Address Info set successfully
#
# clia shelfaddress
IPM Sentry Shelf Manager Command Line Interpreter
Shelf Address Info: "NEW SHELF ADDRESS"
#
```

shmstatus

语法:

```
shmstatus
```

用途:

该命令用于返回冗余配置中的机框管理器状态: 活动或备用。在详细模式下, 该命令将报告更详细的状况: 机框 FRU 信息的状态、RMCP 接口的状态和备用机框管理器的状态 (如果查询的机框管理器是活动机框管理器)。在下列情况下, 运行就绪标志参数显示为 Yes:

- 在活动机框管理器上, 如果找到有效的机框 FRU 信息并成功初始化其 RMCP 接口。
- 在备用机框管理器上, 如果从活动机框管理器成功收到冗余状态信息。

示例:

```
# clia shmstatus -v
IPM Sentry Shelf Manager Command Line Interpreter
Shelf Manager status: "Active"
Ready For Operation: Yes
Detailed State Flags: "Shelf FRU Found" "RMCP Up" "Backup Healthy"
#
```

showhost

语法:

```
showhost slot-number
```

用途:

该命令用于显示 Netra CP3060 板上安装的系统固件的版本。该命令仅对 Netra CT 900 服务器中安装的 Netra CP3060 板有效。

参数 *slot-number* 指定 Netra CP3060 板的插槽编号，*version* 选项用于显示完整的版本信息。

示例:

显示插槽 2 中 Netra CP3060 板的当前固件版本。

```
# clia showhost 2

System Firmware 6.2.5 Netra CP3060 2006/09/15 15:30

Host flash versions:
Hypervisor 1.2.3 2006/08/18 12:25
OBP 4.23.4 2006/08/04 20:46
Netra[TM] CP3060 POST 4.23.4 2006/08/04 21:17

#
```

showunhealthy

语法:

```
showunhealthy
```

用途:

该命令用于显示出现问题的 FRU 的列表。在 PICMG 3.0 环境中，该列表中包括上次热交换状态更改的原因为 Communication Lost、Communication lost due to local failure 和 Unexpected deactivation 的 FRU。

对于每个 FRU，将显示以下信息：IPMB 地址和 FRU 设备 ID、当前热交换状态、先前热交换状态以及上次状态更改的原因。

示例:

显示系统中工作不正常组件的列表。

```
# clia showunhealthy
IPM Sentry Shelf Manager Command Line Interpreter
There are no unhealthy components in the shelf.
#
```

switchover

语法:

```
switchover
```

用途:

该命令用于启动冗余机框管理器实例的切换。可以对机框管理器的活动实例或备用实例执行该命令。

示例:

从活动实例或备用实例启动切换。

```
# clia switchover
This Shelf Manager is now active, but is shutting down to
trigger a switchover.
#
```

terminate

语法:

```
terminate
```

用途:

该命令用于启动在不重新引导机框管理卡的情况下终止机框管理器。如果当前 ShMM 处于活动状态，则将会进行切换。

示例:

从活动实例或备用实例终止机框管理器。

```
# clia terminate
#
```

user

语法:

```
user [subcommand]
```

支持以下子命令:

- add
- delete
- enable
- name
- passwd
- channel

用途:

`user` 命令用于显示有关机框管理器中 RMCP 用户帐户的信息，并提供一种添加、删除和修改用户帐户的简单方式。

以下各小节针对 `user` 命令的不同用途来介绍此命令的语法

显示用户信息

语法:

```
user [-v] [userid]
```

用途:

该命令用于显示有关用户的信息。如果搭配使用 `-v` 选项来启动此命令，则还会显示有关已禁用的用户的信息。（默认情况下，仅列出已启用的用户。）如果指定了用户 ID（可选），则仅显示有关具有该 ID 的用户的信息。

将显示以下信息项:

- 用户 ID
- 用户名
- 每个 IPMI 通道的通道访问信息: 该用户对通道的最大权限级别以及通道访问标志

如果多个通道的通道访问信息都相同，则会联合输出，并显示通道范围。

示例:

