



# Netra™ CT 900 サーバー 管理およびリファレンス マニュアル

---

Sun Microsystems, Inc.  
[www.sun.com](http://www.sun.com)

Part No. 820-0573-10  
2007 年 1 月, Revision A

コメントの送付: <http://www.sun.com/hwdocs/feedback>

Copyright 2007 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします) は、本書に記述されている技術に関する知的所有権を有しています。これら知的所有権には、<http://www.sun.com/patents> に掲載されているひとつまたは複数の米国特許、および米国ならびにその他の国におけるひとつまたは複数の特許または出願中の特許が含まれています。

本書およびそれに付属する製品は著作権法により保護されており、その使用、複製、頒布および逆コンパイルを制限するライセンスのもとにおいて頒布されます。サン・マイクロシステムズ株式会社による事前の許可なく、本製品および本書のいかなる部分も、いかなる方法によっても複製することが禁じられます。

本製品のフォント技術を含む第三者のソフトウェアは、著作権法により保護されており、提供者からライセンスを受けているものです。

本製品の一部は、カリフォルニア大学からライセンスされている Berkeley BSD システムに基づいていることがあります。UNIX は、X/Open Company Limited が独占的にライセンスしている米国ならびに他の国における登録商標です。

本製品は、株式会社モリサワからライセンス供与されたリュウミン L-KL (Ryumin-Light) および中ゴシック BBB (GothicBBB-Medium) のフォント・データを含んでいます。

本製品に含まれる HG 明朝 L と HG ゴシック B は、株式会社リコーがリョービマジクス株式会社からライセンス供与されたタイプフェースマスタをもとに作成されたものです。平成明朝体 W3 は、株式会社リコーが財団法人日本規格協会 文字フォント開発・普及センターからライセンス供与されたタイプフェースマスタをもとに作成されたものです。また、HG 明朝 L と HG ゴシック B の補助漢字部分は、平成明朝体 W3 の補助漢字を使用しています。なお、フォントとして無断複製することは禁止されています。

Sun、Sun Microsystems、Java、AnswerBook2、docs.sun.com、Netra は、米国およびその他の国における米国 Sun Microsystems 社の商標もしくは登録商標です。サンのロゴマークおよび Solaris は、米国 Sun Microsystems 社の登録商標です。

すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している同社の米国およびその他の国における商標または登録商標です。SPARC 商標が付いた製品は、米国 Sun Microsystems 社が開発したアーキテクチャーに基づくものです。

PICMG および PICMG のロゴマーク、AdvancedTCA および AdvancedTCA のロゴマークは、PCI Industrial Computers Manufacturers Group の登録商標です。

OPENLOOK、OpenBoot、JLE は、サン・マイクロシステムズ株式会社の登録商標です。

ATOK は、株式会社ジャストシステムの登録商標です。ATOK8 は、株式会社ジャストシステムの著作物であり、ATOK8 にかかる著作権その他の権利は、すべて株式会社ジャストシステムに帰属します。ATOK Server/ATOK12 は、株式会社ジャストシステムの著作物であり、ATOK Server/ATOK12 にかかる著作権その他の権利は、株式会社ジャストシステムおよび各権利者に帰属します。

本書で参照されている製品やサービスに関しては、該当する会社または組織に直接お問い合わせください。

OPEN LOOK および Sun™ Graphical User Interface は、米国 Sun Microsystems 社が自社のユーザーおよびライセンス実施権者向けに開発しました。米国 Sun Microsystems 社は、コンピュータ産業用のビジュアルまたはグラフィカル・ユーザーインタフェースの概念の研究開発における米国 Xerox 社の先駆者としての成果を認めるものです。米国 Sun Microsystems 社は米国 Xerox 社から Xerox Graphical User Interface の非独占的ライセンスを取得しており、このライセンスは米国 Sun Microsystems 社のライセンス実施権者にも適用されます。

U.S. Government Rights—Commercial use. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含みそれに限定されない、明示的であるか黙示的であるかを問わない、なんらの保証も行われぬものとします。

本書には、技術的な誤りまたは誤植のある可能性があります。また、本書に記載された情報には、定期的に変更が行われ、かかる変更は本書の最新版に反映されます。さらに、米国サンまたは日本サンは、本書に記載された製品またはプログラムを、予告なく改良または変更することがあります。

本製品が、外国為替および外国貿易管理法 (外為法) に定められる戦略物資等 (貨物または役務) に該当する場合、本製品を輸出または日本国外へ持ち出す際には、サン・マイクロシステムズ株式会社の事前の書面による承諾を得ることのほか、外為法および関連法規に基づく輸出手続き、また場合によっては、米国商務省または米国所轄官庁の許可を得ることが必要です。

原典: Netra CT 900 Server Administration and Reference Manual  
Part No: 819-1177-12  
Revision A



# 目次

---

はじめに xvii

## 1. 概要 1

Netra CT 900 サーバーのソフトウェア 1

シェルフマネージャーの概要 6

ATCA での Intelligent Platform Management の概要 6

シェルフマネージャーおよびシェルフ管理カード 8

シェルフマネージャーの機能 8

シェルフマネージャーのスイッチオーバー 9

スイッチオーバーの詳細 10

システム管理者インタフェースのオプション 12

システム管理作業 13

物理アドレスから論理スロットへのマッピング 13

## 2. システムの構成 15

シェルフ管理カードへのアクセス 15

U-Boot の設定 17

U-Boot インタフェース 18

U-Boot 環境変数 18

環境変数への値の割り当て 21

シェルフマネージャー用の環境変数の構成	22
シェルフ管理カードの Ethernet ポートの構成	23
一次 Ethernet インタフェースの使用	23
一次ネットワークインタフェースへの追加 IP アドレスの割り当て	23
RMCP アドレスの伝播	24
二次 Ethernet インタフェースの使用	25
冗長通信のためのデュアル USB ネットワークインタフェースの使用	25
デフォルトの ShMM ネットワークパラメータの変更	26
▼ デフォルトの ShMM ネットワークパラメータを変更する	27
シェルフマネージャー構成ファイルの設定	31
詳細レベルの概要	47
日付と時刻の設定	48
時刻サーバーからの日付および時刻の取得	49
シェルフ管理カードでのユーザーアカウントの設定	50
▼ RMCP アクセス用のユーザーアカウントを追加する	50
ユーザー名の制限	51
パスワード	51
シェルフマネージャーでの OpenHPI の構成	52
/etc/openhpi.conf ファイル	52
▼ /etc/openhpi.conf ファイルを変更する	53
/etc/snmpd.conf ファイル	53
アクセス制御	54
SNMPv3 構成	56
トラップおよび情報の宛先の設定	57
▼ /etc/snmpd.conf ファイルを更新する	57
3. システムの管理	59
IPMI LAN インタフェース	59
IPMI コマンド	60

シェルフマネージャーのコマンド行インタフェース	60
コマンド行インタフェースの開始	61
CLI コマンド	62
システムの監視	66
ボードおよび IPMC の情報の表示	67
FRU 情報の表示	71
IPMI FRU 情報の配置	72
環境 FRU	73
ブレード FRU	73
例	73
シェルフ情報の表示	80
例	80
シェルフマネージャーの再初期化	86
U-Boot 環境の再初期化	86
▼ U-Boot 環境を再初期化する	86
ファイルシステムの再初期化	87
ログインパスワードのリセット	88
シェルフ管理カードの再プログラミング	88
信頼できるファームウェアアップグレード処理	89
フラッシュのパーティション分割	90
/var/upgrade ファイルシステム	91
信頼できるアップグレード処理の状態ファイル	92
信頼できるアップグレードユーティリティ	92
信頼できるアップグレードユーティリティを使用する状況	98
信頼できるアップグレードの例	100
CPLD のプログラミング	110
▼ ShMM の CPLD イメージを再プログラミングする	110
ノードボードのコンソールへの接続	111

シェルフマネージャーとノードボード間のコンソールセッションの確立	112
▼ シェルフマネージャーからコンソールセッションを開始する	112
▼ コンソールセッションを終了する	113
ノードボードの手動による正常な停止	114
▼ ノードボードを停止する	114
A. シェルフマネージャーの CLI コマンド	117
activate	118
alarm	119
board	120
boardreset	122
busres	123
バス型の E-キーイング管理対象のリソースの状態表示	124
指定されたリソースの解放	125
指定されたリソースのロック/ロック解除	125
Bused Resource Control (Query) コマンドの送信	126
リソースの所有者の設定	127
Bused Resource Control (Bus Free) コマンドの送信	128
console	129
deactivate	130
debuglevel	130
exit quit	131
fans	132
flashupdate	133
fru	134
frucontrol	136
frudata	137
frudatar	139

- frudataw 140
- fruinfo 141
- getfanlevel 142
- getfruledstate 143
- gethysteresis 145
- getipmbstate 146
- getlanconfig 147
  - auth\_support 150
  - auth\_enables 151
  - ip 152
  - ip\_source 152
  - mac 153
  - subnet\_mask 154
  - ipv4\_hdr\_param 154
  - pri\_rmcp\_port 155
  - sec\_rmcp\_port 155
  - arp\_control 156
  - arp\_interval 156
  - dft\_gw\_ip 157
  - dft\_gw\_mac 157
  - backup\_gw\_ip 158
  - backup\_gw\_mac 158
  - community 159
  - destination\_count 159
  - destination\_type 160
  - destination\_address 161
- getpefconfig 162
  - control 164

action\_control 165  
startup\_delay 166  
alert\_startup\_delay 166  
event\_filter\_count 167  
event\_filter 167  
event\_filter\_data1 168  
alert\_policy\_count 169  
alert\_policy 170  
system\_guid 170  
alert\_string\_count 171  
alert\_string\_key 172  
alert\_string 172  
oem\_filter\_count 173  
oem\_filter 174  
getsensoreventenable 175  
getthreshold|threshold 176  
help 178  
ipmc 181  
localaddress 183  
minfanlevel 183  
sel 184  
sensor 188  
sensordata 192  
sensorread 194  
session 195  
setextracted 196  
setfanlevel 197  
setfruledstate 198

- sethysteresis 199
- setipmbstate 200
- setlanconfig 201
  - auth\_enables 203
  - ip 203
  - subnet\_mask 204
  - ipv4\_hdr\_param 204
  - arp\_control 205
  - arp\_interval 206
  - dft\_gw\_ip 206
  - backup\_gw\_ip 207
  - community 207
  - destination\_type 208
  - destination\_address 208
- setlocked 209
- setpefconfig 210
  - control 212
  - action\_control 213
  - startup\_delay 213
  - alert\_startup\_delay 214
  - event\_filter 214
  - event\_filter\_data1 215
  - alert\_policy 216
  - system\_guid 217
  - alert\_string\_key 218
  - alert\_string 218
  - oem\_filter 219
- setsensoreventenable 220

- setthreshold 221
- shelf 223
  - シェルフ FRU 情報の表示 224
  - 最大外部使用可能電流の変更 230
  - 最小予測動作電圧の変更 232
  - シェルフマネージャー制御による起動フラグの変更 234
  - シェルフマネージャー制御による停止フラグの変更 239
  - 最大 FRU 電源容量の変更 241
  - 次の電源投入までの遅延の変更 242
  - FRU 起動準備許容時間の変更 244
  - FRU の起動記述子と電源記述子の順序変更 245
  - シェルフ FRU 情報のリフレッシュ 247
  - シェルフ FRU 情報ストレージデバイスの更新 249
- shelfaddress 249
- shmstatus 250
- showhost 251
- showunhealthy 252
- switchover 252
- terminate 253
- user 253
  - ユーザー情報の表示 254
  - 新しいユーザーの追加 255
  - ユーザーの削除 256
  - ユーザーの有効化および無効化 257
  - ユーザー名の変更 258
  - ユーザーのパスワードの変更 259
  - 指定ユーザーおよび指定チャネルのチャネルアクセス設定の変更 260
- version 261

B. Sun の OEM IPMI コマンド	263
Get Version	264
Set Boot Page	266
Get Boot Page	267
Set Front Panel Reset Button State	268
Get Front Panel Reset Button State	269
Set Ethernet Force Front Bit	270
Get Ethernet Force Front Bit	271
Get RTM Status	272



# 図目次

---

図 1-1	Netra CT サーバーのソフトウェアインタフェースおよびハードウェアインタフェースの論理図	5
図 1-2	ATCA シェルフの例	7
図 1-3	シェルフマネージャーのスイッチオーバー信号	9
図 2-1	シェルフアラームパネルのコネクタ	16
図 3-1	IPMI FRU 情報の配置	72



# 表目次

---

表 1-1	システム管理者のための Netra CT サーバーソフトウェア	2
表 1-2	Netra CT 900 システムボードへの接続手段	3
表 1-3	スイッチオーバーをサポートするハードウェア信号およびインタフェース	10
表 1-4	物理アドレスから論理スロットへのマッピング	13
表 2-1	デフォルトの U-Boot 環境変数	18
表 2-2	シェルフマネージャー構成パラメータ	32
表 3-1	Sun の OEM IPMI コマンド	60
表 3-2	シェルフマネージャーの CLI コマンドの概要	62
表 3-3	16M バイトのフラッシュパーティション (reliable_upgrade=y)	90
表 3-4	シェルフマネージャーの CLI コンソール関連のコマンド	112
表 A-1	getlanconfig の LAN 構成パラメータ	148
表 A-2	PEF 構成パラメータ	162
表 A-3	setlanconfig の LAN 構成パラメータ	201
表 A-4	setpefconf の PEF 構成パラメータ	210
表 A-5	shelf コマンドのパラメータ	225
表 B-1	Sun の OEM IPMI コマンド	263



# はじめに

---

『Netra CT 900 サーバー管理およびリファレンスマニュアル』では、Netra™ CT 900 サーバーのシステム管理者向けの構成および管理に関する情報について説明します。また、シェルフマネージャーおよび IPMI コマンドリファレンスに関する情報も記載しています。

このマニュアルは、UNIX の® コマンドとネットワーク、PICMG® 3.x AdvancedTCA® Base Specification、および Intelligent Platform Management Interface (IPMI) を理解していることを前提としています。

---

## マニュアルの構成

第 1 章では、Netra CT 900 サーバーソフトウェアの概要について説明します。

第 2 章では、システムの構成に関する情報について説明します。

第 3 章では、システムの管理方法について説明します。

付録 A では、シェルフマネージャーコマンド行インタフェース (CLI) の各コマンドの構文および使用法について説明します。

付録 B では、Sun 固有の OEM 定義の Intelligent Platform Management Interface (IPMI) コマンドについて説明します。

---

# UNIX コマンド

このマニュアルには、システムの停止、システムの起動、およびデバイスの構成などに使用する基本的な UNIX コマンドと操作手順に関する説明は含まれていない可能性があります。これらについては、以下を参照してください。

- 使用しているシステムに付属のソフトウェアマニュアル
- 下記にある Solaris™ オペレーティングシステム (Solaris OS) のマニュアル  
<http://docs.sun.com>

---

## シェルプロンプトについて

シェル	プロンプト
UNIX の C シェル	<i>machine_name%</i>
UNIX の Bourne シェルと Korn シェル	\$
スーパーユーザー (シェルの種類を問わない)	#

# 書体と記号について

書体または記号*	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例。	.login ファイルを編集します。 ls -a を実行します。 % You have mail.
<b>AaBbCc123</b>	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して表します。	% <b>su</b> Password:
<i>AaBbCc123</i>	コマンド行の可変部分。実際の名前や値と置き換えてください。	rm <i>filename</i> と入力します。
『 』	参照する書名を示します。	『Solaris ユーザーマニュアル』
「 」	参照する章、節、または、強調する語を示します。	第 6 章「データの管理」を参照。 この操作ができるのは「スーパーユーザー」だけです。
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	% <b>grep</b> `^#define` \ <b>XV_VERSION_STRING</b> '

\* 使用しているブラウザにより、これらの設定と異なって表示される場合があります。

## 関連マニュアル

次の表に、Netra CT 900 サーバーのマニュアルを示します。『Important Safety Information for Sun Hardware Systems』を除き、この表に示されているマニュアルはすべて、次の URL から入手できます。

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

タイトル	Part No.
『Netra CT 900 Server Product Notes』	819-1180
『Netra CT 900 サーバー概要』	820-0557
『Netra CT 900 サーバー設置マニュアル』	820-0565
『Netra CT 900 Server Service Manual』	819-1176

タイトル	Part No.
『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』 (印刷版のみ)	816-7190-10

追加の情報については、Solaris OS、OpenBoot™ PROM ファームウェア、Netra CP3010 ボード、Netra CP3020 ボード、および Netra CP3060 ボードの各製品マニュアルの参照が必要になる場合があります。

## マニュアル、サポート、およびトレーニング

Sun のサービス	URL
マニュアル	<a href="http://jp.sun.com/documentation/">http://jp.sun.com/documentation/</a>
サポート	<a href="http://jp.sun.com/support/">http://jp.sun.com/support/</a>
トレーニング	<a href="http://jp.sun.com/training/">http://jp.sun.com/training/</a>

## Sun の技術サポート

このマニュアルに記載されていない、この製品に関する技術的な問い合わせについては、次の URL にアクセスしてください。

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

---

## コメントをお寄せください

マニュアルの品質改善のため、お客様からのご意見およびご要望をお待ちしております。コメントは下記よりお送りください。

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

ご意見をお寄せいただく際には、下記のタイトルと Part No. を記載してください。

『Netra CT 900 サーバー管理およびリファレンスマニュアル』、Part No. 820-0573-10



# 第 1 章

---

## 概要

---

この章は、次のセクションで構成されています。

- 1 ページの「Netra CT 900 サーバーのソフトウェア」
- 6 ページの「シェルフマネージャーの概要」
- 13 ページの「システム管理作業」

---

## Netra CT 900 サーバーのソフトウェア

Netra CT 900 サーバーのソフトウェアには、次のものがあります。

- シェルフマネージャー
- オペレーティングシステムおよびアプリケーション
- ファームウェア

---

**注** – Advanced Telecom Computing Architecture® (ATCA) では、電気通信業の標準的な慣行に合わせて「シェルフ」という用語を採用しています。従来は、基本的に同じ意味の「シャーシ」という用語が使用されていました。

---

表 1-1 にソフトウェアの説明を示し、図 1-1 にハードウェアを含む論理図を示します。

表 1-1 システム管理者のための Netra CT サーバーソフトウェア

カテゴリ	名前	説明
シェルフ管理	IPM Sentry Shelf Manager	シェルフマネージャーソフトウェアは、シェルフ管理カード (ShMM) 上で動作し、出荷時にインストールされています。サーバーを管理するために、IPMI への RMCP (Remote Management Control Protocol) および CLI によるアクセスを提供します。
	コマンド行インタフェース (CLI)	CLI は、シェルフマネージャーへのオンボードのユーザーインタフェースです。
オペレーティングシステムおよびアプリケーション	Solaris オペレーティングシステム (Solaris OS)	Solaris OS は、Netra CP3010、Netra CP3020、Netra CP3060 などの、Sun がサポートする ATCA 準拠のノードボード上で動作します。Solaris 10 は、オプションで、Netra ノードボードにプリインストールされます。Solaris 10 およびその他のバージョンの Solaris OS は、ユーザーがダウンロードしてインストールできます。
	Monta Vista Carrier Grade Linux OS	Netra CP3020 では、Monta Vista Carrier Grade Linux OS を実行することもできます。
ファームウェア	OpenBoot PROM ファームウェア	Netra CP3010 ボードなどの、Sun がサポートするシステムボード上のファームウェアです。起動を制御します。これには診断機能もあります。
	U-Boot	シェルフ管理カード上のファームウェアです。電源投入時自己診断 (POST) を実行し、シェルフ管理カードソフトウェアの起動を制御します。
	Intelligent Platform Management Controller (IPMC)	システム管理コントローラのファームウェアです。Netra CP3010 などの Sun がサポートするノードボード上で、IPMI コントローラを介した通信を可能にします。

Netra CT 900 サーバーには 2 枚のシェルフ管理カード (ShMM) があり、特定のハードウェアおよびソフトウェアイベントに対して、アクティブシェルフ管理カードからスタンバイシェルフ管理カードへの、シェルフ管理カードフェイルオーバーを提供します。「アクティブ」シェルフ管理カードは、ミッドプレーンに接続される多くのコンポーネントのシステムレベルの構成、管理、および制御に使用されます。「スタンバイ」シェルフ管理カードは、アクティブシェルフ管理カードに冗長性とフェイルオーバー機能を提供します。

スイッチングファブリックボードは、シェルフ管理カードとノードボードを内部接続するもので、背面には外部接続用の Ethernet ポートを備えています。

Netra CP3010 ノードボードは、ディスクなどの周辺機器を受け入れ、所有します。このノードボードは、ユーザーアプリケーションの実行も行います。Netra CT 900 サーバーでは、各ノードボードは専用のオペレーティングシステムのコピーを実行するため、それぞれがサーバーと見なされます。システムは、シェルフ管理カード、

ノードボード、スイッチングファブリックボード、およびその他の FRU (現場交換可能ユニット) で構成されます。

**注** – このマニュアルでは、「ノードボード」という用語は、特に指定のないかぎり、Netra CP3010 ボードなどの Sun がサポートする ATCA CPU ボードを指します。

Netra CT 900 サーバーでは、PICMG<sup>®</sup> 3.x に準拠する他社製の ATCA ノードボードを使用できます。このようなボードでは、Solaris OS が動作するとはかぎらず、また Netra CT 900 サーバーのシステム管理ソフトウェアも動作しません。そのため、Netra ノードボードと同じ程度まで管理することはできません。

表 1-2 に、さまざまなボードへの接続手段をまとめます。シェルフ管理カードは、同時に 22 セッション (Tip 接続 1、Telnet 接続 21) をサポートします。

表 1-2 Netra CT 900 システムボードへの接続手段

ボード	接続手段
シェルフアラームパネル (SAP)	フロントパネルには、次のポートがあります。 <ul style="list-style-type: none"><li>• RJ-45 DTE コネクタ付きのシリアル (RS-232) ポート 2 つ。シリアル 1 ポートは、上部のシェルフ管理カード (ShMM1)、つまりデフォルトの「アクティブ」カード用のコンソール接続です。シリアル 2 ポートは、下部のシェルフ管理カード (ShMM2)、つまりデフォルトの「バックアップ」カード用のコンソール接続です。</li><li>• Telco アラームコネクタ (DB-15)</li></ul>
スイッチボード (スロット 7 および 8)	フロントパネル上にある Telnet 接続用の複数の Ethernet ポート。 注 - これらのポートに背面から接続するには、背面切り替えモジュール (Rear Transition Module、RTM) が必要です。背面または前面のポートコネクタはどちらも使用できますが、同時に両方を使用することはできません。両方のポートにケーブルを接続した場合は、前面のポートのみがアクティブになります。
ノードボード (Sun がサポートする CPU ボード) (スロット 1 ~ 6 および 9 ~ 14)	Netra CP3010 ボードでは、フロントパネルに次のポートがあります。 <ul style="list-style-type: none"><li>• Tip または ASCII 端末接続用のシリアルポート (コンソール) 2 つ</li><li>• Telnet 接続用の Ethernet ポート 2 つ</li><li>• 4X Serial Attached SCSI (SAS) ポートコネクタ 1 つ</li></ul> 注 - これらのポートに背面から接続するには、RTM が必要です。背面または前面のポートコネクタはどちらも使用できますが、同時に両方を使用することはできません。両方のポートにケーブルを接続した場合は、前面のポートのみがアクティブになります。 Netra CP3020 および CP3060 ボードでは、フロントパネルに次のポートがあります。 <ul style="list-style-type: none"><li>• Tip または ASCII 端末接続用のシリアルポート (コンソール) 1 つ</li><li>• Telnet 接続用の Ethernet ポート 2 つ</li></ul> 詳細は、特定のノードボードに対応する Netra ATCA CPU ボードのマニュアルを参照してください。

表 1-2      Netra CT 900 システムボードへの接続手段 (続き)

ボード	接続手段
他社製のノードボード (スロット 1 ～ 6 および 9 ～ 14)	他社製のボードによって異なります。

ハードウェアインタフェースには、**Intelligent Platform Management Interface (IPMI)**、ベースインタフェースと拡張インタフェースのほか、シェルフ管理カード上、ノードボード上、およびスイッチングファブリックボード上のネットワークインタフェースがあります。

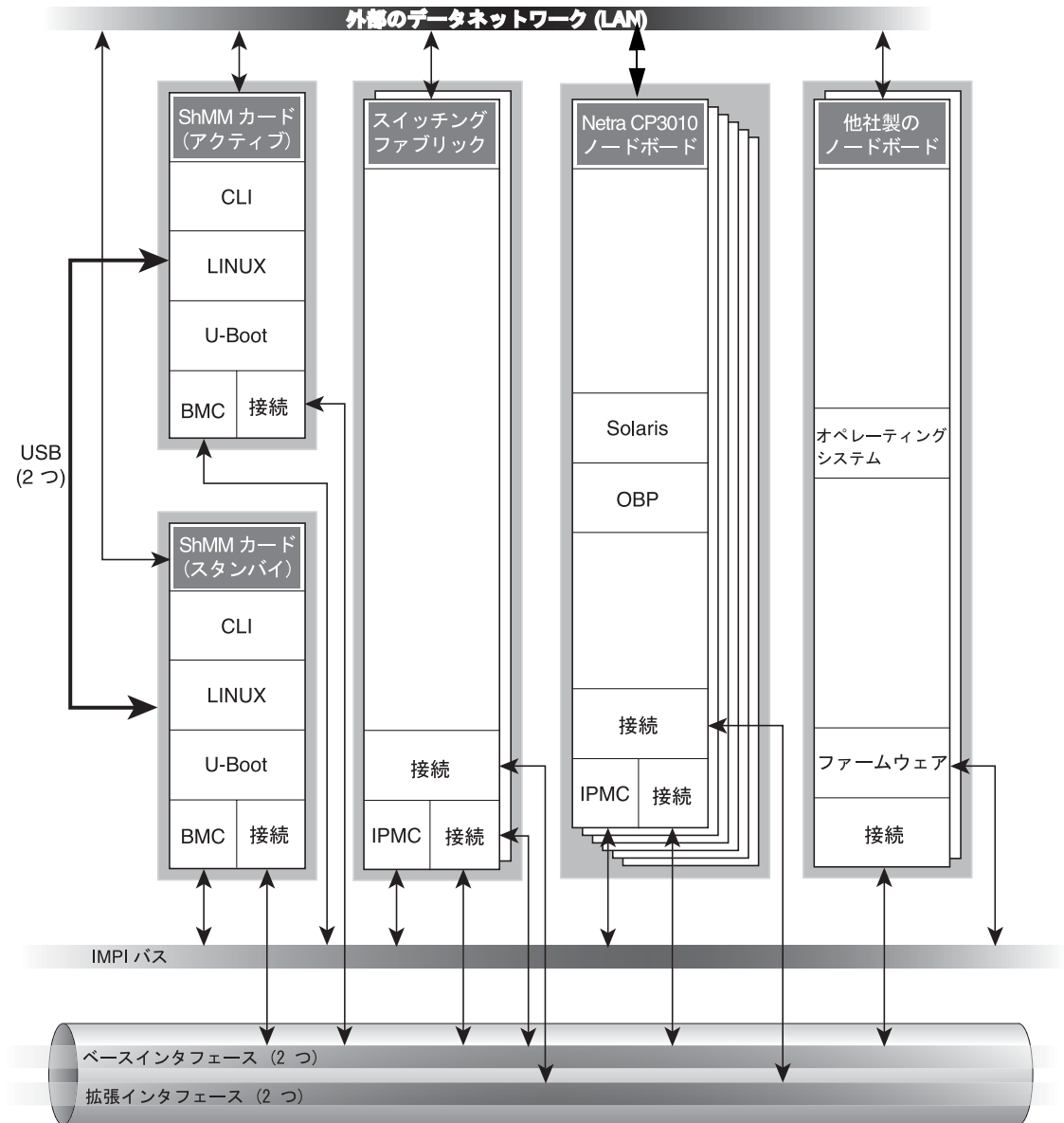


図 1-1 Netra CT サーバーのソフトウェアインタフェースおよびハードウェアインタフェースの論理図

---

## シェルフマネージャーの概要

シェルフマネージャーは、ATCA 製品のためのシェルフレベルの管理ソリューションです。シェルフ管理カードは、ATCA シェルフ内でシェルフマネージャーを動作させるために必要なハードウェアを提供します。この概要説明では、ATCA コンテキストで使用されるすべてのシェルフ管理キャリアに共通する、シェルフマネージャーおよびシェルフ管理カードの特徴を中心に説明します。

## ATCA での Intelligent Platform Management の概要

シェルフマネージャーおよびシェルフ管理カードは、FRU の動的配置とサービス可用性の最大化に重点を置く、ATCA のようなモジュラ型プラットフォーム用に設計された、Intelligent Platform Management (IPM) のビルディングブロックです。IPMI 仕様は、このようなプラットフォームを管理するための堅固な基盤を提供しますが、これらを十分にサポートするために大幅な拡張が必要になります。ATCA 仕様の PICMG 3.0 では、IPMI に必要な拡張について定義しています。

図 1-1 に、サンプルの ATCA シェルフの論理要素を、ATCA 仕様の用語を使用して示します。

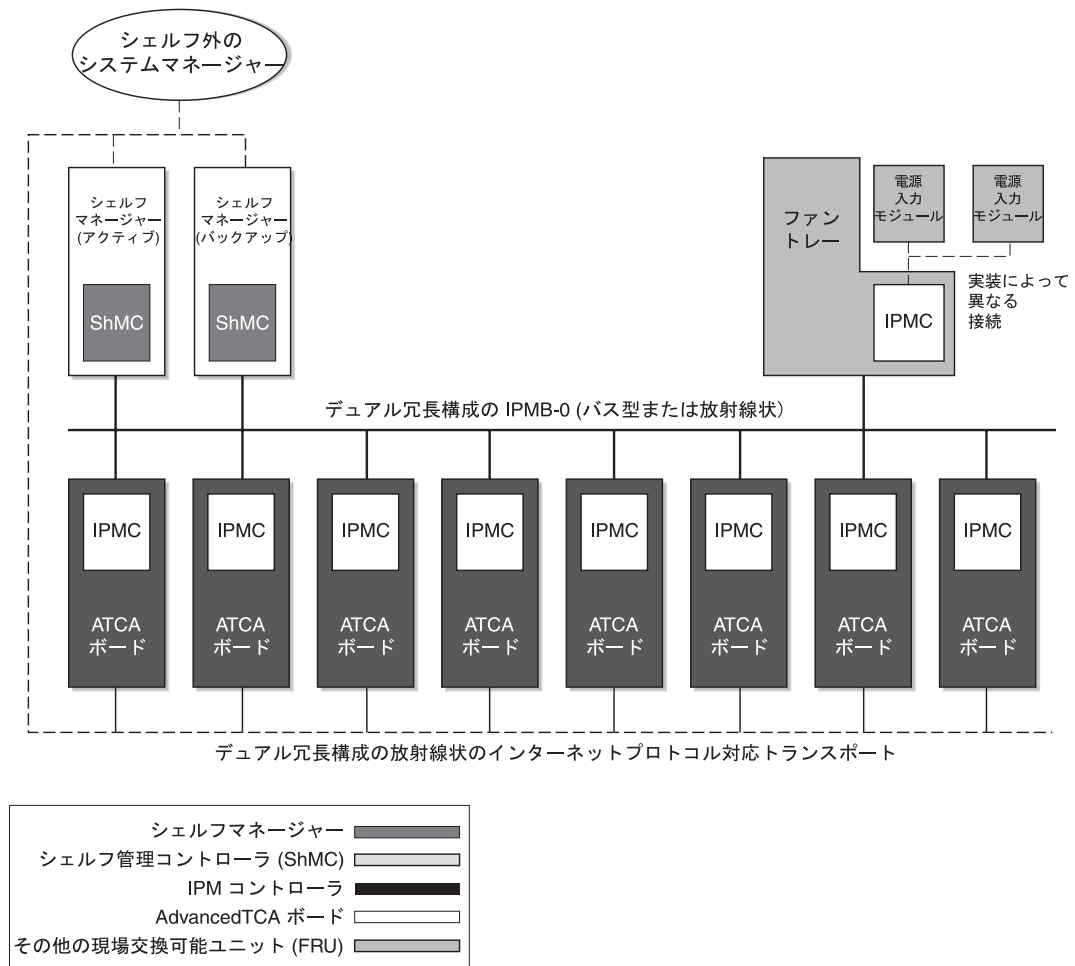


図 1-2 ATCA シェルフの例

総合的なシステムマネージャー (通常はシェルフ外) は、複数のシェルフのアクティビティを調整できます。通常、システムマネージャーは **Ethernet** またはシリアルインタフェースを介して各シェルフマネージャーと通信します。

図 1-2 に、管理の 3 つのレベルであるボード、シェルフ、およびシステムを示します。次のセクションでは、ATCA 準拠のシェルフ管理機能とシェルフ管理コントローラ (ShMC) を実装する、シェルフマネージャーソフトウェアおよびシェルフ管理カードについて説明します。

# シェルフマネージャーおよびシェルフ管理カード

ATCA のシェルフマネージャー要件に適合するシェルフマネージャーには、次に示す 2 つの主要な役割があります。

- FRU の取り付け状況とシェルフの共通インフラストラクチャー (特に電源、冷却、およびインターコネクト用リソースとその使用法) を管理および追跡します。シェルフ内のこのような管理および追跡は、主にシェルフマネージャーと IPM コントローラが Intelligent Platform Management Bus 0 (IPMB-0) を介して対話することによって行われます。
- 総合的なシステムマネージャーを、システムマネージャーインタフェースを使用して管理および追跡に組み込むことを可能にします。通常、これは Ethernet を介して実装されます。

シェルフマネージャーソフトウェアの大部分は、シェルフ電源の投入または切断、FRU の受け入れまたは切り離しなどの日常的に行われる作業に専念します。これには、電源やインターコネクトリソースの割り当てのネゴシエーションも含まれます。また、シェルフ内で例外が発生した場合には、シェルフマネージャーは直接的な対処を行うことができます。たとえば、シェルフマネージャーは温度の例外に対応してファンレベルを上げることができ、この対処が不十分だった場合には、シェルフ内の熱負荷を低減するために FRU の電源切断を開始することもできます。

## シェルフマネージャーの機能

シェルフマネージャーソフトウェアには、次のような機能があります。

- シェルフに適したキャリアボード上に取り付けられる、小型の SO-DIMM フォームファクタモジュールであるシェルフ管理カード上で動作します。
- ATCA 仕様に準拠しています。
- ATCA で規定されたデュアル冗長構成の Intelligent Platform Management Bus (IPMB) を介して、シェルフ内のアクティビティを監視します。
- 温度、電圧などの例外を示す、シェルフ内のインテリジェント FRU から送信されたイベントを受け入れてログに記録し、構成可能な IPMI プラットフォームイベントフィルタに基づいてシェルフ外にアラートを送信します。
- 管理のための可視性を完全に維持しながら、FRU (現場交換可能ユニット) のホットスワップをサポートします。
- シェルフ管理に実装されたドライ接点リレーによる、標準 Telco アラームインフラストラクチャーへのインタフェースとして機能します。
- 高可用性 (HA) を実現する、冗長シェルフマネージャーインスタンスをサポートします。
- ウォッチドッグタイマーを統合し、定期的なストロークがない場合にシェルフ管理カードをリセットします。このリセットで、バックアップシェルフ管理カードへのスイッチオーバーが自動的に発生するように構成できます。

- イベントのタイムスタンプ用に、電池でバックアップされるリアルタイムクロックを内蔵しています。
- RMCP (ATCA に必須) や CLI を含む、Ethernet を介してアクセス可能なシェルフ外インタフェースを豊富に実装しています。

## シェルフマネージャのスイッチオーバー

シェルフマネージャは、最大限の可用性を実現するために、アクティブインスタンスとバックアップインスタンスで構成することができます。図 1-3 に、両方のインスタンスからシステムマネージャにアクセス可能な一方で、常にアクティブインスタンスのみが対話を行う仕組みを示します。これと同様に、IPMB-0 を介してシェルフ内に装着された IPM コントローラと通信を行うのは、アクティブインスタンスのみです。2つのインスタンスは TCP/IP を介して相互に通信し、アクティブインスタンスは状態の増分更新情報をバックアップインスタンスに送信します。そのため、必要な場合に、バックアップはアクティブの役割をただちに引き継ぐことができます。

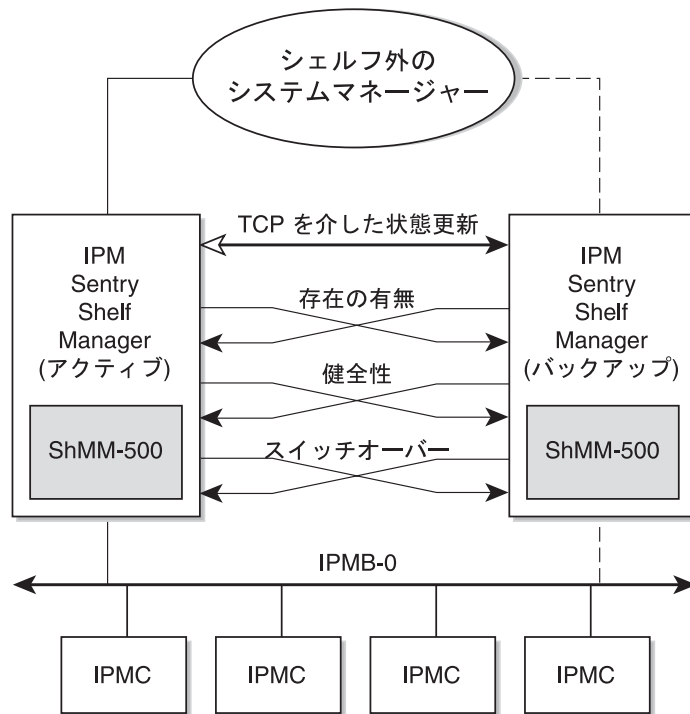


図 1-3 シェルフマネージャのスイッチオーバー信号

表 1-3 に、信号と説明を示します。

表 1-3      スイッチオーバーをサポートするハードウェア信号およびインタフェース

ハードウェア	説明
USB インタフェース	シェルフ管理カード間の主インタフェースで、ハートビートおよび状態同期化のための情報の送信に使用されます。シェルフ管理カードは両方とも、同じ FRU (特定のファントレイ、特定のスロット内のノードボードなど) を、同じ状態 (電源投入済みなど) として認識している必要があります。
#SWITCHOVER	バックアップインスタンスは、必要に応じてスイッチオーバーを強制できます。
#PRSNT	この信号は、シェルフ管理カードが存在することを示します。
#HEALTHY	この信号は、ハードウェアとソフトウェアの両方を含む、シェルフ管理カードの全体的な健全性を示します。

## スイッチオーバーの詳細

アクティブシェルフマネージャーは、IPMB 上で ShMC デバイス (アドレス 20h) を公開し、IPMB および IPM コントローラを管理し、RMCP およびその他のシェルフ外インタフェースを介してシステムマネージャーと対話します。バックアップシェルフマネージャーとの間で開かれた TCP 接続を維持します。また、管理対象オブジェクトの状態に関するすべての変更を、バックアップシェルフマネージャーに通知します。

バックアップシェルフマネージャーが、IPMB 上で ShMC を公開すること、IPMB および IPM コントローラをアクティブに管理すること、シェルフ外インタフェースを介してシステムマネージャーと対話することはありません (後述する 1 つの例外を除く)。その代わりに、管理対象オブジェクトの状態を自身の揮発性および非揮発性のメモリー内に保持し、この状態をアクティブシェルフマネージャーからの指示に従って更新します。

バックアップシェルフマネージャーは、スイッチオーバーの結果としてアクティブになることができます。次に示すように、2 種類のスイッチオーバーが定義されています。

- 協調的スイッチオーバー — アクティブシェルフマネージャーとバックアップシェルフマネージャーが、アクティブからバックアップシェルフマネージャーへの役割の移譲についてネゴシエーションを行います。このモードは、アクティブまたはバックアップシェルフマネージャー上で、CLI の `switchover` コマンドを実行することによってサポートされます。
- 強制的スイッチオーバー — バックアップシェルフマネージャーが、アクティブシェルフマネージャーが動作していないまたは正常でないと判断して、強制的にアクティブシェルフマネージャーの役割を引き継ぎます。

低レベル信号の **Remote Healthy** または **Remote Presence** が停止すると、バックアップシェルフマネージャーは、アクティブシェルフマネージャーが切り離されたと認識します。**Remote Presence** 信号は、ピアシェルフマネージャーの存在を監視するもので、この信号の停止は、ピアシェルフマネージャーをホストするボードがシェルフから取り外されたことを意味します。**Remote Healthy** 信号は、初期化処理中にピアシェルフマネージャーによって設定されるもので、この信号の停止は、遠隔のシェルフマネージャーが正常ではなくなった (通常、電源が切られたかリセットされた) ことを意味します。

