

## Sun Netra CT900 サーバー

管理およびリファレンスマニュアル



Part No.: E22014-01  
2011 年 3 月

Copyright © 2006, 2009, 2010, 2011, Oracle and/or its affiliates. All rights reserved.

このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複写、複製、翻訳、放送、修正、ライセンス供与、送信、配布、発表、実行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アSEMBル、逆コンパイルは互換性のために法律によって規定されている場合を除き、禁止されています。

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オラクル社までご連絡ください。

このソフトウェアまたは関連ドキュメントを、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供する場合は、次の通知が適用されます。

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

このソフトウェアもしくはハードウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアもしくはハードウェアは、危険が伴うアプリケーション(人的傷害を発生させる可能性があるアプリケーションを含む)への用途を目的として開発されていません。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用する場合、安全に使用するために、適切な安全装置、バックアップ、冗長性(redundancy)、その他の対策を講じることは使用者の責任となります。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用したこと起因して損害が発生しても、オラクル社およびその関連会社は一切の責任を負いかねます。

Oracle と Java は Oracle Corporation およびその関連企業の登録商標です。その他の名称は、それぞれの所有者の商標または登録商標です。

AMD, Opteron, AMD ロゴ, AMD Opteron ロゴは、Advanced Micro Devices, Inc. の商標または登録商標です。Intel, Intel Xeon は、Intel Corporation の商標または登録商標です。すべての SPARC の商標はライセンスをもとに使用し、SPARC International, Inc. の商標または登録商標です。UNIX は X/Open Company, Ltd. からライセンスされている登録商標です。

このソフトウェアまたはハードウェア、そしてドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても一切の責任を負いかねます。



リサイクル  
してください



Adobe PostScript

# 目次

---

はじめに xxxv

## 1. 概要 1

Sun Netra CT900 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. システムの設定 15

シェルフ管理カードへのアクセス 16

U-Boot の設定 17

U-Boot インタフェース 18

U-Boot 環境変数 18

環境変数値の表示 21

環境変数への値の割り当て	22
シェルフマネージャーのデフォルトの環境変数	22
U-Boot 変数の変更に関する推奨事項	24
シェルフ管理カードの Ethernet ポートの構成	25
一次 Ethernet インタフェースの使用	26
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	26
RMCP アドレスの伝播	28
二次 Ethernet インタフェースの使用	28
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	28
デフォルトの ShMM ネットワークパラメータの変更	30
▼ デフォルトの ShMM ネットワークパラメータを変更する	30
上部の ShMM と下部の ShMM への異なる IP アドレスの割り当て	33
シェルフマネージャーの構成ファイルの設定	35
キャリア固有の構成ファイル	62
詳細レベルの説明	63
日付と時刻の設定	64
タイムサーバーからの日付と時刻の取得	65
夏時間のタイムゾーンの設定	66
▼ 夏時間のタイムゾーンを設定する	66
シェルフ管理カードでのユーザーアカウントの設定	66
▼ RMCP アクセスのためのユーザーアカウントを追加する	67
ユーザー名の制限	68
パスワード	68
リモート root ログインの無効化	68
シェルフマネージャーでの OpenHPI の構成	70
/etc/openhpi.conf ファイル	70
▼ /etc/openhpi.conf ファイルを変更する	71

/etc/snmpd.conf ファイル	71
アクセス制御	71
SNMPv3 の構成	74
トラップの設定と宛先の通知	75
▼ /etc/snmpd.conf ファイルを更新する	75
OpenHPI デーモンモニターの有効化	75
シェルフマネージャーでの RADIUS の構成	76
3. システムの管理	79
IPMI LAN インタフェース	80
サポートされている IPMI コマンド	80
「自己診断テスト結果の取得」 IPMI コマンド	87
IPMI の OEM 拡張コマンド	87
シェルフマネージャーのコマンド行インタフェース	88
コマンド行インタフェースの開始	88
シェルフマネージャーの CLI コマンド	89
システムの監視	96
ボードおよび IPMC の情報の表示	96
FRU 情報の表示	104
IPMI FRU 情報の配置	104
環境 FRU	105
ブレード FRU	106
例	106
セルフ情報の表示	114
例	114
シェルフマネージャーの再初期化	120
U-Boot 環境の再初期化	120
▼ U-Boot 環境を再初期化する	121
ファイルシステムの再初期化	122
ログインパスワードのリセット	122