```
# clia user
IPM Sentry Shelf Manager Command Line Interpreter
1: ""
    Channels 0-15 Privilege level: "Administrator"
    Flags: "IPMI Messaging"
#
# clia user -v
IPM Sentry Shelf Manager Command Line Interpreter
1: ""
    Channels 0-15 Privilege level: "Administrator"
    Flags: "IPMI Messaging"
7: "TEST1" Disabled
    Channels 0-15 Privilege level: "NO ACCESS"
#
```

添加新用户

语法:

```
user add userid user-name channel-access-flags privilege-level password
```

用途:

该命令用于向系统添加新用户。它可以按照命令中的指定，对所有通道设置相同的最大权限级别和通道访问标志。如果指定的用户不存在，该命令将返回错误。命令参数的含义如下:

- *userid* - 有效的用户 ID
- *user-name* - 用户名（将在没有任何提示的情况下截断为 16 个字符）
- *channel-access-flag* - SetUserInfo 命令的第一个字节（只有第 4 位、第 5 位和第 6 位有意义）
 - 第 6 位 - 已启用 IPMI 消息传送
 - 第 5 位 - 已启用链路验证
 - 第 4 位 - 限于回调
- *privilege-level* - 用户权限级别
- *password* - 用户密码（将在没有任何提示的情况下截断为 16 个字符）

示例:

添加具有名称 root、管理员权限级别和密码 PICMG guru 的用户 9。

```
# clia user
IPM Sentry Shelf Manager Command Line Interpreter
1: ""
           Channels 0-15 Privilege level: "Administrator"
           Flags: "IPMI Messaging"
#
# clia user add 9 "root" 0x40 4 "PICMG guru"
IPM Sentry Shelf Manager Command Line Interpreter
User 9 added successfully
#
# clia user
IPM Sentry Shelf Manager Command Line Interpreter
1: ""
           Channels 0-15 Privilege level: "Administrator"
           Flags: "IPMI Messaging"
9: "root"
           Channels 0-15 Privilege level: "Administrator"
           Flags: "IPMI Messaging"
#
```

删除用户

语法:

```
user delete userid
```

用途:

该命令用于删除 *userid* 所指定的用户。

示例:

删除用户 ID 为 10 的用户。

```
# clia user delete 10
IPM Sentry Shelf Manager Command Line Interpreter
User 10 deleted successful
#
# clia user
IPM Sentry Shelf Manager Command Line Interpreter
1: ""
    Channels 0-15 Privilege level: "Administrator"
    Flags: "IPMI Messaging"
9: "root"
    Channels 0-15 Privilege level: "Administrator"
    Flags: "IPMI Messaging"
#
```

启用和禁用用户

语法:

```
user enable userid 1 | 0
```

用途:

该命令用于按照 *userid* 启用或禁用用户。最后一个命令参数指定请求的操作，如下所示:

- 0 - 禁用指定的用户
- 非零值 - 启用指定的用户

示例:

启用和禁用 *userid* 为 9 的用户。

```
# clia user
IPM Sentry Shelf Manager Command Line Interpreter
1: ""
    Channels 0-15 Privilege level: "Administrator"
    Flags: "IPMI Messaging"
9: "root"
    Channels 0-15 Privilege level: "Administrator"
    Flags: "IPMI Messaging"
#
# clia user enable 9 0
IPM Sentry Shelf Manager Command Line Interpreter
    User 9 disabled successfully
#
# clia user -v
IPM Sentry Shelf Manager Command Line Interpreter
1: ""
    Channels 0-15 Privilege level: "Administrator"
    Flags: "IPMI Messaging"
9: "root" Disabled
    Channels 0-15 Privilege level: "Administrator"
    Flags: "IPMI Messaging"
#
# clia user enable 9 1
IPM Sentry Shelf Manager Command Line Interpreter
    User 9 enabled successfully
#
# clia user
IPM Sentry Shelf Manager Command Line Interpreter
1: ""
    Channels 0-15 Privilege level: "Administrator"
    Flags: "IPMI Messaging"
9: "root"
    Channels 0-15 Privilege level: "Administrator"
    Flags: "IPMI Messaging"
#
```

修改用户名

语法:

```
user name userid user-name
```

用途:

该命令用于修改指定用户的用户名。（用户是通过用户 ID 指定的。）命令参数的含义如下:

- *userid* - 有效的用户 ID
- *user-name* - 用户名（将在没有任何提示的情况下截断为 16 个字符）

示例:

将用户 9 的名称更改为 newby。

```
# cli user
IPM Sentry Shelf Manager Command Line Interpreter
1: ""
    Channels 0-15 Privilege level: "Administrator"
    Flags: "IPMI Messaging"
9: "root"
    Channels 0-15 Privilege level: "Administrator"
    Flags: "IPMI Messaging"
#
# cli user name 9 newby
IPM Sentry Shelf Manager Command Line Interpreter
User 9, name changed successfully
#
# cli user
IPM Sentry Shelf Manager Command Line Interpreter
1: ""
    Channels 0-15 Privilege level: "Administrator"
    Flags: "IPMI Messaging"
9: "newby"
    Channels 0-15 Privilege level: "Administrator"
    Flags: "IPMI Messaging"
#
```

修改用户密码

语法:

```
user passwd userid password
```

用途:

该命令用于修改指定用户的密码。(用户是通过用户 ID 指定的。) 命令参数的含义如下:

- *userid* - 有效的用户 ID
- *password* - 用户密码 (将在没有任何提示的情况下截断为 16 个字符)

示例:

将用户 ID 为 9 的用户的密码更改为 RIP

```
# clia user
IPM Sentry Shelf Manager Command Line Interpreter
1: ""
    Channels 0-15 Privilege level: "Administrator"
    Flags: "IPMI Messaging"
9: "newby"
    Channels 0-15 Privilege level: "Administrator"
    Flags: "IPMI Messaging"
#
# clia user passwd 9 RIP
IPM Sentry Shelf Manager Command Line Interpreter
User 9, password changed successfully
#
# clia user
IPM Sentry Shelf Manager Command Line Interpreter
1: ""
    Channels 0-15 Privilege level: "Administrator"
    Flags: "IPMI Messaging"
9: "newby"
    Channels 0-15 Privilege level: "Administrator"
    Flags: "IPMI Messaging"
#
```

修改指定用户和指定通道的通道访问设置

语法:

```
user channel userid channel-number channel-access-flags privilege-level
```

用途: 该命令用于修改指定通道和用户的通道访问设置。(用户是通过用户 ID 指定的。) 命令参数的含义如下:

- *userid* - 有效的用户 ID
- *channel-number* - 通道号
- *channel-access-flag* - SetUserInfo 命令的第一个字节 (只有第 4 位、第 5 位和第 6 位有意义)
 - 第 6 位 - 已启用 IPMI 消息传送
 - 第 5 位 - 已启用链路验证
 - 第 4 位 - 限于回调
- *privilege-level* - 用户权限级别

示例:

将用户 9 对通道 5 的最大权限级别更改为 User:

```
# clia user 9
IPM Sentry Shelf Manager Command Line Interpreter
9: "newby"
    Channels 0-15 Privilege level: "Administrator"
    Flags: "IPMI Messaging"
#
# clia user channel 9 5 0x60 2
IPM Sentry Shelf Manager Command Line Interpreter
User 9, channel 5 access updated successfully
#
# clia user 9
IPM Sentry Shelf Manager Command Line Interpreter
9: "newby"
    Channels 0-4 Privilege level: "Administrator"
    Flags: "IPMI Messaging"
    Channel 5 Privilege level: "User"
    Flags: "Link Authentication" "IPMI Messaging"
    Channels 6-15 Privilege level: "Administrator"
    Flags: "IPMI Messaging"
#
```

version

语法:

```
version
```

用途:

该命令用于显示机框管理器软件的版本信息。

示例:

```
# clia version
IPM Sentry Shelf Manager Command Line Interpreter
IPM Sentry Shelf Manager ver. 2.1.3
IPM Sentry is a trademark of Pigeon Point Systems.
Copyright (c) 2002-2005 Pigeon Point Systems
Build date/time: April 3 2006 16:39:37
All rights reserved
#
```


Sun OEM IPMI 命令

本附录中介绍的这些命令专用于 Sun Microsystems 设计的 ATCA 板。Internet 号码分配机构 (Internet Assigned Numbers Authority, IANA) 分配给 Sun Microsystems 的号码是 42。

有关 IANA 号码分配的更多信息，请参阅

<http://www.iana.org/assignments/enterprise-numbers>。

这些命令使用的 netfunction (NetFn) 是 0x2E，0x2E 是 IPMI 规范中定义的 OEM netfunction。对于此 netfunction，请求数据包中的前三个数据字节必须是此 IANA 号码，而跟随在完成代码后面的响应数据包中的前三个字节也将是 IANA 号码。对于 Sun ATCA 节点板，这三个字节为 00 00 2A。

表 B-1 列出了 Sun OEM IPMI 命令，随后的几节对这些命令进行了说明。

表 B-1 Sun OEM IPMI 命令

命令	操作码	语法
Get Version	0x80	#GET_VERSION
Set Boot Page [†]	0x81	#SET_BOOT_PAGE
Get Boot Page [†]	0x82	#GET_BOOT_PAGE
Set Front panel reset button state	0x83	#SET_FP_RESET_BUTTON
Get Front panel reset button state	0x84	#GET_FP_RESET_BUTTON_STATE
Set Ethernet Force Front bit	0x85	#SET_ETH_FORCE_FRONT
Get Ethernet Force Front bit	0x86	#GET_ETH_FORCE_FRONT
Get RTM status	0x88	#GET_RTM_STATUS

[†] 仅对 Netra CP3010 节点板有效

Get Version

Get Version 返回 IPM 控制器 (IPM controller, IPMC) 固件版本和备用 CPLD 版本。字节 8、9 和 A 保留供将来使用。

命令	NetFn	操作码	参考
Get Version	0x2E (OEM)	0x80	--

数据字节

类型	字节	数据字段
请求数据	Byte1	00
	Byte2	00
	Byte3	2A
响应数据	Byte1	完成代码： 00 = 正常 C1 = 不支持的命令 CC = 请求中的数据无效 (有关更多完成代码，请参阅 IPMI 规范)
	Byte2	00
	Byte3	00
	Byte4	2A
	Byte5	CPLD 版本
	Byte6	IPMC 固件的 REV1 字节
	Byte7	IPMC 固件的 REV2 字节
	Byte8	保留供将来使用 (忽略)。
	Byte9	保留供将来使用 (忽略)。
	ByteA	保留供将来使用 (忽略)。

示例 (终端模式):

```
[B8 00 80 00 00 2A] <-----请求  
[BC 00 80 00 00 00 2A 02 02 00 00 00 00] <----响应
```

- IPMC 版本按如下方式显示：
REV1 的低半字节 . REV2 的高半字节 . REV2 的低半字节
在上例中，IPMC 版本为 2.0.0。
- CPLD 版本按如下方式显示：
-> CPLD 版本字节的低半字节
在上例中，CPLD 版本为 2。

Set Boot Page

Set Boot Page 设置备用 CPLD 中的引导页面位以便为 Open Boot PROM 选择引导页面。在启动闪存故障恢复中可使用此功能。此命令仅对 Netra CP3010 节点板有效。

命令	NetFn	操作码	参考
Set Boot Page	0x2E (OEM)	0x81	CPLD 规范 1.0

数据字节

类型	字节	数据字段
请求数据	Byte1	00
	Byte2	00
	Byte3	2A
	Byte4	引导页面设置。 位 7 到 2 = 0 位 1 和 0 = 引导页面号码
响应数据	Byte1	完成代码： 00 = 正常 C1 = 不支持的命令 CC = 请求中的数据无效 (有关更多完成代码，请参阅 IPMI 规范)
	Byte2	00
	Byte3	00
	Byte4	2A

示例（终端模式）：