バックアップシェルフマネージャーからの何らかの対処が必要になる別の状況として、シェルフマネージャー間の **TCP** 接続が閉じた場合が挙げられます。これは、2つのシェルフマネージャー間の通信リンクが切断された場合、アクティブシェルフマネージャーのシェルフ管理プロセスが自発的または非自発的に停止した場合、またはソフトウェア例外が発生した場合に発生します。この接続では **TCP** の **keepalive** オプションが有効になっているため、アクティブシェルフ管理カードの電源が切られるかリセットされると、すぐに接続が閉じます。

シェルフマネージャーが停止した場合には、**Remote Healthy** 信号が停止するより先に、**TCP** 接続が閉じることがあります。**TCP** 接続が閉じた理由を判断するため、バックアップシェルフマネージャーはただちに **Remote Healthy** 信号の状態を取得し、この信号がまだアクティブであれば、しばらく待機したあとでもう一度取得を行います。**Remote Healthy** 信号が最終的に停止した場合には、バックアップシェルフマネージャーはアクティブシェルフマネージャーが動作していないと結論付けて、スイッチオーバーを開始します。

**Remote Healthy** 信号がアクティブである場合には、バックアップシェルフマネージャーは、シェルフマネージャー間の通信リンクが切断されたと結論付けます。この場合はスイッチオーバーは開始されず、代わりにバックアップシェルフマネージャーは自身の再初期化を繰り返して、通信リンクが復元されるまでアクティブシェルフマネージャーとの接続の確立を試みます。シェルフ管理カードを再起動し、再起動後にシェルフマネージャーが自動的に再起動されることによって、再初期化が完了します。シェルフマネージャー内の特別なロジックによって、ピアシェルフマネージャーがすでにアクティブである場合には、シェルフマネージャーは起動時にアクティブになる試行をしないことが保証されています。

シェルフマネージャーは、ウォッチドッグタイマーを使用して、永久ループまたはその他のソフトウェアバグによる応答不能を防ぎます。アクティブシェルフマネージャー上のウォッチドッグタイマーがタイムアウトすると、そのシェルフ管理カードはリセットされ、これによってバックアップシェルフ管理カードで **Remote Healthy** 信号が停止して、スイッチオーバーが発生します。

スイッチオーバー後、アクティブになったシェルフマネージャーは再初期化を行なって、キャッシュ内の状態情報を有効にし、**IPMB** 上の **IPM** コントローラから必要な情報を収集します。このアクティブシェルフマネージャーは、次に **IPMB** 上で **ShMC** デバイス (アドレス 20h) を公開し、その **IP** アドレスが以前のアクティブシェルフマネージャーとシステムマネージャー間で **RMCP** およびその他のシェルフ外対話に使用されていたものであると見なします。**RMCP** セッション情報は、アクティブシェ

セルフマネージャーからバックアップセルフマネージャーに伝達されるため、RMCP セッションはスイッチオーバーが発生しても継続できます。RMCP を使用するシステムマネージャーに対しては、スイッチオーバーは影響しません。

スイッチオーバー後は、以前アクティブだったセルフマネージャーを取り外すことも、それ自体をバックアップセルフマネージャーとして再初期化することもできます。バックアップセルフマネージャーとして再初期化するには、以前アクティブだったセルフ管理カード上でオペレーティングシステムを再起動する必要があります。

---

## システム管理者インタフェースのオプション

セルフマネージャーのもう 1 つの主要サブシステムは、システム管理者インタフェースを実装します。システム管理者とは、オペレーションセンター内の人間のオペレータだけでなく、ソフトウェアも含めることのできる論理的な概念です。セルフマネージャーは、次に示す 2 つのシステム管理者インタフェースオプションを提供します。これらは、異なる機構によってセルフ関連の同種の情報および制御機能にアクセスする手段を提供します。

- IPMI ローカルエリアネットワーク (LAN) インタフェース
- コマンド行インタフェース (CLI)

IPMI LAN インタフェースを使用すると、個別に実装されたセルフ製品間の相互運用性を最大限まで高めることができます。このインタフェースは ATCA 仕様では必須とされており、RMCP を介したセルフマネージャーとの IPMI メッセージングをサポートします。セルフとの通信に RMCP を使用すると、システム管理者は ATCA 準拠の任意のセルフマネージャーと対話できるはずです。この低いレベルのインタフェースでは、セルフマネージャーをプロキシとして使用して、システム管理者がセルフ内の IPM コントローラに IPMI コマンドを実行する機能など、セルフの IPMI 関連の側面へのアクセスを提供します。

RMCP は、LAN を介して IPMI コントローラに接続するための標準的なネットワークインタフェースで、IPMI 1.5 仕様で定義されています。

CLI は、物理シリアル接続または Telnet 接続のいずれかを介してセルフマネージャーに実行できる、広範囲にわたるテキストコマンドを提供します。

# システム管理作業

通常、Netra CT 900 サーバシステムの管理には、設置、構成、および管理の各作業が含まれます。

Solaris ユーザーアカウントの追加など、Netra CT 900 サーバ上で行う Solaris OS の管理は、ノードボードにログインして実行します。Netra CT 900 サーバの管理は、シェルフ管理カードにログインし、シェルフ管理カードの CLI を使用して行います。シェルフ管理カードは、構成および管理のために使用する場合は、Netra CT 900 サーバ内の単一のエントリポイントとして使用できます。

システム管理作業については、以降の各章で説明します。

## 物理アドレスから論理スロットへのマッピング

前面側から Netra CT 900 サーバを見た場合、物理スロットには左から右の順に連番が付いています。表 1-4 に、物理スロットから論理スロットへのマッピングおよびアドレスを示します。

表 1-4 物理アドレスから論理スロットへのマッピング

物理スロット	1	2	3	4	5	6	7	8	9	10	11	12	13	14	ShMM #1	ShMM #2
論理スロット	13	11	9	7	5	3	1	2	4	6	8	10	12	14	なし	なし
HW アドレス (16 進数)	4D	4B	49	47	45	43	41	42	44	46	48	4A	4C	4E	8	9
IPMB アドレス (16 進数)	9A	96	92	8E	8A	86	82	84	88	8C	90	94	98	9C	10	12



## 第2章

---

# システムの構成

---

この章では、使用する Netra CT 900 サーバーおよび Netra ATCA ノードボードに、Solaris オペレーティングシステムおよび必要なパッチがすでにインストールされていることを前提としています。

Netra CT 900 システムの構成には、主にアクティブシェルフ管理カードのコマンド行インタフェース (CLI) を使用します。アクティブシェルフ管理カードの CLI では、ノードボード、スイッチングファブリックボード、シェルフ管理カード、電源入力モジュール (PEM)、ファントレーなどを含むシステムレベルの構成、管理、および制御を実行できます。シェルフ管理カードの CLI インタフェースは、ローカルでも遠隔でも使用できます。

この章は、次のセクションで構成されています。

- 15 ページの「シェルフ管理カードへのアクセス」
- 17 ページの「U-Boot の設定」
- 23 ページの「シェルフ管理カードの Ethernet ポートの構成」
- 26 ページの「デフォルトの ShMM ネットワークパラメータの変更」
- 31 ページの「シェルフマネージャー構成ファイルの設定」
- 48 ページの「日付と時刻の設定」
- 50 ページの「シェルフ管理カードでのユーザーアカウントの設定」

---

## シェルフ管理カードへのアクセス

シェルフマネージャーは、Linux の特殊化された実装の最上位で動作します。Linux の最下層は U-Boot ファームウェア監視です。いずれかのシェルフ管理カード (ShMM) にはじめてアクセスするときは、ASCII 端末または Tip プログラムを使用し、シリアルポート (コンソール) を介してアクセスする必要があります。各シェルフ管理カードは、同時に複数のセッション (Tip 接続および Telnet 接続) をサポートします。アクティブシェルフ管理カードのデフォルトの TCP/IP アドレスは、192.168.0.2 です。

シリアルポートを介してシェルフ管理カードに接続する際は、シリアル端末またはエミュレータを、シェルフアラームパネル (SAP) の前面にある 2 つのシリアルポートのいずれかに接続します。シリアル 1 ポートは、デフォルトの「アクティブ」シェルフ管理カードである、上部のシェルフ管理カード (ShMM1) への接続に使用します。シリアル 2 ポートは、デフォルトの「バックアップ」カード (ShMM2) に接続します。図 2-1 に、上部および下部のシェルフ管理カードに接続するシリアルポートの位置を示します。端末またはモデムの設定は、115200、N、8、1 にしてください。

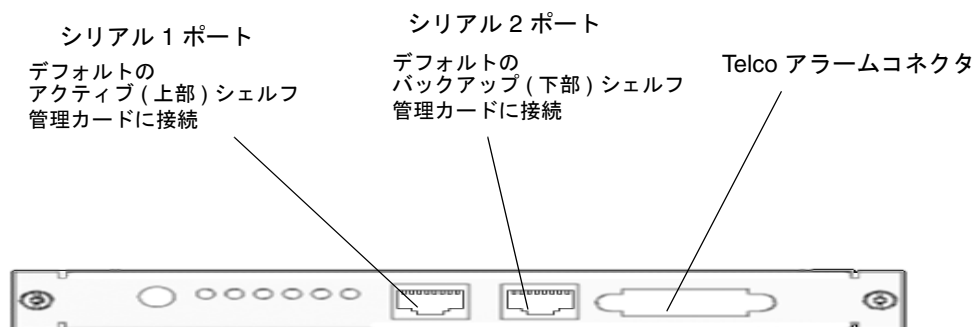


図 2-1 シェルフアラームパネルのコネクタ

はじめてシェルフ管理カードにアクセスする際は、root でログインし、デフォルトのパスワード `sunct900` を使用します。このアカウントにはフル権限 (アクセス権) が設定されています。このアカウントは削除できません。ただし、セキュリティ上の理由から、Netra CT 900 サーバーの運用を開始する前に、このアカウントのパスワードを変更してください。

root のパスワードを変更するには、次のように Linux の `passwd` コマンドを使用します。

```
# 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.
```

```
#
```

ログオンしたら、次の作業に進む前に、`clia shmstatus` コマンドを使用して、「アクティブ」シェルフ管理カードにログオンしていることを確認します。スタンバイシェルフ管理カードにログオンしていた場合は、`clia switchover` コマンドを使用してそのシェルフ管理カードを「アクティブ」に変更するか、または終了してアクティブシェルフ管理カードにログオンします。詳細は、250 ページの「shmstatus」および 252 ページの「switchover」を参照してください。

以降のセクションでは、シェルフ管理カードの CLI を使用した、シェルフ管理カードの Ethernet ポートの構成方法と、ユーザーアカウントおよびパスワードの設定方法について説明します。シェルフ管理カードの CLI の使用法に関する詳細は、第 3 章を参照してください。

---

**注** – このマニュアルでは、特に指定のないかぎり、「シェルフ管理カード」という用語でアクティブまたはスタンバイのいずれかのシェルフ管理カードを指します。このマニュアルでは、両方のカードに対応するプロンプトについても、短縮して `ShMM #` と示します。

---

## U-Boot の設定

シェルフ管理カード (ShMM) の電源投入および再起動を行うと、ハードウェアはフラッシュメモリ内の U-Boot ファームウェアの実行を開始します。このファームウェアは、ShMM の基本的な初期化処理を実行し、ユーザーが明示的に自動起動機能を使用不可にしていないかぎり (つまり、ファームウェアを保守用ユーザーコマンドインタフェースに強制的に切り替えていないかぎり)、Linux カーネルの起動を開始します。Linux は、フラッシュ内にあるカーネルおよび root ファイルシステムイメージから起動されます。U-Boot は、カーネルイメージを RAM に再配置し、カーネルパラメータを設定して、カーネルのエントリーポイントに制御を渡します。

## U-Boot インタフェース

U-Boot は ShMM のシリアルポートを介してアクセス可能で、対象とする運用環境に合わせて構成する必要があります。ShMM に電源を入れると、コンソール上に次の情報が表示されます。

```
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
#
```

# は、ユーザーコマンドを入力できるプロンプトです。

## U-Boot 環境変数

U-Boot には、使用に先立って構成する必要のある一連の環境変数があります。表 2-1 に、使用できる一連のデフォルトの変数の説明を示します。

表 2-1 デフォルトの U-Boot 環境変数

環境変数	説明
addmisc	bootargs に、quiet、reliable_upgrade、および console の設定を付加します。この変数は、通常は変更しません。
baudrate	シリアルポートのボーレートで、デフォルトは 115200 です。
bootargs	Linux カーネルに渡されるコマンド行。ほかの U-Boot 環境変数への参照を含めることができ、参照は実行時に解決されます。デフォルト値は、次のとおりです。 root=/dev/ram rw console=ttyS0,115200 reliable_upgrade=y
bootcmd	自動起動を実施するために実行される U-Boot コマンド。
bootdelay	自動起動の遅延を指定する秒単位の値。

表 2-1 デフォルトの U-Boot 環境変数 (続き)

環境変数	説明
bootfile	net および nfs 起動オプションが使用するカーネルイメージを指定するパラメータ。
console	カーネルおよび init スクリプト用のコンソールポートとボーレートの設定。デフォルトは、console=ttyS0,115200 です。
ethaddr	オンチップの一次 Ethernet コントローラの MAC アドレス。この変数の値は、U-Boot によって自動的に設定されます。このアドレスは、カーネル Ethernet ドライバに渡されます。
eth1addr	二次 Ethernet コントローラの MAC アドレス。この変数の値は、U-Boot によって自動的に設定されます。このアドレスは、カーネル Ethernet ドライバに渡されます。
flash_reset	Linux に、フラッシュファイルシステム (/etc および /var) を消去し、出荷時のデフォルト (y/n) に戻すように指示します。システムの起動スクリプトは、フラッシュの消去後、この変数を n に戻します。デフォルトは n です。
gateway	デフォルトのゲートウェイ IP アドレス。ネットワークインタフェースのルーティングを自動的に構成するために、この変数をカーネルコマンド行の一部として渡すことができます。
hostname	ネットワークホスト名で、デフォルトは sentry です。
io_config	PSC コントローラをデュアルスレーブアドレス構成用に構成するかどうかを決定します (y/n)。デフォルトの設定は、y です。
ipaddr	オンチップの一次 Ethernet インタフェースに使用される IP アドレス。rc_ifconfig 変数が y に設定されている場合は、ipdevice に指定されたネットワークインタフェースを構成するために、この変数が自動的に使用されます。システムの起動スクリプトは、この変数の最下位ビットに ShMM キャリアのハードウェアアドレスの最下位ビットを設定します。つまり、ハードウェアアドレスが偶数値であれば IP アドレスの最下位ビットは 0 に設定され、そうでない場合は 1 に設定されます。これは、冗長 ShMM の同期をとった IP アドレス構成をサポートするために、起動スクリプト /etc/netconfig によって実行されます。この機能を無効にするには、単純に /etc/readhwaddr ファイルを削除します。
ipladdr	二次 Ethernet インタフェースによって使用される IP アドレス。対応するカーネルネットワークインタフェースを自動的に構成するために、この変数をカーネルコマンド行の一部として渡すことができます。
ipdevice	ipaddr に対応するデバイスで、デフォルトは eth0 です。
ipldevice	ipladdr に対応するデバイスで、デフォルトは eth1 です。
kernel_start	フラッシュ内のカーネルイメージの絶対開始アドレス。この変数は、ブートストラップ中に U-Boot によって自動的に設定されます。

表 2-1 デフォルトの U-Boot 環境変数 (続き)

環境変数	説明
logging	メッセージログファイルを ram または flash のどちらかに保持するかを指定します。デフォルトは ram で、このオプションを使用することを推奨します。
net	この変数は、TFTP サーバーからカーネルおよび .rfs イメージを起動する手段として、bootcmd の代わりに使用できます。
netmask	ネットワークネットマスクで、デフォルトは 255.255.255.0 です。
password_reset	Linux に、出荷時のデフォルトのパスワード (ユーザー root の空のパスワード) に戻すよう指示します。デフォルトは n です。
post_normal	起動のたびに実行する POST テストのリストを指定します。指定しない場合は、コンパイル時のデフォルト設定が使用されます。この変数の値にテスト名を列挙する場合は、空白文字で区切って指定します。
post_poweron	起動のたびではなく、電源投入によるリセット時にのみ実行する POST テストのリストを指定します。指定しない場合は、コンパイル時のデフォルト設定が使用されます。この変数の値にテスト名を列挙する場合は、空白文字で区切って指定します。
quiet	起動時にシリアルコンソールへの進捗メッセージを表示しないようカーネルに指示します。デフォルトは、quiet=quiet です。
ramargs	bootargs 変数内のカーネルコマンド行に、RAM ディスクからマウントされる root ファイルシステムに適した値を設定します。
ramdisk	net および nfs の各起動オプションが使用する .rfs イメージを指定します。
ramsize	システムメモリのサイズをバイト単位で指定します。デフォルトの設定は、ビルド時の構成ブロックにエンコーディングされた SDRAM 構成から算出されます。
rc_ifconfig	shelfman ファイルからアドレスを取得するのではなく、/etc/rc スクリプトによって IP アドレスを設定できるようにします。デフォルトは n (shelfman による IP アドレスの設定を許可) です。
rc2	二次 RC スクリプトの起動を指定します。これは、キャリア固有の起動スクリプトです。デフォルトは、/etc/rc.acb3、またはターゲットとするプラットフォーム用のその他の適切なスクリプトです。
reliable_upgrade	ShMM-500 で信頼できるソフトウェアアップグレード処理を使用できるかどうかを指定します (y/n)。デフォルトの設定は y です。この変数での n の設定は、現在サポートされていません。この変数に n を設定すると、次に ShMM を起動したときにエラーメッセージが表示されハングアップします。

表 2-1 デフォルトの U-Boot 環境変数 (続き)

環境変数	説明
rfs_start	フラッシュ内の root ファイルシステムイメージの絶対開始アドレス。この変数は、ブートストラップ中に U-Boot によって自動的に設定されます。
rmcpaddr	RMCP サービス用のデフォルトの IP アドレス。
serverip	TFTP サーバーの IP アドレス。
start_rc2_daemons	二次起動スクリプトに、起動後に snmpd/boa デーモンおよび shelfman デーモンを開始するかどうかを指示します。デフォルトは y です。
time_server	実行時の同期化のための時刻サーバー。この変数を指定しない場合は、システム起動時にハードウェアクロックから時刻が抽出されます。 注: この変数を指定した場合は、ip1device 変数に usb0 を設定して、同期化が正しく行われるようにする必要があります。
timezone	現地時間を CCCn の形式で指定します。n にはグリニッジ標準時 (GMT) からのオフセット (場合によっては負の値) を指定し、CCC にはタイムゾーンを指定します。デフォルトは UTC0 です。

## 環境変数への値の割り当て

環境変数に値を割り当てるには、次の形式のコマンドを使用します。

```
setenv variable_name new_value
```

次に例を示します。

```
# setenv bootdelay 1
```

すべての環境変数を適切に設定したら、フラッシュに保存し直して、ShMM の電源を切ったあとも設定が保持されるようにする必要があります。これには、saveenv コマンドを使用します。

```
# saveenv
```

setenv 機能は、Linux ユーティリティーとしても同じ使用方法で使用できます。シェルスクリプトで U-Boot 変数を表示するには、追加された getenv ユーティリティーを使用します。

## シェルフマネージャー用の環境変数の構成

U-Boot をはじめて起動したときには、次に示すデフォルトの環境変数が定義されています。

```
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
```

これらの環境変数のいくつかは、その **ShMM** を使用するネットワークコンテキストに適した値で再構成する必要があります。

---

# シェルフ管理カードの Ethernet ポートの構成

各シェルフ管理カードは、冗長スイッチカードに接続される 2 つの Ethernet ポートを使用します。ATCA に必要な唯一のシェルフ外インタフェースは RMCP であることから、シェルフの外部 Ethernet ポートは RMCP ポートと呼ばれますが、このポートを介してほかのシェルフの外部インタフェース (Telnet) にアクセスすることも可能です。

接続後は、フルアクセス権が設定されたユーザーアカウントを使用して、シェルフ管理カードにログインする必要があります。CLI コマンドを使用してポートを構成したら、シェルフ管理カードを再起動して変更を有効にします。

## 一次 Ethernet インタフェースの使用

RMCP Ethernet ポートは設置場所のネットワークに直接接続されるため、そのネットワークに適した IP アドレスが設定される必要があります。たとえば、設置場所で 192.168.0.x の範囲の IP アドレスを使用している場合は、RMCP Ethernet ポートに、この範囲内の一意の IP アドレス (192.168.0.2 など) を設定してください。冗長 ShMM 設定では、1 つの ShMM (アクティブ ShMM) のみが、その RMCP Ethernet ポートで利用できる RMCP IP アドレスを持ちます。バックアップ ShMM は RMCP Ethernet ポートに同じ IP アドレスを割り当てますが、この IP アドレスは、そのバックアップ ShMM がアクティブの役割を引き継いだときにのみ有効になります。この方法によって、フェイルオーバーが発生した場合でも RMCP IP アドレスの可用性が維持されます。

## 一次ネットワークインタフェースへの追加 IP アドレスの割り当て

デフォルトの構成では、シェルフマネージャーが起動して RMCP IP が割り当てられるまで、一次ネットワークインタフェースには IP アドレスが割り当てられず、この ShMM にネットワークを介してアクセスすることはできません。しかし状況によっては、オペレーティングシステムの起動後すぐに RMCP ネットワークインタフェースに IP アドレスを割り当て、ネットワークを介して ShMM にアクセスできるようにすると便利ことがあります。このような場合には、シェルフマネージャーの起動時に、もともと割り当てられていた IP アドレスを RMCP IP アドレスに置き換えるのではなく、これらを共存させることも望まれます。

この構成を実現するには、RMCP IP アドレスを一次ネットワークアダプタ自体 (eth0) に割り当ててではなく、最初のエイリアス (eth0:1) に割り当てるように、シェルフマネージャーに指示する必要があります。このようにすると、オペレーティングシステムの起動中に、初期 IP アドレスがネットワークアダプタ自体 (eth0) に割り当てられます。この初期割り当ては、初期化スクリプト /etc/rc 内で行います。これを実現するには、次の手順を実行します。

1. 次のコマンドを実行して、U-Boot 変数 rc\_ifconfig を有効にします。

```
setenv rc_ifconfig y
```

2. 元の IP アドレスを U-Boot 変数 ipaddr に割り当てます。次に例を示します。

```
setenv ipaddr 192.168.1.240
```

3. シェルフマネージャー構成ファイル /etc/shelfman.conf 内の RMCP\_NET\_ADAPTER の値を eth0:1 に変更します。次に例を示します。

```
RMCP_NET_ADAPTER = eth0:1
```

冗長構成の場合、U-Boot 変数 ipaddr は両方の ShMM に対して同じ値を持つことができます。2 つの冗長 ShMM のそれぞれに実際に割り当てられる初期 IP アドレスは、ipaddr の値に基づいて、ShMM のハードウェアアドレスに応じて変更されます。IP アドレスの最下位ビットに、ハードウェアアドレスの最下位ビットが設定されます。前述の例で偶数のハードウェアアドレスを持つ ShMM の IP アドレスは 192.168.1.240 になり、奇数のハードウェアアドレスを持つ ShMM の IP アドレスは 192.168.1.241 になります。このような IP アドレスの変更は、ファイル /etc/readhwaddr を削除することで無効にできます。

## RMCP アドレスの伝播

シェルフマネージャーのオプションの機能として、RMCP IP アドレスとは最下位ビットだけが異なる IP アドレスを使用して、バックアップ ShMM を外部ネットワークに公開することもできます。バックアップ ShMM のネットマスクおよびデフォルトゲートウェイは、アクティブ ShMM と同じ設定になります。たとえば、RMCP IP アドレスが 192.168.0.2 である場合、バックアップ ShMM はこれに対応する IP アドレス 192.168.0.3 を持ち、同じネットマスクとデフォルトゲートウェイを使用します。この機能を有効にするには、シェルフマネージャー構成ファイル /etc/shelfman.conf 内の、シェルフマネージャー構成パラメータ PROPAGATE\_RMCP\_ADDRESS に TRUE を設定する必要があります。

## 二次 Ethernet インタフェースの使用

二次ネットワークインタフェースは、ATCA ネットワークハブボードのいずれかを使用して、シェルフマネージャーに接続します。冗長シェルフマネージャー間の通信には、USB ベースのデュアルネットワークインタフェースが使用されます。

## 冗長通信のためのデュアル USB ネットワークインタフェースの使用

ShMM では、2 つの USB 接続を介して 2 つの追加ネットワークインタフェースが実装されます。この構成では、常に 2 つの冗長シェルフマネージャーが接続されます。インタフェースの名前は、usb0 および usb1 です。インタフェース usb0 は常に存在しますが、インタフェース usb1 は、ピアシェルフマネージャー上で (つまり、ピアシェルフマネージャーが物理的にインストールされ、動作していて) インタフェース usb0 がアクティブである場合にのみ存在します。また、これらのインタフェースはクロス接続されます。つまり、一次シェルフマネージャー上の usb0 は、二次シェルフマネージャー上の usb1 に接続され、同様に逆方向の接続も行います。

シェルフマネージャーは、冗長シェルフマネージャー間の通信での USB ネットワークインタフェースの使用をサポートします。この機能を使用するには、次に示すように、シェルフマネージャー構成ファイル `/etc/shelfman.conf` 内に 2 つの冗長ネットワークアダプタを定義する必要があります。

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

冗長ネットワークインタフェースのサブネットマスクの定義に関して、考慮事項がもう 1 つあります。従来の方法では、冗長ネットワークアダプタが 1 つのみ使用される場合には、`/etc/shelfman.conf` に指定された冗長 IP アドレスから 2 つの異なる IP アドレスが派生します。これらは冗長接続の 2 つの終端に割り当てられ、最下位ビットのみが異なります。

ただし、2 つの冗長ネットワークアダプタが使用される場合には、4 つの異なる IP アドレスが使用され、各終端に 1 つずつ割り当てられます (2 つの冗長シェルフマネージャーのそれぞれの 2 つの終端)。正常な動作を確実に実現するため、同じシェルフマネージャー上の 2 つの終端 (usb0 および usb1) は異なる論理ネットワークに属している必要があり、一方で、あるシェルフマネージャー上の usb0 は、もう一つのシェルフマネージャー上の usb1 と同じ論理ネットワークに属している必要があります。この考慮事項に基づき、`/etc/shelfman.conf` に指定した冗長 IP アドレスのサブネットマスクの最下位ビットを入れ替えることによって、2 つの追加 IP アドレスを派生させます。そのため、サブネットマスクは、指定された冗長 IP アドレスクラスのデフォルトより厳密である必要があります。サブネットマスクが指定されていない場合は、デフォルトで 255.255.255.128 が設定されます。この値は、USB ネットワークインタフェースを使用して冗長性を実現する場合の、`/etc/shelfman.conf` 内のこのパラメータの推奨値でもあります。

次に、USB ネットワークインタフェースの IP アドレスの派生の例を示します。

/etc/shelfman.conf に、次の定義があるとします。

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

「偶数」のハードウェアアドレスを持つ ShMM では、割り当てられる IP アドレスは次のようになります。

```
usb0: 192.168.1.2 (変更なし)  
usb1: 192.168.1.130 (ネットマスクの最下位ビットを入れ替え)
```

「奇数」のハードウェアアドレスを持つ ShMM では、割り当てられる IP アドレスは次のようになります。

```
usb0: 192.168.1.131 (IP アドレスの最下位ビットおよびネットマスクの最下位ビットを入れ替え)  
usb1: 192.168.1.3 (IP アドレスの最下位ビットを入れ替え)
```

## デフォルトの ShMM ネットワークパラメータの変更

ShMM を特定のネットワーク環境で動作させるように構成するには、次に示すネットワークパラメータの変更が必要です。

- RMCP IP アドレス
- RMCP ゲートウェイアドレス
- RMCP ネットマスク

RMCP ネットワークパラメータを変更するには、複数の手順による処理を行います。U-Boot ネットワーク環境変数を更新し、次に、起動されたアクティブ ShMM モジュールのネットワーク設定をシェルフマネージャー CLI を使用して更新する必要があります。

## ▼ デフォルトの ShMM ネットワークパラメータを変更する

1. ShMM モジュールに、シリアルポートコンソールを接続します。

通常、これは 115200 ボー、N/8/1 です。ShMM のキャリアを再起動し、スペースバーを押して自動起動プロセスに割り込みます。次の画面が表示されます。

```
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. 現在のネットワーク設定を表示します。

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

3. 設定を変更し、非揮発性ストレージに格納して確定します。

```
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. ShMM を完全に機能する状態で起動し、root ユーザーとしてログインします。

```
ShMM # reset

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 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
## 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. ShMM の起動を許可します。

---

注 – U-Boot ファームウェアで変更された設定は、必ずしも Linux 環境に伝播される  
とはかぎりません。これは、フェイルオーバー状況を管理するために、シェルフマ  
ネージャーが自身のネットワーク構成データのコピーを維持する必要があるためで  
す。

---

これがシェルフマネージャーのはじめての起動である場合、または起動に先立ってフラッシュデバイスが出荷時のデフォルトにリセットされていた場合には、シェルフマネージャーは **U-Boot** が提供するネットワーク設定を使用して、このネットワークコンテキストを設定します。そのため、**U-Boot** での変更がその先へと伝播されます。

これ以外の場合に、シェルフマネージャーコンテキストのネットワーク設定を構成するには、次の手順を実行する必要があります。

## 6. アクティブシェルフマネージャーであるかどうかを確認します。

変更は、`cpld` コマンドを使用して、アクティブシェルフ管理カードに対してのみ行う必要があります。この操作によって、冗長インタフェースを介して、バックアップカードもこのネットワーク構成変更で更新されます。アクティブ **ShMM** でない場合は、もう一方の **ShMM** デバイスに接続してから手順 7 に進みます。

```
# cpld
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. 現在の IP 設定を取得します。

```
# 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. 次に示すように、IP 設定を変更します。

```
# 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
#
```

---

## シェルフマネージャー構成ファイルの設定

シェルフマネージャー構成ファイル `shelfman.conf` は、`/etc` ディレクトリ内にあります。このファイルの各行は、`#` で始まるコメント行、または構成パラメータへの割り当てを示す `name = value` の組み合わせのいずれかです。`name` と `value` は、等号 (=) で区切ります。

構成パラメータの名前では、大文字と小文字は区別されません。各構成パラメータの型は、ブール型、数値、文字列、または IP アドレスのいずれかです。

構成パラメータの型に適合する値の形式は、次のとおりです。

ブール型	ブール型は、文字列 <code>FALSE</code> または <code>TRUE</code> で表すか、それぞれの数値表現である <code>0</code> または <code>1</code> で表すことができます。
数値	整数値で、符号を付けることもあります。16 進表記法「 <code>0x...</code> 」もサポートされます。
文字列	引用符付き (二重引用符 <code>"</code> を使用) または引用符なしの文字列。引用符付きの文字列には空白文字を含めることができますが、引用符なしの文字列では最初の空白文字が終端となります。文字列の最大長は、文字列を使用する構成パラメータごとに個別に指定されています。
IP アドレス	ドット付き 10 進表記 ( <code>xxx.xxx.xxx.xxx</code> ) の IP アドレス。

\$envvar という表記を使用すると、環境変数の値を構成パラメータの値として指定することができます。このように指定すると、構成ファイルを読み取る際に、変数 envvar の値に置換されます。次に例を示します。

```
DEFAULT_RMCP_IP_ADDRESS = $IPADDR
```

シェルフマネージャーがはじめて起動したあとは、IP アドレスが IPMI LAN 構成パラメータとともに保存されます。LAN 構成パラメータは、RMCP インタフェースまたは CLI シェルフ外インタフェースのいずれかの方法でアクセスまたは変更でき、シェルフマネージャーの再起動時には shelfman 構成ファイルよりも優先されます。そのため、これらのインタフェースを介して LAN の IP アドレスまたはゲートウェイに対して行なった変更が確実に持続されます。ただし、シェルフ FRU 情報内のシェルフマネージャー IP 接続の記録に IP アドレスが含まれている場合には、この情報がほかのすべての設定 (シェルフ外または RMCP の IP アドレス) より優先されます。シェルフ FRU 情報には、このアドレスを指定しないか、または 0.0.0.0 を設定して、アドレスをシェルフマネージャー構成ファイルおよび IPMI LAN 構成パラメータによって確実に制御できるようにすることをお勧めします。

次に、現在サポートされている構成パラメータを示します。

表 2-2 シェルフマネージャー構成パラメータ

名前	型	デフォルト	説明
2_X_SYSTEM	ブール型	なし	指定した場合、このパラメータは、現在のシステムが <b>AdvancedTCA</b> であることを明示します (FALSE の場合)。指定しない場合 (TRUE) は、システムのタイプは自動的に選択されます。誤ったハードウェア検出アルゴリズムによるシステムタイプを上書きする必要がある場合以外に、このパラメータを指定することはお勧めしません。
ALARM_CUTOFF_TIMEOUT	数値	600 秒 (10 分)	アラームカットオフのタイムアウト値 (アラームカットオフが非アクティブになるまでの時間) を秒単位で指定します。
ALLOW_CLEARING_CRITICAL_ALARM	ブール型	FALSE	TRUE に設定すると、CLI コマンド <code>clia alarm clear</code> によって、クリティカルアラーム状態をクリアできるようになります。
ALTERNATE_CONTROLLER	ブール型	TRUE	<code>address = ShMM hardware address</code> と指定して、シェルフマネージャーの代替コントローラを使用します。
AUTO_SEND_MESSAGE	ブール型	TRUE	シェルフマネージャー以外の IPMB アドレスに送信された RMCP 要求を、そのアドレス宛ての Send Message 要求に自動変換します。

表 2-2 シェルフマネージャー構成パラメータ (続き)

名前	型	デフォルト	説明
CARRIER	文字列 (16)	PPS	ShMM が取り付けられた特定のキャリアボードの名前。
CARRIER_OPTIONS	文字列 (256)	" "	キャリア固有のオプションで、サポートされるキャリアごとに個別に定義します。
CONSOLE_LOGGING_ENABLED	ブール型	FALSE	シェルフマネージャーを起動したコンソールにログメッセージを出力します。
COOLING_IGNORE_LOCAL_CONTROL	ブール型	FALSE	ファンデバイスのローカル制御機能を使用不可にし、シェルフマネージャーがファンレベルを管理することを明示します。
COOLING_POLL_TIMEOUT	数値	30 秒	冷却監視と管理スレッドを続けて起動する際の間隔の最長時間 (秒単位)。
CTCA_FRU_RESET_TIMEOUT	数値	なし	なし
CTCA_HEALTHY_TIMEOUT	数値	なし	なし
CTCA_INITIAL_FAN_LEVEL	数値	なし	なし
DEFAULT_GATEWAY_IP_ADDRESS	IP アドレス	なし	チャンネル 1 の IPMI LAN 構成パラメータで、これに対応するパラメータに 0.0.0.0 が設定されている場合、シェルフ外 (RMCP ベース) 通信のゲートウェイに使用されるデフォルトの IP アドレス。LAN 構成パラメータによってゼロ以外のゲートウェイ IP アドレスが提供される場合は、シェルフマネージャー構成ファイルによって提供される値は無視されます。
DEFAULT_RMCP_IP_ADDRESS	IP アドレス	なし	シェルフ外 (RMCP ベース) の通信に使用されるデフォルトの IP アドレス。シェルフマネージャーの冗長インスタンス間でスイッチオーバーされます。チャンネル 1 の IPMI LAN 構成パラメータと、シェルフ FRU 情報のシェルフマネージャー IP 接続記録で、これに対応するパラメータに 0.0.0.0 が設定されている場合にのみ、この IP アドレスが使用されます。LAN 構成パラメータまたはシェルフ FRU 情報、あるいはその両方によってゼロ以外の IP アドレスが提供される場合は、シェルフマネージャー構成ファイルによって提供される値は無視されます。

表 2-2 シェルフマネージャー構成パラメータ (続き)

名前	型	デフォルト	説明
DEVICE_POLL_TIMEOUT	数値	10 秒	Get Device ID コマンドによって、シェルフマネージャーが IPMB デバイスに連続ポーリングを行う際の時間間隔 (秒単位)。
EXIT_IF_NO_SHELF_FRU	ブール型	FALSE	TRUE を設定すると、シェルフ FRU が見つからない場合に、シェルフマネージャーが終了 (多くの場合は ShMM をリセット) します。
INITIAL_FAN_LEVEL	数値	5	シェルフマネージャーがファントレーに適用する初期ファンレベル。通常、ファンレベルの値は 0 ～ 15 で指定し、0 がもっとも遅く、15 が可能なかぎり速いファン速度です。
IPMB_ADDRESS	数値	0	シェルフマネージャーの IPMB アドレスで、ハードウェアアドレスを上書きします。0 の値を指定すると、シェルフマネージャーはハードウェアからハードウェアアドレスを読み取り、IPMB アドレスにハードウェアアドレス * 2 を設定します。
IPMB_RETRIES	数値	3	要求に対する応答を受信できない場合に IPMB 要求の再送を試みる回数で、この回数に達すると再送を停止します。
IPMB_RETRY_TIMEOUT	数値	4 秒	IPMB 要求を送信したあとシェルフマネージャーが応答を待つ時間の長さで、この時間を過ぎると要求を再送します。
LOCAL_SHELF_FRU	ブール型	TRUE	シェルフマネージャー上で、シェルフ FRU 情報を公開するローカル FRU 1 を作成します。シェルフ FRU 情報は、ファイル /var/nvdata/shelf_fru_info から取得します。
M7_TIMEOUT	数値	-1 (秒)	FRU が M7 状態にある最長時間 (秒単位)。この時間を過ぎると、FRU は自動的に M0 に移行します。-1 (デフォルト) は、期限なしを意味します。このパラメータを 0 に設定すると、FRU が M7 状態になることを完全に抑制します。
MAX_ALERT_POLICIES	数値	64	使用できる PEF アラートポリシーの最大数。
MAX_ALERT_STRINGS	数値	64	使用できる PEF アラート文字列の最大数。
MAX_DEFERRED_ALERTS	数値	32	未処理の PEF アラートの最大数。

表 2-2 シェルフマネージャ構成パラメータ (続き)

名前	型	デフォルト	説明
MAX_EVENT_FILTERS	数値	64	使用できる PEF イベントフィルタの最大数。
MAX_EVENT_SUBSCRIBERS	数値	64	シェルフマネージャからイベント通知を受信するために、同時に登録できるエンティティの最大数。
MAX_EVENT_SUBSCRIBER_IDLE_TIME	数値	60 秒	イベント受信登録者のタイムアウトの最長時間 (秒単位) で、イベントが到着してから、登録者がシェルフマネージャからそのイベントを取得するまでの時間です。このタイムアウト時間を過ぎると登録者は不在とみなされ、自動的に登録を解除されます。
MAX_PENDING_EVENT_NOTIFICATIONS	数値	1024	有効な登録者ごとの未処理のイベント通知の最大数。
MAX_PENDING_IPMB_REQUESTS	数値	64	応答を待つ保留状態の IPMB 要求の最大数。
MAX_SEL_ENTRIES	数値	1024	システムイベントログ (SEL) 内のエントリの最大数。
MAX_SESSIONS	数値	32	IPMI の同時セッションの最大数。
MAX_USERS	数値	32	IPMI ユーザーの最大数。
MIN_FAN_LEVEL	数値	1	ファンレベルの最小値。ファンレベルを自動制御しているとき、冷却制御コードは、いずれかのファンのファンレベルをこの値より低くすることはできません。
MIN_SHELF_FRUS	数値	2	正常に起動するためにシェルフマネージャが検出する必要がある、シェルフ内のシェルフ FRU の最小数。
PHYSICAL_SENSORS	ブール型	TRUE	ADM1026 チップおよび LM75 チップがホストする物理センサーに基づいて IPMI センサーを作成します。
POWER_UNLISTED_FRUS	ブール型	TRUE	シェルフ FRU 情報の電源管理テーブルにリストされていない FRU をアクティブにし、電源を入れることを許可します。

表 2-2 シェルフマネージャー構成パラメータ (続き)

名前	型	デフォルト	説明
PROPAGATE_RMCP_ADDRESS	ブール型	FALSE	TRUE の場合、アクティブシェルフマネージャーは RMCP IP アドレスをバックアップシェルフマネージャーに伝達します。バックアップシェルフマネージャーは、この IP アドレスを最下位ビットだけ反転させて使用し、RMCP_NET_ADAPTER 変数に指定されたネットワークインタフェースを構成します。
REDUNDANCY_ENABLED	ブール型	TRUE	シェルフマネージャーを詳細モードで実行します。
REDUNDANCY_NET_ADAPTER	文字列 (16)	usb0	シェルフマネージャーの冗長インスタンス間の通信に使用されるネットワークアダプタの名前。
REDUNDANCY_NET_ADAPTER2	文字列 (16)	usb1	シェルフマネージャーの冗長インスタンス間の通信に使用される、二次ネットワークアダプタの名前 (デュアル USB ネットワークインタフェースがこの目的で使用される場合)。
REDUNDANCY_NETMASK	数値	0	冗長 IP アドレスに割り当てるネットマスク。デフォルト (0) では、ネットマスクは IP アドレスのクラスから自動的に決定されます。
REDUNDANCY_PORT	数値	1040	シェルフマネージャーの冗長インスタンス間の対話に使用される TCP ポート。
REDUNDANT_IP_ADDRESS	IP アドレス	なし	冗長通信に使用される IP アドレス。このアドレスは、実際には、最下位ビットだけが異なる一組の IP アドレスを示します。これらは、ハードウェアアドレスに従って冗長シェルフマネージャーに割り当てられます。
RESERVATION_RETRIES	数値	10	シェルフマネージャーが Reserve Device SDR コマンドを再試行する最大回数。
RMCP_NET_ADAPTER	文字列 (16)	eth0	RMCP ベースの通信に使用されるネットワークアダプタの名前。
RMCP_NET_ADAPTER2	文字列 (16)	なし	ハードウェアによってクロス接続リンクがサポートされる場合に、RMCP ベースの通信に使用される代替ネットワークアダプタの名前。
SDR_READ_RETRIES	数値	3	シェルフマネージャーが Read Device SDR コマンドを再試行する最大回数。

表 2-2 シェルフマネージャー構成パラメータ (続き)

名前	型	デフォルト	説明
SEL_HIGH_WATERMARK	数値	0	SEL の自動削除を制御するアルゴリズムのための「高位境界値」。SEL 内の空きエントリの実際のパーセンテージがこの値を下回った場合、または SEL がオーバーフローした場合に、シェルフマネージャーは、SEL 内の古い記録を古いものから順に削除するスレッドを起動します。
SEL_LOW_WATERMARK	数値	0	SEL の自動削除を制御するアルゴリズムのための「低位境界値」。SEL から古い記録を削除するスレッドが起動した場合、このスレッドは、SEL 内の使用済みエントリのパーセンテージがこの値を下回るまで記録を削除します。
SHELF_FRU_IN_EEPROM	ブール型	TRUE	TRUE を指定すると、バックプレーン上の EEPROM からキャリア固有の方法でシェルフ FRU 情報が取得され、FALSE を指定すると、フラッシュファイルシステム上のファイルからシェルフ FRU 情報が取得されます。
SHELF_FRU_TIMEOUT	数値	5 秒	初期化中に、シェルフマネージャーがシェルフ FRU 情報のデバイスが検出されるのを待機する時間。
SHORT_SEND_MSG_RESPONSE	ブール型	TRUE	シェルフマネージャーが提供する Send Message 応答のタイプを決定します。PICMG 3.0 ECR の要件に合わせるか (TRUE の場合)、またはシェルフマネージャーの以前のバージョンとの互換性を維持します (FALSE の場合)。
SWITCHOVER_TIMEOUT_ON_BROKEN_LINK	数値	10 秒	このパラメータは、シェルフマネージャーとシステムマネージャーとの間の物理ネットワークリンク (RMCP リンク) が切断された場合の、シェルフマネージャーによるスイッチオーバー開始の有無とその時期に影響を与えます。リンクの切断がこのパラメータに指定された秒数以上に継続した場合は、スイッチオーバーが行われ、リンクがこのタイムアウト時間内に復元された場合は、スイッチオーバーは行われません。このパラメータの値が -1 である場合は、RMCP リンクが切断されても自動スイッチオーバーは行われません。
SYSLOG_LOGGING_ENABLED	ブール型	TRUE	システムログにログメッセージを出力します。

表 2-2      シェルフマネージャー構成パラメータ (続き)

名前	型	デフォルト	説明
TASKLET_RETRIES	数値	3	各シェルフマネージャータスクレット (アクティブまたは非アクティブへの切り替え、情報の取得) を再試行する回数で、この回数を超えると再試行を停止します。
VERBOSITY	数値	7	シェルフマネージャーの詳細レベル。
VERIFY_SHELF_FRU_CHECKSUM	ブール型	TRUE	シェルフ FRU 情報記録のチェックサムの検証を有効にします。FALSE を設定すると、シェルフマネージャーはチェックサムを無視します。
WATCHDOG_ENABLED	ブール型	TRUE	CPLD によってサポートされる、ハードウェアウォッチドッグタイマーを使用します。

デフォルトでは、ShMM がはじめて起動したときに、この構成ファイルの変数が自動的に使用されます。デフォルトの構成ファイルは、次に示す数個の U-Boot によって設定された環境変数をインポートします。

\$CARRIER_OPTIONS	デフォルトのキャリア固有のオプション
\$IPADDR	デフォルトの RMCP IP アドレス
\$IPDEVICE	デフォルトの RMCP ネットワークアダプタ
\$IP1ADDR	デフォルトの冗長 IP アドレス
\$IP1DEVICE	デフォルトの冗長ネットワークアダプタ
\$GATEWAY	RMCP 通信に使用されるデフォルトのゲートウェイ

必要に応じて、シェルフマネージャーを出荷時のデフォルトのパラメータ値にリセットできます。コード例 2-1 に、デフォルトの構成ファイルを示します。

コード例 2-1      デフォルトの shelfman.conf ファイル

```
# /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
```

## コード例 2-1 デフォルトの shelfman.conf ファイル (続き)

```
# 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.
```

## コード例 2-1 デフォルトの shelfman.conf ファイル (続き)

```
#
# 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

# 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
```

## コード例 2-1 デフォルトの shelfman.conf ファイル (続き)

```
# 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.
#
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
```

## コード例 2-1 デフォルトの shelfman.conf ファイル (続き)

```
# 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.
#
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.
#
```

## コード例 2-1 デフォルトの shelfman.conf ファイル (続き)

```
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
# 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.
#
```

## コード例 2-1 デフォルトの shelfman.conf ファイル (続き)

```
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
# 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
```

## コード例 2-1 デフォルトの shelfman.conf ファイル (続き)

```
# 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.
#
#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.
```

## コード例 2-1 デフォルトの shelfman.conf ファイル (続き)

```
# 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.
#
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 holds the BD_SEL# line low in order to reset a
# CompactPCI board. Default is 500 milliseconds.
```

## コード例 2-1 デフォルトの shelfman.conf ファイル (続き)

```
#
# 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
```

## 詳細レベルの概要

詳細レベルを指定すると、構成パラメータ `CONSOLE_LOGGING_ENABLED` および `SYSLOG_LOGGING_ENABLED` の設定に応じて、コンソールまたは `syslog` に追加出力を送信できます。 `VERBOSITY` 構成パラメータは 16 進数のビットマスクで、各ビットで次に示す特定のタイプのメッセージ出力が可能になります。

0x01	エラーメッセージ
0x02	警告メッセージ
0x04	情報メッセージ

0x08	詳細情報メッセージ
0x10	トレースメッセージ
0x20	詳細トレースメッセージ
0x40	IPM コントローラの初期化中に、IPM コントローラに送信される重要なコマンドに関して表示されたメッセージ
0x80	内部ロックの獲得および解放に関する詳細メッセージ

デフォルトのデバッグレベルは 7 で、エラー、警告、および情報の各メッセージの出力が可能になります。

## 日付と時刻の設定

システムをはじめて起動したときにはクロックはセットされていないため、初期化する必要があります。クロックは、初期状態では 1970 年 1 月 1 日に設定されています。日付は、シリアルコンソールを使用して操作できます。

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

日付を変更するには、date アプリケーションを使用して正しい日付を入力します。date コマンドの形式は MMDDHHMMSSYYYY で、次の意味があります。

MM	月
DD	日
HH	時 (24 時間制の表記法を使用)
MM	分
SS	秒
YYYY	年

次に例を示します。

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

この日付を継続して使用するには、`hwclock` アプリケーションを使用して保存する必要があります。

```
# hwclock -systohc
```

場合によっては、次のエラーメッセージが表示されることがあります。

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

このエラーは無視できます。これは、元の日付が未初期化状態であるために発生したものです。

## 時刻サーバーからの日付および時刻の取得

シェルフ管理カードにリアルタイムクロック (RTC) 用の電池がない場合は、システムの起動時に時刻サーバーから日付および時刻を取得し、その後も定期的に同期化することができます。時刻サーバーには、TCP に関して RFC 868 の規定をサポートするものを選択する必要があります。これは、`rdate` ユーティリティの要件です。この機能を有効にするには、`U-Boot` 変数 `time_server` と、任意で追加の変数 `timezone` を定義する必要があります。

`time_server` 変数には、起動後にシェルフマネージャーによるシステム時刻の照会先となる時刻サーバーの IP アドレスを入力します。この変数は、環境変数 `TIMESERVER` として Linux レベルに伝達されます。この変数を設定した場合、起動スクリプト `/etc/netconfig` はスクリプト `/etc/timesync` をデーモンとして起動します。このデーモンは、永久ループの中で、デフォルトでは 300 秒間隔で時刻サーバーに照会を実行します。この間隔を変更するには、スクリプト `/etc/timesync` を編集して、変数 `INTERVAL` の値を変更します。

---

**注** – `time_server` 変数を指定した場合は、正しく同期化を行うため、`ip1device` 変数に `usb0` を設定する必要があります。

---

変数 `timezone` には、現在のタイムゾーンの名前と、続けてグリニッジ標準時 (GMT) からのオフセットを指定します。オフセットは、グリニッジより西では正の値、グリニッジより東では負の値になります。この変数は、環境変数 `TZ` として Linux レベルに伝達されます。この変数のデフォルト値は `UTC0` です。これは協定世界時 (UTC) で、グリニッジ時刻と同じです。

時刻サーバーによって送信される時刻は GMT です。シェルフマネージャーのタイムゾーンが設定されていない場合、または設定が不正である場合には、時刻サーバーから取得した時刻は正しく解釈されません。タイムゾーンを示す 3 文字の名前は、シェルフマネージャーでは使用されませんが、Linux タイムゾーンの設定のために伝達さ

れます。たとえば、タイムゾーン名に XXX0 を使用している場合、date コマンドは、Thu Sep 9 21:24:24 XXX 2004 のような出力を生成します。夏時間はサポートされません。

次に、米国東部時間に対応する timezone の例を示します。

```
timezone = EST5
```

ここで、「5」という数字は、GMT から西に 5 時間のタイムゾーンであることを示しています。EST は、任意の 3 文字に置き換えることができます。この文字列は、たとえば Linux の date コマンドの出力で、タイムゾーンを識別するために使用されます。

## シェルフ管理カードでのユーザーアカウントの設定

RMCP アクセス用のユーザーアカウントは、シェルフマネージャーの CLI を使用して設定します。アクティブシェルフ管理カードで入力したユーザー情報は、ただちにスタンバイシェルフ管理カードにミラー化され、共有されるようになります。シェルフ管理カードは、32 のアカウントと対応するパスワードをサポートします。

### ▼ RMCP アクセス用のユーザーアカウントを追加する

1. アクティブシェルフ管理カードにログインします。
2. ユーザーを次のように追加します。

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

変数のパラメータの意味は、次のとおりです。

*userid* — 有効なユーザー ID

*user-name* — ユーザー名 (最大 16 文字)

*channel-access-flag* — SetUserInfo コマンドの先頭バイト (意味を持つのはビット 4、5、および 6 のみ)

- ビット 6 — IPMI メッセージングを使用可能
- ビット 5 — リンク認証を使用可能

- ビット 4 – コールバックに限定

*privilege-level* – ユーザーの特権レベル

*password* – ユーザーのパスワード (通知なしで 16 文字に切り詰められる)

次に、特権レベルが管理者で、パスワードに PICMG guru が設定された、root という名前のユーザー 9 を追加する例を示します。

```
# 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"
#
```

アクセス権の詳細は、253 ページの「user」を参照してください。また、clia user コマンドを参照してください。

## ユーザー名の制限

*username* フィールドには、最大 16 文字を入力できます。これには 1 文字以上の英小文字を含める必要があり、先頭文字は英字にする必要があります。

*username* に使用できる文字は、次のとおりです。

- 英字
- 数字
- ピリオド (.)
- 下線 (\_)
- ハイフン (-)

## パスワード

パスワードには最大で 16 文字を使用でき、16 を超える文字は切り詰められます。

---

## シェルフマネージャーでの OpenHPI の構成

シェルフマネージャーには、SA Forum のハードウェアプラットフォームインタフェース (HPI) 実装のオープンソースである OpenHPI のサポートが含まれています。HPI は、通常はシャーシおよびラックベースのサーバーに、管理コンピュータハードウェアへのインタフェースを提供します。HPI には、SNMP MIB を使用して OpenHPI SNMP サブエージェントを介してアクセスします。

システム管理者が注意する必要がある構成ファイルは、次の 2 つです。

- /etc/openhpi.conf – OpenHPI 構成ファイル
- /etc/snmpd.conf – SNMP サブエージェント構成ファイル

### /etc/openhpi.conf ファイル

OpenHPI 構成ファイル /etc/openhpi.conf は、ShMM の適切な IP アドレスを提供するために更新する必要があります。構成ファイルを更新したあと、ShMM をリセットして変更を有効にする必要があります。

## ▼ /etc/openhpi.conf ファイルを変更する

1. /etc/openhpi.conf ファイルを編集して、libipmidirect の addr パラメータの値を ShMM の IP アドレスに変更します。

次に、/etc/openhpi.conf ファイルの例を示します。

```
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. プロンプトで reboot コマンドを実行して、ShMM を再起動します。

次に例を示します。

```
# reboot
```

## /etc/snmpd.conf ファイル

SNMP サブエージェント構成ファイル /etc/snmpd.conf は、SNMP サブエージェントの動作について定義し、アクセス制御およびトラップ設定に関する指示子を含んでいます。アクセス制御、SNMPv3 構成、およびトラップ設定については、以降のセクションで説明します。

## アクセス制御

SNMP サブエージェントは、RFC 2575 で定義された、ビューに基づくアクセス制御モデル (VACM) をサポートします。この目的のため、サブエージェントは構成ファイルの次のキーワードを認識します。

- `com2sec`
- `group`
- `access`
- `view`

また、次に示す、容易に使用できるいくつかのラッパー指示子を認識します。

- `rocommunity`
- `rwcommunity`
- `rouser`
- `rwuser`

このセクションでは、さまざまなタイプおよびレベルのアクセスを受け入れる `snmpd` プログラムの構成方法について明示します。

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

VACM アクセス構成テーブルに、SNMPv3 の USM ユーザーを作成します。`group`、`access`、および `view` 指示子を組み合わせて使用すると、より効率的で高機能ですが、ここに示したラッパー指示子を使用する方がたいへん簡易です。

ユーザーが使用する必要のある最低レベルの認証およびプライバシーは、最初のトークンで指定されます (デフォルトは `auth`)。OID パラメータは、ユーザーのアクセスを、指定された OID 下のものに制限します。

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

エージェントへのアクセスに使用できる、読み取り専用コミュニティおよび読み取り/書き込み可能コミュニティを作成します。これらは簡易的なラッパーで、より複雑で高機能な `com2sec`、`group`、`access`、および `view` 指示子行を含んでいます。ラッパーは、これらの指示子ほど効率的ではないうえ、グループが作成されないためにテーブルがより大きくなる可能性があります。複雑な環境ではこれらの指示子を使用することは、お勧めしません。環境が比較的単純である場合、またはパフォーマンスに小さな悪影響があっても維持できる場合に、これらの指示子を使用してください。

`source` トークンの形式については、次の `com2sec` 指示子のセクションで説明します。OID トークンは、コミュニティのアクセスを、指定された OID 下のものに制限します。

```
com2sec name source community
```

*source/community* のペアから、セキュリティー保護された *name* へのマッピングを指定します。*source* には、ホスト名、サブネット、または *default* という語を指定できます。サブネットは、IP/マスクまたは IP/ビット数の形式で指定できます。着信パケットにマッチする最初の *source/community* の組み合わせが選択されます。

#### *group name model security*

*securitymodel/securityname* から *group* へのマッピングを定義します。*model* は、*v1*、*v2c*、または *usm* のいずれかです。

#### *access name context model level prefix read write notify*

*group/security* および *model/security* レベルからビューにマッピングします。*model* は、*any*、*v1*、*v2c*、または *usm* のいずれかです。*level* は、*noauth*、*auth*、または *priv* のいずれかです。*prefix* は、着信 PDU のコンテキストに対する *context* のマッチング方法を指定する、*exact* または *prefix* のいずれかです。*read*、*write*、および *notify* には、対応するアクセスに使用されるビューを指定します。*v1* または *v2c* アクセスでは、*level* は *noauth* で、*context* は空白にします。

#### *view name type subtree [mask]*

名前付きのビューを定義します。*type* は、*included* または *excluded* のいずれかです。*mask* は、16 進のオクテットをピリオド (.) またはコロン (:) で区切ったリストです。マスクを指定しない場合のデフォルトは、*ff* です。*mask* を使用すると、比較的容易にテーブル内の 1 つの行へのアクセスを制御できます。たとえば、ISP は、各顧客に自身のインタフェースにアクセスする機能を提供することを検討できます。

```
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
```

これらのエントリには行インデックスが隠されて含まれていますが、ユーザーは行のそのフィールドを変更することもできます。

次に、VACM の例を示します。

```
# 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 構成

`engineID` *string*

SNMPv3 メッセージに応答できるようにするには、`engineID` を使用してサブエージェントを構成する必要があります。この構成ファイル行を指定すると、*string* から `engineID` が設定されます。`engineID` のデフォルト値は、そのマシンのホスト名に対応する IP アドレスで最初に見つかったものに設定されます。

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

MD5 および SHA は使用する認証タイプですが、SHA を使用するには、インストールされた **OpenSSL** を使用してパッケージを構築しておく必要があります。現在サポートされているプライバシープロトコルは、DES のみです。*privpassphrase* を指定しない場合は、*authpassphrase* と同じと見なされます。

---

注 – 作成したユーザーは、前述の VACM アクセス制御テーブルにも追加しなければ使用できません。

---

---

注 – パスフレーズは 8 文字以上で指定します。

---

## トラップおよび情報の宛先の設定

`trapcommunity string`

トラップを送信する際に使用される、デフォルトのコミュニティー文字列を定義します。このコマンドは、このコミュニティー文字列を使用する次の 3 つのコマンドより先に使用する必要があることに注意してください。

`trapsink host [community [port]]`

`trap2sink host [community [port]]`

`informsink host [community [port]]`

トラップを受信するホスト、または `informsink` で通知を報告するホストを定義します。このデーモンは、起動時にコールドスタートトラップを送信します。有効であれば、認証が失敗した場合にもトラップを送信します。複数の `trapsink`、`trap2sink`、および `informsink` 行を指定して、複数の宛先を指定することができます。 `trap2sink` は SNMPv2 トラップの送信に使用し、`informsink` は情報通知の送信に使用します。 `community` を指定しない場合は、先に指定した `trapcommunity` 指示子の文字列が使用されます。 `port` を指定しない場合は、既知の SNMP トラップポート (162) が使用されます。

`trapssess [snmpcmdargs] host`

より汎用的なトラップ構成トークンで、SNMP の任意のバージョンで任意のタイプのトラップ宛先を指定できます。これには、バージョン番号 `v2c` または `v3` を指定する必要があります。

## ▼ /etc/snmpd.conf ファイルを更新する

1. `/etc/snmpd.conf` ファイルを編集して、必要に応じて指示子を追加、変更、または削除します。
2. プロンプトで `reboot` コマンドを実行して、ShMM を再起動します。



## 第3章

---

# システムの管理

---

システムの管理は、シェルフ管理カードのコマンド行インタフェース (CLI) を使用するか、または RMCP インタフェースを使用して Ethernet 経由で行います。

この章は、次のセクションで構成されています。

- 60 ページの「シェルフマネージャのコマンド行インタフェース」
- 66 ページの「システムの監視」
- 86 ページの「シェルフマネージャの再初期化」
- 88 ページの「シェルフ管理カードの再プログラミング」

---

## IPMI LAN インタフェース

IPMI LAN インタフェースは、ATCA 仕様に対応するために必要なもので、RMCP (Remote Management Control Protocol) を介したシェルフマネージャとの IPMI メッセージングをサポートします。シェルフとの通信に RMCP を使用すると、システム管理者は ATCA 準拠の任意のシェルフマネージャと対話できます。この低いレベルのインタフェースでは、シェルフマネージャをプロキシとして使用して、システム管理者がシェルフ内の IPM コントローラに IPMI コマンドを実行する機能など、シェルフの IPMI の側面へのアクセスが提供されます。

## IPMI コマンド

標準の IPMI コマンドは、PICMG 3.0 (ATCA 仕様) に記載されています。また、この仕様にはカスタム OEM の IPMI コマンドも記述されています。Sun Microsystems が設計する ATCA ボードでは、これらのコマンドの独自セットが提供されています。これらのコマンドについては、表 3-1 にその一覧を示し、付録 B で説明します。

表 3-1 Sun の OEM IPMI コマンド

コマンド	オペレーション コード	構文
Get Version	0x80	#GET_VERSION
Set Boot Page <sup>†</sup>	0x81	#SET_BOOT_PAGE
Get Boot Page <sup>†</sup>	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

<sup>†</sup> Sun Netra™ CP3010 ノードボードでのみ有効

## シェルフマネージャーのコマンド行インタフェース

シェルフマネージャーのコマンド行インタフェース (CLI) を使用すると、シェルフのインテリジェント管理コントローラ、ボード、およびシェルフマネージャー自体と、テキストコマンドを介して通信できます。この CLI は IPMI をベースとしたコマンドセットで、直接アクセスすることも、より高いレベルの管理アプリケーションまたはスクリプトからアクセスすることもできます。管理者は、Telnet 接続またはシェルフ管理カードのシリアルポートを介して CLI にアクセスできます。オペレータは、CLI を使用して現在の FRU の取り付け状況、現在のセンサーの値、しきい値の設定、最新のイベント、シェルフ全体の健全性など、現在のシェルフの状態に関する情報にアクセスできます。

## コマンド行インタフェースの開始

CLI を使用するには、まずシェルフ管理カード上の Linux オペレーティングシステムにログインします。ログインしたら、コマンド行から実行可能ファイル `clia` を、特定のパラメータを指定して実行します。1 つめのパラメータはコマンド動詞です。`clia` 実行可能ファイルは、シェルフ管理カード上で動作している Linux によって維持管理される仮想の root ファイルシステム上にあります。`clia` 実行可能ファイルは、メインのシェルフマネージャーソフトウェアプロセスに接続してコマンド情報を渡し、結果を取得します。CLI を開始するには、事前にシェルフマネージャーが動作している必要があります。

次に例を示します。

```
# 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)
#
```

パラメータを指定せずに開始すると、`clia` は対話モードになります。このモードでは、プログラムは端末にプロンプトを表示し、ユーザーの入力をパラメータとともに次のコマンドとして受け付け、そのコマンドを実行し、端末に結果を表示します。これは、ユーザーが `exit` コマンドまたは `quit` コマンドを入力するまで繰り返されません。次に例を示します。

```
# 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 コマンド

CLI には、次のコマンドが実装されています。これらのコマンドは、付録 A の 117 ページの「シェルフマネージャの CLI コマンド」に、コマンド名のアルファベット順に、それぞれサブセクションに分けて詳細に記載されています。

表 3-2 シェルフマネージャの CLI コマンドの概要

コマンド	パラメータ	説明
activate	IPMB のアドレス FRU のデバイス ID	指定された FRU を起動します。
alarm	アラームのタイプ	Telco アラームをアクティブにするか、またはクリアします。
board	スロット番号 (省略可能)	ボードに関する情報を表示します。
boardreset	スロット番号	指定された ATCA ボードをリセットします。
busres	サブコマンド (パラメータを指定する)	指定された操作を、バス型の E-キーイング管理対象のリソースで実行します。
console	スロット番号	指定されたスロットのノードボードのコンソールセッションを開きます。
deactivate	IPMB のアドレス FRU のデバイス ID	指定された FRU を停止します。
debuglevel	新しいデバッグレベル (省略可能)	シェルフマネージャの現在のデバッグレベルを取得するか、または新しいデバッグレベルを設定します。
exit/quit		対話モードのインタプリタを終了します。
fans	IPMB のアドレス (省略可能) FRU のデバイス ID (省略可能)	ファンに関する情報を表示します。
flashupdate	サーバーの IP アドレス ファームウェアイメージへのパス名	Netra CP3060 ノードボード上のシステムファームウェアを、指定されたサーバーおよびパス名からダウンロードし、更新します。
fru	IPMB のアドレス (省略可能) FRU のデバイス ID (省略可能) FRU のタイプ (省略可能)	シェルフ内の単体の FRU または FRU のグループに関する情報を表示します。 FRU はタイプまたは親 IPM コントローラに基づいて選択されます。
frucontrol	IPMB のアドレス FRU のデバイス ID オプション	特定の FRU に FRU 制御コマンドを送信します。

表 3-2 シェルフマネージャの CLI コマンドの概要 (続き)

コマンド	パラメータ	説明
frudata	IPMB のアドレス (省略可能) FRU のデバイス ID (省略可能) ブロックオフセットまたはバイトオフセット (省略可能) データ (省略可能)	指定された FRU 上の FRU 情報への raw アクセスを提供します。
frudatar	IPMB のアドレス FRU のデバイス ID ファイル名	指定された FRU の FRU データ領域を読み取り、指定されたファイルにデータを保存します。
frudataw	IPMB のアドレス FRU のデバイス ID ファイル名	指定されたファイルの FRU データを、指定された FRU の FRU データ領域に書き込みます。
fruinfo	IPMB のアドレス FRU のデバイス ID	ユーザーにわかりやすい FRU 情報の出力を提供します。
getfanlevel	IPMB のアドレス (省略可能) FRU のデバイス ID (省略可能)	指定された FRU で制御されるファンの現在のレベルを表示します。
getfruledstate	IPMB のアドレス (省略可能) FRU のデバイス ID (省略可能) LED ID または ALL (省略可能)	FRU の LED の状態を表示します。
gethysteresis	IPMB のアドレス (省略可能) センサー名 (省略可能) センサー番号 (省略可能)	指定されたセンサーの正および負の両方のヒステリシスを表示します。
getipmbstate	IPMB のアドレス IPMB リンク番号 (省略可能)	対象アドレスでの IPMB-0 の現在の状態を表示します。リンク番号が指定されていて、対象の IPMC が IPMB ハブである場合、特定のリンクに関する情報が表示されます。
getlanconfig	チャンネル番号 パラメータ名または番号 (省略可能) 設定セクタ (省略可能)	特定のチャンネルの LAN 構成パラメータを取得して表示します。
getpefconfig	パラメータ名または番号 (省略可能) 設定セクタ (省略可能)	PEF 構成パラメータを取得して表示します。
getsensoreventenable	IPMB のアドレス (省略可能) センサー名 (省略可能) センサー番号 (省略可能)	指定されたセンサーでサポートされているイベントの現在のセンサーイベントマスク値を表示します。
getthreshold、threshold	IPMB のアドレス (省略可能) センサー名 (省略可能) センサー番号 (省略可能)	特定のセンサーに関するしきい値情報を表示します。

表 3-2 シェルフマネージャーの CLI コマンドの概要 (続き)

コマンド	パラメータ	説明
help		サポートされているコマンドのリストを表示します。
ipmc	IPMB のアドレス (省略可能)	シェルフ内の単体またはすべての IPM コントローラに関する情報を表示します。
localaddress		現在のシェルフマネージャーの IPMB アドレスを取得します。
minfanlevel	ファンのレベル (省略可能)	ファンの最小レベルを表示または設定します。
sel	IPMB のアドレス (省略可能) 項目数 (省略可能)	対象の IPM コントローラで維持管理されているシステムイベントログから、最新の項目をいくつか表示します。
sensor	IPMB のアドレス (省略可能) センサー名 (省略可能) センサー番号 (省略可能)	単体のセンサーまたはセンサーのグループに関する情報を表示します。センサーは IPM コントローラのアドレス、番号、または名前で選択されます。
sensordata	IPMB のアドレス (省略可能) センサー名 (省略可能) センサー番号 (省略可能)	特定のセンサーに関する値の情報を表示します。
sensorread	IPMB のアドレス センサーの番号	特定のセンサーに関する raw 値の情報を表示します。センサーの説明が記述されたセンサーデータレコード (SDR) は無視されます。
session		アクティブな RMCP セッションに関する情報を表示します。
setextracted	IPMB のアドレス FRU のデバイス ID	指定された FRU が物理的にシェルフから引き出されていることをシェルフマネージャーに通知します。
setfanlevel	IPMB のアドレス FRU のデバイス ID レベル	指定された FRU で制御されるファンに新しいレベルを設定します。
setfruledstate	IPMB のアドレス FRU のデバイス ID LED ID または ALL LED の操作 LED の色 (省略可能)	指定された FRU の特定の LED またはすべての LED の状態を設定します。

表 3-2 シェルフマネージャの CLI コマンドの概要 (続き)

コマンド	パラメータ	説明
sethysteresis	IPMB のアドレス センサー名またはセンサー番号 設定するヒステリシス (pos または neg) ヒステリシスの値	指定されたセンサーに新しいヒステリシス値を設定します。
setipmbstate	IPMB のアドレス IPMB のバス名 (A または B) IPMB リンク番号 (省略可能) 実行する処理	対象の IPM コントローラ上の IPMB-A または IPMB-B (あるいは特定の IPMB リンク) を、使用可能または使用不可に切り替えます。
setlanconfig	チャンネル パラメータ名または番号 追加パラメータ	指定されたチャンネルの LAN 構成パラメータの値を設定します。
setlocked	IPMB のアドレス FRU のデバイス ID 状態	指定された FRU のロックビットを、指定された状態に設定します (0 – ロック解除、1 – ロック)。
setpefconfig	パラメータ名または番号 設定セクタ (省略可能) パラメータ値	PEF 構成パラメータの新しい値を設定します。
setpowerlevel	IPMB のアドレス FRU のデバイス ID 電力レベルまたは OFF Copy	指定された FRU の電力レベルを設定し、FRU の電源を切り、目的のレベルを現在のレベルにコピーします。
setsensoreventenable	IPMB のアドレス センサー名 センサー番号 グローバルフラグ 表明イベントマスク (省略可能) 非表明イベントマスク (省略可能)	特定のセンサーのイベント有効化マスクを変更します。
setthreshold	IPMB のアドレス センサー名 センサー番号 しきい値のタイプ しきい値	特定のセンサーの特定のしきい値 (上限 / 下限、クリティカル / 非クリティカル / 回復不能) を変更します。
shelf	サブコマンド (パラメータを指定する)	シェルフに関する一般情報を表示します。いくつかのサブコマンドを使用すると、シェルフの属性の設定、および特定領域に関する追加情報の取得が可能です。

表 3-2      シェルフマネージャーの CLI コマンドの概要 (続き)

コマンド	パラメータ	説明
shelfaddress	シェルフのアドレス文字列 (省略可能)	シェルフの FRU 情報内のアドレステーブルのシェルフアドレスフィールドを取得または設定します。
shmstatus		シェルフマネージャーの状態がアクティブとバックアップのどちらであるかを表示します。
showhost	スロット番号	Netra CP3060 ボード上のファームウェアのバージョン情報を表示します。
showunhealthy		シェルフの異常なコンポーネントを表示します。
switchover		バックアップシェルフマネージャーへのスイッチオーバーを開始します。
terminate		シェルフ管理カードを再起動せずにシェルフマネージャーを終了します。
user	サブコマンド (パラメータを指定する)	シェルフマネージャー上の RMCP ユーザーアカウントに関する情報を表示し、ユーザーアカウントを追加、削除、および変更する簡単な方法を提供します。
version		シェルフマネージャーのバージョン情報を表示します。

ほとんどの情報コマンドでは、簡略モードおよび詳細モードの実行形式をサポートしています。これらのモードでは、提供される情報量が異なります。簡略モードがデフォルト (標準) です。詳細モードを選択するには、コマンド行でオプション `-v` を使用します。このオプションは、コマンドの直後でかつ位置を示す引数の前に指定します。



## システムの監視

シェルフマネージャーの CLI には、システムの監視およびシステムの状態表示を行うコマンドが多数用意されています。このセクションでは、システムを監視するさまざまな方法について説明します。詳細は、62 ページの「CLI コマンド」または付録 A の 117 ページの「シェルフマネージャーの CLI コマンド」を参照してください。

## ボードおよび IPMC の情報の表示

ボードの情報には、ATCA スロットに割り当てられている IPMB アドレスの範囲内の各 IPM コントローラに関する情報、およびこれらのコントローラが制御する追加の各 FRU に関する情報が含まれます。ボード上に IPM コントローラが搭載されている PICMG 3.0 システムの場合、IPMB アドレスの範囲は 82h ~ A0h です。

次のタスクについて、使用するコマンドとその出力を例として示します。

- サーバー内のすべてのボードに関する標準の情報を表示する
- ボードに関する詳細情報を表示する
- ボード上のセンサーのリストを表示する
- ボード上のセンサーからのデータを表示する
- サーバー内のすべての IPMC のリストを表示する
- 特定の IPM コントローラに関する情報を表示する
- IPM コントローラに関する詳細情報を表示する

### ● サーバー内のすべてのボードに関する標準の情報を表示する

この例では、物理スロット 1 および 14 にのみボードが存在しています。

```
# 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"
#
```

- ボードに関する詳細情報を表示する

この例では、物理スロット 14 のボードに関する詳細情報を表示します。

```
# 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
#
```

- ボード上のセンサーのリストを表示する

この例では、IPMB アドレス 92 のボード上のセンサーのリストが表示されています。

```
# 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]
#

```

- ボード上のセンサーからのデータを表示する

この例では、IPMB アドレス 92 のボード上のセンサー番号 3 (CPU2 Temp) に関する情報が表示されています。

```

# 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]

```

- サーバー内のすべての IPMC のリストを表示する

この例では、ipmc コマンドの標準的な出力を表示します。

```
# 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)
```

- 特定の IPM コントローラに関する情報を表示する

この例では、アドレス 9C の IPM コントローラに関する基本的な情報が表示されています。

```
# 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)
#
```

## ● IPM コントローラに関する詳細情報を表示する

この例では、アドレス 9C の IPM コントローラに関する詳細情報が表示されています。

```
# 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"
#
```

## FRU 情報の表示

パラメータを指定せずに `clia fru` コマンドを実行してシステム内のすべての FRU に関する情報を表示するか、または FRU のアドレスや、必要に応じて FRU ID を指定して特定の FRU に関する情報を表示することができます。シャースのスロット番号と物理アドレスおよび IMPB アドレスとのマッピングについては、13 ページの「物理アドレスから論理スロットへのマッピング」を参照してください。

次の例では、「20: FRU # 1」の行の 20 はミッドプレーンの IPMB アドレスで、1 は FRU のデバイス ID です。

```
# 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 情報の配置

図 3-1 に、IPMI FRU 情報の構成を示します。各パーティションは、特定のタイプのデータを提供します。

- 共通ヘッダーには、領域オフセットが含まれます。
- 内部使用領域は、独自データの格納に使用されます。
- シャーシ情報領域には、シャーシタイプ、パーツ番号、シリアル番号が含まれます。
- ボード情報領域には、メーカーのタイムスタンプ、メーカー名、製品名、パーツ/シリアル番号が含まれます。
- 製品情報領域には、メーカー名、製品名、パーツ/シリアル番号、バージョンが含まれます。
- マルチレコード領域には、動的データが含まれます。

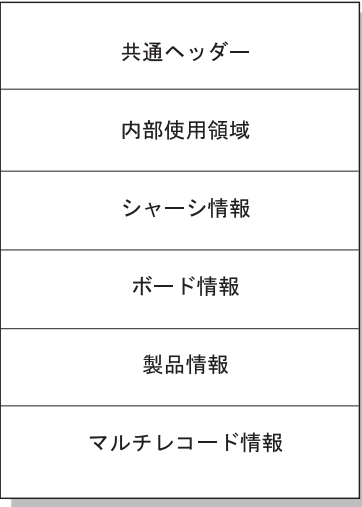


図 3-1 IPMI FRU 情報の配置

## 環境 FRU

環境 FRU には、ミッドプレーン、シェルフ管理カード、ファン、電源入力モジュール (PEM)、およびシェルフアラームパネル (SAP) が含まれます。すべての環境 FRU には、IPMI FRU 情報のみが構成されています。この情報は、他社ベンダーによって提供され、プログラムされています。

ミッドプレーンの FRU 情報には、他社のパーツ番号だけでなく Sun のパーツ番号も含まれています。また、ミッドプレーン FRU 情報のマルチレコード領域には、スロット、VLAN、vtag、その他のデータなど、システムの付加情報が Sun によって追加されます。ミッドプレーン FRU 情報は、2 つの同一の EEPROM に格納されています。一方の EEPROM の内容が変更されると、もう一方の EEPROM も自動的に変更されます。

## ブレード FRU

Netra CT 900 サーバーのスロット 7 および 8 にあるハブボードには、IPMI FRU 情報のみが構成されています。Sun のノードボードには 2 つの別個の EEPROM があり、一方には IPMI FRU 情報が、もう一方には Sun の FRU 情報が保存されています。

## 例

次のタスクについて、使用するコマンドとその出力を例として示します。

- シェルフ内のすべての FRU に関する標準の情報を表示する
  - アドレス 9C にあるすべての FRU に関する標準の情報を表示する
  - アドレス 20 にある FRU 1 に関する詳細情報を表示する
  - FRU 情報を raw 形式で表示する
  - FRU 情報をユーザーにわかりやすい形式で表示する
- シェルフ内のすべての FRU に関する標準の情報を表示する

この例では、引数を指定せずに fru コマンドを実行した場合に表示される標準の FRU 情報を示します。

```
# 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"

```

- アドレス 9C にあるすべての FRU に関する標準の情報を表示する

この例では、物理アドレス 9C のすべての FRU に関する FRU 情報のみが表示されています。

```

# 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"
#

```

- アドレス 20 にある FRU 1 に関する詳細情報を表示する

この例では、物理アドレス 20 の FRU デバイス ID 1 に関して、詳細な FRU 情報が表示されています。

```
# 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

```

### ● FRU 情報を raw 形式で表示する

この例では、すべての FRU および特定の FRU に関する FRU 情報が raw 形式で表示されています。

```

# 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

```

### ● FRU 情報をユーザーにわかりやすい形式で表示する

この例では、FRU 情報をユーザーにわかりやすい形式で示します。

```

# 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

## シェルフ情報の表示

clia shelf コマンドに有効なパラメータの 1 つを指定して実行すると、主要なシェルフ FRU の FRU 情報を表示できます。また、シェルフの現在の動作データを取得することも、シェルフ FRU 情報のいくつかのフィールドを変更することもできます。有効な shelf パラメータは次のとおりです。

- cooling\_state または cs
- fans\_state または fs
- address\_table または at
- power\_distribution または pd
- power\_management または pm
- pci\_connectivity または pcic
- ha\_connectivity または ha
- h110\_connectivity または h1110c
- point-to-point\_connectivity または ppc

詳細は、224 ページの「シェルフ FRU 情報の表示」を参照してください。

### 例

次のタスクについて、使用するコマンドとその出力を例として示します。

- シェルフの冷却状態を表示する
- シェルフのファンの状態を表示する
- アドレステーブルを表示する
- 電源管理情報を表示する
- 配電情報を表示する

#### ● シェルフの冷却状態を表示する

この例では、シェルフの冷却状態を表示するコマンドとその出力を示します。

```
# 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)
```

- シェルフのファンの状態を表示する

この例では、ファンの状態を表示するコマンドとその出力を示します。

```
# 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)

#
```

- アドレステーブルを表示する

この例では、シェルフのアドレステーブルを表示するコマンドとその出力を示します。

```
# 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
```

## ● 電源管理情報を表示する

この例では、コマンドとその出力の例を示します。

```
# 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

```

#

### ● 配電情報を表示する

この例では、シェルフの配電情報を取得するコマンドとその出力を示します。

```

# 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: 0xfe  
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

```

#

# シェルフマネージャの再初期化

このセクションでは、U-Boot 環境変数、フラッシュメモリー内のファイルシステム、およびシェルフ管理カード上のログインパスワードを再初期化する方法について説明します。

## U-Boot 環境の再初期化

U-Boot 環境変数は、シェルフ管理カードの EEPROM に格納されています。U-Boot 環境変数を出荷時のデフォルト設定に戻すには、まず EEPROM に格納されている環境変数を消去してから、シェルフ管理カードのリセットまたは電源の再投入を実行する必要があります。

### ▼ U-Boot 環境を再初期化する

1. U-Boot プロンプトで次のコマンドを入力して、EEPROM を消去します。

```
ShMM # eeeprom write 80400000 0 1000

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

2. 次のように、シェルフ管理カードをリセットします。

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

CPU: Au1550 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: Au1X00 ETHERNET
Hit any key to stop autoboot: 0

ShMM #
```

3. 次のコマンドを使用して、これらの環境設定を保存します。

```
ShMM # saveenv
```

## ファイルシステムの再初期化

ファイルシステムはフラッシュメモリに格納されているため、非常に簡単に出荷時のデフォルト設定にリセットすることができます。U-Boot には、`flash_reset` という環境変数があります。この変数を `y` に設定してシステムを起動すると、ファイルシステムが出荷時のデフォルト設定に再初期化されます。

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

`flash_reset` 変数は、フラッシュメモリの再初期化後のシステムの起動時に、自動的に `n` に設定されます。`bootcmd` コマンドを実行すると、Linux カーネルの起動が開始されます。ファイルシステムの再初期化は、このプロセス中に行われます。次の出力が、コンソールに表示されます。

```
/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...
```