```
[B8 00 81 00 00 2A 02] <-----请求  
[BC 00 81 00 00 00 2A] <-----响应
```

Get Boot Page

Get Boot Page 返回所选 Open Boot PROM 引导页面的当前设置以及选择引导页面的硬件交换机的当前设置。此命令仅对 Netra CP3010 节点板有效。

命令	NetFn	操作码	参考
Get Boot Page	0x2E (OEM)	0x82	--

数据字节

类型	字节	数据字段
请求数据	Byte1	00
	Byte2	00
	Byte3	2A
响应数据	Byte1	完成代码： 00 = 正常 C1 = 不支持的命令 CC = 请求中的数据无效 (有关更多完成代码，请参阅 IPMI 规范)
	Byte2	00
	Byte3	00
	Byte4	2A
	Byte5	引导页面设置 位 7 到 4 = 忽略。将为零。 位 3、2 = 硬件交换机设置。 位 1、0 = 当前设置的引导页面。

示例（终端模式）：

```
[B8 00 82 00 00 2A] <-----请求  
[BC 00 82 00 00 00 2A 02] <-----响应
```

Set Front Panel Reset Button State

软件使用 Set Front panel reset button state 来更改按下此按钮时 CPLD 处理前面板复位的方式。CPLD 电源打开的默认值为 10。

命令	NetFn	操作码	参考
Set Front panel reset button state	0x2E (OEM)	0x83	CPLD 规范 1.0

数据字节

类型	字节	数据字段
请求数据	Byte1	00
	Byte2	00
	Byte3	2A
	Byte4	前面板复位按钮设置。 位 7 到 2 = 0 位 1 和 0 = 前面板按钮状态： 00 = 使 IPMC 复位并声明 CPU 的 POR 01 = CPU 的 XIR 10 = CPU 的 POR 11 = 禁用前面板复位按钮
响应数据	Byte1	完成代码： 00 = 正常 C1 = 不支持的命令 CC = 请求中的数据无效 (有关更多完成代码，请参阅 IPMI 规范)
	Byte2	00
	Byte3	00
	Byte4	2A

示例（终端模式）：

```
[B8 00 83 00 00 2A 02] <-----请求  
[BC 00 83 00 00 00 2A] <-----响应
```

Get Front Panel Reset Button State

Get front panel reset button state 返回前面板复位按钮处理方式的当前设置。默认情况下，打开 CPLD 电源时会采用 10，即按下此按钮会导致 CPU 加电复位。

命令	NetFn	操作码	参考
Get front panel reset button state	0x2E (OEM)	0x84	CPLD 规范 1.0

数据字节

类型	字节	数据字段
请求数据	Byte1	00
	Byte2	00
	Byte3	2A
响应数据	Byte1	完成代码： 00 = 正常 C1 = 不支持的命令 CC = 请求中的数据无效 (有关更多完成代码，请参阅 IPMI 规范)
	Byte2	00
	Byte3	00
	Byte4	2A
	Byte5	前面板复位按钮设置。 位 7 到 2 = 零。 位 1 和 0 = 前面板按钮状态： 00 = 使 IPMC 复位并声明 CPU 的 POR 01 = CPU 的 XIR 10 = CPU 的 POR 11 = 禁用前面板复位按钮

示例（终端模式）：