## ログインパスワードのリセット

シェルフ管理カードの出荷時のデフォルトのログイン設定は、ユーザー ID が `root` で、パスワードは設定されていません。ユーザーは、シェルフマネージャーの構成時にパスワードを変更するように勧められます。新しいパスワードを忘れた場合は、U-Boot 変数の `password_reset` を使用してパスワードを出荷時のデフォルト設定にリセットすることができます。この変数を `y` に設定してシステムを起動すると、`root` のパスワードが消去されます。

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

起動中に、次の出力がコンソールに表示されます。

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

---

## シェルフ管理カードの再プログラミング

シェルフ管理カードの再プログラミングには、このカードに格納されているいくつかのファームウェアイメージの再プログラミングが含まれます。イメージを再プログラミングする方法によって、イメージのセットを概念的に 2 つのグループに分けることができます。これらのグループは次のとおりです。

- U-Boot、Linux カーネル、および Linux のルートファイルシステム (RFS) のイメージ

これらのイメージは、通常、信頼できるアップグレード処理を使用して再プログラミングされます。また、カーネルおよび RFS については、TFTP サーバーからロードすることで U-Boot プロンプトから再プログラミングすることもできます。

- CPLD (Complex Programmable Logic Device) のイメージ

このイメージは、シェルフ管理カードのコマンド行から、特別なコマンド行ユーティリティー `cpldtool` を使用して再プログラミングされます。

以降では、これらのイメージを再プログラミングする方法について詳細に説明します。

# 信頼できるファームウェアアップグレード処理

Monterey Linux オペレーティングシステムでは、動作および機能しているシェルフ管理カード上のファームウェアイメージに対して、信頼できるアップグレード処理が提供されています。この手続きは、U-Boot ファームウェア、Linux カーネル、および Linux RFS (またはこれらの 3 つのイメージの任意の組み合わせ) のアップグレードをサポートしています。問題のある U-Boot ファームウェアイメージのインストール、シェルフ管理カードの起動不可、シェルフマネージャーの起動不可などが原因で、ソフトウェアのアップグレードの試みが失敗すると、信頼できるアップグレード処理は、永続フラッシュメモリー内にある以前のバージョンのファームウェアに自動的に戻します。

シェルフ管理カードのフラッシュメモリーは、2 つの領域に分けられます。これらの領域のいずれかに安定したファームウェアセットが確立されると、この領域は「永続」領域として指定されます。新しいファームウェアがインストールされると、もう一方の領域に入ります。この領域は、最初は「暫定」として指定されます。暫定領域内の新しいファームウェアセットの妥当性検査が行われると、この領域が永続領域として指定され、その後のアップグレードサイクルでこの処理が最初からやり直されるまで、この領域が使用され続けます。

信頼できるアップグレードのハードウェア機構では、暫定フラッシュにインストールされている内容にかかわらず、シェルフ管理カードが常に、完全に機能するソフトウェアコピーまたは十分に正常なソフトウェアコピーから起動し、アップグレードセッションで問題が発生した場合はこれを判断し、その結果、永続フラッシュ内の安全なソフトウェアコピーに戻すための適切な修正措置を実行することが保証されます。

信頼できるアップグレードのハードウェア機構は、より高いレベルで、`/var/upgrade/status` 内の非揮発性ファイルへのアップグレードセッションの状態のログに基づいて、ソフトウェアプロトコルによってサポートされます (92 ページの「信頼できるアップグレード処理の状態ファイル」を参照)。ソフトウェアプロトコルは、特定のアプリケーションが必要とする場合があるカスタム「フック」スクリプトで定義された処理など、必要なすべての処理が正常に完了するまで、信頼できるアップグレードが終了しないことを保証します。

Monterey Linux に実装されている信頼できるアップグレード処理は、シェルフ管理カード上で動作する組み込みアプリケーションには影響を与えません。この処理では十分な「フック」セットが提供されており、これによって、シェルフ管理カード上で動作する特定のアプリケーションが、信頼できるアップグレードの適切な時点でカスタム処理を確実に実行することができます。このセクションの以降の部分では、これらのフックを使用して実装された IPM Sentry Shelf Manager ファームウェアの信頼できるアップグレードの準備について説明します。

## フラッシュのパーティション分割

シェルフ管理カードは、MIPS プロセッサ上で動作しているソフトウェアの制御下で、システムメモリーマップ内のフラッシュの下位半分と上位半分の交換を可能にするハードウェア機構を提供します。この機能は、フラッシュ内のソフトウェアイメージに対する信頼できるアップグレード処理をサポートするために実装されています。信頼できるソフトウェアアップグレード処理では、ソフトウェアの 2 つのコピーがフラッシュデバイスの上位半分と下位半分に 1 つずつ含まれていることを前提としています。すべてのシェルフ管理カードは、パーティション分割された状態で出荷されます。フラッシュデバイスは 2 等分され、それぞれがシェルフマネージャーソフトウェアのコピーを 1 つ保持するために使用されます。

U-Boot 環境変数 `reliable_upgrade` (18 ページの「U-Boot 環境変数」を参照) は、Linux 層で使用され、信頼できるアップグレード処理が使用できるかどうかを決定します。この変数に `y` を設定して、`bootargs` カーネルパラメータ文字列で Linux カーネルに渡す必要があります。Linux Flash MTD 層は、フラッシュパーティションの初期化時に `reliable_upgrade` パラメータを確認し、パラメータの値 (およびシェルフ管理カードに搭載されているフラッシュデバイスのサイズ) に応じて、適切な方法でフラッシュデバイスをパーティション分割します。

このセクションは、シェルフ管理カードが、2 つの別個のフラッシュ領域を含めて、信頼できるアップグレードをサポートするように構成されていることを前提としています。表 3-3 に、この構成 (`reliable_upgrade=y`) のシェルフ管理カードで維持管理されるフラッシュパーティションの概要を示します。

表 3-3 16M バイトのフラッシュパーティション (`reliable_upgrade=y`)

フラッシュ内のオフセット (M バイト単位)	サイズ (M バイト単位)	デバイスノード	マウントの場所 (起動時)	内容
0	0.5	<code>/dev/mtdchar10、 /dev/mtdblock10</code>	<code>/var/upgrade</code>	<code>/var/upgrade</code> JFFS2 ファイルシステムの後半
$0.5 + (\text{FLASH\_SIZE} - 16) / 2$	1.5	<code>/dev/mtdchar5、 /dev/mtdblock5</code>	マウントされま せん	もう一方の <code>/var</code> JFFS2 ファイルシステム
$\text{FLASH\_SIZE} / 2 - 62$	1	<code>/dev/mtdchar6、 /dev/mtdblock6</code>	マウントされま せん	もう一方の <code>/etc</code> JFFS2 ファイルシステム
$\text{FLASH\_SIZE} / 2 - 53$	1	<code>/dev/mtdchar7</code>	マウントされま せん	もう一方の Linux カー ネルイメージ
$\text{FLASH\_SIZE} / 2 - 44$	0.25	<code>/dev/mtdchar8</code>	マウントされま せん	もう一方の U-Boot ファームウェアイメージ
$\text{FLASH\_SIZE} / 2 - 3.754.25$	3.75	<code>/dev/mtdchar9</code>	マウントされま せん	もう一方の Linux ルー トファイルシステム (rfs) イメージ

表 3-3 16M バイトのフラッシュパーティション (reliable\_upgrade=y) (続き)

フラッシュ内のオフセット (M バイト単位)	サイズ (M バイト単位)	デバイスノード	マウントの場所 (起動時)	内容
FLASH_SIZE/28	0.5	/dev/mtdchar10、 /dev/mtdblock10	/var/upgrade	/var/upgrade JFFS2 ファイルシステムの前半
FLASH_SIZE - 7.58.5	1	/dev/mtdchar0、 /dev/mtdblock0	/var	/var JFFS2 ファイルシ ステム
FLASH_SIZE - 610	1	/dev/mtdchar1、 /dev/mtdblock1	/etc	/etc JFFS2 ファイルシ ステム
FLASH_SIZE - 5.11	1	/dev/mtdchar2	マウントされま せん	Linux カーネルイメー ジ
FLASH_SIZE - 412	0.25	/dev/mtdchar3	マウントされま せん	U-Boot ファームウェア イメージ
FLASH_SIZE - 3.712.255	3.75	/dev/mtdchar4	マウントされま せん	Linux ルートファイルシ ステム (rfs) イメージ

## /var/upgrade ファイルシステム

89 ページの「信頼できるファームウェアアップグレード処理」で説明したように、reliable\_upgrade が y の場合、Monterey Linux は /var/upgrade に JFFS2 ファイルシステムとして 1M バイトのパーティションをマウントします。このファイルシステムは、信頼できるアップグレード処理の状態ファイルをホストするために使用されます (92 ページの「信頼できるアップグレード処理の状態ファイル」を参照)。

/var/upgrade の JFFS2 パーティションは隣接しない 2 つのフラッシュブロック (各 0.5M バイト) で構成されており、1 つはフラッシュデバイスの下位半分に、もう 1 つは上位半分に存在することに注意することが重要です。このように /var/upgrade を実装するために、Monterey Linux では Linux MTD 層および JFFS2 層の機能を利用して、隣接しないフラッシュセクター内のファイルシステムをサポートします。

信頼できるアップグレード処理を目的として /var/upgrade が機能するための JFFS2 ファイルシステムのもう 1 つの特徴は、フラッシュセクター番号またはフラッシュ内の絶対オフセットに基づく依存関係 (リンクされたリストなど) が JFFS2 内部構造では作成されないことです。代わりに、パーティション上にファイルシステムをマウントするときに、JFFS2 はパーティションを構成するすべてのフラッシュセクターを走査し、内部の RAM 内表現でファイルシステムの論理的内容を再作成します。この特徴によって、ShMM の起動元がフラッシュのどちらの半分にかかわらず、Linux が /var/upgrade を JFFS2 ファイルシステムとしてマウントし、ファイルシステムの以前の内容を使用できることが保証されます。

## 信頼できるアップグレード処理の状態ファイル

信頼できるソフトウェアアップグレード処理では、最新のアップグレード処理セッションの状態が、専用のファイルシステム (`/var/upgrade`) 内に存在する `/var/upgrade/status` に維持管理されます。これは、どのフラッシュから **ShMM** が起動したかにかかわらず、**Linux** によってマウントされます。このファイルが存在する場合、ファイルには、現在進行中か、または最近完了したいずれかのアップグレード処理セッションの状態が格納されています。

`/var/upgrade/status` は、改行を終端とするレコードが 1 つ以上含まれる **ASCII** 形式のファイルで、各レコードにはアップグレード処理の特定の手順の状態が記述されています。レコード行の形式は、次のとおりです。

`<step>: <status>`

`step` は、1 ～ 14 の範囲の整数で、手順 14 は完了したアップグレードセッションに対応します。`status` は、人間が読み取り可能な文字列で、アップグレード処理セッションの現在の手順の状態を表します。

信頼できるアップグレードユーティリティ (92 ページの「信頼できるアップグレードユーティリティ」を参照) はこの状態ファイルを使用して、信頼できるアップグレード処理のハードウェア機構の上にソフトウェアプロトコルを維持管理し、アップグレード処理の状態を確実に判断して適切に処理を進めます。

## 信頼できるアップグレードユーティリティ

最新のアップグレードの状態を確認するだけでなく、信頼できるアップグレード処理を実行する手段として、特別なユーザー空間の信頼できるアップグレードユーティリティが提供されています。

このユーティリティは、スーパーユーザー (`root`) アカウントからのみ呼び出すことができます。スーパーユーザー以外のアカウントからこのユーティリティを実行しようとしても、拒否されます。

このユーティリティは、実行の最初の手順として、**U-Boot** 環境変数の `reliable_upgrade` (90 ページの「フラッシュのパーティション分割」を参照) がカーネルパラメータ文字列として **U-Boot** から **Linux** カーネルに渡されるときに、`y` に設定されていることを確認します。この確認に失敗すると、ユーティリティはただちに処理を中止し、適切なエラーコードで終了します。

`-s`、`-c`、または `-f` オプションのいずれかを指定して呼び出すと、ユーティリティは信頼できるアップグレード処理の実行に使用されます。アップグレード処理の間、ユーティリティは、アップグレード処理の手順を進める際に実行した各操作の状態を `/var/upgrade/status` に記録します。ユーティリティが失敗を検出すると、`/var/upgrade/status` にアップグレード処理が正常に完了しなかったことを示すレコードを追加して、適切なエラーコードで終了し、信頼できるアップグレード処理が打ち切られます。

このユーティリティーは情報メッセージを stdout に出力します。-v 指示子をサポートする任意のオプションにこの指示子を指定すると、情報メッセージがより詳細になります。このユーティリティーは、エラーメッセージをすべて stderr に出力します。

このユーティリティーでは、次の構文が使用されます。

- `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`

ここでは、パラメータは次のように定義されます。

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

信頼できるアップグレード処理を開始します。シェルフマネージャーがサポートされている場合、この手順では次の処理も実行されます。

- コピーするイメージをローカルにまたはネットワークを介して取得する
- イメージを暫定フラッシュにコピーする
- ShMM でシェルフマネージャーのインスタンスが動作している場合は、これを終了する
- 非揮発性データを暫定フラッシュにコピーする
- シェルフ管理カードをリセットし、暫定フラッシュから起動するように指示する

最後の手順のため、通常は `rupgrade_tool -s` の呼び出しからは復帰せず、代わりにシェルフ管理カードがリセットされます。`rupgrade_tool -s` から復帰する場合は、シェルフ管理カードをリセットし、暫定フラッシュから起動する手順に進む前に、信頼できるアップグレード処理が失敗し、処理が打ち切られたことを示しています。

ユーティリティーがアップグレード処理の最初の手順を開始する前に、`/var/upgrade/status` ファイルは削除されます (92 ページの「信頼できるアップグレード処理の状態ファイル」を参照)。つまり、`rupgrade_tool -s` が呼び出されるとすぐに、前回のアップグレード処理セッションの状態は失われ (存在する場合)、この内容は失われ、新しいアップグレード処理セッションの状態を上書きされます。

`rupgrade_tool -s` の呼び出しに、`--dst=src` 指示子を 1 つ以上指定できます。このような指示子はそれぞれ、インストールするアップグレードイメージファイルの名前や、ファイルをインストールするシェルフ管理カードのフラッシュの場所を定義します。

*dst* は、新しくインストールするアップグレードイメージの宛先を定義するもので、次の値を指定できます。

- **u** – 暫定 U-Boot ファームウェアイメージのパーティション (/dev/mtdchar3) 内の U-Boot イメージをアップグレードします。
- **k** – 暫定 Linux カーネルイメージのパーティション (/dev/mtdchar2) 内の Linux カーネルイメージをアップグレードします。
- **r** – 暫定ルートファイルシステムイメージのパーティション (/dev/mtdchar4) 内のルートファイルシステムイメージをアップグレードします。

*src* には、*dst* で指定した暫定フラッシュパーティションにコピーするアップグレードイメージファイルを指定します。

イメージのアップグレードは次のように動作します。指定したそれぞれの *src* イメージが、指定したコピープロトコルを使用してシェルフ管理カードにコピーされます。-d 指示子を指定しない場合は、まずイメージがシェルフ管理カードの RAM ファイルシステム (具体的には /tmp ディレクトリ) にコピーされてから、フラッシュに移動されます。つまり、フラッシュ内の宛先パーティションにコピーされ、その後 RAM ファイルシステムから削除されます。rupgrade\_tool -s の呼び出しに -d 指示子を指定した場合、/tmp ディレクトリへの中間コピーは省略され、イメージはフラッシュ内の宛先に直接コピーされます。この指示子は、RAM ファイルシステムへの中間コピーに必要な実行時メモリーがシェルフ管理カードで不足しているような状況で使用します。

-d 指示子が指定されていない場合、信頼できるアップグレード処理は特別なスクリプトを呼び出します。このスクリプトの主な目的は、/tmp ディレクトリにイメージがコピーされたあとで、それらのイメージの妥当性検査を行うことです。-d 指示子が指定されていると、このような妥当性検査は実行されません。

現在、シェルフマネージャーで提供されているスクリプト

/etc/upgrade/step4vshm では、この特別なイメージの妥当性検査手順は実行されません。ただし、現在の rupgrade\_tool の呼び出しでイメージが提供されていないフラッシュパーティションにデータを書き込む役割を担います。これは、部分アップグレードの状況で発生します。これらのパーティションは、現在の永続フラッシュから暫定フラッシュにコピーされます。たとえば、現在の部分アップグレードで新しい RFS イメージのみが提供されている場合、このスクリプトは U-Boot およびカーネルパーティションを永続フラッシュから暫定フラッシュにコピーします。

最初のイメージが宛先にインストールされるとすぐに、ユーティリティーは 2 つめ以降のイメージ (存在する場合) に順に進み、提供されたイメージファイルがすべて正常にフラッシュにインストールされるまで処理を続けます。イメージの正常なインストールに失敗すると、失敗したイメージを省略して次のイメージに進むことはせずに、アップグレード処理はただちに終了します。

この方法によって、ユーザーはシェルフ管理カードファームウェアの 3 つの部分 (U-Boot、カーネル、および RFS イメージ) を別個にアップグレードできます。ただし、明示的に更新しない部分は、永続フラッシュからコピーされることに注意してください。

次のいずれかのアップグレード方法を使用することをお勧めします。

- 3つのすべてのパーティションを明示的にアップグレードします。
- 2つまでのパーティションを明示的にアップグレードする場合は、`-d` 指示子を省略します。この場合、前述した特別なスクリプトによって、事実上、どのアップグレードも、3つのすべてのパーティションを対象にした完全アップグレードになることが自動的に保証されます。

*protocol* には、指定した各 *src* ファイルをシェルフマネージャーに転送するために使用されるファイルコピープロトコルを指定します。次の値を指定できます。

- `no` – コピーは実行されません。このプロトコルは、信頼できるアップグレード処理が開始される前に、指定したすべての *src* ファイルが `/tmp` ディレクトリに転送されていることを前提としています。このプロトコルの選択は、`-d` オプションとともに使用することはできません。
- `cp:dir` – 簡単なコピー。このプロトコルは、指定したすべての *src* ファイルがシェルフマネージャーのローカルファイルシステム内の指定したディレクトリから、`cp` コマンドによってコピーされることを前提としています。たとえば、NFS がマウントされたファイルシステムからのアップグレードイメージのインストール、または JFFS2 ファイルシステムからのアップグレードイメージのインストールの場合にも、このプロトコルは役立つ可能性があります。
- `ftp:server:dir:user[:pwd]` – 遠隔 FTP サーバーからのコピー。このプロトコルは、指定したすべての *src* ファイルが、*server* でホスト名または IP アドレスとして指定された FTP サーバーホストから、シェルフ管理カードにコピーされることを前提としています。すべてのイメージは、遠隔 FTP サーバー上の *dir* で指定したディレクトリに存在する必要があります。FTP 接続は、ユーザーパラメータで指定したアカウントと、オプションの `pwd` パラメータで指定したパスワードを使用して確立されます。`pwd` を指定しないと、ユーティリティーがパスワードを求めるプロンプトを表示します。

シェルフマネージャーへのイメージのコピーに失敗すると、ユーティリティーは失敗したイメージを省略して次のイメージに進むことはせずに、そのアップグレード処理を終了します。

`-s` オプションを指定して各暫定フラッシュパーティションをアップグレードする場合は、イメージの妥当性が確認されてから *src* イメージがフラッシュに移動される直前に、アップグレードされるパーティションに書き込み権限が与えられます。この書き込み権限は、イメージ全体がフラッシュに移動された直後に、パーティションから削除されます。シェルフマネージャーの起動時には、U-Boot、Linux カーネル、およびルートファイルシステムのイメージを含むすべてのパーティションが読み取り専用であるという事実と合わせて考えると、アプリケーションが重要な起動イメージを誤って消去する可能性はないことが保証されます。

指定したイメージがフラッシュメモリー内のそれぞれの宛先にすべてインストールされたあとで、ユーティリティーは「フック」スクリプトを起動します。このスクリプトは、ShMM のアップグレードウォッチドッグタイマー (WDT) をアクティブにして、アップグレードイメージがフラッシュにすでにインストールされ

ていても、アップグレード処理が信頼できるアップグレード処理のハードウェア機構をまだ開始していない時点で、アプリケーションが必要とするカスタム処理を使用可能にします。

フックスクリプト (`/etc/upgrade/step4hshm`) は、シェルフマネージャーに付属しています。次の処理を実行します。

- シェルフマネージャーを終了し、シェルフの再起動を行わずにバックアップ `ShMM` へのスイッチオーバーを実行します。ATCA ウォッチドッグタイマーが停止します。
- 暫定の `/etc` および `/var` フラッシュパーティションをマウントし、パーティション上のすべてのファイルを消去します。
- 必要に応じて、`/etc` ディレクトリの現在の内容を暫定 `/etc` フラッシュパーティションにコピーします。
- 必要に応じて、現在の非揮発性のシェルフマネージャー情報をディレクトリ `/var/nvdata` から暫定 `/var` ファイルシステムにコピーします。または、必要に応じて `/var` ディレクトリ全体を暫定 `/var` フラッシュパーティションにコピーします。
- 次の起動まで一時的に起動遅延を 0 に設定します。これは、次の起動時間を最小限にし、信頼できるアップグレードのウォッチドッグタイマーが途中で期限切れになることを回避するために行います。

このスクリプトは、サブシェルとして起動され、1 つのパラメータが渡されます。これは、`args` で指定される文字列、または `args` なし (空の文字列) のいずれかです。指定するパラメータはスクリプトの動作モードを定義するもので、これによって永続フラッシュパーティションから暫定フラッシュパーティションへの非揮発性情報のコピーなどの操作が決定します。このスクリプトは、次の `args` 値を受け取り、対応する処理を実行します。

- パラメータの指定なし - スクリプトは暫定 `/etc` と暫定 `/var` の両方のディレクトリを消去してから、シェルフマネージャーの非揮発性情報を `/var/nvdata` ディレクトリから暫定 `/var` パーティションにコピーします。これは、デフォルトの動作モードです。この場合、非揮発性データは保持されますが、シェルフマネージャーの構成ファイルは新しい RFS イメージから取得されます。
- `erase` - スクリプトは暫定 `/etc` と暫定 `/var` の両方のディレクトリを消去します。これらのディレクトリは次の起動時に RFS のデフォルト値から復元されます。現在のシェルフマネージャーの非揮発性データおよび構成は保持されません。
- `etc_copy` - スクリプトは暫定 `/etc` と暫定 `/var` の両方のディレクトリを消去してから、`/etc` の内容と、`/var/nvdata` ディレクトリの非揮発性情報を、暫定フラッシュパーティションにコピーします。この場合、非揮発性データとシェルフマネージャーの構成ファイルの両方が保持されます。
- `copy` - スクリプトは暫定 `/etc` および暫定 `/var` の両方のディレクトリを消去してから、`/etc` および `/var` ディレクトリのすべての内容を暫定パーティションにコピーします。この場合、構成ファイルだけでなく、`/var/bin` に配

置された実行可能ファイルもコピーされ、RFS イメージからの同名の実行可能ファイルを上書きします。この動作モードは、/var/bin ディレクトリに、アップグレード後も保持する必要があるいくつかの特別な実行可能ファイル (たとえば、特定のバージョンのシェルフマネージャーまたはその他のユーティリティ) が含まれている場合に、便利です。

このスクリプトは正常に終了すると 0 を返し、失敗すると 0 以外の値を返します。0 以外の値が返されると、アップグレード処理は終了します。

ユーティリティは、タイムアウト期間が 12.8 秒に指定されたアップグレード WDT を開始します。このタイムアウト期間は、リセット後に起動する任意のソフトウェアが、`rupgrade_tool -c` (アップグレード WDT がアクティブの場合にストローブ信号を送信する) を呼び出すことができる時点まで進むために十分な時間だと考えられており、それまでにアップグレード WDT にストローブ信号を送信する必要はありません。ユーティリティは、ShMM をリセットする直前にアップグレード WDT のストローブを実行します。

#### ■ -c [-v]

ShMM が暫定フラッシュから起動されたあとで、信頼できるアップグレード処理に進みます。`rupgrade_tool -c` の呼び出しは、/etc/rc スクリプトから実行されます。以降で説明するように、`rupgrade_tool -c` で検出される特定の状況が、アップグレード処理での失敗を示すことがあり、修正措置が必要になります。その中には、結果として ShMM のソフトリセットが必要になるものもあります。つまり、`rupgrade_tool -c` の呼び出しから復帰せずに、代わりに ShMM のリセットが発生する場合があります。リセットが発生すると、ShMM は永続フラッシュにインストールされているソフトウェアに戻されます。

アップグレード WDT がアクティブで、`rupgrade -c` を呼び出す前の任意の手順でタイムアウトが発生している場合は、ShMM がすでに永続フラッシュ内のソフトウェアに戻されていることを意味します。この状況では、ユーティリティはアップグレード WDT を無効にして永続フラッシュの使用に戻り、アップグレード処理を終了します。

アップグレード WDT がアクティブでタイムアウトしていない場合は、この時点までに ShMM が暫定フラッシュから正常に起動されたことを意味します。ユーティリティは、アップグレード WDT をストローブし、アップグレード処理セッションが進行中であることを示すリターンコード 0 で終了します。

アップグレード WDT がアクティブでなくても、/var/upgrade/status ファイルの内容がアップグレード処理がまだ進行中であることを示している場合は、アップグレード処理のいずれかの手順で電源の再投入が発生したために ShMM が再起動されたことを意味します。この状況では、ユーティリティは、アップグレード WDT がアクティブでタイムアウトした場合と同じ修正措置を実行します。

最後に、アップグレード WDT がアクティブでなく、/var/upgrade/status が存在しないか、アップグレード処理が完了したこと (成功または失敗のいずれか) を示している場合、ユーティリティは進行中のアップグレード処理が存在していないことを示す戻り値 1 で終了します。

■ **-f [--hook=args] [-v]**

アップグレード処理を完了します。rupgrade\_tool -f の呼び出しは、シェルフマネージャーが自身の初期化を正常に完了したあとに、シェルフマネージャー内部から実行されます。シェルフマネージャーが自動的に起動されない場合、この呼び出しは /etc/rc スクリプトの最後に実行されます。

rupgrade\_tool -f は呼び出されるとすぐに、アップグレード WDT をストローブし、新しい永続フラッシュの確立およびアップグレード WDT の無効化に進みます。

完了の前に、ユーティリティはアップグレード処理の正常な完了を示すレコードで /var/upgrade/status を更新し、値 0 で終了します。

■ **-w [-f]**

最新のアップグレード処理の現在の状態を出力します。基本的に、このオプションは /var/upgrade/status ファイルの内容を stdout にダンプします。

rupgrade\_tool -w は、アップグレード処理が正常に完了した場合には値 0 を、アップグレード処理が失敗した場合には値 1 を、/var/upgrade/status が存在しない場合にはエラーコードを返します。

-f 指示子を指定すると、rupgrade\_tool -w は、終了前に /var/upgrade/status ファイルを削除します。

■ **-S [-v]**

アップグレード WDT をストローブします。rupgrade\_tool -S は、新しくインストールされたソフトウェアが自身の妥当性を検査する際のシェルレベルのインタフェースとして使用されます。

rupgrade\_tool -S は値 0 を返します。

■ **-u**

正常なアップグレードセッションを取り消して、以前の永続フラッシュデバイスに戻します。

rupgrade\_tool -u を使用すると、ShMM が再起動されます。

■ **-h**

stdout にヘルプを出力します。

## 信頼できるアップグレードユーティリティを使用する状況

ShMM のアップグレードを実行するには、次の順番で信頼できるアップグレードユーティリティが使用されます。

1. ユーザーが rupgrade\_tool -s を呼び出して、アップグレード処理を開始します。この呼び出しは、ShMM シリアルコンソールからローカルに実行すること、あるいは telnet、rsh、ssh、または同等の任意のユーティリティを使用してネットワーク経由で遠隔から実行することもできます。

2. ユーザーは、`rupgrade_tool -s` による ShMM の再起動を待機します。ユーザーがシリアルコンソールにローカルに接続している場合は、U-Boot ファームウェアや Linux がシリアルコンソールに出力するメッセージによって、再起動の状態がよくわかります。遠隔から ShMM に接続している場合、再起動の状態はあまり明確ではありません。たとえば、Telnet 接続がシェルフ管理カードの再起動時にタイムアウトします。ユーザーはアップグレード処理が正常に実行されたと想定するか、またはアップグレードセッションの完了に必要な一定の時間を待機してから、前述の遠隔シェルフツールのいずれかを使用して再度遠隔から `rupgrade_tool -w` を呼び出し、アップグレードセッションの状態を調べることもできます。待機する時間は、イメージの妥当性検査スクリプトが実行する処理だけでなく、アップグレードイメージのサイズ、およびシェルフ管理カードへのイメージの送信に使用されるコピープロトコルに応じて異なります。
3. シェルフ管理カードで、起動スクリプト `/etc/rc` が無条件に `rupgrade_tool -c` を呼び出します。この呼び出しに対して、進行中のアップグレードが存在しないことを示す値 1、またはアップグレードセッションが失敗したことを示すエラーコード値が返されると、起動スクリプトは通常モードの起動処理に進みます。ただし、進行中のアップグレードセッションが存在することを示す値 0 が返されると、起動スクリプトは新しくインストールされたソフトウェアの妥当性検査に進み、妥当性検査の時間がアップグレード WDT のタイムアウト期間よりも長くなる場合に備えて、処理の途中で `rupgrade_tool -S` を呼び出してアップグレード WDT をストローブします。最後に、シェルフマネージャーを起動して最終的な妥当性検査を実行します。ウォッチドッグタイマーの間隔は 12.8 秒に設定されているため、`rupgrade_tool -c` の呼び出しから WDT のストローブまでの間、および WDT のストローブからシェルフマネージャーの起動までの間の `/etc/rc` スクリプトの処理時間が、それぞれ 12.8 秒を超えないようにする必要があります。
4. 初期化中に、シェルフマネージャーはアップグレード WDT をもう一度ストローブしてから、接続先のシェルフマネージャーとのネットワーク接続の確立を試みます。ネットワーク接続の確立には、最大 6 秒かかる可能性があります。その後、初期化が正常に完了し、新しい構成の妥当性が示されたあとで、シェルフマネージャーは `rupgrade_tool -f` を呼び出してアップグレード処理を完了します。
5. ユーザーは、必要に応じて `rupgrade_tool -w` を呼び出して、アップグレードセッションの状態を確認します。前述したように、このオプションは、ローカルアップグレードの場合のように、アップグレードの進行状況をシリアルコンソールに出力されるメッセージで直接確認することができない遠隔アップグレードセッションの場合に特に役立つ可能性があります。

信頼できるアップグレードの完了後、新しいイメージが何らかの理由で受け入れられなかった場合、ユーザーは元のイメージに戻すことができます。元に戻すには、ユーザーは `rupgrade_tool -u` を呼び出します。

必要に応じて、遠隔ネットワークホスト上で実行するように設計された簡単なスクリプトを作成すると、前述の手順を容易に自動化できます。代わりに、オペレータが信頼できるアップグレードをローカルのシリアルコンソールから、またはネットワークを介して遠隔から手動で実行することもできます。

## 信頼できるアップグレードの例

### 例 1:

ここでは、/etc および /var/nvdata 非揮発性ディレクトリを暫定フラッシュにコピーして、3つのすべてのコンポーネント (U-Boot、カーネル、および RFS イメージ) の信頼できるアップグレードを実行する例を示します。すべてのイメージはローカルの /tmp から取得されます。つまり、これらのイメージは何らかの不特定な方法でこの場所にすでにコピーされているものとします。U-Boot イメージは /tmp/u-boot.bin から、カーネルイメージは /tmp/sentry.kernel から、RFS イメージは /tmp/sentry.rfs から取得されます。アップグレード処理は、シリアルコンソールから開始されます。アップグレード処理の各手順の付加的背景を説明するために、コンソールログの間にコメントが示されています。

まず、コマンドプロンプトから `rupgrade_tool` が起動されます。パラメータは、3つのすべてのフラッシュイメージが更新され、シェルフマネージャーの非揮発性データおよび構成ファイルは保持されることを示しています。

```
# 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)
```

アップグレードユーティリティは、妥当性検査スクリプトを起動して、現在 /tmp に提供されているイメージを検査します。指定されたファイル指示子のいずれかが /tmp に見つからない場合、ユーティリティは停止し、次のようなメッセージが生成されます。

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

ユーティリティは、暫定フラッシュ内の指定された宛先へのイメージのコピーに進みます。

```
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]
```

この時点で、step4hshm フックスクリプトが起動され、シェルフマネージャーを停止し、非揮発性情報を暫定フラッシュにコピーします。

```
/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.
```

ここで、信頼できるアップグレード処理は ShMM をリセットします。これによって、暫定フラッシュから U-Boot が起動されます。

```
* 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
```

この時点で、rc スクリプトが `rupgrade_tool -c` を起動して、信頼できるアップグレードが進行中かどうかを確認します。ツールが 0 を返し、アップグレードが進行中であることを確認できます。この結果を前提として、rc スクリプトは起動プロセスを続行します。

```
/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
```

信頼できるアップグレードが進行中のため、rc スクリプトでもう一度ウォッチドッグタイマーがストローブされます。

```
/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 ***
```

シェルフマネージャーが起動し、`rupgrade_tool -f` を実行して信頼できるアップグレードが終了します。

```
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)
```

その後、ユーザーは `rupgrade_tool -w` コマンドを実行して、信頼できるアップグレードの状態を確認します。

```
# 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
#
```

## 例 2:

ここでは、/etc および /var/nvdata 非揮発性ディレクトリを暫定フラッシュにコピーして、RFS イメージのみの信頼できるアップグレードを実行する例を示します。RFS イメージは、IP アドレス 192.168.1.253 の FTP サーバーから取得されます。FTP サーバー上の RFS イメージへのパスは、/tftpboot/ru-mips/sentry.mips.rfs です。アップグレード処理は、Telnet セッションから開始されます。

---

**注** – RFS イメージは明示的に更新されるため、U-Boot およびカーネルのイメージは、自動的に永続フラッシュパーティションから暫定フラッシュパーティションにコピーされます。

---

ローカルのシステムはネットワークを介して FTP サーバーにアクセスする必要があります。つまり、システムのネットワークアダプタが動作および構成されており、ShMM から FTP サーバーへの送信経路が存在する必要があります。この例では、ShMM はネットワークアドレス 192.168.1.174 (FTP サーバーと同じネットワーク内) で構成されています。

```
# 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 に指定したパラメータは、RFS のみをアップグレードすることと、コピープロトコルが FTP で、指定された IP アドレスおよびファイルにユーザー admin がパスワードなしでアクセスすることを示しています。

```
# 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
```

ここで、ユーザーは FTP サイトへのパスワードを求められます。パスワードは手動で入力します。

```
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.
```

次の手順では、特別なスクリプト step4vshmm が起動され、U-Boot およびカーネルのイメージが永続フラッシュから暫定フラッシュにコピーされます。その後、アップグレードユーティリティーは、暫定フラッシュ内の指定された宛先への RFS イメージのコピーに進みます。

```
rupgrade_tool: invoking scripts (step4v*) [--r=sentry.mips.rfs --
proto=ftp:192.168.1.253:/tftpboot/ru-mips:admin --hook=etc_copy]
/etc/upgrade/step4vshmm: Erasing /dev/mtdchar7...Done
etc/upgrade/step4vshmm: Copying Kernel from /dev/mtdchar2 to
/dev/mtdchar7...Done
/etc/upgrade/step4vshmm: Erasing /dev/mtdchar8...Done
/etc/upgrade/step4vshmm: Copying U-Boot from /dev/mtdchar3 to
/dev/mtdchar8...Done
rupgrade_tool: copying sentry.mips.rfs from /tmp to /dev/mtdchar9
using 'cp' protocol
```

step4hshm フックスクリプトが起動されます。シェルフマネージャーが停止し、非揮発性データは保持されます。その後、ユーティリティーはアップグレード WDT を開始し、再起動します。

```
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.
```

この時点で、一定の非活動期間のあとで Telnet セッションが閉じます。数秒後に、rupgrade\_tool -w を呼び出してターゲットに再接続し、信頼できるアップグレードの状態を確認することができます。

```
# 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
#
```

### 例 3:

ここでは、信頼できるアップグレードの失敗例を示します。暫定フラッシュからの起動のあと、信頼できるアップグレードが完了する前に、電源が切断されています。電源を入れ直したあとで、永続フラッシュへのロールバックが発生します。この信頼できるアップグレードは、シリアルコンソールから開始されます。3つのすべてのイメージが、すでに /tmp に存在することを前提としています。

```
# 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.
```

ここで、信頼できるアップグレード処理は ShMM をリセットし、暫定フラッシュから U-Boot を起動します。

```
** Resetting Integrated Peripherals

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

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: 2005-04-11 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
```

ここで、電源が切断されます。しばらくしてから、電源がふたたび投入されます。電力の損失により暫定フラッシュの割り当てが失われたため、システムは永続フラッシュに戻ります。

```
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 スクリプトの次の手順は、`rupgrade_tool -c` を呼び出して、信頼できるアップグレードが進行中かどうかを確認することです。この確認によって、試行された信頼できるアップグレードが失敗したことがわかります。メッセージ「restoring ADM1060 EEPROM to RAM」は、ShMM システム監視デバイス (ADM1060) を示しています。このデバイスは ShMM の起動プロセスを監視し、信頼できるアップグレードのハードウェア面に関するサポートをいくつか実現するものです。このメッセージは、起動プロセスに影響を与える主要な変数が、信頼できるアップグレードが試行される前の状態に戻されていることを示しています。

```
/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
```

## CPLD のプログラミング

ShMM には、CPLD デバイスが組み込まれています。この CPLD デバイスは、ハードウェアレベルの冗長性インタフェースなど、ShMM 操作に関するいくつかの主な側面を制御する役割をします。

### ▼ ShMM の CPLD イメージを再プログラミングする

1. シェルフ管理カードの `/var/bin` ディレクトリに、次のファイルをアップロードします。
  - `cpld_tool`

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

2. 次のように CPLD を消去します。

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

3. 指定されたイメージで CPLD をプログラミングします。

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

4. CPLD イメージの妥当性を検査します。

ユーザー ID は 0x33623030 と等しいはずです。

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

---

## ノードボードのコンソールへの接続

シェルフマネージャーは、ノードボードに接続し、アクティブのシェルフ管理カード (ShMM) からコンソールセッションを開く機能を提供します。まず、シリアルポートまたは Ethernet ポートのいずれかを使用して、アクティブの ShMM にログインします。

---

**注** – コンソール機能を使用するには、一次 (上部) シェルフ管理カードが「アクティブ」のシェルフ管理カードである必要があります。また、スイッチカードは Netra CT 900 サーバーのミッドプレーンのスロット 7 に取り付けられている必要があります。

---

いったんノードボードでコンソールセッションを確立すると、passwd などのシステム管理コマンドの実行、状態メッセージおよびエラーメッセージの読み取り、またはその特定スロットにあるボードの停止を実行できます。

---

**注** – コンソールまたはシリアルケーブルがノードボードのシリアルポートに接続されている場合は、ケーブルが接続されたときに ShMM のコンソールセッションがアクティブな状態であったとしても、コンソールの出力は ShMM 上のコンソールセッションではなく、ケーブル接続されたコンソールに表示されます。

---

# シェルフマネージャとノードボード間のコンソールセッションの確立

コンソールを使用するようにシステムを構成すると、アクティブな ShMM にログインし、スロットのコンソールを開くことができます。Netra CT システムでは、ノードボードのスロットごとにコンソールセッションを 1 つ開くことができます。

表 3-4 に、シェルフマネージャの CLI コンソール関連のコマンドを示します。これらのコマンドは、シェルフ管理カード上の現在のログインセッションから実行できます。

表 3-4      シェルフマネージャの CLI コンソール関連のコマンド

コマンド	説明
<code>clia console slot_no</code>	コンソールモードに入り、 <i>slot_no</i> のノードボードに接続します。 <i>slot_no</i> は、ノードボードが取り付けられているスロットの番号です。
<code>~q</code> または <code>~.</code>	現在のコンソールセッションから切り離します。

## ▼ シェルフマネージャからコンソールセッションを開始する

1. 一次 (上部) ShMM にログインします。

一次 (上部) ShMM には、シリアルポート接続または Ethernet ポート接続のいずれかに接続された端末を介してログインできます。

2. 一次 ShMM がアクティブ ShMM であるかどうかを確認します。

ログオンしてから、次の作業に進む前に、`clia shmstatus` コマンドを使用してアクティブ ShMM にログオンしていることを確認します。「スタンバイ」ShMM にログオンしている場合は、`clia switchover` コマンドを使用してその ShMM を Active に変更します。詳細は、117 ページの「シェルフマネージャの CLI コマンド」の「shmstatus」および「switchover」を参照してください。

### 3. ノードボードへのコンソールセッションを開きます。

```
# clia console slot_no
```

*slot\_no* は、1 ～ 6 および 9 ～ 14 です。たとえば、スロット 4 のボードに対してコンソールを開くには、次のコマンドを入力します。

```
# clia console 4
```

ここで、スロット 4 のノードボードにアクセスできるようになります。その特定のスロットのボードの状態や前回のユーザーがシェルからログアウトしたかどうかに応じて、次のいずれかのプロンプトが表示されます。

- console login% (Solaris レベル)
- # (Solaris レベル。前回のユーザーがスーパーユーザーとしてログインし、コンソールから切り離される前にログアウトしていない)
- ok (OpenBoot PROM レベル)
- # (Monta Vista Linux)

## ▼ コンソールセッションを終了する

1. (省略可能) OS シェルをログアウトします。
2. プロンプトで、エスケープシーケンス `~q` または `~.` (チルダとピリオド) を入力してコンソールから切り離します。

```
prompt ~q
```

コンソールから切り離しても、遠隔ホストから自動的にログアウトされることはありません。遠隔ホストからログアウトしないと、ボードに接続する次のコンソールユーザーには、前回のセッションのシェルプロンプトが表示されます。

---

**注** – 処理が完了したら、コンソールセッションから必ずログアウトしてください。

---

## ノードボードの手動による正常な停止

このリリースでは、ノードボードまたは FRU を誤って取り外すことを防ぐために、手動による正常な停止機能が提供されています。正常な停止は、そのノードボードのペイロード上で動作しているすべてのアプリケーションと OS、およびペイロード自体を強制的でない方法で休止 (停止) することを意味します。

**Netra** ノードボードのホットスワップまたは取り外しを行う前に、そのノードボード上で動作しているアプリケーションおよびオペレーティングシステムを正常に停止しておくことをお勧めします。シェルフマネージャーのネットコンソール機能を使用すると、システム管理者はノードボードのコンソールセッションを開始し、その設置場所の停止手順に従って正常な停止を実行できます。

この停止手順が完了したら、システム管理者は **FRU** またはノードボードの停止機能を有効にして、ボードのホットスワップラッチを開くことができます。ノードボードの交換または再取り付けが完了したら、管理者は **FRU** またはノードボードの停止機能を無効にして、ホットスワップラッチが開いたときに異常な方法で停止することを防止できます。

このプロセスに含まれる手順については、次の処理の中で詳細に説明します。

### ▼ ノードボードを停止する

この手順では、ノードボードのホットスワップラッチが閉じている必要があります。

---

**注** – ホットスワップラッチが開いていると、青色のホットスワップ LED が点滅を続け、ボードがホットスワップ可能な状態 (青色の LED の常時点灯で示される) になりません。この状態を修正するには、この手順を開始する前にホットスワップラッチを閉じてください。

---

1. アクティブ ShMM にログインします。
2. ノードボードへのコンソールセッションを開始します。  
112 ページの「シェルフマネージャーからコンソールセッションを開始する」を参照してください。
3. ノードボード上のアプリケーションおよびオペレーティングシステムを停止します。  
設置場所に適した正常な停止手順を実行します。停止が完了したら、コンソールセッションを閉じます (113 ページの「コンソールセッションを終了する」を参照)。

4. シェルフマネージャーの次の CLI コマンドを使用して、ノードボード上でシェルフマネージャーの制御による停止機能を有効にします。

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

次に例を示します。

```
# 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. シェルフマネージャーの次の CLI コマンドを使用して、シェルフマネージャーの制御による停止機能が有効であるかどうかを確認します。

```
# clia shelf pm
```

「Shelf Manager Controlled Deactivation: Enabled」というメッセージを探します。

次に例を示します。

```
# 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. ノードボードのホットスワップラッチを開きます。
7. ノードボードの青色のホットスワップ LED が点灯している場合は、ノードボードを取り外すか、または交換します。



---

**注意** – モジュールを取り扱う際は、必ずアースされた静電気防止用リストストラップを着用してください。

---

8. ノードボードの交換または再取り付けを行なった場合は、シェルフマネージャーの次の CLI コマンドを使用して、ノードボードに対してシェルフマネージャーの制御による停止機能を無効にします。

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

次に例を示します。

```
# 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  
#
```

# シェルフマネージャーの CLI コマンド

---

この章では、各 CLI コマンドと、使用可能な各コマンドの構文および使用法について説明します。CLI は、AdvancedTCA シェルフのコンテキストをサポートしています。

便宜上、主な種類のシェルフコンポーネントは、IPMB アドレスおよび数値で表される FRU 識別子に基づく表記法での参照の代わりに、次の方法でも参照できます。

- board *n* | b *n*
- fan\_tray *n* | ft *n*
- shm 1 | 2

---

**注** – shm 1 および shm 2 の表記法を使用すると、シェルフ FRU のアドレステーブルに記述されている冗長構成のシェルフマネージャーにアクセスできます。このマニュアルでは、shm 1 がハードウェアアドレスの数値が小さい方のシェルフマネージャーを表し、shm 2 がハードウェアアドレスの数値が大きい方のシェルフマネージャーを表します。

---

冗長構成では、これ以降に示すすべてのコマンドがバックアップシェルフマネージャーでサポートされるわけではありません。バックアップシェルフマネージャーは、次のコマンドのみ認識します。

- debuglevel
- localaddress
- shmstatus
- switchover

ほとんどの情報コマンドでは、簡略モードおよび詳細モードの実行形式をサポートしています。これらのモードでは、提供される情報量が異なります。簡略モードがデフォルト (標準) です。詳細モードを選択するには、コマンド行でオプション `-v` を使用します。このオプションは、コマンドの直後でかつ位置を示す引数の前に指定します。

以降のコマンド構文では、オプション要素は角括弧 ([, ]) で囲んで示し、IPMB アドレス、FRU デバイス ID などのコマンド行の変数要素は斜体で示します。垂直バー (|) は、パラメータの選択肢を区切るために使用しています。

## activate

### 構文:

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

### 用途:

このコマンドは、指定された FRU に対して IPMI コマンド Set FRU Activation (Activate FRU) を送信します。FRU は、それを所有する IPM コントローラの IPMB アドレスと FRU デバイス ID で指定します。PICMG 3.0 コンテキストでは、FRU デバイス ID 0 は IPM コントローラ自体を指定します。

PICMG 3.0 コンテキストでは、このコマンドは主に、シェルフ FRU 情報の電源管理テーブルにリストされなかった FRU、または Shelf Manager Controlled Activation 属性が Disabled に設定されている FRU に対して役立ちます。これらの FRU は、シェルフマネージャーによって自動的に起動されることはないため、状態 M2 (起動要求) にとどまります。これら以外の FRU が状態 M2 になった場合には、シェルフマネージャーがそれらを自動的に起動します。状態 M2 ではない FRU の起動を試みても、状態は変わりません。

### 例:

アドレス 9C の IPM コントローラ自体を起動します。

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

---

# alarm

## 構文:

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

## 用途:

このコマンドを使用すると、Telco アラームの出力にアクセスできます。パラメータ minor、major、および critical を指定すると、ユーザーはそれに対応するアラームの出力を設定できます。これらの処理は累積されるため、コマンド `clia alarm minor` および `clia alarm major` を実行したあとは、マイナーおよびメジャーの両方のアラームが設定されます。clear 処理は、マイナーアラームおよびメジャーアラームの出力を解除しますが、クリティカルアラームの出力は解除できません。

パラメータを設定せずにコマンドを呼び出すと、Telco アラーム出力の状態が返されます。

## 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

## 構文:

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

## 用途:

このコマンドおよび boardreset コマンドは、ATCA ボードで動作し、引数として IPM コントローラのアドレスおよび FRU デバイス ID ではなく物理スロット番号を受け入れます。この点がほかの一連のコマンドとは異なります。コマンド board は、ATCA スロットに割り当てられた IPMB アドレスの範囲内の各 IPM コントローラに関する情報、およびこれらのコントローラが制御する追加の各 FRU に関する情報を表示します。表示される項目の一覧は、134 ページの「fru」および 181 ページの「ipmc」セクションで示します。ボード上に IPM コントローラが搭載されている PICMG 3.0 システムの場合、IPMB アドレスの範囲は 82h ~ A0h です。

物理アドレスは、10 進数で指定します。PICMG 3.0 システムでは、物理アドレスおよび IPMB アドレスの対応関係は、シェルフ FRU 情報で指定されます。シェルフ FRU 情報にアドレステーブルが含まれない場合は、次のマッピングテーブル (論理スロット番号のマッピング) が使用されます。

スロット番号	IPMB アドレス
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

例:

システム内のすべてのボードに関する標準的な情報を取得します。ここでは、物理スロット 1 および 14 のみにボードが取り付けられているものとします。

```
# 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"
#
```

物理スロット 14 のボードに関する詳細情報を取得します。

```
# 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

構文:

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

用途:

このコマンドは、指定された物理スロットのボードをリセットし、そのボードに IPMI コマンド FRU Control (Cold Reset) を送信します。

物理アドレスは、10 進数で指定します。PICMG 3.0 システムでは、物理アドレスおよび IPMB アドレスの対応関係は、シェルフ FRU 情報で指定されます。シェルフ FRU 情報にアドレステーブルが含まれない場合は、次のマッピングテーブル (論理スロット番号のマッピング) が使用されます。FRU デバイス ID は 0 です。

スロット番号	IPMB アドレス
1	9A
2	96
3	92
4	8E
5	8A
6	86
7	82
8	84
9	88

スロット番号	IPMB アドレス
10	8C
11	90
12	94
13	98
14	9C

例:

物理スロット 14 のボードをリセットします (IPMB アドレス 9C、FRU 0)。

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

## busres

構文:

`busres subcommand`

次のサブコマンドがサポートされています。

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

用途:

このコマンドは、バス型の E-キーイングで管理されるリソースの現在の状態に関する情報を表示し、その状態の変更を可能にします。

すべてのサブコマンドは、パラメータの 1 つとしてリソース ID を受け付けます。リソース ID は、0 から始まるリソース番号またはリソース名の略称のいずれかです。次のリソース名およびリソース番号が定義されています。

番号	略称	説明
0	mtb1	Metalic Test Bus ペア 1
1	mtb2	Metalic Test Bus ペア 2
2	clk1	Synch Clock グループ 1
3	clk2	Synch Clock グループ 2
4	clk3	Synch Clock グループ 3

以降のサブセクションでは、複数のキーを使用する `busres` コマンドの構文について説明します。

## バス型の E-キーイング管理対象のリソースの状態表示

構文:

```
busres info [resource]
```

用途:

このコマンドは、指定されたリソース、またはリソース ID が指定されていない場合はすべてのリソースの現在の状態に関する情報を表示します。

パラメータ *resource* には、リソース ID を指定します。サポートされるリソース ID の一覧は、123 ページの「`busres`」に示されています。

例:

Metalic Test Bus ペア 2 の状態に関する情報を取得します。

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

## 指定されたリソースの解放

構文:

```
busres release | force resource
```

用途:

このコマンドは、リソースの現在の所有者に Bused Resource Control 要求を送信して、リソースの解放を指示します。コマンド構文が `busres release resource` の場合は、Bused Resource Control (Release) コマンドが送信されます。コマンド構文が `busres force resource` の場合は、Bused Resource Control (Force) コマンドが送信されます。これらの ATCA コマンドに関する説明の詳細は、PICMG 3.0 R1.0 仕様のセクション 3.7.3.4 を参照してください。

パラメータ `resource` には、リソース ID を指定します。サポートされるリソース ID の一覧は、123 ページの「busres」に示されています。

例:

現在の所有者に Metalic Test Bus ペア 2 の解放を強制します。

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

## 指定されたリソースのロック/ロック解除

構文:

```
busres lock | unlock resource
```

用途:

このコマンドは、指定されたリソースのロック (`busres lock resource`) またはロック解除 (`busres unlock resource`) を実行します。リソースがロックされている場合に、別の IPM コントローラがシェルフマネージャーに Bused Resource Control (Request) コマンドを送信すると、シェルフマネージャーは拒否状態の応答を返します。リソースがロック解除されている場合に、別の IPM コントローラがシェルフマネージャーに Bused Resource Control (Request) コマンドを送信すると、シェルフマネージャーはビジー状態の応答を返し、現在の所有者に Bused Resource Control (Release) を送信します。現在の所有者がリソースを解放すると、このリソースは次の要求で要求者の所有となります。

ロックできるのは、IPM コントローラが所有するリソースのみです。現在の所有者がリソースを解放するとすぐに、このリソースのロックも解除されます。

パラメータ *resource* には、リソース ID を指定します。サポートされるリソース ID の一覧は、123 ページの「busres」に示されています。

例:

Synch Clock グループ 3 をロックします。

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

## Bused Resource Control (Query) コマンドの送信

構文:

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

用途:

このコマンドは、指定された IPM コントローラに Bused Resource Control (Query) 要求を送信します。コマンド行で IPM コントローラが指定されない場合、要求はリソースの現在の所有者に送信されます。応答を受信した時点で、リソーステーブルに適切な変更が行われます。たとえば、現在の所有者と考えられる IPM コントローラが制御なし状態の応答を返すと、その事実を反映してテーブルが変更されます。ただし、これは *noupdate* フラグを指定していない場合に限られます。コマンド行で *noupdate* フラグを渡すと、受信した情報に基づくリソーステーブルの変更は行われません。

パラメータ *resource* には、リソース ID を指定します。サポートされるリソース ID の一覧は、123 ページの「busres」に示されています。

パラメータ *target* には、要求の送信先である IPM コントローラの IPMB アドレスを指定します。

フラグ *noupdate* を指定した場合は、クエリー要求への応答として受信した情報でリソーステーブルが更新されないことを示します。

現在のバージョンのシェルフマネージャーでは、*-v* フラグを指定しても追加情報は表示されません。

例:

Metalic Test Bus ペア 1 に対する照会をアドレス 0x82 の IPM コントローラに送信します。この応答に基づくリソーステーブルの更新は行いません。

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

## リソースの所有者の設定

構文:

```
busres setowner resource target
```

用途:



---

**注意** – このコマンドは、経験のあるユーザーが自己の判断で使用してください。

---

このコマンドは、リソーステーブルで指定されたリソースの所有者を直接設定します。コマンドの実行前に別の所有者がそのリソースを所有していても、**Bused Resource Control** コマンドは一切送信されません。これは低レベルのコマンドで、テストおよび復旧の目的にのみに使用されるはずです。

パラメータ *resource* には、リソース ID を指定します。サポートされるリソース ID の一覧は、123 ページの「busres」に示されています。

パラメータ *target* には、リソースの所有者として設定する IPM コントローラの IPMB アドレスを指定します。リソースがいずれの IPM コントローラにも所有されないように指定するには、IPMB アドレスに 0 を使用してください。

例:

Metalic Test Bus ペア 1 の新しい所有者として、ボード 1 を設定します。

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

## Bused Resource Control (Bus Free) コマンドの送信

構文:

```
busres sendbusfree resource target
```

用途:



---

**注意** – このコマンドは、経験豊のあるユーザーが自己の判断で使用してください。

---

このコマンドは、指定された IPM コントローラに Bused Resource Control (Bus Free) 要求を送信します。異なる IPM コントローラがリソースを所有している場合でも、要求を送信する前にそのリソース上で処理は行われません。ただし、リソーステーブルはこの要求への応答に基づいて更新されます。つまり、IPM コントローラがリソースの所有を受け入れると、その IPM コントローラはリソーステーブルで新しい所有者として設定されます。これは低レベルのコマンドで、テストおよび復旧の目的にのみ使用されるはずです。

パラメータ *resource* には、リソース ID を指定します。サポートされるリソース ID の一覧は、123 ページの「busres」に示されています。

パラメータ *target* には、要求の送信先である IPM コントローラの IPMB アドレスを指定します。リソースがどの IPM コントローラにも所有されないように指定するには、IPMB アドレスに 0 を使用します。

例:

Metalic Test Bus ペア 1 に対するバス解放要求を、アドレス 0x82 の IPM コントローラに送信します。

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

# console

## 構文:

`console slot-number`

## 用途:

このコマンドは、指定された物理スロットのノードボード上でコンソール端末セッションを確立します。シェルフマネージャーでは、ノードボードごとに 1 つのコンソールセッションを開くことができます。有効なスロット番号は、1 ～ 6 および 9 ～ 14 です。

---

**注** – コンソール機能を使用するには、一次 (上部) シェルフ管理が「アクティブ」のシェルフ管理カードである必要があります。また、スイッチカードは Netra CT 900 サーバーのミッドプレーンのスロット 7 に取り付けられている必要があります。

---

ノードボードでコンソールセッションを確立すると、`passwd` などのシステム管理コマンドの実行、状態メッセージおよびエラーメッセージの読み取り、またはその特定のスロットにあるボードの停止を実行できます。

---

**注** – コンソールまたはシリアルケーブルがノードボードのシリアルポートに接続されている場合は、ケーブルが接続されたときに ShMM のコンソールセッションがアクティブな状態であったとしても、コンソールの出力は ShMM 上のコンソールセッションではなく、ケーブル接続されたコンソールに表示されます。

---

現在のコンソールセッションから切り離すには、`~q` または `~.` (チルダとピリオド) を入力します。

## 例:

物理スロット 4 のノードボード上でコンソールセッションを開始します。

```
# clia console 4  
prompt
```

---

# deactivate

## 構文:

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

## 用途:

このコマンドは、指定された FRU に対して IPMI コマンド Set FRU Activation (Deactivate FRU) を送信します。FRU は、それを所有する IPM コントローラの IPMB アドレスと FRU デバイス ID で指定します。PICMG 3.0 のコンテキストでは、FRU デバイス ID 0 は IPM コントローラ自体を指定します。

## 例:

アドレス 9C の IPM コントローラ自体を停止します。

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

---

# debuglevel

## 構文:

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

## 用途:

このコマンドは、IPM Sentry Shelf Manager の現在のデバッグレベルを表示するか、または新しい値が指定された場合にデバッグレベルをその値に設定します。

デバッグレベルは、ビットマスクとして処理される 0x0000 ~ 0x00FF の範囲の 16 進数で表されます。マスクの各ビットが設定されると、特定の種類のデバッグ出力が有効になります。

- 0x0001 — エラーメッセージ

- 0x0002 — 警告メッセージ
- 0x0004 — 情報メッセージ
- 0x0008 — 詳細情報メッセージ
- 0x0010 — トレースメッセージ
- 0x0020 — 詳細トレースメッセージ
- 0x0040 — IPM コントローラの初期化中に IPM コントローラに送信される重要なコマンドに関して表示されるメッセージ
- 0x0080 — 内部ロックの獲得および解放に関する詳細メッセージ

シェルフマネージャーのデフォルトのデバッグレベルは 0x0007 です。ただし、コマンド行で `-v` オプションを指定すると、シェルフマネージャーの起動中にこの値を上書きできます。CLI では、実行時にデバッグレベルを変更する追加機能が提供されています。

例:

現在のデバッグレベルを取得し、次にデバッグレベルを 0x001F に設定します。

```
# 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

構文:

```
exit
quit
```

用途:

コマンド `exit` または `quit` は、CLI の対話モードを終了します。対話モードに入るには、パラメータを指定しないで `clia` を実行します。

例:

```
# exit
#
```

---

# fans

## 構文:

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

## 用途:

このコマンドは、指定されたファン FRU に関する情報を表示します。FRU デバイス ID を省略すると、指定したアドレスの IPM コントローラで制御されるすべてのファン FRU に関する情報が表示されます。IPMB アドレスも省略すると、シェルフマネージャーに認識されるすべてのファンの FRU に関する情報が表示されます。表示される情報は、次のとおりです。

- IPMB アドレスおよび FRU デバイス ID
- 最低速度レベル
- 最高速度レベル
- 最高持続速度レベル
- 現在のレベル (両方使用可能な場合は、オーバーライドレベルおよびローカル制御レベル)

## 例:

IPMB アドレス 20 のすべてのファン FRU に関するファン情報を取得します。

```
# 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

## 構文:

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

## 用途:

このコマンドを使用すると、Netra CP3060 ボードのシステムファームウェアを指定した場所のファームウェアイメージで更新できます。このコマンドは、Netra CT 900 サーバーに取り付けられた Netra CP3060 ボードのみに有効です。更新する Netra CP3060 ボードのシステムファームウェアには、ALOM-CMT、Hypervisor、OBP、Post、および VBSC のファームウェアが含まれています。

---

**注** – Netra CP3060 ファームウェアのダウンロードサイトへのリンクは、<http://www.sun.com/downloads/> にあります。

---

このコマンドを使用するには、次の情報が必要です。

- ファームウェアイメージのダウンロード元の FTP サーバーの IP アドレス
- プロンプトで入力する FTP サーバーのユーザー名およびパスワード
- イメージの格納先のパス

*slot-number* には Netra CP3060 ボードのスロット番号を指定し、**-s server-ip-address** 引数にはファームウェアイメージのダウンロード元のサーバーの IP アドレスを指定します。**-f fw-image-pathname** には、ファームウェアイメージを格納するフルパス名を指定します。

## 例:

Netra CP3060 ボードのシステムファームウェアをダウンロードし、更新します。このプロセスが完了するまで数分かかる可能性があることに注意してください。プロセスが正常に完了したら、boardreset コマンドを実行してボードをリセットしてください。

```
# 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

### 構文:

```
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:

このコマンドは、特定の FRU に関する情報を表示します。FRU デバイス ID を省略すると、指定したアドレスの IPM コントローラで制御されるすべての FRU に関する情報が表示されます。IPMB アドレスも省略すると、シェルフマネージャーに認識されるすべての FRU に関する情報が表示されます。

また、設置場所のタイプで FRU を選択することもできます。設置場所のタイプは、16 進数でコマンドパラメータに指定します。FRU と設置場所のタイプの関連付けは、シェルフ FRU 情報に格納されます。設置場所のタイプは、PICMG 3.0 仕様で次のように定義されています。

- 00h — AdvancedTCA ボード
- 01h — 電源入力モジュール
- 02h — シェルフ FRU 情報
- 03h — 専用 ShMC
- 04h — ファントレー
- 05h — ファンフィルタトレイ
- 06h — アラーム
- 07h — AdvancedTCA モジュール (メザニン)
- 08h — PMC
- 09h — 背面切り替えモジュール
- C0h ~ CFh — OEM 定義
- その他の値はすべて予約されています。

標準モードでは、FRU に関する次の情報が表示されます。

- IPMB アドレスおよび FRU デバイス ID
- エンティティ ID、エンティティインスタンス
- 設置場所のタイプおよび設置場所の番号 (認識される場合)

- 現在のホットスワップの状態、前回のホットスワップの状態、および前回の FRU の状態変更の原因。ホットスワップの状態 M0 ～ M7 は、PICMG 3.0 仕様で次のように定義されています。
  - M0 – 取り付けられていない
  - M1 – 休止状態
  - M2 – 起動要求
  - M3 – 起動中
  - M4 – FRU は動作中
  - M5 – 停止要求
  - M6 – 停止中
  - M7 – 通信不可

詳細モードの場合のみ、FRU に関する次の情報が表示されます。

- FRU デバイスタイプ、デバイスタイプの修飾子 (FRU-device-ID != 0 の場合のみ)。これは、FRU センサーデータレコード (SDR) から取得される情報で、IPMI 仕様のセクション 37.12 に準拠しています。
- FRU SDR のデバイス ID 文字列
- 現在の FRU 電源レベルおよび最大 FRU 電源レベル、現在の割り当て電源消費量 (W 単位)

例:

アドレス 9C のすべての FRU に関する標準の情報を取得します。

```
# 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"
#
```

アドレス 9C のすべての FRU に関する詳細情報を取得します。

```
# 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  
#
```

アドレス 20 の FRU 1 に関する詳細情報を取得します。

```
# 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

### 構文:

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

```
frucontrol board n option
```

```
frucontrol shm n option
```

```
frucontrol fan_tray n option
```

### 用途:

このコマンドは、FRU 制御コマンドを指定された FRU に送信し、指定された処理を FRU ペイロード上で実行します。FRU は、それを所有する IPM コントローラの IPMB アドレスと FRU デバイス ID で指定します。PICMG 3.0 のコンテキストでは、FRU デバイス ID 0 は IPM コントローラ自体を指定します。

パラメータ *option* には、使用する FRU 制御コマンドのオプションを指定します。次の記号値のいずれかを指定できます。

- cold\_reset (省略形は cr) — FRU ペイロードのコールドリセット
- warm\_reset (省略形は wr) — FRU ペイロードのウォームリセット
- graceful\_reboot (省略形は gr) — FRU ペイロードの正常な再起動
- diagnostic\_interrupt (省略形は di) — 診断の中断

例:

IPMB アドレス 9C の FRU 0 に対してコールドリセットコマンドを実行します。

```
# 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

構文:

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

*addr fru\_id* の代わりに、次の方法でも参照できます。

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

用途:

このコマンドは、**raw** 形式の FRU 情報にアクセスできます。コマンドの形式によって、FRU 情報の読み取りまたは書き込みに使用できます。

読み取り形式では、コマンドはオプションの 32 バイトのブロック番号を受け入れます。

書き込み形式では、コマンドはバイトオフセットパラメータを必要とします。ユーザーは最大 65535 バイトの FRU 情報を変更できます。

**frudataw** および **frudatar** は **frudata** コマンドのバリエーションです。**frudataw** を使用すると、ユーザーは ShMM フラッシュファイルシステム上のファイルを、シェルフ内の特定の FRU の FRU 情報ストレージに書き込むことができます (139 ページの「**frudatar**」を参照)。**frudatar** を使用すると、ユーザーは特定の FRU の FRU 情報ストレージの中身を、ShMM フラッシュファイルシステム上のファイルに転送できます (139 ページの「**frudatar**」を参照)。

例:

すべての FRU の標準の FRU データを表示します。

```
# 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
. . .
#
```

この例では、FRU データを表示する方法および FRU にデータを書き込む方法を示します。

```
# 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

### 構文:

```
frudatar addr fru_id file_name
frudatar addr fru_id byte_offset byte1 [byte2 ... [byte 16] ...]
```

*addr fru\_id* の代わりに、次の方法でも参照できます。

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

### 用途:

このコマンドは、指定された FRU から FRU 情報を読み取り、その情報を ShMM フラッシュファイルシステム上のファイルに **raw** 形式で格納します。つまり、FRU 情報を指定された FRU からフラッシュファイルにアップロードします。パラメータ *file name* には、アップロード先のファイルへのパスを指定します。FRU から読み取られてアップロード先のファイルに書き込まれるバイト数は、指定された FRU に対する IPMI コマンド Get FRU Inventory Area Info への応答で返されるバイト数と同じです。

### 例:

特定の FRU の FRU データを読み取り、指定されたファイルにそのデータを格納します。

```
# 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

### 構文:

```

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

```

*addr fru\_id* の代わりに、次の方法でも参照できます。

```

board n
shm n
fan_tray n

```

### 用途:

このコマンドは、ShMM フラッシュファイルシステム上のファイルから指定された FRU に FRU 情報をダウンロードします。ファイルには、FRU 情報の raw バイナリイメージが含まれます。パラメータ *file name* には、ソースファイルへのパスを指定します。

### 例:

FRU データを、ファイルから指定された FRU に書き込みます。

```

# 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

### 構文:

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

*addr fru\_id* の代わりに、次の方法でも参照できます。

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

### 用途:

このコマンドは、ユーザーにわかりやすい形式で FRU 情報を表示します。

### 例:

特定の FRU に関する FRU 情報を表示します。

```
# 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

### 構文:

```

getfanlevel IPMB-address fru-id
getfanlevel fan_tray n

```

### 用途:

このコマンドは、コマンドパラメータで指定された FRU が制御するファンの現在のレベルを表示します。

例:

IPMB アドレス 0x20 の FRU #2 にあるファンのファンレベルを取得します。

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

---

## getfruledstate

構文:

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

用途:

このコマンドは、現在の FRU の LED の状態を、その LED に対して有効なすべての制御レベルで表示します。詳細モードでは、LED がサポートする色に関する情報も表示されます。

指定された FRU の特定の LED またはすべての LED に関する情報を表示できます。対象の LED の IPMB アドレスおよび FRU ID を省略することも可能です。FRU ID を省略すると、指定された IPM コントローラのすべての FRU にある全 LED に関する情報が表示されます。IPMB アドレスも省略すると、シェルフ内で認識されるすべての LED に関する情報が表示されます。

例:

IPMB アドレス FCh の IPM コントローラ上にあるすべての LED の LED 状態を表示します。

```
# 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
```

IPMB アドレス FCh の IPM コントローラ上にある、すべての LED の LED 状態に関する詳細情報を表示します。

```
# clia getfruiledstate -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
```

IPMB アドレス 20h にある IPM コントローラの FRU #0 に関する LED 状態を表示します。

```
# 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
```

IPMB アドレス 20h にある IPM コントローラの FRU #0 の LED #1 に関する LED 状態を表示します。

```
# 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

### 構文:

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

### 用途:

このコマンドは、指定されたセンサーの現在のヒステリシス値を表示します。センサーは、しきい値ベースのセンサーである必要があります。raw 値および処理された値の両方が表示されます。

対象のコントローラが複数の LUN 上のセンサーをサポートしている場合、ユーザーはセンサー番号を論理ユニット番号 (LUN) で修飾できます。LUN を省略すると、指定されたセンサー番号を持つすべてのセンサーの現在のヒステリシス値が表示されます。*lun* には、値 0、1、または 3 を指定できます。LUN 2 は予約されています。センサー名を指定する際は、LUN 番号で修飾しません。通常、センサー名はコントローラ内で一意であることが前提とされるためです。ただし、コントローラ内に同じ名前のセンサーが複数存在する場合は、それらすべてのセンサーに関する情報が表示されます。*IPMB-address* を省略すると、指定された IPMB アドレスのすべてのセンサーの、現在のヒステリシスレベルが表示されます。

例:

IPMB アドレス FCh の IPM コントローラ上にある、センサー # 2 のヒステリシス値を表示します。

```
# 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

構文:

`getipmbstate IPMB-address [link]` (放射線状の IPMB-0 環境)  
`getipmbstate IPMB-address` (バス型の IPMB-0 環境)

用途:

このコマンドは、対象の IPM コントローラ上にある IPMB-0 の現在の状態を表示します。この状態は、対象の IPMC の IPMB リンクセンサー (センサータイプ F1) が提供するセンサーデータから取得されます。バス A および バス B の両方に関する情報が出力されます。

このコマンドは、バス型と放射線状の接続環境では動作が異なります。バス型の接続環境、または対象の IPMC が IPMB ハブ以外の場合の放射線状の接続環境では、引数 *link* は使用されません。対象の IPM コントローラの IPMB-A および IPMB-B の状態に関する情報が表示されます。

放射線状の接続環境で、対象の IPM コントローラが IPMB ハブである場合、コマンドは次のように機能します。

- *link* を省略すると、コマンドは放射線状のすべての IPMB リンクの状態に関する情報を出力します。状態は、IPM コントローラ上の複数の IPMB リンクセンサーのセンサーデータから取得されます。
- *link* を指定すると、コマンドは特定の放射線状の IPMB リンク (1 ~ 95) に関する情報を出力します。リンクの状態は、IPM コントローラ上の対応する IPMB リンクセンサーの状態から取得されます。

どちらの場合も、IPMB-A および IPMB-B の両方の状態に関する情報が表示されます。

例:

IPMB アドレス 92h の IPMC 上にある、IPMB-0 の現在の状態を表示します。

```
# 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)
```

放射線状の接続環境のシェルフマネージャーに関して、リンク 8 の現在の状態を表示します。

```
# 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

構文:

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

## 用途:

このコマンドは、指定されたチャネルの、指定された LAN 構成パラメータの値を表示します。構成パラメータの名前または番号の指定がない場合、指定されたチャネルのすべての構成パラメータが表示されます。

表 A-1 に、getlanconfig コマンドでサポートされる LAN 構成パラメータの名前および番号の一覧を示します。

表 A-1 getlanconfig の LAN 構成パラメータ

パラメータ名	番号	説明
auth_support	1	LAN チャネルの認証タイプサポートフラグを含む 8 ビットの値。
auth_enables	2	LAN チャネルのコールバック、ユーザー、オペレータ、管理者、および OEM の各特権レベルの認証タイプ有効化フラグを含む 8 ビットの 5 つの値。
ip	3	LAN チャネルに割り当てられた IP アドレスを含む、ドット区切りの 10 進表記の文字列値 (例: 192.168.0.15)。
ip_source	4	割り当てられた IP アドレスのソースをエンコードする値。
mac	5	LAN チャネルに割り当てられた MAC アドレスを、: 記号で区切られた 16 進数の 6 つのバイトとして含む文字列値 (例: 00:A0:24:C6:18:2F)。
subnet_mask	6	LAN チャネルに割り当てられたサブネットマスクを含む、ドット区切りの 10 進表記の文字列値 (例: 255.255.255.0)。
ipv4_hdr_param	7	RMCP パケットを送信するためのさまざまな IPv4 ヘッダーパラメータを含む、次に示す 8 ビットの 3 つの値。 <ul style="list-style-type: none"><li>• 生存期間</li><li>• IP ヘッダーフラグ (ビット [7:5])</li><li>• 優先順位 (ビット [7:5]) およびサービスタイプ (ビット [4:1])</li></ul>
pri_rmcp_port	8	一次 RMCP のポート番号を含む 16 ビットの値 (通常の RMCP 通信に使用されるポート)。
sec_rmcp_port	9	二次 RMCP のポート番号を含む 16 ビットの値 (セキュリティ保護された RMCP 通信に使用されるポート)。
arp_control	10	LAN チャネル上でアドレス解決プロトコル (ARP) の動作を制御する、次に示す 2 つのフラグ。 <ul style="list-style-type: none"><li>• ARP 要求への応答を有効にする</li><li>• Gratuitous ARP 送信を有効にする</li></ul>

表 A-1 getlanconfig の LAN 構成パラメータ (続き)

arp_interval	11	Gratuitous ARP の間隔。固定小数点形式で秒単位で示します。小数点以下が含まれる場合があります。
dft_gw_ip	12	デフォルトゲートウェイの IP アドレスを含む、ドット区切りの 10 進表記の文字列値。
dft_gw_mac	13	デフォルトゲートウェイの MAC アドレスを含む、コロン (:) で区切られた 16 進数の 6 つのバイトの文字列値。
backup_gw_ip	14	バックアップゲートウェイの IP アドレスを含む、ドット区切りの 10 進表記の文字列値。
backup_gw_mac	15	バックアップゲートウェイの MAC アドレスを含む、コロン (:) で区切られた 16 進数の 6 つのバイトの文字列値。
community	16	PET トラップの Community String フィールドに格納される文字列値 (最大 18 記号)。
destination_count	17	LAN チャンネルでサポートされる LAN アラートの宛先の最大数。
destination_type	18	指定された設定セレクトで識別される宛先タイプ。設定セレクトを指定しない場合、すべての宛先タイプが表示されます。各宛先タイプのエントリには、次のフィールドが含まれます。 <ul style="list-style-type: none"> <li>宛先タイプ (0 ~ 7)</li> <li>アラート確認応答フラグ</li> <li>アラート確認応答タイムアウト/再試行間隔の秒数 (1 ~ 256)</li> <li>再試行の回数 (0 ~ 7)</li> </ul>
destination_address	19	指定された設定セレクトに関連付けられた宛先アドレス。設定セレクトを指定しない場合、すべての宛先アドレスが表示されます。各宛先アドレスのエントリには、次のフィールドが含まれます。 <ul style="list-style-type: none"> <li>ゲートウェイセレクト: 0 はデフォルトを使用、1 はバックアップを使用</li> <li>IP アドレス (ドット区切りの 10 進表記の文字列)</li> <li>MAC アドレス (コロン [:] で区切られた 16 進数の 6 つのバイトの文字列)</li> </ul>

例:

以降のサブセクションでは、サポートされる各パラメータについて、より詳細に説明します。

チャンネル 1 の LAN パラメータの表を取得して表示します。

```
# 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

### 構文:

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

### 用途:

このコマンドは、LAN パラメータ `auth_support` の現在の値を表示します。このパラメータは、シェルフマネージャーでサポートされる認証タイプを示します。これはシングルバイトで表され、次のように定義されるビットでビットマスクとして処理されます。

- 0x01 — なし
- 0x02 — MD2

- 0x04 – MD5
- 0x10 – 単純パスワード/キー
- 0x20 – OEM 所有

その他のビットは予約されているため、0 に設定されている必要があります。

16 進数の **raw** 値とともに、設定されているビットの記号値も表示されます。

例:

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

## auth\_enables

構文:

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

用途:

このコマンドは、LAN パラメータ **auth\_enables** の現在の値を表示します。このパラメータは、サポートされる 5 つの各特権レベル (コールバック、ユーザー、管理者、オペレータ、および OEM) に対して、現在シェルフマネージャーによって有効化されている認証タイプを示します。パラメータは連続する 5 バイトで表され、各バイトがそれぞれの特権レベルに対応し、次のように定義されるビットでビットマスクとして処理されます。

- 0x01 – なし
- 0x02 – MD2
- 0x04 – MD5
- 0x10 – 単純パスワード/キー
- 0x20 – OEM 所有

その他のビットは予約されているため、0 に設定されている必要があります。

16 進数の **raw** 値とともに、設定されているビットの記号値も表示されます。

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

構文:

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

用途:

このコマンドは、チャンネルで使用される 現在の IP アドレスを、ドット区切りの 10 進表記で表示します。

例:

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

## ip\_source

構文:

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

用途:

このコマンドは、LAN パラメータ `ip_source` の現在の値を表示します。このパラメータは、シェルフマネージャーが使用する IP アドレスのソースを、シングルバイトで示します。次の値のいずれかになります。

- 0 – 指定なし

- 1 – 静的アドレス (手動設定)
- 2 – DHCP を実行するシェルフマネージャーによって取得されたアドレス
- 3 – BIOS またはシステムソフトウェアによってロードされたアドレス
- 4 – ほかのアドレス割り当てプロトコルを実行するシェルフマネージャーによって取得されたアドレス

その他の値は予約されています。

16 進数の raw 値とともに、記号値も表示されます。

例:

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

## mac

構文:

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

用途:

このコマンドは、チャンネルで使用される現在の MAC アドレスを、コロンで区切られた 16 進数の 6 つのバイトの形式で表示します。

例:

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

## subnet\_mask

### 構文:

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

### 用途:

このコマンドは、チャネルで使用する現在の IP サブネットマスクを、ドット区切りの 10 進表記で表示します。

### 例:

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

## ipv4\_hdr\_param

### 構文:

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

### 用途:

このコマンドは、現在の IP 4 ヘッダーパラメータを表示します。これらは、16 進表記の 3 つのシングルバイト値として、コロンで区切って表されます。バイトの内容は、IPMI 1.5 仕様のセクション 19.2 に準拠しています。

### 例:

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

## pri\_rmcp\_port

### 構文:

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

### 用途:

このコマンドは、チャンネルで使用する現在の RMCP の一次ポートを 16 進数で表示します。これは、RMCP 経由の通常の対話に使用されるポートです。

### 例:

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

## sec\_rmcp\_port

### 構文:

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

### 用途:

このコマンドは、チャンネルで使用する現在の RMCP の二次ポートを 16 進数で表示します。これは、RMCP 経由のセキュリティー保護された対話に使用されるポートです。

### 例:

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

## arp\_control

### 構文:

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

### 用途:

このコマンドは、LAN パラメータ `arp_control` の現在の値を表示します。このパラメータは、シェルフマネージャーによって提供される追加の **ARP** サポートを示します。また、シングルバイトで表され、次のように定義されるビットでビットマスクとして処理されます。

- 1 – シェルフマネージャーが生成する **Gratuitous ARP** を有効にする
- 2 – シェルフマネージャーが生成する **ARP 応答** を有効にする

その他のビットは予約されているため、0 に設定されている必要があります。

16 進数の **raw** 値とともに、設定されたビットの記号値も表示されます。

### 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

### 構文:

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

### 用途:

このコマンドは、チャンネルで使用する現在の **ARP** 間隔を表示します。値は、固定小数点形式の秒数で表示されます。

例:

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

## dft\_gw\_ip

構文:

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

用途:

このコマンドは、チャンネルで使用されるデフォルトゲートウェイの IP アドレスを、ドット区切りの 10 進表記で表示します。

例:

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

## dft\_gw\_mac

構文:

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

用途:

このコマンドは、チャンネルで使用されるデフォルトゲートウェイの MAC アドレスを、16 進数の 6 つのバイトをコロンで区切った形式で表示します。

例:

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

## backup\_gw\_ip

構文:

```
getlanconfig channel backup_gw_ip  
getlanconfig channel 14
```

用途:

このコマンドは、チャネルで使用するバックアップゲートウェイの IP アドレスを、ドット区切りの 10 進表記で表示します。

例:

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

## backup\_gw\_mac

構文:

```
getlanconfig channel backup_gw_mac  
getlanconfig channel 15
```

用途:

このコマンドは、チャネルで使用するバックアップゲートウェイの MAC アドレスを、16 進数の 6 つのバイトをコロンで区切った形式で表示します。

例:

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

## community

構文:

```
getlanconfig channel community  
getlanconfig channel 16
```

用途:

このコマンドは、PET トラップで使用されるコミュニティ文字列パラメータを表示します。

例:

```
# clia getlanconfig 1 community  
IPM Sentry Shelf Manager Command Line Interpreter  
Community String: "public"  
#
```

## destination\_count

構文:

```
getlanconfig channel destination_count  
getlanconfig channel 17
```

用途:

このコマンドは、チャンネルで使用可能なアラートの宛先の最大数を表示します。これは、IPM Sentry Shelf Manager の構成パラメータで、shelfman 構成ファイルを介してのみ変更できます。

例:

```
# clia getlanconfig 1 destination_count  
IPM Sentry Shelf Manager Command Line Interpreter  
Number of Destinations: 16  
#
```

## destination\_type

構文:

```
getlanconfig channel destination_type [set-selector]  
getlanconfig channel 18 [set-selector]
```

用途:

このコマンドは、*set-selector* と等しいインデックスを持つ宛先テーブルの要素を表示します。インデックスは、0 から始まります。セクタ 0 は、揮発性の宛先を指定するために使用します。宛先に関する次の情報が表示されます。

- 宛先セクタ
- アラートの宛先タイプ (PET トラップまたは OEM の宛先、アラートが確認される必要があるかどうか)
- アラートの確認応答タイムアウト
- 再試行の回数

設定セクタを省略すると、すべてのアクティブな宛先が、その番号とともに表示されます。

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

構文:

```
getlanconfig channel destination_address [set-selector] getlanconfig
channel 19 [set-selector]
```

用途:

このコマンドは、*set-selector* と等しいインデックスを持つ宛先アドレステーブルの要素を表示します。インデックスは、0 から始まります。セクタ 0 は、揮発性の宛先を指定するために使用します。宛先に関する次の情報が表示されます。

- 宛先セクタ
- アドレス形式 (デフォルトは IP+MAC)
- 宛先の IP アドレス
- 宛先の MAC アドレス
- 使用するゲートウェイ (デフォルトまたはバックアップ)

設定セクタを省略すると、すべてのアクティブな宛先アドレスがその番号とともに表示されます。

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

構文:

```
getpefconfig
getpefconfig parameter-name [additional-parameters]
getpefconfig parameter-number [additional-parameters]
```

用途:

このコマンドは、指定された PEF 構成パラメータの値を表示します。構成パラメータの名前とパラメータ番号のどちらも指定しない場合、すべての PEF 構成パラメータが表示されます。

表 A-2 に、PEF 構成パラメータの名前および番号の一覧を示します。

表 A-2 PEF 構成パラメータ

パラメータ名	番号	説明
control	1	PEF の制御フラグを表す 8 ビットの値 (PEF の有効化、PEF 起動遅延の有効化など)。
action_control	2	PEF 処理のグローバル制御フラグを表す 8 ビットの値 (リセットの有効化、電源切断の有効化など)。
startup_delay	3	システムの電源投入およびリセット後の PEF の遅延時間 (秒単位)。
alert_startup_delay	4	システムの電源投入およびリセット後のアラートの遅延時間 (秒単位)。
event_filter_count	5	イベントフィルタの最大数。
event_filter	6	指定された設定セレクトで識別されるイベントフィルタテーブルのエントリ。設定セレクトを指定しない場合、アクティブなイベントフィルタがすべて表示されます。

表 A-2 PEF 構成パラメータ (続き)

event_filter_data1	7	指定された設定セクタで識別されるイベントフィルタテーブルのエントリの先頭バイト。設定セクタを指定しない場合、アクティブなイベントフィルタがすべて表示されます。
alert_policy_count	8	アラートポリシーの最大数。
alert_policy	9	指定された設定セクタで識別されるアラートポリシーテーブルのエントリ。設定セクタを指定しない場合、アクティブなアラートポリシーがすべて表示されます。
system_guid	10	PET トラップの GUID フィールドへの値の入力に使用される GUID。
alert_string_count	11	アラート文字列の最大数。
alert_string_key	12	指定された設定セクタで識別されるアラート文字列キー。設定セクタを指定しない場合、すべてのアラート文字列キーが表示されます。
alert_string	13	指定された設定セクタで識別されるアラート文字列。設定セクタを指定しない場合、すべてのアラート文字列が表示されます。
oem_filter_count	96	OEM フィルタの最大数。
oem_filter	97	指定された設定セクタで識別される OEM フィルタテーブルのエントリ。設定セクタを指定しない場合、アクティブなイベントフィルタがすべて表示されます。

例:

PEF パラメータテーブル全体を、取得して表示します。

```
# 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
#
```

以降のサブセクションでは、サポートされる各パラメータについて、より詳細に説明します。

## control

### 構文:

```
getpefconfig control
getpefconfig 1
```

### 用途:

このコマンドは、PEF パラメータ `control` の現在の値を表示します。このパラメータはシングルバイトで、次のように定義されるビットでビットマスクとして処理されます。

- 0x01 — PEF を有効にする
- 0x02 — PEF 処理に対するイベントメッセージ生成を有効にする
- 0x04 — システムの電源投入およびリセット時の PEF の起動遅延を有効にする
- 0x08 — PEF のアラート起動遅延を有効にする

その他のビットは予約されているため、0 に設定されている必要があります。

### 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

## 構文:

```
getpefconfig action_control  
getpefconfig 2
```

## 用途:

このコマンドは、PEF パラメータ `action_control` の現在の値を表示します。このパラメータはシングルバイトで、次のように定義されるビットでビットマスクとして処理されます。

- 0x01 — アラート処理を有効にする
- 0x02 — 電源切断処理を有効にする
- 0x04 — リセット処理を有効にする
- 0x08 — 電源の再投入処理を有効にする
- 0x10 — OEM 処理を有効にする
- 0x20 — 診断の中断を有効にする

その他のビットは予約されているため、0 に設定されている必要があります。

## 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

### 構文:

```
getpefconfig startup_delay  
getpefconfig 3
```

### 用途:

このコマンドは、PEF パラメータ `startup_delay` の現在の値を表示します。このパラメータはシングルバイトで、PEF 機能が起動時に遅延する秒数を表します。

### 例:

```
# clia getpefconfig startup_delay  
IPM Sentry Shelf Manager Command Line Interpreter  
    PEF Startup Delay: 60 seconds  
#
```

## alert\_startup\_delay

### 構文:

```
getpefconfig startup_delay  
getpefconfig 4
```

### 用途:

このコマンドは、PEF パラメータ `alert_startup_delay` の現在の値を表示します。このパラメータはシングルバイトで、アラート機能が起動時に遅延する秒数を表します。

### 例:

```
# clia getpefconfig alert_startup_delay  
IPM Sentry Shelf Manager Command Line Interpreter  
    PEF Alert Startup Delay: 60 seconds  
#
```

## event\_filter\_count

### 構文:

```
getpefconfig event_filter_count  
getpefconfig 5
```

### 用途:

このコマンドは、PEF パラメータ `event_filter_count` の現在の値を表示します。この読み取り専用の値は、イベントフィルタテーブルのサイズです。この値は、IPM Sentry Shelf Manager の構成パラメータで、`shelfman` 構成ファイルを介してのみ変更できます。

### 例:

```
# clia getpefconfig event_filter_count  
IPM Sentry Shelf Manager Command Line Interpreter  
PEF Number of Event Filters: 64  
#
```

## event\_filter

### 構文:

```
getpefconfig event_filter [set-selector]  
getpefconfig 6 [set-selector]
```

### 用途:

このコマンドは、*set-selector* と等しいインデックスを持つイベントフィルタテーブルの要素を表示します。インデックスは、1 から始まります。各イベントフィルタに関する次の情報が表示されます。

- フィルタ構成: フィルタがソフトウェアで構成されるか、またはメーカーで事前構成されているか
- イベントフィルタ処理マスク
- アラートポリシー番号
- イベントの重要度
- 一致するイベントのソースアドレス (255 = すべてのアドレス)
- 一致するソースチャネル/LUN (255 = すべてのソースチャネル/LUN に一致)
- 一致するセンサータイプ

- 一致するセンサー番号
- 一致するイベントトリガー (イベント/読み取りタイプ)
- イベントオフセットマスク
- イベントデータバイト 1、2、および 3 に使用する AND、コンペア 1 (CMP1)、およびコンペア 2 (CMP2) マスク

設定セレクタを省略すると、すべてのアクティブなイベントフィルタテーブルのエントリがその番号とともに表示されます。

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

構文:

```
getpefconfig event_filter_data1 [set-selector]
getpefconfig 7 [set-selector]
```

用途:

このコマンドは、*set-selector* と等しいインデックスを持つイベントフィルタテーブルの要素の先頭バイトを表示します。インデックスは、1 から始まります。このバイトは、16 進数で表示されます。このバイトのビットには、次の意味があります。

- 0x80 — このフィルタは有効。
- 0x40 — このフィルタはメーカーで事前構成されているため、ソフトウェアで変更するべきではない。

その他のビットは予約されているため、0 に設定されている必要があります。

設定セクタを省略すると、アクティブな各イベントフィルタテーブルのエントリの先頭バイトが、対応するフィルタ番号とともに表示されます。

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

構文:

```
getpefconfig alert_policy_count
getpefconfig 8
```

用途:

このコマンドは、PEF パラメータ `alert_policy_count` の現在の値を表示します。この読み取り専用の値は、アラートポリシーテーブルのサイズです。この値は、IPM Sentry Shelf Manager の構成パラメータで、`shelfman` 構成ファイルを介してのみ変更できます。

例:

```
# clia getpefconfig alert_policy_count
IPM Sentry Shelf Manager Command Line Interpreter

    Alert Policies Count: 64
#
```

## alert\_policy

### 構文:

```
getpefconfig alert_policy [set-selector]  
getpefconfig 9 [set-selector]
```

### 用途:

このコマンドは、*set-selector* と等しいインデックスを持つアラートポリシーテーブルの要素を表示します。インデックスは、1 から始まります。各アラートポリシーに関する次の情報が表示されます。

- ポリシー番号
- ポリシータイプ (前回の宛先に送信されたアラートに関連するポリシータイプ)
- 宛先のチャンネル番号
- 宛先セクタ
- アラート文字列キー

設定セクタを省略すると、すべてのアクティブなアラートポリシーテーブルのエントリが、その番号とともに表示されます。

### 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

### 構文:

```
getpefconfig system_guid  
getpefconfig 10
```

### 用途:

このコマンドは、PEF パラメータ *system\_guid* の現在の値を表示します。このパラメータは、PET トラップ PDU でアラートの宛先に送信された GUID を表します。この GUID は、別個の GUID として、または Get System GUID IPMI コマンドで取得できるシステム GUID と同じ GUID として定義できます。

例:

```
# clia getpefconfig system_guid  
IPM Sentry Shelf Manager Command Line Interpreter  
    PEF GUID: 23662f7f-ba1b-4b65-8808-94ca09c9bbb0  
#
```

## alert\_string\_count

構文:

```
getpefconfig alert_string_count  
getpefconfig 11
```

用途:

このコマンドは、PEF パラメータ `alert_string_count` の現在の値を表示します。この読み取り専用の値は、アラート文字列テーブルのサイズです。これは、同時に使用できるアラート文字列の最大数です。この値は、IPM Sentry Shelf Manager の構成パラメータで、`shelfman` 構成ファイルを介してのみ変更できます。

例:

```
# clia getpefconfig alert_string_count  
  
IPM Sentry Shelf Manager Command Line Interpreter  
  
    Alert Strings Count: 64  
#
```

## alert\_string\_key

### 構文:

```
getpefconfig alert_string_key [set-selector]  
getpefconfig 12 [set-selector]
```

### 用途:

このコマンドは、*set-selector* と等しいインデックスを持つアラート文字列キーテーブルの要素を表示します。インデックスは、1 から始まります。インデックス 0 は、揮発性のアラート文字列を指定するために使用できます。各キーは、アラートを生成するためにイベントフィルタとアラート文字列を関連付けます。各アラート文字列キーに関する次の情報が表示されます。

- アラート文字列キー番号
- 関連付けられたイベントフィルタ番号
- 関連付けられたアラート文字列番号

設定セクタを省略すると、すべてのアクティブなアラート文字列キーテーブルのエントリが、その番号とともに表示されます。

### 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

### 構文:

```
getpefconfig alert_string [set-selector]  
getpefconfig 13 [set-selector]
```

### 用途:

このコマンドは、*set-selector* と等しいインデックスを持つアラート文字列テーブルの要素を表示します。インデックスは、1 から始まります。インデックス 0 は、揮発性のアラート文字列を指定するために使用できます。このコマンドでは、文字列全体が一度に表示されます。

設定セクタを省略すると、すべての定義済みアラート文字列が、その番号とともに表示されます。

例:

```
# clia getpefconfig alert_string 2
IPM Sentry Shelf Manager Command Line Interpreter
Alert Strings:
    0x02: "This is the alert string"
#
```

## oem\_filter\_count

構文:

```
getpefconfig oem_filter_count
getpefconfig 96
```

用途:

このコマンドは、PEF パラメータ `oem_filter_count` の現在の値を表示します。この読み取り専用の値は、OEM フィルタテーブルのサイズです。この値は、IPM Sentry Shelf Manager の構成パラメータで、`shelfman` 構成ファイルを介してのみ変更できます。

OEM フィルタテーブルは、Pigeon Point Systems 社で定義された IPMI 仕様の OEM 拡張機能です。このテーブルによって、プラットフォームイベントのほかに OEM のタイプスタンプ付きおよびタイムスタンプなしの SEL エントリにも、PEF を適用できます。レコードタイプの範囲は、C0h ~ FFh です。

例:

```
# clia getpefconfig oem_filter_count
IPM Sentry Shelf Manager Command Line Interpreter
PEF Number of OEM Filters: 16
#
```

## oem\_filter

### 構文:

```
getpefconfig oem_filter [set-selector]  
getpefconfig 97 [set-selector]
```

### 用途:

OEM フィルタテーブルは、Pigeon Point Systems 社で定義された IPMI 仕様の OEM 拡張機能です。このテーブルによって、プラットフォームイベントのほかに OEM のタイプスタンプ付きおよびタイムスタンプなしの SEL エントリにも、PEF を適用できます。レコードタイプの範囲は、C0h ~ FFh です。

OEM フィルタテーブルの各エントリは、この OEM フィルタが適用されるレコードタイプの範囲を OEM レコードタイプの範囲内で定義し、レコードタイプが一致するレコードが SEL に記録されたときに呼び出されるアラートポリシー番号を定義します。

このコマンドは、*set-selector* と等しいインデックスを持つ OEM フィルタテーブルの要素を表示します。インデックスは、1 から始まります。各 OEM フィルタに関する次の情報が表示されます。

- バイト 1: SEL レコードタイプの範囲の下限
- バイト 2: SEL レコードタイプの範囲の上限
- バイト 3: レコードタイプがバイト 1 およびバイト 2 で指定した範囲に一致する SEL エントリに対して、呼び出されるアラートポリシー番号

設定セレクトを省略すると、すべてのアクティブな OEM フィルタテーブルのエントリが、その番号とともに表示されます。

### 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

## 構文:

```
getsensoreventenable [IPMB-address [sensor-name | [lun:]sensor-number]]
getsensoreventenable board n [sensor-name | [lun:]sensor-number]]
getsensoreventenable shm n [sensor-name | [lun:]sensor-number]]
```

このコマンドは、指定されたセンサーの現在のイベント有効化マスクの値を表示します。

対象のコントローラが複数の LUN 上のセンサーをサポートしている場合、ユーザーはセンサー番号を論理ユニット番号 (LUN) で修飾できます。LUN を省略すると、すべての LUN 上にある指定されたセンサー番号のセンサーに関する情報が表示されます。lun には、値 0、1、または 3 を指定できます。LUN 2 は予約されています。

センサー名を指定する際は、LUN 番号で修飾しません。通常、センサー名はコントローラ内で一意であることが前提とされるためです。ただし、コントローラ内に同じ名前のセンサーが複数存在する場合は、それらすべてのセンサーに関する情報が表示されます。

このコマンドは、指定されたセンサーでサポートされるイベントの現在のセンサーイベントマスク値を表示します。次に示す各センサーの属性も表示されます。

- センサーを所有する IPM コントローラの IPMB アドレス
- センサー番号、センサー名 (SDR のデバイス ID 文字列)、およびそのセンサーにアクセスできる LUN
- センサータイプ

## 例:

IPM コントローラ FE 上の温度センサー Local Temp のイベント有効化値を取得します。

```
# 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
#
```

同じセンサーのイベント有効化情報を取得しますが、センサーの LUN および番号を指定します。

```
# 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

構文:

```
getthreshold [IPMB-address [sensor-name | [lun:]sensor-number]]
getthreshold board n [sensor-name | [lun:]sensor-number]]
getthreshold shm n [sensor-name | [lun:]sensor-number]]
```

getthreshold の代わりに、動詞 threshold を使用することもできます。

## 用途:

このコマンドは、指定されたセンサーでサポートされるしきい値の、現在のしきい値を表示します。センサーは、しきい値ベースのセンサーである必要があります。**raw** 値および処理された値の両方が表示されます。次に示す各センサーの属性も表示されます。

- センサーを所有する IPM コントローラの IPMB アドレス
- センサー番号、センサー名 (SDR のデバイス ID 文字列)、およびそのセンサーにアクセスできる LUN
- センサータイプおよびイベント/読み取りタイプコード

対象のコントローラが複数の LUN 上のセンサーをサポートしている場合、ユーザーはセンサー番号を論理ユニット番号 (LUN) で修飾できます。LUN を省略すると、すべての LUN 上にある指定されたセンサー番号のセンサーに関する情報が表示されます。*lun* には、値 0、1、または 3 を指定できます。LUN 2 は予約されています。

センサー名を指定する際は、LUN 番号で修飾しません。通常、センサー名はコントローラ内で一意であることが前提とされるためです。ただし、コントローラ内に同じ名前のセンサーが複数存在する場合は、それらすべてのセンサーに関する情報が表示されます。

## 例:

IPM コントローラ FE 上の温度センサー Local Temp のしきい値を取得します。

```
# 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
#
```

同じセンサーのしきい値情報を取得しますが、センサーの LUN および番号を指定します。

```
# 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

### 構文:

help [*command* [*subcommand*]]

### 用途:

このコマンドは、サポートされるコマンドおよびそれらの構文に関するヘルプ情報を表示します。

### 例:

コマンドおよびその構文の一覧を取得します。

```
# 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> [noupdate]]
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
showunhealthy
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

```

```
# cli help shelf pwrreorder
```

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

```
Set the Power Order
```

```
PwrReorder <addr1> <fru_id1> before/after <addr2> <fru_id2>
```

```
#
```

特定のコマンドのヘルプを表示します。

```
# cli help shelf pwrreorder
```

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

```
Set the Power Order
```

```
PwrReorder <addr1> <fru_id1> before/after <addr2> <fru_id2>
```

```
#
```

---

# ipmc

## 構文:

```
ipmc [-v] [IPMB-address]  
ipmc board n  
ipmc fan_tray n
```

## 用途:

このコマンドは、指定されたアドレスの IPM コントローラに関する情報を表示します。または、*IPMB-address* が省略された場合に、シェルフマネージャーに認識されるすべての IPM コントローラに関する情報を表示します。

標準モードでは、IPM コントローラに関する次の情報が表示されます。

- 2 桁の 16 進数で表されるコントローラの IPMB アドレス。
- IPM コントローラのエンティティ ID およびエンティティインスタンス
- IPM コントローラで可能な最大 FRU デバイス ID
- PICMG 拡張バージョン。PICMG 3.0 準拠の IPM コントローラの場合、このバージョンは 2.0 に設定してください。
- 現在のホットスワップ状態、前のホットスワップ状態、および IPM コントローラの FRU デバイス 0 (IPM コントローラ自体を表す) の前回の状態変更の原因。ホットスワップ状態 M0 ～ M7 は、PICMG 3.0 仕様で次のように定義されています。
  - M0 – 取り付けられていない
  - M1 – 休止状態
  - M2 – 起動要求
  - M3 – 起動中
  - M4 – FRU は動作中
  - M5 – 停止要求
  - M6 – 停止中
  - M7 – 通信不可

詳細モードでは、IPM コントローラに関する次の追加情報が表示されます。

- Get Device ID IPMI コマンドで返される情報。メーカー ID、製品 ID、デバイス ID、デバイスのファームウェアバージョン、およびサポートされる IPMI バージョンなど
- コントローラ SDR のデバイス ID 文字列
- 16 進数で表される、コントローラ SDR の電源状態通知属性
- 16 進数で表される、コントローラ SDR のグローバル初期化属性
- 16 進数で表される、コントローラ SDR のデバイス機能属性

- コントローラがデバイス SDR を提供するかどうか
- サポートされる機能マスクと各ビットのテキスト形式の説明
- E-キーイングの対象となるポートおよびその状態 (有効/無効) の一覧

例:

アドレス 9C の IPM コントローラに関する情報を取得します。

```
# 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)
#
```

アドレス 9C の IPM コントローラに関する詳細情報を取得します。

```
# 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

## 構文:

localaddress

## 用途:

このコマンドは、現在のシェルフマネージャーの IPMB アドレスを、ハードウェアアドレスに基づいて表示します。汎用的な BMC アドレス 0x20 ではありません。BMC アドレスが冗長シェルフマネージャー間で共有されるのに対して、これらのアドレスは冗長シェルフマネージャー間で異なります。

## 例:

```
# clia localaddress
IPM Sentry Shelf Manager Command Line Interpreter
Local IPMB Address = 0xFC
#
```

---

# minfanlevel

## 構文:

minfanlevel [*level*]

## 用途:

このコマンドは、ファンレベルの最小値を表示または設定します。通常の状態では、冷却制御アルゴリズムは、正常な温度状態が維持されている間にシステム内のファンのレベルを徐々に下げます。ただし、冷却制御アルゴリズムは構成パラメータ MIN\_FAN\_LEVEL またはこのコマンドで指定されたレベルの最小値よりもファンレベルを下げようとはしません。

ファンレベルの最小値のデフォルト値は 1 です。ファンレベルの最小値をそれより大きい値に設定しても、RMCP を介して実行されるコマンド clia setfanlevel または ATCA コマンド SetFanLevel でその値よりも低いファンレベルが設定されることは防止できません。ファンレベルの最小値は、冷却制御機能によるファンレベルの自動管理のみに適用されます。

パラメータを指定しないでこのコマンドを実行すると、現在のファンレベルの最小値が表示されます。

整数のパラメータを指定してこのコマンドを実行すると、指定したパラメータの値がファンレベルの最小値に設定されます。

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

構文:

```
sel [-v] [IPMB-address [record-count [starting-entry]]]
sel clear [IPMB-address]
sel info [IPMB-address]
```

*IPMB-address* は、board *n* または shm *n* の省略形に置き換えることができます。

用途:

このコマンドは、指定された IPM コントローラ (デフォルトでは IPMB アドレス 20h) のシステムイベントログ (SEL) の内容を表示します。オプションのパラメータ *record-count* を使用すると、SEL 内のレコード番号 *starting-entry* から何個のレコードを表示するかを指定できます。オプションのパラメータ *starting-entry* は、最初に出る SEL レコードの、SEL の先頭からの相対的なエントリ番号です。*record-count* および *starting-entry* は、どちらも 1 から SEL のレコード総数までの範囲内で指定する必要があります。オプションのパラメータ *starting-entry* のデフォルト値は 1 です。*starting-entry* は、SEL レコードの RecordID フィールドとは無関係です。

各 SEL レコードに関連する次の情報フィールドが表示されます。

- レコード ID
- レコードタイプ (現在はイベントのみがサポートされており、Event という語が表示される)

- タイムスタンプ (タイムスタンプ付きのレコード用)
- ソースアドレスパラメータ (IPMB アドレス、LUN、およびチャンネル番号)
- イベントを生成するセンサーのタイプおよび番号
- イベント/読み取りタイプコード
- raw 形式および使用可能な場合は処理された形式の 3 バイトのイベントデータ

コマンド `sel clear` は、指定された IPM コントローラ (デフォルトでは IPMB アドレス 20h) の SEL を消去します。

-v オプションを指定すると、SEL エントリの出力がよりユーザーにわかりやすい形式で表示されます。

例:

シェルフマネージャー上の SEL を読み取ります。

```
# 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  
#
```

エントリ #15 (0x0f) から始まる 5 つの SEL エントリを取得します。

```
# 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  
#
```

SEL を消去します。

```
# 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

## 構文:

```
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:

このコマンドは、特定のセンサーに関する情報を表示します。対象のセンサーは、IPM コントローラの IPMB アドレス、センサー番号、またはセンサー名で選択します。センサー名は、センサー SDR のデバイス ID 文字列を二重引用符で囲んで指定します。センサー名またはセンサー番号のどちらも指定しない場合、指定した IPM コントローラ上のすべてのセンサーに関する情報が表示されます。パラメータを指定しない場合は、認識されるすべてのセンサーに関する情報が表示されます。

対象のコントローラが複数の LUN 上のセンサーをサポートしている場合、ユーザーはセンサー番号を論理ユニット番号 (LUN) で修飾できます。LUN を省略すると、すべての LUN 上にある指定されたセンサー番号のセンサーに関する情報が表示されます。*lun* には、値 0、1、または 3 を指定できます。LUN 2 は予約されています。

センサー名を指定する際は、LUN 番号で修飾しません。通常、センサー名はコントローラ内で一意であることが前提とされるためです。ただし、コントローラ内に同じ名前のセンサーが複数存在する場合は、それらすべてのセンサーに関する情報が表示されます。

標準モードでは、各センサーに関する次の情報が表示されます。

- センサーを所有する IPM コントローラの IPMB アドレス
- センサー番号、センサー名 (SDR のデバイス ID 文字列)、およびそのセンサーにアクセスできる LUN
- センサータイプおよびイベント/読み取りタイプコード
- 関連するエンティティのエンティティ ID、エンティティインスタンス (センサーが FRU に関連付けられている場合は FRU デバイス ID)

詳細モードの場合のみ、センサーに関する次の情報が表示されます。これらの属性については、IPMI 仕様を参照してください。

- 表明マスク
- 非表明マスク
- センサー状態の設定可能/読み取り可能マスク (ディスクリットセンサーの場合)、またはしきい値 (しきい値ベースのセンサーの場合)

しきい値ベースのセンサーの場合のみ、詳細モードで次の情報が表示されます。

- センサーの単位: 基本単位および変更単位
- 単位のパーセンテージ、修飾子、およびレート
- アナログ形式およびフラグ
- 線形化パラメータ、M、B、K1、K2 係数
- 許容係数および精度係数
- 公称値、正常な最大値、正常な最小値、最大値、最小値
- しきい値の上限: 非クリティカル、クリティカル、回復不能
- しきい値の下限: 非クリティカル、クリティカル、回復不能
- ヒステリシス値: 正の値および負の値

例:

IPM コントローラ FE 上のセンサー FAN 4 に関する標準の情報を取得します。

```
# 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]
#
```

IPM コントローラ 9C 上の センサー 2 に関する詳細情報を取得します。

```
# 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)
#

```

前述の例と同じコマンドを、センサーの LUN を明示的に指定して実行します。

```

# 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

### 構文:

```
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:

このコマンドは、しきい値ベースのセンサーの場合は指定されたセンサーの実際の値を、またはディスクリットセンサーの場合は現在表明されている状態を表示します。対象のセンサーは、IPM コントローラの IPMB アドレス、センサー番号、またはセンサー名で選択します。センサー名は、センサー SDR のデバイス ID 文字列を二重引用符で囲んで指定します。センサー名とセンサー番号をどちらも指定しない場合、指定された IPM コントローラ上のすべてのセンサーの値が表示されます。パラメータを指定しない場合は、認識されるすべてのセンサーの値が表示されます。

対象のコントローラが複数の LUN 上のセンサーをサポートしている場合、ユーザーはセンサー番号を論理ユニット番号 (LUN) で修飾できます。LUN を省略すると、すべての LUN 上にある指定されたセンサー番号のセンサーに関する情報が表示されます。lun には、値 0、1、または 3 を指定できます。LUN 2 は予約されています。

センサー名を指定する際は、LUN 番号で修飾しません。通常、センサー名はコントローラ内で一意であることが前提とされるためです。ただし、コントローラ内に同じ名前のセンサーが複数存在する場合は、それらすべてのセンサーに関する情報が表示されます。

各センサーに関する次の情報が表示されます。

- センサーを所有する IPM コントローラの IPMB アドレス
- センサー番号、センサー名 (SDR のデバイス ID 文字列)、およびそのセンサーにアクセスできる LUN
- センサータイプおよびイベント/読み取りタイプコード
- センサーの値 (しきい値ベースのセンサーの場合)、または現在表明されている状態の raw 形式のマスク (ディスクリットセンサーの場合)
- デコードされた 16 進表記で表されるしきい値超過状態

値または表明された状態が、raw 形式および処理された形式の両方で表示されます。処理された形式では、アナログ値が M、B、および R に従って変換され、単位名とともに「27 degrees」のように表示されます。ディスクリット値は、イベント/読み取りコードタイプに従って注釈が付けられます。たとえば、イベント/読み取りコード 2 では、表明された状態 0 が「Transition to Idle」として表示されます。

例:

IPM コントローラ FE 上の温度センサー Local Temp のセンサーデータの値を取得します。

```
# 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
```

IPM コントローラ 9C 上のディスクリット (ホットスワップ) センサー #0 のセンサーデータの値を取得します。

```
# 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
```

同じセンサーのセンサーデータの値を取得しますが、明示的に LUN で修飾します。

```
# 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

### 構文:

```
sensorread IPMB-address [lun:]sensor-number
```

### 用途:

このコマンドは、指定されたセンサーの **raw** 値を表示します。コマンド **sensorread** と **sensordata** で異なるのは、コマンド **sensorread** が対象の IPM コントローラの存在またはセンサー番号の有効性を確認せずに、IPMB を介して直接 Get Sensor Reading 要求を送信する点だけです。このコマンドではセンサーの SDR を取得しないため、取得したデータを処理することはできません。

対象のコントローラが複数の LUN 上のセンサーをサポートしている場合、ユーザーはセンサー番号を論理ユニット番号 (LUN) で修飾できます。LUN を省略すると、LUN 0 が使用されます。*lun* には、値 0、1、または 3 を指定できます。LUN 2 は予約されています。

各センサーに関する次の情報が表示されます。

- センサーを所有する IPM コントローラの IPMB アドレス
- センサー番号、センサー名 (SDR のデバイス ID 文字列)、およびそのセンサーにアクセスできる LUN
- センサータイプおよびイベント/読み取りタイプコード
- センサーの値 (しきい値ベースのセンサーの場合)、または現在表明されている状態の raw 形式のマスク (ディスクリットセンサーの場合)

### 例:

IPM コントローラ FC 上のセンサー 4 のセンサーデータの値を取得します。**sensorread** コマンドでは、未処理のセンサー値のみが提供される点に注意してください。また、コマンド例では LUN を明示的に指定している点にも注目してください。

```
# 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

### 構文:

```
session
```

### 用途:

このコマンドは、アクティブな RMCP セッションに関する情報を表示します。情報には次の項目が含まれます。

- 可能な最大セッション数および現在アクティブなセッション数
- 現在アクティブな各セッションに関する次の情報
  - セッションハンドル
  - セッションをアクティブにする間に使用されるユーザー ID およびユーザー名
  - セッションの最大特権レベル
  - IPMI チャネル番号およびタイプ
  - LAN セッションの場合は接続先の IP アドレスおよびポート番号

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

構文:

```
setextracted IPMB-address fru-id
```

用途:

このコマンドは、指定された FRU が物理的にシェルフから引き出されていることをシェルフマネージャーに通知します。指定された FRU が状態 M7 にある場合、シェルフマネージャーはその FRU を状態 M0 (FRU が物理的に存在しない) にします。

例:

```
# clia setextracted 9c 0
IPM Sentry Shelf Manager Command Line Interpreter
Set FRU extracted state successfully
#
```

---

## setfanlevel

構文:

```
setfanlevel IPMB-address fru-id level
setfanlevel fan_tray n level
setfanlevel all level
```

用途:

このコマンドは、コマンドパラメータで指定された FRU が制御するファンに対して、新しいレベルを設定します。最小レベルは 1 で、最大レベルは 15 です。

all 修飾子を使用すると、このコマンドはシェルフ内で認識されるすべてのファンを同じレベルに設定しようと試みます。

例:

IPMB アドレス 0x20 の FRU #2 にあるファンのファンレベルを 5 に設定します。

```
# clia setfanlevel 20 2 5

IPM Sentry Shelf Manager Command Line Interpreter

20: FRU # 2 Set Fan Level to: 5
#
```

シェルフ内で認識されるすべてのファンに対して、ファンレベルを 4 に設定します。

```
# 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

## 構文:

```
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:

このコマンドを使用すると、ユーザーは特定の LED または指定した FRU のすべての LED の状態を設定できます。

最初の引数 *IPMB-address* には、IPM コントローラの IPMB アドレスを指定します。2 番目の引数 *fru-id* には、FRU デバイス ID を指定します。3 番目の引数には、LED ID (数値) または ALL のいずれかを指定できます。ALL を指定した場合は、指定した操作がすべての LED に適用されます。

引数 *LedOp* には、PICMG 3.0 仕様に基づいて FRU に適用される操作を指定します。操作は次のように定義されています。

- ON — LED を点灯する
- OFF — LED を消灯する
- LOCAL — LED のローカル制御に戻る
- BLINK — *onTime* ミリ秒間の点灯と *offTime* ミリ秒間の消灯を繰り返して、LED を点滅させる
- TEST — *onTime* ミリ秒間のランプテストを実行する

TEST 操作では、*onTime* を 12800 ミリ秒 (12.8 秒) 未満の時間に設定し、BLINK 操作では、*onTime* および *offTime* の両方の値を 10 ～ 2500 ミリ秒の範囲内で設定する必要があります。

オプションのパラメータ *LedColor* には、記号名または 10 進数の値のいずれかを使用して色を指定します。PICMG 3.0 仕様に準拠して、色の記号名は次に示す 10 進数の値に対応しています。パラメータを指定しない場合は、デフォルトの LED の色が使用されます。

- BLUE = 1
- RED = 2
- GREEN = 3
- AMBER = 4
- ORANGE = 6
- NONE = 14 (色を変更しない)

例:

IPMB アドレス 20h の IPM コントローラの FRU #0 にある LED #1 を消灯します。

```
# clia setfruledstate 20 0 1 OFF
IPM Sentry Shelf Manager Command Line Interpreter

    Setting FRU's led state completed successfully, status = 0x0
#
```

IPMB アドレス 20h の IPM コントローラの FRU #0 にある LED #1 のローカル制御を有効にします。

```
# clia setfruledstate 20 0 1 LOCAL
IPM Sentry Shelf Manager Command Line Interpreter

    Setting FRU's led state completed successfully, status = 0x0
#
```

IPMB アドレス 20h の IPM コントローラの FRU #0 にある LED #1 の点滅を有効にします。点滅は、デフォルトの色で行われます。点灯時間は 100 ミリ秒、消灯時間は 200 ミリ秒です。

```
# 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

構文:

```
sethysteresis IPMB-address [lun:] sensor_id | sensor_name pos | neg [-r]
value
```

用途:

このコマンドは、指定されたセンサーに対して、指定されたヒステリシス値を設定します。センサーは、しきい値ベースのセンサーである必要があります。センサーは、指定されたしきい値のヒステリシスをサポートし、そのヒステリシスは設定可能である必要があります。

対象のコントローラが複数の LUN 上のセンサーをサポートしている場合、ユーザーはセンサー番号を論理ユニット番号 (LUN) で修飾できます。コマンドは、pos 引数が指定されると正のヒステリシスを設定し、neg 引数が指定されると負のヒステリシスを設定します。

例:

IPMB アドレス 0xFC の IPM コントローラのセンサー #2 に正のヒステリシスを設定します。

```
# clia sethysteresis FC 2 pos 10
IPM Sentry Shelf Manager Command Line Interpreter

        Positive hysteresis set successfully to 0xA, previous: 0x0
#
```

---

## setipmbstate

構文:

```
setipmbstate IPMB-address A|B [link] 1|0 (放射線状の IPMB-0 環境)
setipmbstate IPMB-address A|B 1|0 (バス型の IPMB-0 環境)
```

用途:

このコマンドは、対象の IPM コントローラ上で IPMB リンクを有効または無効にします。2 番目の引数には、有効または無効にするバス (IPMB-A または IPMB-B) を定義します。最後の引数には、実行する操作を定義します。リンクを有効にするには 1 を指定し、リンクを無効にするには 0 を指定します。

このコマンドは、バス型と放射線状の接続環境では動作が異なります。バス型の接続環境、および対象の IPM コントローラが IPMB ハブ以外の場合の放射線状の接続環境では、引数 *link* は使用されません。放射線状の接続環境の IPMB ハブのコントローラの場合、引数 *link* はオプションです。

コマンドで *link* を指定すると、特定の放射線状の IPMB リンク (1 ~ 95) が有効化または無効化されます。*link* を省略すると、放射線状のシステムの IPMB ハブ上のすべてのリンクが有効または無効になります。

例:

IPMB アドレス 92h の IPM コントローラ上の IPMB-A リンクを無効にします。

```
# clia setipmbstate 92 A 0
IPM Sentry Shelf Manager Command Line Interpreter

Command executed successfully
```

シェルフマネージャーのバス B (IPMB ハブ) の放射線状の IPMB リンク 3 を有効にします。

```
# clia setipmbstate 20 B 3 1
IPM Sentry Shelf Manager Command Line Interpreter

Command executed successfully
```



# setlanconfig

構文:

```
setlanconfig channel parameter-name additional-parameters
setlanconfig channel parameter-number additional-parameters
```

用途:

このコマンドは、指定されたチャンネル上で指定された LAN 構成パラメータの値を設定します。チャンネル番号、構成パラメータの名前または番号、およびパラメータ値は、明示的に指定します。

表 A-3 に、setlanconfig コマンドでサポートされる LAN 構成パラメータの名前および番号の一覧を示します。

表 A-3      setlanconfig の LAN 構成パラメータ

パラメータ名	番号	説明
auth_enables	2	LAN チャンネルのコールバック、ユーザー、オペレータ、管理者、および OEM の認証タイプ有効化フラグを含む 8 ビットの 5 つの値。
ip	3	LAN チャンネルに割り当てられた IP アドレスを含む、ドット区切りの 10 進表記の文字列値。

表 A-3 setlanconfig の LAN 構成パラメータ (続き)

subnet_mask	6	LAN チャンネルに割り当てられたサブネットマスクを含む、ドット区切りの 10 進表記の文字列値。
ipv4_hdr_param	7	<p>RMCP パケットを送信するためのさまざまな IPv4 ヘッダーパラメータを含む、次に示す 8 ビットの 3 つの値。</p> <ul style="list-style-type: none"> <li>• 生存期間</li> <li>• IP ヘッダーフラグ (ビット [7:5])</li> <li>• 優先順位 (ビット [7:5]) およびサービスタイプ (ビット [4:1])</li> </ul>
arp_control	10	<p>LAN チャンネルで ARP 動作を制御する、次に示す 2 つのフラグ。</p> <ul style="list-style-type: none"> <li>• ARP 要求への応答を有効にする</li> <li>• Gratuitous ARP の送信を有効にする</li> </ul>
arp_interval	11	固定小数点形式で表す Gratuitous ARP の間隔。整数部分は秒数を表し、小数点以下の部分はミリ秒を表します。
dft_gw_ip	12	デフォルトゲートウェイの IP アドレスを含む、ドット区切りの 10 進表記の文字列値。
backup_gw_ip	14	バックアップゲートウェイの IP アドレスを含む、ドット区切りの 10 進表記の文字列値。
community	16	PET トラップの Community String フィールドに格納される文字列値 (最大 18 記号)。
destination_type	18	<p>指定された設定セレクトで識別される宛先タイプ。このパラメータには、設定セレクトを指定してください。各宛先タイプのエントリには、次のフィールドが含まれます。</p> <ul style="list-style-type: none"> <li>• 宛先タイプ (0 ～ 7)</li> <li>• アラート確認応答フラグ</li> <li>• アラート確認応答タイムアウト/再試行間隔の秒数 (1 ～ 256)</li> <li>• 再試行の回数 (0 ～ 7)</li> </ul>
destination_address	19	<p>指定された設定セレクトに関連付けられた宛先アドレス。このパラメータには、設定セレクトを指定してください。各宛先アドレスのエントリには、次のフィールドが含まれます。</p> <ul style="list-style-type: none"> <li>• ゲートウェイセレクト: 0 はデフォルトを使用、1 はバックアップを使用</li> <li>• IP アドレス (ドット区切りの 10 進表記の文字列)</li> <li>• MAC アドレス (コロン [:] で区切られた 16 進数の 6 つのバイトの文字列)</li> </ul>

## auth\_enables

### 構文:

```
setlanconfig channel auth_enables value1 value2 value3 value4 value5
setlanconfig channel 2 value1 value2 value3 value4 value5
```

### 用途:

このコマンドは、LAN パラメータ `auth_enables` の現在の値を設定します。このパラメータは、サポートされる 5 つの各特権レベル (コールバック、ユーザー、管理者、オペレータ、および OEM) に対して、現在シェルフマネージャーによって有効になる認証タイプを指定します。パラメータは連続する 5 バイトで表され、各バイトがそれぞれの特権レベルに対応し、次のように定義されるビットでビットマスクとして処理されます。

- 0x01 — なし
- 0x02 — MD2
- 0x04 — MD
- 0x10 — 単純パスワード/キー
- 0x20 — OEM 所有

パラメータ `value1` から `value5` には、これらのバイトの値を 16 進数で指定します。シェルフマネージャーは現在コールバックおよび OEM の特権レベルをサポートしていません。そのため、これらの特権レベルに対応する `value1` および `value5` パラメータには、0 を指定してください。

### 例:

```
# clia setlanconfig 1 auth_enables 0 1 1 1 0
IPM Sentry Shelf Manager Command Line Interpreter
Authentication Type Enables set successfully
#
```

## ip

### 構文:

```
setlanconfig channel ip value
setlanconfig channel 3 value
```

### 用途:

このコマンドは、チャンネルで使用される現在の IP アドレスを設定します。値は、ドット区切りの 10 進表記の IP アドレスです。

例:

```
# clia setlanconfig 1 ip 172.16.2.203
IPM Sentry Shelf Manager Command Line Interpreter
IP Address set successfully
#
```

## subnet\_mask

構文:

```
setlanconfig channel subnet_mask value
setlanconfig channel 6 value
```

用途:

このコマンドは、チャネルで使用する現在の IP サブネットマスクを設定します。値は、ドット区切りの 10 進表記のサブネットマスクです。

例:

```
# clia setlanconfig 1 subnet_mask 255.255.255.0
IPM Sentry Shelf Manager Command Line Interpreter
Subnet Mask set successfully
#
```

## ipv4\_hdr\_param

構文:

```
setlanconfig channel ipv4_hdr_param value1 value2 value3
setlanconfig channel 7 value1 value2 value3
```

用途:

このコマンドは、シェルフマネージャの IP 4 ヘッダーパラメータを設定します。3 つのパラメータ *value1*、*value2*、および *value3* には、シングルバイトの値を 16 進表記で指定します。バイトの内容は IPMI 1.5 仕様のセクション 19.2 に準拠し、次の属性が含まれています。

- バイト 1: 生存期間
- バイト 2: IP ヘッダーフラグ (ビット [7:5])
- バイト 3: 優先順位 (ビット [7:5]) およびサービスタイプ (ビット [4:1])

例:

```
# clia setlanconfig 1 ipv4_hdr_param 37 E0 11
IPM Sentry Shelf Manager Command Line Interpreter
IPv4 Header Parameters set successfully
#
```

## arp\_control

構文:

```
setlanconfig channel arp_control value
setlanconfig channel 10 value
```

用途:

このコマンドは、LAN パラメータ `arp_control` の現在の値を設定します。このパラメータは、シェルフマネージャーによって提供される追加の ARP サポートを指定します。また、シングルバイトで表され、次のように定義されるビットでビットマスクとして処理されます。

- 0x01 — シェルフマネージャーが生成する Gratuitous ARP を有効にする
- 0x02 — シェルフマネージャーが生成する ARP 応答を有効にする

その他のビットは予約されているため、0 に設定されている必要があります。

例:

```
# clia setlanconfig 1 arp_control 3
IPM Sentry Shelf Manager Command Line Interpreter
BMC-generated ARP control set successfully
#
```

## arp\_interval

### 構文:

```
setlanconfig channel arp_interval value  
setlanconfig channel 11 value
```

### 用途:

このコマンドは、チャンネルで使用する現在の ARP 間隔を設定します。値は、固定小数点形式の秒数/ミリ秒数で表し、小数点以下が含まれる場合もあります。IPMI でこのパラメータの定義に従って、値は 500 ミリ秒単位で長い時間の方に切り上げられます。

### 例:

```
# clia setlanconfig 1 arp_interval 3.5  
IPM Sentry Shelf Manager Command Line Interpreter  
Gratuitous ARP interval set successfully  
#
```

## dft\_gw\_ip

### 構文:

```
setlanconfig channel dft_gw_ip value  
setlanconfig channel 12 value
```

### 用途:

このコマンドは、チャンネルで使用するデフォルトゲートウェイの IP アドレスを設定します。値は、ドット区切りの 10 進表記の IP アドレスです。

### 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

### 構文:

```
setlanconfig channel backup_gw_ip value  
setlanconfig channel 14 value
```

### 用途:

このコマンドは、チャンネルで使用するバックアップゲートウェイの IP アドレスを設定します。値は、ドット区切りの 10 進表記の IP アドレスです。

### 例:

```
# clia setlanconfig 1 backup_gw_ip 172.16.2.100  
IPM Sentry Shelf Manager Command Line Interpreter  
Backup Gateway Address set successfully  
#
```

## community

### 構文:

```
setlanconfig channel community value  
setlanconfig channel 16 value
```

### 用途:

このコマンドは、PET トラップで使用されるコミュニティー文字列パラメータを設定します。値には、二重引用符で囲んだ文字列を指定します。

### 例:

```
# clia setlanconfig 1 community "Community"  
IPM Sentry Shelf Manager Command Line Interpreter  
Community string set successfully  
#
```

## destination\_type

### 構文:

```
setlanconfig channel destination_type set-selector value1 value2 value3
setlanconfig channel 18 set-selector value1 value2 value3
```

### 用途:

このコマンドは、*set-selector* と等しいインデックスを持つ宛先テーブルの要素を設定します。インデックスは、0 から始まります。セクタ 0 は、揮発性の宛先を指定するために使用します。値 *value1*、*value2*、および *value3* には、IPMI 仕様のセクション 19.2 に従って、新しい宛先に関する情報を指定します。次の情報を指定します。

- アラートの宛先タイプ (PET トラップまたは OEM の宛先、アラートが確認される必要があるかどうか)
- アラートの確認応答タイムアウト
- 再試行の回数

### 例:

```
# clia setlanconfig 1 destination_type 2 80 3 5
IPM Sentry Shelf Manager Command Line Interpreter
Destination Type set successfully
#
```

## destination\_address

### 構文:

```
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:

このコマンドは、*set-selector* と等しいインデックスを持つ宛先アドレステーブルの要素を設定します。インデックスは、0 から始まります。セクタ 0 は、揮発性の宛先を指定するために使用します。コマンドパラメータには、必要な情報を指定します。

- *gateway-sel* — 使用するゲートウェイ。0 はデフォルトゲートウェイの場合、1 はバックアップゲートウェイの場合
- *IP-address* — 宛先 IP アドレス (ドット区切りの 10 進表記)
- *MAC-address* — 宛先 MAC アドレス (コロンで区切られた 16 進数の 6 つのバイト)

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

構文:

```
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:

このコマンドは、指定された FRU のロックビットを、指定された状態に設定します。0 はロック解除状態、1 はロック状態を指定します。FRU は、それを所有する IPM コントローラの IPMB アドレスと FRU デバイス ID で指定します。PICMG 3.0 のコンテキストでは、FRU デバイス ID 0 は IPM コントローラ自体を指定します。

ロックビットは、PICMG 3.0 仕様に従って、状態 M1 (休止状態) から状態 M2 (起動要求) への FRU の自立的な移行を許可するかどうかを制御します。ロックビットが設定されると、この移行は許可されません。シェルフマネージャーが Deactivate コマンドを FRU に送信すると、その FRU は状態 M1 に移行し、ロックビットを設定してそれ以降の状態移行を回避します。

このコマンドで、前に手動で停止された FRU のロックビットをクリアすると、その FRU を再度起動できます。

例:

アドレス 9C の IPM コントローラ本体のロックビットをクリアして、IPM コントローラを再度起動可能にします。

```
# clia setlocked 9c 0 0
IPM Sentry Shelf Manager Command Line Interpreter
Lock set successfully to 0x0
#
```

# setpefconfig

## 構文:

```
setpefconfig parameter-name additional-parameters  
setpefconfig parameter-number additional-parameters
```

## 用途:

このコマンドは、指定された PEF 構成パラメータの新しい値を設定します。表 A-4 に、このコマンドで設定可能な PEF 構成パラメータの名前と番号を示します。

表 A-4 setpefconf の PEF 構成パラメータ

パラメータ名	番号	説明
control	1	PEF の制御フラグを表す 8 ビットの値 (PEF の有効化、PEF 起動遅延の有効化など)。
action_control	2	PEF 処理のグローバル制御フラグを表す 8 ビットの値 (リセットの有効化、電源切断の有効化など)。
startup_delay	3	システムの電源投入およびリセット後の PEF の遅延時間 (秒単位)。
alert_startup_delay	4	システムの電源投入およびリセット後のアラートの遅延時間 (秒単位)。

表 A-4 setpefconf の PEF 構成パラメータ (続き)

パラメータ名	番号	説明
event_filter	6	<p>指定された設定セクタで識別されるイベントフィルタテーブルのエントリ。次に示す、IPMI 仕様 version 1.5 の表 15-2 の定義に従ってエンコードされた、19 個の 16 進数の数値で構成されます。</p> <ul style="list-style-type: none"> <li>• フィルタ構成</li> <li>• イベントフィルタ処理</li> <li>• アラートポリシー番号</li> <li>• イベントの重要度</li> <li>• 生成 ID バイト 1</li> <li>• 生成 ID バイト 2</li> <li>• センサータイプ</li> <li>• センサー番号</li> <li>• イベントトリガー (イベント/読み取りタイプ)</li> <li>• イベントデータ 1 イベントオフセットマスク</li> <li>• イベントデータ 1 AND マスク</li> <li>• イベントデータ 1 コンペア 1</li> <li>• イベントデータ 1 コンペア 2</li> <li>• イベントデータ 2 AND マスク</li> <li>• イベントデータ 2 コンペア 1</li> <li>• イベントデータ 2 コンペア 2</li> <li>• イベントデータ 3 AND マスク</li> <li>• イベントデータ 3 コンペア 1</li> <li>• イベントデータ 3 コンペア 2</li> </ul>
event_filter_data1	7	<p>指定された設定セクタで識別されるイベントフィルタのテーブルのエントリの先頭バイト。</p>
alert_policy	9	<p>指定された設定セクタで識別されるアラートポリシーテーブルのエントリ。次に示す、IPMI 1.5 の表 15-4 の定義に従ってエンコードされた、5 つの 16 進数の数値で構成されます。</p> <ul style="list-style-type: none"> <li>• ポリシー番号 (4 ビットの値)</li> <li>• 有効化/無効化ビットを含むポリシー (4 ビットの値)</li> <li>• チャネル番号 (4 ビットの値)</li> <li>• 宛先セクタ (4 ビットの値)</li> <li>• アラート文字列セット/セクタ</li> </ul>
system-guid	10	<p>PET トラップの GUID フィールドへの値の入力に使用される GUID。</p>
alert_string_key	12	<p>指定された設定セクタで識別されるアラート文字列キー。8 ビットの 2 つの値、つまりイベントフィルタ番号およびアラート文字列セットで構成されます。</p>

表 A-4 setpefconf の PEF 構成パラメータ (続き)

パラメータ名	番号	説明
alert_string	13	指定された設定セクタで識別されるアラート文字列。
oem_filter	97	指定された設定セクタで識別される OEM フィルタテーブルのエントリ。次の 3 つの数値で構成されます。 <ul style="list-style-type: none"> <li>• バイト 1: SEL レコードタイプの範囲の下限</li> <li>• バイト 2: SEL レコードタイプの範囲の上限</li> <li>• バイト 3: 前述の範囲に一致するレコードタイプを持つ SEL エントリに対して呼び出されるアラートポリシー番号。</li> </ul>

## control

### 構文:

```
setpefconfig control value
setpefconfig 1 value
```

### 用途:

このコマンドは、PEF パラメータ `control` の新しい値を設定します。このパラメータはシングルバイトで、次のように定義されるビットでビットマスクとして処理されます。

- 0x01 — PEF を有効にする
- 0x02 — PEF 処理に対するイベントメッセージの生成を有効にする
- 0x04 — システムの電源投入およびリセット時の PEF の起動遅延を有効にする
- 0x08 — PEF のアラート起動遅延を有効にする

その他のビットは予約されているため、0 に設定されている必要があります。値は 16 進数で入力します。

### 例:

```
# clia setpefconfig control 7
IPM Sentry Shelf Manager Command Line Interpreter
PEF control set successfully
#
```

## action\_control

### 構文:

```
setpefconfig action_control value  
setpefconfig 2 value
```

### 用途:

このコマンドは、PEF パラメータ `action_control` の新しい値を設定します。このパラメータはシングルバイトで、次のように定義されるビットでビットマスクとして処理されます。

- 0x01 — アラート処理を有効にする
- 0x02 — 電源切断処理を有効にする
- 0x04 — リセット処理を有効にする
- 0x08 — 電源の再投入処理を有効にする
- 0x10 — OEM 処理を有効にする
- 0x20 — 診断の中断を有効にする

その他のビットは予約されているため、0 に設定されている必要があります。値は 16 進数で入力します。

### 例:

```
# clia setpefconfig action_control 3f  
IPM Sentry Shelf Manager Command Line Interpreter  
PEF action control set successfully  
#
```

## startup\_delay

### 構文:

```
setpefconfig startup_delay value  
setpefconfig 3 value
```

### 用途:

このコマンドは、PEF パラメータ `startup_delay` の新しい値を設定します。このパラメータはシングルバイトで、PEF 機能が起動時に遅延する秒数を表します。値は 10 進数の秒数で指定します。

例:

```
# clia setpefconfig startup_delay 45  
IPM Sentry Shelf Manager Command Line Interpreter  
PEF startup delay set successfully  
#
```

## alert\_startup\_delay

構文:

```
setpefconfig startup_delay value  
setpefconfig 4 value
```

用途:

このコマンドは、PEF パラメータ `alert_startup_delay` の現在の値を設定します。このパラメータはシングルバイトで、アラート機能が起動時に遅延する秒数を表します。値は 10 進数の秒数で指定します。

例:

```
# clia setpefconfig alert_startup_delay 45  
IPM Sentry Shelf Manager Command Line Interpreter  
Alert startup delay set successfully  
#
```

## event\_filter

構文:

```
setpefconfig event_filter set-selector value1 ... <value19>  
setpefconfig 6 set-selector value1 ... <value19>
```

用途:

このコマンドは、*set-selector* と等しいインデックスを持つイベントフィルタテーブルの要素を設定します。インデックスは、1 から始まります。次に示す新しい要素の内容は、*value1* から <value19> までの 19 個の数値で指定します。この数値は、IPMI 仕様 version 1.5 の表 15-2 の定義に従ってエンコードされた 16 進数です。

- フィルタ構成
- イベントフィルタ処理

- アラートポリシー番号
- イベントの重要度
- 生成 ID バイト 1
- 生成 ID バイト 2
- センサータイプ
- センサー番号
- イベントトリガー (イベント/読み取りタイプ)
- イベントデータ 1 イベントオフセットマスク
- イベントデータ 1 AND マスク
- イベントデータ 1 コンペア 1
- イベントデータ 1 コンペア 2
- イベントデータ 2 AND マスク
- イベントデータ 2 コンペア 1
- イベントデータ 2 コンペア 2
- イベントデータ 3 AND マスク
- イベントデータ 3 コンペア 1
- イベントデータ 3 コンペア 2

例:

イベントフィルタ 2 を設定して、アドレス 9C の IPM コントローラの FRU 0 が状態 M0 に移行したときに、アラート処理をトリガーします。アラートは、アラートポリシー 1 に従って送信されます。

```
# 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

構文:

```
setpefconfig event_filter_data1 set-selector value
setpefconfig 7 set-selector value
```

用途:

このコマンドは、*set-selector* と等しいインデックスを持つイベントフィルタテーブルの要素の先頭バイトを設定します。インデックスは、1 から始まります。このバイトは、16 進数で指定します。このバイトのビットには、次の意味があります。

- 0x80 — このフィルタは有効

- 0x40 — このフィルタはメーカーで事前構成されているため、ソフトウェアで変更するべきではない

その他のビットは予約されているため、0 に設定されている必要があります。

このコマンドを使用すると、イベントフィルタの有効/無効の状態をすばやく切り替えることができます。つまり、テーブルエントリ全体を書き換えなくても、イベントフィルタを有効および無効にできます。

例:

イベントフィルタ 2 を有効にします。

```
# clia setpefconfig event_filter_data1 2 80
IPM Sentry Shelf Manager Command Line Interpreter
Event filter data1 set successfully
#
```

イベントフィルタ 2 を無効にします。

```
# clia setpefconfig event_filter_data1 2 0
IPM Sentry Shelf Manager Command Line Interpreter
Event filter data1 set successfully
#
```

## alert\_policy

構文:

```
setpefconfig alert_policy set-selector value1 value2 value3 value4 value5
setpefconfig 9 set-selector value1 value2 value3 value4 value5
```

用途:

このコマンドは、指定された設定セクタで識別されるアラートポリシーテーブルのエントリを設定します。新しい要素の内容は、次の *value1* から *value5* までの 5 つの数値で指定します。この数値は、IPMI 1.5 の表 15-4 の定義に従ってエンコードされた 16 進数です。

- ポリシー番号 (4 ビットの値)
- 有効化/無効化ビットを含むポリシー (4 ビットの値)
- チャンネル番号 (4 ビットの値)
- 宛先セクタ (4 ビットの値)
- アラート文字列セット/セクタ

### 例:

次の例では、アラートポリシーテーブルのエントリ 2 を、次に示す属性で設定します。

- ポリシー番号 = 5
- 有効化ポリシー = 常にこの宛先にアラートを送信する
- 宛先チャンネル = 1
- 宛先セクタ = 1
- アラート文字列セクタ = すべてのイベントに対して文字列 1 を使用する

```
# clia setpefconfig alert_policy 2 5 8 1 1 1
IPM Sentry Shelf Manager Command Line Interpreter
Policy set successfully
#
```

## system\_guid

### 構文:

```
setpefconfig system_guid guid-value
setpefconfig 10 guid-value
setpefconfig system_guid none
setpefconfig 10 none
```

### 用途:

このコマンドは、PEF パラメータ `system_guid` の現在の値を設定します。このパラメータは、PET トラップ PDU でアラートの宛先に送信される GUID を表します。この GUID は、別個の GUID として、またはシステム GUID と同じ GUID として定義できます。

`guid-value` には、標準的な GUID 形式 `xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx` に従った実際の GUID、または記号値 `none` を指定できます。前者の場合、PEF 機能は指定された GUID を PET トラップに使用します。後者の場合、PEF 機能は、IPMI の Get System GUID コマンドの出力結果であるシステム GUID を、デフォルトで PET トラップに使用します。

例:

```
# clia setpefconfig system_guid 23662F7F-BA1B-4b65-8808-94CA09C9BBB0
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

構文:

```
setpefconfig alert_string_key set-selector value1 value2
setpefconfig 12 set-selector value1 value2
```

用途:

このコマンドは、*set-selector* と等しいインデックスを持つアラート文字列キーテーブルの要素を設定します。インデックスは、1 から始まります。設定セクタ 0 は、揮発性のアラート文字列を指定するために使用できます。各キーは、アラートを生成するためにイベントフィルタとアラート文字列を関連付ける役割を果たし、イベントフィルタ番号およびアラート文字列番号で構成されています。どちらの値も 8 ビット値で、それぞれパラメータ *value1* および *value2* に 16 進数で指定します。

例:

```
# clia setpefconfig alert_string_key 2 10 11
IPM Sentry Shelf Manager Command Line Interpreter
Alert string keys set successfully
#
```

## alert\_string

構文:

```
setpefconfig alert_string set-selector <string-value>
setpefconfig 13 set-selector <string-value>
```

## 用途:

このコマンドは、*set-selector* と等しいインデックスを持つアラート文字列テーブルの要素を設定します。インデックスは、1 から始まります。インデックス 0 は、揮発性のアラート文字列を指定するために使用できます。文字列の値は二重引用符 (") で囲む必要があり、引用符内には特殊文字および改行を含めることができます。

## 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

## 構文:

```
setpefconfig oem_filter set-selector value1 value2 value3
setpefconfig 97 set-selector value1 value2 value3
```

## 用途:

OEM フィルタテーブルは、Pigeon Point Systems 社で定義された IPMI 仕様の OEM 拡張機能です。このテーブルによって、プラットフォームイベントのほかに OEM のタイプスタンプ付きおよびタイムスタンプなしの SEL エントリにも、PEF を適用できます。レコードタイプの範囲は、C0h ~ FFh です。

OEM フィルタテーブルの各エントリは、この OEM フィルタが適用されるレコードタイプの範囲を OEM レコードタイプの範囲内で定義し、レコードタイプが一致するレコードが SEL に記録されたときに呼び出されるアラートポリシー番号を定義します。

このコマンドは、OEM フィルタテーブルのエントリを設定します。エントリ番号は、指定された設定セクタで特定されます。エントリは、次の 3 つの数値で構成されます。

- バイト 1: SEL レコードタイプの範囲の下限
- バイト 2: SEL レコードタイプの範囲の上限
- バイト 3: 前述の範囲に一致するレコードタイプを持つ SEL エントリに対して呼び出されるアラートポリシー番号

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

構文:

```
setsensoreventenable IPMB-address sensor-name global
[assertion_events [deassertion_events]]
```

```
setsensoreventenable IPMB-address [lun:]sensor-number global
[assertion_events [deassertion_events]]
```

*IPMB-address* の代わりに、次の方法でも参照できます。

```
board n
shm n
```

用途:

このコマンドは、指定されたセンサーのイベント有効化マスクを変更します。センサーは、所有 IPM コントローラの IPMB アドレスおよびセンサー名、またはセンサー番号で指定します。別の方法として、ボード番号または専用のシェルフマネージャー番号を使用して対象の IPM コントローラを指定できます。

対象のコントローラが複数の LUN 上のセンサーをサポートしている場合、ユーザーはセンサー番号を論理ユニット番号 (LUN) で修飾できます。*lun* には、値 0、1、または 3 を指定できます。LUN 2 は予約されています。LUN を省略すると、もっとも小さい番号の LUN 上にある、指定されたセンサー番号を持つセンサーに対してコマ

ンドが適用されます。たとえば、コマンドで明示的に LUN で修飾せずにセンサー 3 を指定した場合、対象のコントローラは LUN 1 上のセンサー 3 と、LUN 3 上のもう 1 つのセンサー 3 を公開しますが、コマンドは LUN 1 上のセンサー 3 に適用されま

例:

IPM コントローラ FE 上の温度センサー Local Temp 上で、Lower Non-Critical Going Low イベントを有効にします。

```
# 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
#
```

同じセンサーに対して同じ処理を実行しますが、LUN とセンサー番号を使用してセンサーを指定します。

```
# clia setsensoreventenable fe 0:3 0x90 0x01 0x00
IPM Sentry Shelf Manager Command Line Interpreter
Event enable mask set successfully
#
```

---

## setthreshold

構文:

```
setthreshold IPMB-address sensor-name threshold-type [-r] value
setthreshold IPMB-address [lun:]sensor-number threshold-type [-r] value
```

IPMB-address の代わりに、次の方法でも参照できます。

```
board n
shm n
```

## 用途:

このコマンドは、指定されたセンサーの指定されたしきい値に関して、現在のしきい値を変更します。センサーは、それを所有する IPM コントローラの IPMB アドレスと、センサー名またはセンサー番号で指定します。対象のセンサーは、しきい値ベースのセンサーである必要があります。パラメータ *threshold-type* は、次のいずれかの記号値で指定できます。

- upper\_non\_recoverable (unr に省略可能)
- upper\_critical (uc に省略可能)
- upper\_non\_critical (unc に省略可能)
- lower\_non\_recoverable (lnr に省略可能)
- lower\_critical (lc に省略可能)
- lower\_non\_critical (lnc に省略可能)

デフォルトでは、処理された形式で目的の値を指定します。つまり、電圧センサーの場合はボルト、または温度センサーの場合は摂氏温度で指定します。オプション *-r* を指定する場合は、代わりに **raw** 値を使用します。通常は 1 バイトサイズの数量で、センサー固有の規則に従って変換されます。

対象のコントローラが複数の LUN 上のセンサーをサポートしている場合、ユーザーはセンサー番号を論理ユニット番号 (LUN) で修飾できます。*lun* には、値 0、1、または 3 を指定できます。LUN 2 は予約されています。LUN を省略すると、もっとも小さい番号の LUN 上にある、指定されたセンサー番号を持つセンサーに対してコマンドが適用されます。たとえば、コマンドで明示的に LUN で修飾せずにセンサー 3 を指定した場合、対象のコントローラは LUN 1 上のセンサー 3 と、LUN 3 上のもう 1 つのセンサー 3 を公開しますが、コマンドは LUN 1 上のセンサー 3 に適用されます。

## 例:

IPM コントローラ 9C 上の温度センサー *emulated temp* の非クリティカルなしきい値の上限を、摂氏 99 °C に設定します。

```
# 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

### 構文:

`shelf subcommand`

次のサブコマンドがサポートされています。

- `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:

コマンド `shelf` を使用すると、主要なシェルフ FRU 情報のほか、シェルフに関して選択した現在の動作データを表示し、シェルフ FRU 情報のいくつかのフィールドを変更することができます。このコマンドで表示または変更する情報の種類は、コマンドパラメータで指定します。

以降のサブセクションでは、`shelf` コマンドのさまざまな適用目的に対応する構文について説明します。

## シェルフ FRU 情報の表示

#### 構文:

```
shelf [cooling_state | fans_state | address_table
      | power_distribution | power_management
      | pci_connectivity | ha_connectivity
      | h110_connectivity | point-to-point_connectivity]
```

#### 用途:

コマンド `shelf` のこの構文では、主要なシェルフ FRU 情報のほか、シェルフに関して選択した現在の動作データが表示されます。このコマンドで表示する情報の種類は、コマンドパラメータで指定します。表 A-5 に、`shelf` コマンドでサポートされるパラメータの一覧を示します。

表 A-5 shelf コマンドのパラメータ

コマンドパラメータ	表示される情報
cooling_state (cs に省略可能)	<p>シェルフの現在の冷却状態。</p> <ul style="list-style-type: none"> <li>• <b>Normal</b> — すべての温度センサーが正常な動作温度を示しています。</li> <li>• <b>Minor Alert</b> — 1 つ以上の温度センサーがマイナーアラート状態です。いずれのセンサーもメジャーまたはクリティカルアラート状態ではありません。</li> <li>• <b>Major Alert</b> — 1 つ以上の温度センサーがメジャーアラート状態です。いずれのセンサーもクリティカルアラート状態ではありません。</li> <li>• <b>Critical Alert</b> — 1 つ以上の温度センサーがクリティカルアラート状態です。</li> </ul>
fans_state (fs に省略可能)	<p>シェルフ内のファンの回転速度計の現在の状態。</p> <ul style="list-style-type: none"> <li>• <b>Normal</b> — すべてのファンの回転速度計センサーは正常な動作速度を示しています。</li> <li>• <b>Minor Alert</b> — 1 つ以上のファンの回転速度計センサーがマイナーアラート状態です。いずれのセンサーもメジャーまたはクリティカルアラート状態ではありません。</li> <li>• <b>Major Alert</b> — 1 つ以上のファンの回転速度計センサーがメジャーアラート状態です。いずれのセンサーもクリティカルアラート状態ではありません。</li> <li>• <b>Critical Alert</b> — 1 つ以上のファンの回転速度計センサーがクリティカルアラート状態です。</li> </ul>
address_table (at に省略可能)	<p>シェルフ FRU 情報のアドレステーブルレコード。次の情報が表示されます。</p> <ul style="list-style-type: none"> <li>• シェルフのアドレス (タイプ別に表示)</li> <li>• アドレステーブルのエントリと、各エントリのハードウェアアドレス、設置場所の番号、および設置場所のタイプの一覧</li> </ul>
power_distribution (pd に省略可能)	<p>各電源フィードに関する次に示す情報。大部分はシェルフ FRU 情報のシェルフ配電レコードから取得されます。</p> <ul style="list-style-type: none"> <li>• 最大外部使用可能電流</li> <li>• 最大内部電流</li> <li>• 最小予測動作電圧</li> <li>• 実際の使用可能電力</li> <li>• 現在の消費電力</li> <li>• 電源フィードに接続されている FRU と、各 FRU のハードウェアアドレスおよび FRU デバイス ID の一覧</li> </ul>

表 A-5 shelf コマンドのパラメータ (続き)

power_management (pm に省略可能)	<p>シェルフ FRU 情報のシェルフ電源管理レコード。このレコードには、FRU の電源記述子の一覧が含まれます。各記述子に対して、次の情報が表示されます。</p> <ul style="list-style-type: none"> <li>• ハードウェアアドレス</li> <li>• FRU デバイス ID</li> <li>• FRU 電源容量の最大値</li> <li>• シェルフマネージャー制御による起動</li> <li>• 次の電源投入までの遅延</li> </ul>
pci_connectivity (pcic に省略可能)	<p>シェルフ FRU 情報のシェルフ PCI 接続レコード。次の情報が表示されます。</p> <ul style="list-style-type: none"> <li>• PCI スロット記述子</li> <li>• DSEL 接続</li> <li>• セグメント ID</li> <li>• 拡張 PCI スロット記述子</li> <li>• 地理的なアドレス</li> <li>• インタフェース番号</li> <li>• システムスロット対応</li> </ul>
ha_connectivity (ha に省略可能)	<p>シェルフ FRU 情報のシェルフ HA 接続レコード。次の情報が表示されます。</p> <ul style="list-style-type: none"> <li>• 放射線状の接続サポート</li> </ul>
h110_connectivity (h110c に省略可能)	<p>シェルフ FRU 情報のシェルフ H110 接続レコード。次の情報が表示されます。</p> <ul style="list-style-type: none"> <li>• 地理的なアドレス</li> <li>• セグメント ID</li> </ul>
point-to-point_connectivity (ppc に省略可能)	<p>シェルフ FRU 情報のシェルフポイントツーポイント接続レコード。次の情報が表示されます。</p> <ul style="list-style-type: none"> <li>• チャネルタイプ</li> <li>• チャネル数</li> <li>• スロット/ハードウェアアドレス</li> <li>• チャネル記述子</li> </ul>

コマンドパラメータ `cooling_state` および `fans_state` では、詳細オプション `-v` が使用できます。このオプションは、`clia shelf -v cooling_state` のように、コマンドパラメータの前に指定する必要があります。コマンドでこのオプションを使用すると、現在の状態の情報源である温度センサーまたはファンの回転速度計センサーの一覧が表示されます。各センサーについて、*IPMB-address* およびセンサー番号がセットで表示されます。

例:

シェルフの冷却状態を取得します。

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

シェルフのファンの回転速度計の状態を取得します (詳細)。

```
# 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)
#
```

アドレステーブルを取得します。

```
# 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
#
```

配電情報を取得します。

```
# 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
#
```

電源管理情報を取得します。

```
# 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
#
```

## 最大外部使用可能電流の変更

構文:

```
shelf maxcurrent [feed] current
```

用途:

このコマンドは、指定された電源フィード番号の最大外部使用可能電流を設定し、シェルフ内で認識されるすべてのシェルフ FRU 情報のインスタンスを更新します。*feed* パラメータを省略すると、シェルフ FRU 情報の最初の電源フィードである電源フィード 0 に対して値が設定されます。

パラメータ *feed* には、電源フィード番号を指定します。これは、シェルフ FRU 情報に含まれる 0 から始まる番号で、電源フィードが記述されている順番に基づいています。

パラメータ *current* には、設定する現在の値をアンペアで指定します。

例:

電源フィード 0 に対する最大外部使用可能電流を 50 A から 99 A に変更します。

```
# 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
#
```

## 最小予測動作電圧の変更

### 構文:

```
shelf minvoltage [feed] voltage
```

### 用途:

このコマンドは、指定された電源フィード番号の最小予測動作電圧を設定し、シェルフ内で認識されるすべてのシェルフ FRU 情報のインスタンスを更新します。*feed* パラメータを省略すると、シェルフ FRU 情報の最初の電源フィード (電源フィード 0) に対して値が設定されます。

パラメータ *feed* には、電源フィード番号を指定します。これは、シェルフ FRU 情報に含まれる 0 から始まる番号で、電源フィードが記述されている順番に基づいています。

*voltage* には、設定する値を指定します。

### 例:

電源フィード 0 の最小予測動作電圧を変更します。

```
# 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
```

#

## シェルフマネージャー制御による起動フラグの変更

構文:

```
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:

このコマンドは、指定された IPM コントローラの指定された FRU に関して、Shelf Manager Controlled Activation (シェルフマネージャー制御による起動) フィールドを表示または変更します。このコマンドは、Shelf Activation and Power Management (シェルフの起動および電源管理) レコードの既存エントリでのみ、Shelf Manager Controlled Activation フラグを変更します。このコマンドは、シェルフマネージャーが使用する、キャッシュに格納されたシェルフ FRU 情報も更新します。そのため、Shelf Manager Controlled Activation フィールドの新しい値は、シェルフマネージャーを再起動しなくてもすぐに有効になります。

パラメータ *hardware-addr* には、7 ビットのハードウェアアドレスを 16 進表記で指定します。

パラメータ *fru-id* には、FRU ID を 16 進表記で指定します。0xFE は、指定されたハードウェアアドレスのすべての FRU を意味します。

例:

ハードウェアアドレス 0x42 (IPMB アドレス 0x84) の IPM コントローラ上で、シェルフマネージャー制御による起動機能を有効にします。

```
# 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
```

```
#
```

## シェルフマネージャー制御による停止フラグの変更

### 構文:

```
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:

このコマンドは、指定された IPM コントローラの指定された FRU に対して、Shelf Manager Controlled Deactivation (シェルフマネージャー制御による停止) フィールドを表示または変更します。このコマンドは、Shelf Activation and Power Management (シェルフの起動および電源管理) レコードの既存エントリに対してのみ、Shelf Manager Controlled Deactivation フラグを変更します。このコマンドは、シェルフマネージャーが使用する、キャッシュに格納されたシェルフ FRU 情報も更新します。そのため、Shelf Manager Controlled Deactivation フィールドの新しい値は、シェルフマネージャーを再起動しなくてもすぐに有効になります。

パラメータ *hardware-addr* には、7 ビットのハードウェアアドレスを 16 進表記で指定します。

パラメータ *fru-id* には、FRU ID を 16 進表記で指定します。0xFE は、指定されたハードウェアアドレスのすべての FRU を意味します。

値 0 を設定すると、シェルフマネージャー制御による停止機能が有効になり、ボードの停止をシェルフマネージャーが制御します。値 1 を設定すると、シェルフマネージャー制御による停止機能が無効になり、シェルフマネージャーが自動的にボードを停止しないようにできます。シェルフマネージャー制御による停止機能を無効にした場合、ボードの停止はシステム管理者が行います。

### 例:

ハードウェアアドレス 0x41 (IPMB アドレス 0x82) の IPM コントローラ上で、シェルフマネージャー制御による停止機能を有効にします。

```
# 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

#

```

## 最大 FRU 電源容量の変更

構文:

```
shelf pwrcapability hardware-addr fru-id value
shelf pwrcapability board n value
shelf pwrcapability fan_tray n value
```

用途:

このコマンドは、指定された IPM コントローラの指定された FRU に関して、Maximum FRU Power Capability (最大 FRU 電源容量) フィールドを変更します。

---

**注** – Maximum FRU Power Capability フィールドに設定する値は、シェルフ環境で安全な値を決して超えないようにしてください。

---

このコマンドは、Shelf Activation and Power Management (シェルフの起動および電源管理) レコードの既存エントリのフィールドのみを変更します。このコマンドは、シェルフマネージャーが使用する、キャッシュに格納されたシェルフ FRU 情報も更新します。そのため、Maximum FRU Power Capability フィールドの新しい値は、シェルフマネージャーを再起動しなくてもすぐに有効になります。

- パラメータ *hardware-addr* には、7 ビットのハードウェアアドレスを 16 進表記で指定します。
- パラメータ *fru-id* には、FRU ID を 16 進表記で指定します。0xFE は、指定されたハードウェアアドレスのすべての FRU を意味します。
- パラメータ *value* には、フィールドの新しい値を W 単位で指定します。指定できる値の範囲は、0 ~ 65535 です。

例:

ハードウェアアドレス 0x42 (IPMB アドレス 0x84) の IPM コントローラ上で、最大 FRU 電源容量を 150 W に設定します。

```
# 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

#

```

## 次回の電源投入までの遅延の変更

### 構文:

```

shelf pwrdelay hardware-addr fru-id value
shelf pwrdelay board n value
shelf pwrdelay fan_tray n value

```

### 用途:

このコマンドは、指定された IPM コントローラの指定された FRU に関して、Delay Before Next Power On (次回の電源投入までの遅延) フィールドを変更します。このコマンドは、Shelf Activation and Power Management (シェルフの起動および電源管理) レコードの、既存エントリのフィールドのみを変更します。このコマンドは、シェルフマネージャーが使用する、キャッシュに格納されたシェルフ FRU 情報も更新します。そのため、Delay Before Next Power On フィールドの新しい値は、シェルフマネージャーを再起動しなくてもすぐに有効になります。

- パラメータ *hardware-addr* には、7 ビットのハードウェアアドレスを 16 進表記で指定します。
- パラメータ *fru-id* には、FRU ID を 16 進表記で指定します。0xFE は、指定されたハードウェアアドレスのすべての FRU を意味します。
- パラメータ *value* には、フィールドの新しい値を 0.1 秒単位で指定します。指定できる値の範囲は、0 ～ 63 です。

例:

ハードウェアアドレス 0x42 (IPMB アドレス 0x84) の IPM コントローラ上で次の電源投入までの遅延を 5 秒に設定します。

```
# 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
#
```

## FRU 起動準備許容時間の変更

構文:

```
shelf allowance value
```

用途:

このコマンドは、Allowance for FRU Activation Readiness (FRU 起動準備許容時間) パラメータを変更します。

パラメータ *value* には、フィールドの新しい値を秒単位で指定します。指定できる値の範囲は、0 ～ 255 です。

例:

FRU 起動準備許容時間を 5 秒に設定します。

```
# 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
```

```
#
```

## FRU の起動記述子と電源記述子の順序変更

### 構文:

```
shelf pwrreorder hardware-addr-1 fru-id-1 before|after hardware-addr-2  
fru-id-2
```

*hardware-address fru-id* の代わりに、次の方法でも参照できます。

```
board n  
fan_tray n
```

### 用途:

このコマンドは、シェルフ FRU 情報内の FRU の起動記述子および電源記述子の順序を変更します。このコマンドでは、既存の記述子の順序のみを変更できます。また、現在の実装では、記述子の順序の変更は、単一の **Shelf Activation and Power Management** (シェルフの起動および電源管理) レコード内のみに制限されています。このコマンドは、シェルフマネージャーが使用する、キャッシュに格納されたシェルフ FRU 情報も更新します。そのため、記述子の新しい順序は、シェルフマネージャーを再起動しなくてもすぐに有効になります。

- パラメータ *hardware-addr-1* には、移動する必要がある記述子の 7 ビットのハードウェアアドレスを 16 進表記で指定します。
- パラメータ *fru-id-1* には、移動する必要がある記述子の FRU ID を 16 進表記で指定します。0xFE は、指定されたハードウェアアドレスのすべての FRU を意味します。
- パラメータ *hardware-addr-2* には、記述子の 7 ビットのハードウェアアドレスを 16 進表記で指定します。このパラメータの前後のいずれかに、*hardware-addr-1/fru-id-1* 記述子が配置されます。
- パラメータ *fru-id-2* には、記述子の FRU ID を 16 進表記で指定します。このパラメータの前後のいずれかに、*hardware-addr-1/fru-id-1* 記述子が配置されます。

例:

ハードウェアアドレス 0x42 (IPMB アドレス 0x84) の IPM コントローラの記述子を、ハードウェアアドレス 0x41 (IPMB アドレス 0x82) の IPM コントローラの記述子の前に配置します。

```
# 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
#
```

# シェルフ FRU 情報のリフレッシュ

構文:

```
shelf info_refresh
```

用途:

このコマンドを使用すると、シェルフマネージャーはシェルフ内で以前検出されたシェルフ FRU 情報のソースを再度読み取り、どのソースに有効なシェルフ FRU 情報が含まれているかを再評価します。有効なシェルフ FRU 情報が確認された場合は、すべてのシェルフ FRU 情報ストレージデバイスと、キャッシュに格納されたシェルフ FRU 情報のマスターコピーが、新しいシェルフ FRU 情報の内容で更新されます。

PICMG 3.0 のセクション 3.6.4 で指定されているように、シェルフマネージャーは初期化中に、存在する可能性のあるシェルフ FRU 情報ストレージデバイスの検出を試みます。有効なシェルフ FRU 情報を含む FRU 情報デバイスが 2 台以上検出された場合、シェルフマネージャーは election を実行して、使用するシェルフ FRU 情報のソースを決定します。この選択は、シェルフ FRU 情報ストレージデバイス内のデータの検証およびその内容の比較に基づいて行われます。選択が正常に行われると、シェルフマネージャーはキャッシュに格納されたシェルフ FRU 情報のマスターコピーを揮発性メモリー内に作成します。これは、シェルフ FRU 情報のソースが更新されるたびに使用され、シェルフ FRU 情報の唯一のソースとして扱われます。そのため、シェルフ FRU 情報に関連したすべての処理はマスターコピーで行われ、マスターコピーの変更は増分更新として自動的にすべてのシェルフ FRU 情報のソースデバイスに伝播されます。