```
[B8 00 84 00 00 2A] <-----请求  
[BC 00 84 00 00 00 2A 02] <-----响应
```

Set Ethernet Force Front Bit

软件使用 Set Ethernet Force Front Bit 来强制建立与前 I/O 面板的以太网连接，即使系统中有后部转换模块也是如此。将此位设置为 1 会强制建立与前 I/O 面板的以太网连接。

命令	NetFn	操作码	参考
Set Ethernet Force Front Bit	0x2E (OEM)	0x85	CPLD 规范 1.0

数据字节

类型	字节	数据字段
请求数据	Byte1	00
	Byte2	00
	Byte3	2A
	Byte4	Ethernet Force Front Bit 的设置 位 7 到 1 = 0 位 0 = 强制前面板状态 (1 = 强制建立与前面板的以太网连接)
响应数据	Byte1	完成代码： 00 = 正常 C1 = 不支持的命令 CC = 请求中的数据无效 (有关更多完成代码，请参阅 IPMI 规范)
	Byte2	00
	Byte3	00
	Byte4	2A

示例（终端模式）：

```
[B8 00 85 00 00 2A 01] <----- 请求
[BC 00 85 00 00 00 2A] <----- 响应
```

Get Ethernet Force Front Bit

Get Ethernet Force Front Bit 返回 Ethernet Force Front Bit 的当前设置。

命令	NetFn	操作码	参考
Get Ethernet Force Front Bit	0x2E (OEM)	0x86	CPLD 规范 1.0

数据字节

类型	字节	数据字段
请求数据	Byte1	00
	Byte2	00
	Byte3	2A
响应数据	Byte1	完成代码： 00 = 正常 C1 = 不支持的命令 CC = 请求中的数据无效 (有关更多完成代码，请参阅 IPMI 规范)
	Byte2	00
	Byte3	00
	Byte4	2A
	Byte5	Ethernet Force Front Bit 的设置 位 7 到 1 = 0 位 0 = Ethernet Force Front Bit 的设置 (1=强制建立与前面板的以太网连接)

示例（终端模式）：

```
[B8 00 86 00 00 2A] <-----请求  
[BC 00 86 00 00 00 2A 01] <-----响应
```

Get RTM Status

Get RTM Status 用于检测系统中是否存在后部转换模块 (rear transition module, RTM)。

命令	NetFn	操作码	参考
Get RTM Status	0x2E (OEM)	0x88	CPLD 规范

数据字节

类型	字节	数据字段
请求数据	Byte1	00
	Byte2	00
	Byte3	2A
响应数据	Byte1	完成代码: 00 = 正常 C1 = 不支持的命令 CC = 请求中的数据无效
	Byte2	00
	Byte3	00
	Byte4	2A
	Byte5	RTM 存在 位 7 到 1 = 0 位 0 = RTM 存在 (0=未检测到 RTM, 1=检测到 RTM)

示例（终端模式）：