ただし、動的再構成はサポートされていません。新しいシェルフ FRU 情報が以前のシェルフ FRU 情報と異なる場合、その変更内容はシェルフマネージャーを再起動したあとにのみ完全に有効になります。

例:

シェルフ FRU 情報の 2 つの一致するソースが存在する場合の正常なリフレッシュを示します。

```
# 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
```

```
#
```

次に、両方のデータソースに一致しないデータまたは無効なデータが含まれている場合のリフレッシュの失敗を示します。

```
# 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
```

```
#
```

# シェルフ FRU 情報ストレージデバイスの更新

構文:

```
shelf info_force_update
```

用途:

このコマンドは、シェルフ FRU 情報のソースデバイスを確認し、シェルフ FRU 情報のマスターコピーの内容を、それらすべてのデバイスにコピーします。このコマンドは、シェルフ FRU 情報のマスターコピーと非揮発性ソースデバイス間に不一致があり、その不一致が自動的に解決されない場合に役立ちます。たとえば、EEPROM とマスターコピーの両方が互いに一致しない場合などです。このような場合、オペレータはこのコマンドを使用して、EEPROM をマスターコピーの内容に強制的に同期させることができます。また、このコマンドは元の不一致により発生したエラー状態をクリアします。つまり、このコマンドを実行したあとは、シェルフ FRU 情報へのそれ以降の更新が、ふたたび EEPROM に伝達され始めます。

このコマンドは、シェルフ FRU 情報のソースデバイスの更新を非同期的に開始します。

例:

```
# clia shelf info_force_update

IPM Sentry Shelf Manager Command Line Interpreter

Starting the Shelf FRU Info source device update
#
```

---

## shelfaddress

構文:

```
shelfaddress [最大 30 文字のシェルフのアドレス]
```

用途:

このコマンドは、シェルフ FRU 情報内のアドレステーブルのシェルフアドレスフィールドを取得または設定します。このコマンドは 6 ビットの packets 値を使用するため、大文字と数字のみが許可されます。

小文字は自動的に大文字に変換されます。

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

Syntax:

shmstatus

用途:

このコマンドは、冗長構成のシェルフマネージャーに関して、アクティブまたはバックアップの状態を返します。詳細モードでは、より詳細な情報としてシェルフ FRU 情報の状態、RMCP インタフェースの状態、およびバックアップシェルフマネージャーの状態 (照会先のシェルフマネージャーがアクティブの場合) が報告されます。Ready For Operation (動作可能状態) フラグは、次の場合に Yes と表示されるパラメータです。

- 照会先のアクティブシェルフマネージャーが、有効なシェルフ FRU 情報を検出し、その RMCP インタフェースを正常に初期化した場合。
- 照会先のバックアップシェルフマネージャーが、アクティブシェルフマネージャーから冗長状態の情報を正常に受信した場合。

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

構文:

`showhost slot-number`

用途:

このコマンドは、Netra CP3060 ボードにインストールされたシステムファームウェアのバージョンを表示します。このコマンドは、Netra CT 900 サーバーに取り付けられた Netra CP3060 ボードのみに有効です。

パラメータ *slot-number* には、Netra CP3060 ボードのスロット番号を指定します。*version* オプションを指定すると、完全なバージョン情報が表示されます。

例:

スロット 2 の Netra CP3060 ボードの現在のファームウェアバージョンを表示します。

```
# 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

## 構文:

```
showunhealthy
```

## 用途:

このコマンドは、問題があると認識される FRU の一覧を表示します。PICMG 3.0 のコンテキストでは、前回のホットスワップの状態変更が発生した原因が Communication Lost、Communication lost due to local failure、Unexpected deactivation である FRU が、一覧に含まれます。

FRU ごとに、IPMB アドレスと FRU デバイス ID、現在のホットスワップ状態、前のホットスワップの状態、および前回の状態変更の原因が表示されます。

## 例:

システム内の異常なコンポーネントを表示します。

```
# clia showunhealthy
IPM Sentry Shelf Manager Command Line Interpreter
There are no unhealthy components in the shelf.
#
```

---

# switchover

## 構文:

```
switchover
```

## 用途:

このコマンドは、冗長シェルフマネージャーのインスタンスのスイッチオーバーを開始します。このコマンドは、シェルフマネージャーのアクティブインスタンスでも、バックアップインスタンスでも実行できます。

例:

アクティブまたはバックアップのいずれかのインスタンスから、スイッチオーバーを開始します。

```
# clia switchover
    This Shelf Manager is now active, but is shutting down to
    trigger a switchover.
#
```

---

## terminate

構文:

terminate

用途:

このコマンドは、シェルフ管理カードを再起動せずに シェルフマネージャー を終了します。現在の ShMM がアクティブの場合は、スイッチオーバーが行われます。

例:

アクティブまたはバックアップのいずれかのインスタンスから、シェルフマネージャーを終了します。

```
# clia terminate

#
```

---

## user

構文:

user [subcommand]

次のサブコマンドがサポートされています。

- add
- delete

- enable
- name
- passwd
- channel

#### 用途:

`user` コマンドは、シェルフマネージャー上の **RMCP** ユーザーアカウントに関する情報を表示し、そのユーザーアカウントを追加、削除、および変更する簡単な手段を提供します。

以降のサブセクションでは、`user` コマンドのさまざまな適用目的に対応する構文について説明します。

## ユーザー情報の表示

#### 構文:

```
user [-v] [userid]
```

#### 用途:

このコマンドは、ユーザーに関する情報を表示します。`-v` オプションを指定してこのコマンドを実行すると、無効なユーザーに関する情報も表示されます。デフォルトでは、有効なユーザーのみが一覧表示されます。オプションとしてユーザー ID を指定すると、指定 ID を持つユーザーに関する情報のみが表示されます。

次の項目に関する情報が表示されます。

- ユーザー ID
- ユーザー名
- IPMI チャンネルごとのチャンネルアクセス情報。そのチャンネル上での指定ユーザーの最大特権レベル、およびチャンネルアクセスフラグ

複数のチャンネルのチャンネルアクセス情報が同じ場合は、出力が統合されてチャンネル範囲が示されます。

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"
#
```

## 新しいユーザーの追加

構文:

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

用途:

このコマンドは、システムに新しいユーザーを追加します。このコマンドは、コマンドへの指定に従って、同じ最大特権レベルとチャンネルアクセスフラグをすべてのチャンネルに設定します。指定されたユーザーが存在しない場合、コマンドはエラーを返します。次にコマンドパラメータの意味を示します。

- *userid* — 有効なユーザー ID
- *user-name* — ユーザー名 (通知なしで 16 文字に切り詰められる)
- *channel-access-flag* — **SetUserInfo** コマンドの先頭バイト (意味を持つのはビット 4、5、および 6 のみ)
  - ビット 6 — IPMI メッセージングを使用可能
  - ビット 5 — リンク認証を使用可能
  - ビット 4 — コールバックに限定
- *privilege-level* — ユーザーの特権レベル
- *password* — ユーザーパスワード (通知なしで 16 文字に切り詰められる)

例:

ユーザー 9 を追加します。ユーザー名を root、特権レベルを管理者、パスワードを PICMG guru とします。

```
# 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"
#
```

## ユーザーの削除

構文:

```
user delete userid
```

用途:

このコマンドは、*userid* に指定されたユーザーを削除します。

例:

ユーザー ID が 10 のユーザーを削除します。

```
# 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"
#
```

## ユーザーの有効化および無効化

構文:

```
user enable userid 1 | 0
```

用途:

このコマンドは、*userid* に指定したユーザーを有効または無効にします。末尾のコマンドパラメータには、次の要求する処理を指定します。

- 0 – 指定されたユーザーを無効にする
- 0 以外 – 指定されたユーザーを有効にする

例:

*userid* が 9 であるユーザーを無効にしてから有効にします。

```
# 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"
#

```

## ユーザー名の変更

構文:

`user name userid user-name`

用途:

このコマンドは、指定されたユーザーのユーザー名を変更します。ユーザーは、ユーザー ID で指定します。次にコマンドパラメータの意味を示します。

- *userid* — 有効なユーザー ID
- *user-name* — ユーザー名 (通知なしで 16 文字に切り詰められる)

例:

ユーザー 9 の名前を newby に変更します。

```

# 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 name 9 newby
IPM Sentry Shelf Manager Command Line Interpreter
User 9, name 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"
#
```

## ユーザーのパスワードの変更

構文:

```
user passwd userid password
```

用途:

このコマンドは、指定されたユーザーのパスワードを変更します。ユーザーは、ユーザー ID で指定します。次にコマンドパラメータの意味を示します。

- *userid* — 有効なユーザー ID
- *password* — ユーザーパスワード (通知なしで 16 文字に切り詰められる)

例:

ユーザー 9 のパスワードを RIP に変更します。

```
# 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"
#

```

## 指定ユーザーおよび指定チャネルのチャネルアクセス設定の変更

構文:

```
user channel userid channel-number channel-access-flags privilege-level
```

用途:

このコマンドは、指定されたチャネルおよびユーザーのチャネルアクセス設定を変更します。ユーザーは、ユーザー ID で指定します。次にコマンドパラメータの意味を示します。

- *userid* — 有効なユーザー ID
- *channel-number* — チャネル番号
- *channel-access-flags* — SetUserInfo コマンドの先頭バイト (意味を持つのはビット 4、5、および 6 のみ)
  - ビット 6 — IPMI メッセージングを使用可能
  - ビット 5 — リンク認証を使用可能
  - ビット 4 — コールバックに限定
- *privilege-level* — ユーザーの特権レベル

例:

チャネル 5 のユーザー 9 の最大特権レベルを User に変更します。

```

# 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

### 構文:

```
version
```

### 用途:

このコマンドは、シェルフマネージャソフトウェアのバージョン情報を表示します。

### 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 の OEM IPMI コマンド

この付録では、Sun Microsystems が設計した ATCA ボードに固有のコマンドについて説明します。Sun Microsystems に割り当てられている IANA (Internet Assigned Numbers Authority) 番号は 42 です。

IANA 番号の割り当ての詳細は、  
<http://www.iana.org/assignments/enterprise-numbers> を参照してください。

これらのコマンドに使用されるネットワーク関数 (NetFn) は 0x2E です。これは、IPMI 仕様の定義に従った OEM ネットワーク関数です。このネットワーク関数では、要求パケットの先頭 3 データバイトはこの IANA 番号である必要があり、完了コードに続く応答パケットの先頭 3 バイトはこの IANA 番号になります。Sun の ATCA ノードボードの場合、この 3 バイトは 00 00 2A です。

表 B-1 に Sun の OEM IPMI コマンドの一覧を示し、以降のセクションで詳細に説明します。

表 B-1 Sun の OEM IPMI コマンド

コマンド	オペレー ション コード	構文
Get Version	0x80	#GET_VERSION
Set Boot Page <sup>†</sup>	0x81	#SET_BOOT_PAGE
Get Boot Page <sup>†</sup>	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

<sup>†</sup> Netra CP3010 ノードボードでのみ有効

# Get Version

Get Version は、IPM コントローラ (IPMC) のファームウェアのバージョンとスタンバイ CPLD のバージョンを返します。バイト 8、9、および A は、将来の使用のために予約されています。

コマンド	NetFn	オペレーション コード	参照
Get Version	0x2E (OEM)	0x80	--

データバイト

タイプ	バイト	データフィールド
要求データ	バイト 1	00
	バイト 2	00
	バイト 3	2A
応答データ	バイト 1	完了コード: 00 = OK C1 = コマンドがサポートされていません CC = 要求内に無効なデータがあります (その他の完了コードについては、IPMI 仕様を参照)
	バイト 2	00
	バイト 3	00
	バイト 4	2A
	バイト 5	CPLD のバージョン
	バイト 6	IPMC ファームウェアの REV1 バイト
	バイト 7	IPMC ファームウェアの REV2 バイト
	バイト 8	将来の使用のために予約されています (無視)。
	バイト 9	将来の使用のために予約されています (無視)。
	バイト A	将来の使用のために予約されています (無視)。

## 例 (端末モード):

```
[B8 00 80 00 00 2A] <-----要求
[BC 00 80 00 00 00 2A 02 02 00 00 00 00] <----応答
```

- IPMC のバージョンは、次のように読み取ります。

REV1 の下位ニブル. REV2 の上位ニブル. REV2 の下位ニブル  
前述の例では、IPMC のバージョンは 2.0.0 です

- CPLD のバージョンは、次のように読み取ります。

-> CPLD バージョンバイトの下位ニブル  
この例では、CPLD バージョンは 2 です

## Set Boot Page

Set Boot Page は、スタンバイ CPLD のブートページのビットを設定して、OpenBoot PROM のブートページを選択します。この機能は、ブートフラッシュの障害からの回復に使用できます。このコマンドは、Netra CP3010 ノードボードでのみ有効です。

コマンド	NetFn	オペレーション コード	参照
Set Boot Page	0x2E (OEM)	0x81	CPLD 仕様 1.0

### データバイト

タイプ	バイト	データフィールド
要求データ	バイト 1	00
	バイト 2	00
	バイト 3	2A
	バイト 4	ブートページの設定。 ビット 7 ~ 2 = 0 ビット 1 および 0 = ブートページの番号
応答データ	バイト 1	完了コード: 00 = OK C1 = コマンドがサポートされていません CC = 要求内に無効なデータがあります (その他の完了コードについては、IPMI 仕様を参照)
	バイト 2	00
	バイト 3	00
	バイト 4	2A

### 例 (端末モード):

```
[B8 00 81 00 00 2A 02] <-----要求
[BC 00 81 00 00 00 2A] <-----応答
```

# Get Boot Page

Get Boot Page は、選択された OpenBoot PROM ブートページの現在の設定と、ブートページを選択するハードウェアスイッチの現在の設定を返します。このコマンドは、Netra CP3010 ノードボードでのみ有効です。

コマンド	NetFn	オペレーション コード	参照
Get Boot Page	0x2E (OEM)	0x82	--

データバイト

タイプ	バイト	データフィールド
要求データ	バイト 1	00
	バイト 2	00
	バイト 3	2A
応答データ	バイト 1	完了コード: 00 = OK C1 = コマンドがサポートされていません CC = 要求内に無効なデータがあります (その他の完了コードについては、IPMI 仕様を参照)
	バイト 2	00
	バイト 3	00
	バイト 4	2A
	バイト 5	ブートページの設定 ビット 7 ~ 4 = 無視。0 になるはずです。 ビット 3、2 = ハードウェアスイッチの設定。 ビット 1、0 = 現在設定されているブートページ。

## 例 (端末モード):

```
[B8 00 82 00 00 2A] <-----要求
[BC 00 82 00 00 00 2A 02] <-----応答
```

# Set Front Panel Reset Button State

Set Front panel reset button state は、フロントパネルのリセットボタンが押されたときに CPLD によってフロントパネルのリセットが処理される方法を変更するために、ソフトウェアによって使用されます。CPLD 電源投入時のデフォルト値は、10 です。

コマンド	NetFn	オペレーション コード	参照
Set Front panel reset button state	0x2E (OEM)	0x83	CPLD 仕様 1.0

## データバイト

タイプ	バイト	データフィールド
要求データ	バイト 1	00
	バイト 2	00
	バイト 3	2A
	バイト 4	フロントパネルのリセットボタンの設定。 ビット 7 ~ 2 = 0 ビット 1 および 0 = フロントパネルのボタンの状態: 00 = IPMC をリセットし、CPU に POR を表明 01 = CPU に対する XIR 10 = CPU に対する POR 11 = フロントパネルのリセットボタンは無効
応答データ	バイト 1	完了コード: 00 = OK C1 = コマンドはサポートされていません CC = 要求内に無効なデータがあります (その他の完了コードについては、IPMI 仕様を参照)
	バイト 2	00
	バイト 3	00
	バイト 4	2A

## 例 (端末モード):

```
[B8 00 83 00 00 2A 02] <-----要求
[BC 00 83 00 00 00 2A] <-----応答
```

# Get Front Panel Reset Button State

Get Front panel reset button state は、フロントパネルのリセットボタンの処理に関する現在の設定を返します。デフォルトでは、CPLD 電源投入時の値は 10 です。つまり、このボタンを押すと CPU に対して電源投入リセットが発生します。

コマンド	NetFn	オペレーション コード	参照 コード
Get Front panel reset button state	0x2E (OEM)	0x84	CPLD 仕様 1.0

データバイト

タイプ	バイト	データフィールド
要求データ	バイト 1	00
	バイト 2	00
	バイト 3	2A
応答データ	バイト 1	完了コード: 00 = OK C1 = コマンドはサポートされていません CC = 要求内に無効なデータがあります (その他の完了コードについては、IPMI 仕様を参照)
	バイト 2	00
	バイト 3	00
	バイト 4	2A
	バイト 5	フロントパネルのリセットボタンの設定。 ビット 7 ~ 2 = 0。 ビット 1 および 0 = フロントパネルのボタンの状態: 00 = IPMC をリセットし、CPU に POR を表明 01 = CPU に対する XIR 10 = CPU に対する POR 11 = フロントパネルのリセットボタンは無効

## 例 (端末モード):

```
[B8 00 84 00 00 2A] <-----要求
[BC 00 84 00 00 00 2A 02] <-----応答
```

## Set Ethernet Force Front Bit

Set Ethernet Force Front Bit は、背面切り替えモジュールがシステムに取り付けられている場合でも、強制的にフロント I/O パネルへの Ethernet 接続を行うために、ソフトウェアによって使用されます。このビットを 1 に設定すると、強制的にフロント I/O パネルへの Ethernet 接続が行われます。

コマンド	NetFn	オペレーション コード	参照
Set Ethernet Force Front Bit	0x2E (OEM)	0x85	CPLD 仕様 1.0

### データバイト

タイプ	バイト	データフィールド
要求データ	バイト 1	00
	バイト 2	00
	バイト 3	2A
	バイト 4	フロントへの Ethernet 接続強制ビットの設定 ビット 7 ~ 1 = 0 ビット 0 = フロントへの強制の状態 (1 = フロントへの強制的な Ethernet 接続)
応答データ	バイト 1	完了コード: 00 = OK C1 = コマンドがサポートされていません CC = 要求内に無効なデータがあります (その他の完了コードについては、IPMI 仕様を参照)
	バイト 2	00
	バイト 3	00
	バイト 4	2A

### 例 (端末モード):

```
[B8 00 85 00 00 2A 01] <-----要求  
[BC 00 85 00 00 00 2A] <-----応答
```

# Get Ethernet Force Front Bit

Get Ethernet Force Front Bit は、フロントへの Ethernet 接続強制ビットの現在の設定を返します。

コマンド	NetFn	オペレーション コード	参照
Get Ethernet Force Front Bit	0x2E (OEM)	0x86	CPLD 仕様 1.0

## データバイト

タイプ	バイト	データフィールド
要求データ	バイト 1	00
	バイト 2	00
	バイト 3	2A
応答データ	バイト 1	完了コード: 00 = OK C1 = コマンドがサポートされていません CC = 要求内に無効なデータがあります (その他の完了コードについては、IPMI 仕様を参照)
	バイト 2	00
	バイト 3	00
	バイト 4	2A
	バイト 5	フロントへの Ethernet 接続強制ビットの設定 ビット 7 ~ 1 = 0 ビット 0 = フロントへの Ethernet 接続強制ビットの設定 (1 = フロントへの強制的な Ethernet 接続)

## 例 (端末モード):

[B8 00 86 00 00 2A] <-----要求  
[BC 00 86 00 00 00 2A 01] <-----応答

# Get RTM Status

Get RTM Status は、システム内の背面切り替えモジュール (RTM) の有無を検出するために使用できます。

コマンド	NetFn	オペレーション コード	参照
Get RTM Status	0x2E (OEM)	0x88	CPLD 仕様

## データバイト

タイプ	バイト	データフィールド
要求データ	バイト 1	00
	バイト 2	00
	バイト 3	2A
応答データ	バイト 1	完了コード: 00 = OK C1 = コマンドがサポートされていません CC = 要求内に無効なデータがあります
	バイト 2	00
	バイト 3	00
	バイト 4	2A
	バイト 5	RTM の有無 ビット 7 ~ 1 = 0 ビット 0 = RTM の有無 (0 = RTM が検出されていない、 1 = RTM が検出された)

## 例 (端末モード):

[B8 00 88 00 00 2A] <-----要求  
[BC 00 88 00 00 00 2A 01] <-----応答

# 用語集

---

次の用語および頭字語の知識は、Netra CT 900 サーバーの管理に役立ちます。

---

## A

**ATCA** Advanced Telecom Computing Architecture の略語で、AdvancedTCA と呼ばれます。キャリアグレードの次世代通信機器のための一連の業界標準仕様です。AdvancedTCA は、高速インターコネクトテクノロジー、次世代プロセッサ、および高度な信頼性/管理容易性/保守性における最新の傾向を取り入れて、通信用に最適化された新しいブレード (ボード) とシャーシ (シェルフ) フォームファクタを、標準化することで低コストで実現します。

---

## E

**ETSI** ヨーロッパ電気通信標準化協会 (European Telecommunications Standards Institute)。

---

## I

**I<sup>2</sup>C** Inter-Integrated Circuit Bus の略語で、現在の IPMB の基礎として使用されるマルチマスターの 2 線式シリアルバス。

**IPM コントローラ (IPMC)** ATCA IPMB-0 へのインタフェースになる FRU の一部で、その FRU および FRU に従属するデバイスを表します。

- IPMB** Intelligent Platform Management Bus の略語で、Intelligent Platform Management Bus Communications Protocol 仕様に記載されているように、もっとも低いレベルのハードウェア管理バス。
- IPMB-0 ハブ** システム内の各種 FRU に複数の放射線状の IPMB-0 リンクを提供するハブデバイス。たとえば、IPMB-0 ハブは、放射線状の IPMB-0 リンクを持つ ShMC 内にあります。
- IPMB-0 リンク** 放射線状トポロジでの、IPMB-0 ハブと 1 つの FRU との間の物理的な IPMB-0 セグメント。IPMB-0 ハブ上の各 IPMB-0 リンクは、通常、個別の IPMB-0 センサーに関連付けられています。IPMB-0 リンクは、複数の FRU にバス型のトポロジで接続することもできます。
- IPMI** Intelligent Platform Management Interface の略語で、コンピュータシステムの各要素に対してインベントリ管理、監視、ロギング、および制御を行うための仕様および機構。Intelligent Platform Management Interface 仕様に定義されています。

---

## N

- NEBS** Network Equipment/Building System の略語で、アメリカ合衆国内の電気通信制御施設に設置されている装置に関する一連の要件。これらの要件は、人員の安全、資産の保護、および操作の継続性を対象としています。NEBS の試験には、さまざまな振動負荷、火災、およびその他の環境と品質に関する測定基準によって装置に影響を与える試験が含まれています。NEBS コンプライアンスには 3 つのレベルがあり、それぞれ前のレベルを包含します。もっとも高いレベルの NEBS レベル 3 では、「極限的な環境」に装置を安全に配置できることを保証しています。電気通信の中央局は、極限的な環境と考えられます。
- NEBS 規格は、Telcordia Technologies, Inc. (以前の Bellcore) によって管理されています。

---

## P

- PCI** Peripheral Component Interconnect の略語で、周辺装置をコンピュータに接続するための規格。これは 20 ~ 33 MHz で動作し、124 ピンコネクタでは一度に 32 ビットを、または 188 ピンコネクタでは 64 ビットを伝送します。アドレスは 1 サイクルで送信され、そのあとに 1 ワード (バーストモードでは複数ワード) のデータが続きます。

技術的には、PCI はバスではなく、ブリッジまたはメザニンです。CPU と比較的低速の周辺装置とを分離するためのバッファが含まれており、これらを非同期に処理することができます。

PICMG PCI Industrial Computer Manufacturers Group の略語で、CompactPCI 規格など、電気通信用および工業用コンピュータアプリケーションのオープン仕様を開発する企業コンソーシアム。

---

## S

ShMC Shelf Management Controller の略語で、シェルフマネージャーに要求される機能もサポートできる IPMC。

SNMP Simple Network Management Protocol の略語。

---

## U

U 44.45 mm (1.75 インチ) に相当する測定単位。

---

## あ

### アップデートチャンネル インタフェース

アップデートチャンネルとも呼ばれます。10 組の差動信号で構成される接続を 2 つのボード間に提供するゾーン 2 インタフェースです。この 2 つのボード間の直接接続は、状態情報の同期をとるためにも使用できます。ボード上のアップデートチャンネルで実現されるトランスポートは、定義されていません。アップデートチャンネルは、同一ベンダーが製造した同等の機能を持つ 2 つのボードでのみ使用できます。電子キーイングは、アップデートチャンネルの終端のプロトコルが、ドライバを有効にする前に割り当てられたトランスポートプロトコルと一致していることを保証するために使用されます。ミッドプレーンは、アップデートチャンネルをサポートしている必要があります。ボードは、アップデートチャンネルをサポートできます。

---

# け

## 現場交換可能ユニット (FRU)

保守の観点から見て、それ以上分解できないサーバー要素。FRU の例としては、ディスクドライブ、I/O カード、電源入力モジュールがあります。カードやその他のコンポーネントをすべて搭載したサーバーは FRU ではありません。ただし、空のサーバーは FRU です。

---

# し

## シェルフ

ミッドプレーン、フロントボード、冷却デバイス、背面切り替えモジュール、および電源入力モジュールで構成されるコンポーネントの集合。シェルフは、従来はシャーシと呼ばれていました。

## シェルフアース

フレームに接続される安全アースおよびアースリターンで、すべてのボードで使用できます。

## シェルフアドレス

管理ドメイン内の各シェルフに対して一意の識別子を提供する、最大 20 バイトの可変長、可変書式の記述子。

## シェルフマネージャー

AdvancedTCA シェルフ内の電力、冷却、およびインターコネクト (電子キーイングを使用) を管理する役割を持つシステムの構成要素。また、シェルフマネージャーは、システムマネージャーインタフェースと IPMB-0 間のメッセージの経路指定、システムリポジトリへのインタフェースの提供、およびイベントメッセージに対する応答も行います。シェルフマネージャーは、ShMC またはシステムマネージャーハードウェアの一部分または全体に配置できます。

## システム

ノードやスイッチ、シェルフ、およびフレームなどのコンポーネントを 1 つ以上含めることができる管理対象。

## 信頼性、可用性、保守性 (RAS)

サーバーの信頼性、可用性、および保守性を実現または改善するためのハードウェアおよびソフトウェア機能。

---

## す

- スイッチ** ミッドプレーン内の多くのノードボードへの接続を提供する、スタートポロジミッドプレーンでの使用を目的としたボード。スイッチは、ベースインタフェースおよびファブリックインタフェースのいずれかまたは両方をサポートできます。ファブリックインタフェースを利用するボードは、通常、使用可能な 15 のすべてのファブリックチャネルに対してスイッチング用リソースを提供します。ベースインタフェースをサポートするスイッチは、論理スロット 1 と 2 に取り付けられ、16 のすべてのベースチャネルを使用して、最大 14 のノードボードともう 1 つのスイッチに 10/100/1000BASE-T Ethernet のスイッチング用リソースを提供します。1 つのベースチャネルは、シェルフ管理カードへの接続のサポートに割り当てられます。
- スイッチスロット** スタートポロジミッドプレーンでは、スイッチスロットは論理スロット 1 および 2 に存在する必要があります。スイッチスロットは、ベースインタフェースおよびファブリックインタフェースの両方をサポートします。論理スロット 1 と 2 にあるスイッチスロットは、ベースインタフェースおよびファブリックインタフェースの両方のスイッチをサポートすることができます。論理スロット 1 および 2 は、ファブリックトポロジに関係なく、常にスイッチスロットです。これらのスロットは、最大 16 のベースチャネルと最大 15 のファブリックチャネルをそれぞれサポートします。
- スタートポロジ** サポートされるノードスロット間を接続する、1 つ以上のハブスロットを備えたミッドプレーントポロジ。

---

## そ

- ゾーン 1** 電力、管理、およびその他の補助的な機能に割り当てられる、ATCA スロットの高さの相に沿った線形の領域。
- ゾーン 2** データトランスポートインタフェースに割り当てられる、ATCA スロットの高さの相に沿った線形の領域。
- ゾーン 3** ユーザー定義の接続用、または背面操作システムでの背面切り替えモジュールへのインターコネクト用、あるいはその両方に予約された、ATCA スロットの高さの相に沿った線形の領域。

---

## て

### データトランスポートインタフェース

スイッチおよびノードボード上のペイロード間のインターコネクトの提供を目的とする、ポイントツーポイントインタフェースおよびバス接続された信号の集合。

### デュアルスタートポロジ

2つのスイッチリソースがネットワーク内のすべての終端に冗長接続を提供する、インターコネクトファブリックポロジ。1組のスイッチは、ノードボード間の冗長インターコネクトを提供します。

### 電子キーイングまたはE-キーイング

ベースインタフェース、ファブリックインタフェース、アップデートチャンネルインタフェース、およびフロントボードの同期クロック接続の間の互換性を表現するために使用されるプロトコル。

---

## の

### ノードスロット

ノードボードのみをサポートするミッドプレーン内のスロット。ノードスロットはスイッチをサポートできないため、ノードボードが論理スロット1および2を使用することはありません。ノードスロットは、スタートポロジをサポートするように設計されたミッドプレーンにのみ適用されます。ノードスロットは、ベースインタフェースおよびファブリックインタフェースの両方をサポートします。通常、ノードスロットは、2つまたは4つのファブリックチャンネルと、ベースチャンネル1および2をサポートします。2つのチャンネルノードスロットは、それぞれ論理スロット1および2への接続を個々に確立します。4つのチャンネルノードスロットは、論理スロット1、2、3、および4への接続を個々に確立します。

### ノードボード

ミッドプレーン内のスイッチへの接続性のある、スタートポロジミッドプレーンでの使用を目的としたボード。ノードボードは、ベースインタフェースおよびファブリックインタフェースのいずれかまたは両方をサポートすることができます。ファブリックインタフェースをサポートするボードは、ファブリックチャンネル1および2を使用します。ベースインタフェースをサポートするボードは、ベースチャンネル1および2のみを使用して、10/100/1000BASE-T Ethernet をサポートします。

---

## は

### 背面切り替えモジュール

Netra CT 900 サーバーの背面操作モデルでのみ使用されるカードで、シェルフの背面までコネクタを拡張します。

### 背面操作

すべてのケーブルがシェルフの背面から出てくるようにするための、Netra CT 900 サーバーの構成オプション。

### バックアップシェルフ管理カード

シェルフマネージャー機能のサポートを引き継ぐことができる任意のシェルフ管理カード。

---

## ふ

### ファブリックインタフェース

ボードまたはスロットごとに 15 の接続を提供するゾーン 2 インタフェースで、最大 8 組の差動信号 (チャンネル) で構成され、最大 15 のほかのスロットまたはボードとの接続をサポートします。ミッドプレーンは、フルメッシュトポロジ、デュアルスタートポロジなど、さまざまな構成のファブリックインタフェースをサポートできます。ファブリックインタフェースをサポートするボードは、ファブリックノードボード、ファブリックスイッチ、またはメッシュ対応ボードとして構成できます。ファブリックインタフェースのボード実装は、PICMG 3.x 補足仕様に定義されています。

### ファブリックチャンネル

ファブリックチャンネルは 2 列の信号の組で構成され、チャンネルあたり合計 8 組の信号に対応します。このようにして、各コネクタは、ボード間の接続に使用できる最大 5 つのチャンネルをサポートします。また、1 つのチャンネルは、4 つの 2 ペアポートで構成されるととらえられる場合もあります。

### 物理アドレス

FRU の物理スロットの位置を定義するアドレス。物理アドレスは、設置場所のタイプと設置場所の番号で構成されます。

### フルチャンネル

終端間で 8 組の差動信号をすべて使用するファブリックチャンネル接続。

### フルメッシュトポロジ

ファブリックインタフェース内でサポートできるフルメッシュ型の構成で、シェルフ内の各スロットペア間の接続に 1 つの専用チャンネルを提供します。フルメッシュ構成のミッドプレーンは、デュアルスター型の配置で取り付けられた、メッシュ対応のボードまたはスイッチ、およびノードボードをサポートできます。

### フレーム

1 台以上のシェルフを収納できる物理的または論理的な実体。ラックまたはキャビネット (密閉型の場合) と呼ばれることもあります。

**フロントボード**    PCB およびパネルを含む、PICMG 3.0 機械仕様 (8 U × 280 mm) に準拠するボード。フロントボードは、ゾーン 1 とゾーン 2 のミッドプレーンコネクタと接続します。任意で、ゾーン 3 のミッドプレーンコネクタに接続するか、または背面切り替えモジュールコネクタに直接接続することができ、シェルフの前面部分に取り付けられます。

---

## へ

**ベースインタフェース**    シェルフ内のノードボードとスイッチ間の 10/100 または 1000BASE-T 接続をサポートするために使用されるインタフェース。すべてのノードボードスロットと各スイッチスロット間で 4 組の差動信号を経路指定してベースインタフェースをサポートする場合には、ミッドプレーンが必要になります。Netra CT 900 サーバーでは、ベーススイッチのスロットは物理スロット 7 および 8 (論理スロット 1 および 2) です。

**ベーススイッチ**    ベースインタフェースをサポートするスイッチ。ベーススイッチは、シェルフに取り付けられているすべてのノードボードに 10/100/1000BASE-T のパケット交換サービスを提供します。Netra CT 900 サーバーでは、ベーススイッチはシェルフ内の物理スロット 7 および 8 (論理スロット 1 および 2) にあり、すべてのノードスロットおよびボードへの接続をサポートします。ファブリックインタフェースおよびベースインタフェースをサポートするボードは、「スイッチ」とも呼ばれます。

**ベースチャネル**    最大 4 組の差動信号で構成されるベースインタフェース内の物理接続。各ベースチャネルは、ベースインタフェース内のスロット間接続の終端です。

---

## ほ

**ホットスワップ**    システムの処理を中断せずに、周辺装置またはその他のコンポーネントの接続および切り離しを行うこと。この機能は、設計上、ハードウェアおよびソフトウェアの両方に関わる場合があります。

---

## み

**ミッドプレーン**    機能的には、バックプレーンと同等です。ミッドプレーンは、サーバーの背面に固定されています。CPU カード、I/O カード、およびストレージデバイスは前面からミッドプレーンに接続し、背面切り替えモジュールは背面からミッドプレーンに接続します。

---

## め

### メッシュ対応ボード

ミッドプレーン内のほかのすべてのボードへの接続を提供するボード。メッシュ対応ボードはファブリックインタフェースをサポートし、ベースインタフェースもサポートできます。メッシュ対応ボードは、2 ～ 15 のファブリックインタフェースチャネル (通常は 15 チャネルすべて) を使用して、シェルフ内のほかのすべてのボードへの直接接続をサポートします。サポートされるチャネルの数によって、シェルフ内に接続できるボードの最大数が決まります。ベースインタフェースを使用しないメッシュ対応ボードは、もともと下の使用可能な論理スロットに取り付けることができます。ベースインタフェースをサポートするメッシュ対応ボードは、ベーススイッチになることができます。この場合、このボードはベースチャネル 1 および 2 をサポートでき、論理スロット 3 ～ 16 に取り付けることができます。ベースインタフェースをサポートするボードは、ベースチャネル 1 および 2 のみを使用して、10/100/1000BASE-T Ethernet をサポートします。

---

## ろ

### 論理アース

ボード間で伝送される論理レベル信号の参照パスと戻りパスとして、ボード上およびミッドプレーン上で使用されるシェルフ全体の電気ネットワーク。



# 索引

---

## A

ATCA (Advanced Telecommunications Computing Architecture), 1  
ATCA シェルフ, 6

## C

CPLD  
cpldtool ユーティリティー, 88  
再プログラミング, 110

## E

Ethernet ポート, 3, 23  
E-キーイング, 123

## F

FRU 情報, 71, 73, 75, 76  
IPMI, 72  
Sun, 73

## I

IANA (Internet Assigned Numbers Authority), 263  
IP アドレス, 23  
RMCP, 24

## IPMB

スロット番号, 120  
論理スロット番号, 120

IPMB アドレス, 120, 122  
範囲, 120

IPMC, 2

## IPMI

FRU 情報, 72  
FRU 情報の配置, 72  
LAN インタフェース, 12, 59  
概要, 6

IPMI コマンド, 60

## O

OpenBoot PROM ファームウェア, 2  
OpenHPI, 52  
/etc/openhpi.conf, 52  
libipmddirect パラメータ, 53  
構成, 52

## P

POST, 2

## R

RMCP, 12, 23

アドレス, 24

RMCP (Remote Management Control Protocol), 59  
rupgrade\_tool, 93, 95, 97

## S

SAP (シェルフアラームパネル), 3

Telco アラーム, 3

アクセス, 3

コネクタ, 16

シリアルポート, 3

SNMP, 52

/etc/snmpd.conf ファイルの更新, 57

アクセス制御, 54

トラップの設定, 57

SNMP サブエージェント, 53

SNMP サブエージェント構成ファイル, 52

SNMPv3 構成, 56

Solaris オペレーティングシステム, 2

Sun の FRU 情報, 73

Sun の OEM IPMI コマンド, 60, 263

## T

Telco アラーム, 8

telnet, 15

tip プログラム, 15

## U

U-Boot, 2, 17, 24

アクセス, 18

環境変数, 18, 22

デフォルトの復元, 86

変数, 24

USB

インタフェース, 10

接続, 25

## W

WDT (ウォッチドッグタイマー), 96, 97, 98

## か

拡張インタフェース, 4

環境 FRU, 73

管理作業, 13

## き

強制的スイッチオーバー, 10

協調的スイッチオーバー, 10

## く

クロックの設定, 48

## け

現場交換可能ユニット (FRU), 3

## こ

コネクタ

シェルフアラームパネル, 16

コマンド

clia shmstatus, 17, 112

console, 112, 113

showcpustate, 67

switchover, 10

useradd, 50

コマンド行インタフェース, 2

コンソール, 15, 111 ~ 113

コンソールセッション, 111, 129

## さ

再初期化, 11

## し

- シェルフ管理カード, 2, 6, 8
  - Ethernet ポート, 23
  - アクセス, 15
  - アクティブ, 16
  - コンソール, 15, 111 ~ 113
  - セッション, 3
  - 説明, 2, 13
  - バックアップ, 16
  - ユーザーアカウント, 50
  - リセット, 86
  - ログイン, 16
- シェルフ情報, 80
- シェルフマネージャー, 2, 6, 8
  - CLI コマンド, 117, 117 ~ 261
  - CLI コマンドの概要, 62
  - CLI の開始, 61
  - IP アドレス, 15
  - shelfman.conf ファイル, 31
  - インタフェースオプション, 12
  - 概要, 6
  - 機能, 8
  - 構成パラメータ, 32
  - 構成ファイル, 25
  - コマンド行インタフェース, 60
  - 再プログラミング, 88
  - 詳細レベル, 47
  - スイッチオーバー, 9
  - ソフトウェア, 2
  - デバッグレベル, 130
  - ネットワークパラメータ, 26
  - バックアップ, 11
  - コマンド行インタフェース (CLI), 2
- シェルフマネージャーの再プログラミング, 88
- 時刻サーバー, 49
- システムマネージャー, 7
- シリアルポート, 15
- 信号
  - Remote Healthy, 11
  - Remote Presence, 11
  - スイッチオーバー, 10
- 信頼できるアップグレード, 89

状況, 98

状態ファイル, 92

フラッシュのパーティション分割, 90

ユーティリティ, 92

例, 100

信頼できるアップグレード処理, 89, 92

信頼できるアップグレードユーティリティ, 98

## す

スイッチオーバー, 10, 12

強制的, 10

協調的, 10

スイッチカード, 111, 129

スイッチボード

Ethernet ポート, 3

アクセス, 3

スイッチングファブリックボード

説明, 2

スクリプト

/etc/upgrade/step4hshm, 96

## せ

制御された停止, 115, 116

正常な停止, 114

## て

停止

手動による正常な停止, 114

ノードボード, 114

## ね

ネットコンソール, 114

ネットワークインタフェース, 4

ネットワーク関数 (NetFn), 263

## の

- ノードボード
  - Ethernet ポート, 3
  - SAS ポート, 3
  - アクセス, 3
  - コンソール, 111
  - シリアルポート, 3
  - 説明, 3
  - 停止, 114
  - 「ノードボード、他社製」も参照
- ノードボード、他社製, 4
- 説明, 3

## は

- ハードウェアインタフェース, 4, 5
- 背面切り替えモジュール, 3
- パスワード, 51

## ふ

- ファームウェア
  - OpenBoot PROM, 2
- ファイルシステム
  - 再初期化, 87
- 物理スロットから論理スロットへのマッピング, 13
- フラッシュのパーティション, 90
- フラッシュメモリー, 89

## へ

- ベースインタフェース, 4

## ま

- マッピング
  - 物理アドレス, 122
- マニュアル, xix

## ゆ

- ユーザーアカウント, 13
- ユーティリティー
  - 信頼できるアップグレード, 92

## ろ

- ログイン
  - 出荷時のデフォルト設定, 88
  - パスワードのリセット, 88
- ログイン、シェルフ管理カード, 16