```
[B8 00 88 00 00 2A] <-----请求  
[BC 00 88 00 00 00 2A 01] <-----响应
```


词汇表

熟悉以下术语和首字母缩略词有助于对 Netra CT 900 服务器进行管理。

A

ATCA (Advanced Telecom Computing Architecture, 高级电信计算体系结构) 也称为 AdvancedTCA。用于下一代电信级通信设备的一系列行业标准规范。AdvancedTCA 引入了高速互连技术、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, 这种标准化使得刀片 (板) 和机箱 (机框) 的形状规格得以优化, 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)

任何能够支持机框管理器功能的机框管理卡。

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 分组交换服务。在 Netra CT 900 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。基接口需要由中间背板提供支持, 以便为所有节点板插槽和每个交换机插槽间的四个不同信号对提供路由 (在 Netra CT 900 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负荷间提供互连。

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

(光纤通道)

一条光纤通道由两排信号对组成（每条通道一共有八个信号对）。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组双对端口组成。

Fabric interface

(交换接口)

一种 Zone 2（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 个差动信号对（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节点板、光纤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，安装有自身各种卡和其他部件的服务器不是 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 Zone 1 (区域 1) 和 Zone 2 (区域 2) 中间背板连接器相连接。前板可以 (可选) 与 Zone 3 (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) 一种光纤通道连接, 进行连接时会使用端点间的全部八个差动信号对。
- 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) 在不中断系统操作的情况下对外设部件或其他部件进行连接和断开其连接。这项功能的设计中既考虑到了硬件, 也考虑到了软件。

I

- I²C** 内部集成电路总线。目前的 IPMB 使用多主控、两线串行总线作为基础。
- 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 提供多种辐射型 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, 用于为计算机系统的元素提供库存管理、监控、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)

一种机框范围的电气网络，用于板和中间背板上，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，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 3-16 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、财产保护以及操作连续性。NEBS 测试包括使设备接受多种振动应力、防火以及其他环境和质量等衡量标准方面的测试。有三种级别的 NEBS 法规遵从性，每一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。节点插槽仅适用于设计为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) 一种用于将外设部件连接至计算机的标准。它的运行频率为 20 - 33 MHz，可通过 124 管脚连接器一次传输 32 位数据或通过 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。

从技术上讲，PCI 不是总线，而是一种网桥或夹层。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。

physical address
(物理地址)

定义 FRU 的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) 一些公司组成的协会，该协会开发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)

Netra CT 900 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面引出。

rear transition module
(后部转换模块)

一种仅在后部接入式 Netra CT 900 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) 一些部件的集合, 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) 一种长度、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) 安全接地和连接至框架的接地回线, 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和 IPMB-0 之间传递消息, 为系统信息库提供接口, 并对事件消息作出响应。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 个节点板和其他交换机提供 10/100/1000BASE-T 以太网交换资源。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 个光纤通道。
- system** (系统) 可以包含一个或多个以下部件的管理实体: 节点和交换机、机框以及框架。

U

U 一种测量单位，等于 1.75 英寸（44.45 毫米）。

update channel interface

（更新通道接口）

也称为更新通道。一种 Zone 2（区域 2）接口，为两个板间的十个差分信号对提供连接。两个板间的这种直接连接可用于同步状态信息。在板上为更新通道实现的传输尚未定义。更新通道只能由一个供应商创建的两个功能类似的板使用。电子钥控用于确保在启用驱动程序之前，更新通道端点已映射了相匹配的传输协议。中间背板必须支持更新通道。板可以支持更新通道。

Z

- Zone 1（区域 1）** 沿着（为电源、管理和其他辅助功能分配的） ATCA 插槽的高度尺寸方向的线性空间。
- Zone 2（区域 2）** 沿着（为数据传输接口分配的） ATCA 插槽的高度尺寸方向的线性空间。
- Zone 3（区域 3）** 沿着（为用户定义的连接和/或与后部接入式系统的后部转换模块的互连所保留的） ATCA 插槽的高度尺寸方向的线性空间。

索引

A

- ATCA 机框, 5
- ATCA (Advanced Telecommunications Computing Architecture, 高级电信计算体系结构), 1

C

- CPLD
 - cpldtool 实用程序, 80
 - 重新编程, 99
- 重新初始化, 10
- 串行端口, 14

D

- 登录
 - 重新设置密码, 79
 - 出厂默认值, 79
- 登录, 机框管理卡, 14
- 电子钥控, 111
- 对机框管理器进行重新编程, 80

F

- FRU 信息, 63, 64, 66, 67
 - IPMI, 63
 - Sun, 64
- FRU (field-replaceable unit, 现场可更换单元), 2

G

- 固件
 - Open Boot PROM, 2
- 关闭
 - 节点板, 102
 - 手动正常, 101
- 管理任务, 11

H

- 后部转换卡, 3
- 环境 FRU, 64

I

- Internet 号码分配机构 (Internet Assigned Numbers Authority, IANA), 243
- IP 地址, 20
 - RMCP, 21
- IPMB
 - 插槽编号, 108
 - 逻辑插槽编号, 108
- IPMB 地址, 108, 110
 - 范围, 108
- IPMC, 2
- IPMI
 - FRU 信息, 63
 - FRU 信息布局, 63
 - LAN 接口, 10, 51
 - 概述, 5
- IPMI 命令, 52

J

- 基接口, 3
- 机框管理器, 2, 5, 7
 - CLI 命令, 105, 105 - 241
 - CLI 命令汇总, 54
 - IP 地址, 14
 - shelfman.conf 文件, 26
 - 备用, 10
 - 重新编程, 80
 - 功能, 7
 - 简介, 5
 - 接口选项, 10
 - 命令行界面, 52
 - 命令行界面 (Command-line interface, CLI), 2
 - 配置参数, 27
 - 配置文件, 21
 - 启动 CLI, 53
 - 切换, 8
 - 软件, 2
 - 调试级别, 117
 - 网络参数, 22
 - 详细级别, 40
- 机框管理卡, 2, 5, 7
 - 备用, 14
 - 登录, 14
 - 访问, 14
 - 复位, 78
 - 会话, 3
 - 活动, 14
 - 控制台, 14, 99 - 101
 - 说明, 2, 11
 - 以太网端口, 20
 - 用户帐户, 43
- 机框信息, 71
- 脚本
 - /etc/upgrade/step4hshm, 85
- 交换机板
 - 接入, 3
 - 以太网端口, 3
- 交换机卡, 99, 116
- 交换结构板
 - 说明, 2

节点板

- SAS 端口, 3
 - 串行端口, 3
 - 关闭, 102
 - 接入, 3
 - 控制台, 99
 - 另请参见节点板, 第三方说明, 2
 - 以太网端口, 3
- ## 节点板, 第三方, 3
- 说明, 2

K

- 可靠升级, 80
 - 方案, 87
 - 闪存分区, 81
 - 示例, 88
 - 实用程序, 83
 - 状态文件, 82
- 可靠升级过程, 80, 83
- 可靠升级实用程序, 87
- 控制的取消激活, 102, 103
- 控制台, 14, 99 - 101
- 控制台会话, 99, 116
- 扩展接口, 3

L

- 连接器
 - 机框报警面板, 14

M

- 密码, 44
- 命令
 - cli showstatus, 15, 100
 - console, 100
 - showcpustate, 58
 - switchover, 9
 - useradd, 43
- 命令行界面, 2

N

netfunction (NetFn), 243

O

OpenBoot PROM 固件, 2

OpenHPI, 44

 /etc/openhpi.conf, 44

 libipmdirect 参数, 45

 配置, 44

P

POST, 2

Q

强制切换, 9

切换, 9, 10

 强制, 9

 协同, 9

R

RMCP, 10, 20

 地址, 21

rupgrade_tool, 83, 84, 86

S

SAP (shelf alarm panel, 机框报警面板), 3

 Telco 报警, 3

 串行端口, 3

 接入, 3

 连接器, 14

SNMP, 44

 访问控制, 46

 更新 /etc/snmpd.conf 文件, 49

 设置陷阱, 48

SNMP 子代理, 45

SNMP 子代理配置文件, 44

SNMPv3 配置, 48

Solaris 操作系统, 2

Sun FRU 信息, 64

Sun OEM IPMI 命令, 52, 243

闪存, 80

闪存分区, 81

设置时钟, 41

时间服务器, 42

实用程序

 可靠升级, 83

T

Telco 报警, 7

telnet, 14

tip 程序, 14

U

U-Boot, 2, 15, 20

 变量, 20

 访问, 16

 环境变量, 16, 19

 恢复默认值, 77

USB

 接口, 9

 连接, 21

W

WDT (watchdog timer, 监视程序计时器), 85, 86

网络接口, 3

网络控制台, 101

文档, xix

文件系统

 重新初始化, 78

物理与逻辑插槽的对应关系, 11

X

- 系统管理器, 6
- 协同切换, 9
- 信号
 - Remote Healthy, 9, 10
 - Remote Presence, 9
 - 切换, 9

Y

- 以太网端口, 3, 20
- 硬件接口, 3, 4

映射

- 物理地址, 110
- 用户帐户, 11
- 远程管理控制协议 (Remote Management Control Protocol, RMCP), 51

Z

- 正常关闭, 101