シェルフ管理カードの再プログラミング	123
信頼できるファームウェアアップグレード処理	123
フラッシュのパーティション分割	124
/var/upgrade ファイルシステム	127
信頼できるアップグレード処理の状態ファイル	127
信頼できるアップグレードユーティリティ	128
信頼できるアップグレードユーティリティの使用シナリオ	134
信頼できるアップグレードの例	135
ノードボードコンソールへの接続	146
シェルフマネージャーとノードボード間でのコンソールセッションの確立	146
▼ 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	147
▼ コンソールセッションを終了する	148
コンソールのエスケープ文字の変更	148
コンソールセッション中のボードのユーザーラベルの表示	148
手動によるノードボードの正常な停止	149
▼ ノードボードを停止する	150
A. シェルフマネージャーの CLI コマンド	153
activate	154
構文:	154
目的:	154
例:	154
alarm	155
構文:	155
目的:	155
例:	155
airfilterreplaced	156
構文:	156
目的:	156
例:	157

amcportstate	157
構文:	157
目的:	157
例:	157
board	159
構文:	159
目的:	159
例:	160
boardreset	162
構文:	162
目的:	162
例:	163
busres	163
構文:	163
目的:	164
バス型 E-キーイング管理リソースの状態の表示	164
構文:	164
目的:	164
例:	165
指定したリソースの解放	165
構文:	165
目的:	165
例:	165
指定したリソースのロック/ロック解除	166
構文:	166
目的:	166
例:	166

「バス型に接続された資源の制御 (クエリー)」コマンドの送信 167

構文: 167

目的: 167

例: 167

リソースの所有者の設定 168

構文: 168

目的: 168

例: 168

「バス型に接続された資源の制御 (バスによって制限されない)」コマンドの送信 169

構文: 169

目的: 169

例: 169

console 170

構文: 170

目的: 170

例: 170

deactivate 171

構文: 171

目的: 171

例: 171

debuglevel 171

構文: 171

目的: 171

例: 172

exit | quit 173

構文: 173

目的: 173

例: 173



fans 174  
    構文: 174  
    目的: 174  
    例: 174

flashupdate 175  
    構文: 175  
    目的: 175  
    例: 175

fru 176  
    構文: 176  
    目的: 176  
    例: 177

frucontrol 178  
    構文: 178  
    目的: 178  
    例: 179

frudata 180  
    構文: 180  
    目的: 180  
    例: 181

frudatar 182  
    構文: 182  
    目的: 182  
    例: 183

frudataw 183  
    構文: 183  
    目的: 183  
    例: 184

fruinfo 184  
    構文: 184  
    目的: 184  
    例: 185

getacousticlevel 192  
    構文: 192  
    目的: 192  
    例: 193

getbootdev 193  
    構文: 193  
    目的: 193  
    例: 193

getfanlevel 194  
    構文: 194  
    目的: 194  
    例: 194

getfanpolicy 195  
    構文: 195  
    目的: 195  
    例: 195

getfruledstate 196  
    構文: 196  
    目的: 196  
    例: 196

gethysteresis 199  
    構文: 199  
    目的: 199  
    例: 199

getipmbstate 200  
    構文: 200  
    目的: 200  
    例: 200

getlanconfig 201  
    構文: 201  
    目的: 201  
    例: 203

auth\_support 204  
    構文: 204  
    目的: 204  
    例: 204

auth\_enables 205  
    構文: 205  
    目的: 205  
    例: 205

ip 206  
    構文: 206  
    目的: 206  
    例: 206

ip\_source 206  
    構文: 206  
    目的: 206  
    例: 207

mac 207  
    構文: 207  
    目的: 207  
    例: 207

subnet\_mask 207

構文: 207

目的: 207

例: 208

ipv4\_hdr\_param 208

構文: 208

目的: 208

例: 208

pri\_rmcp\_port 208

構文: 208

目的: 208

例: 209

sec\_rmcp\_port 209

構文: 209

目的: 209

例: 209

arp\_control 209

構文: 209

目的: 210

例: 210

arp\_interval 210

構文: 210

目的: 210

例: 210

dft\_gw\_ip 211

構文: 211

目的: 211

例: 211

dft_gw_mac	211
構文:	211
目的:	211
例:	211
backup_gw_ip	212
構文:	212
目的:	212
例:	212
backup_gw_mac	212
構文:	212
目的:	212
例:	212
community	213
構文:	213
目的:	213
例:	213
destination_count	213
構文:	213
目的:	213
例:	213
destination_type	214
構文:	214
目的:	214
例:	214
destination_address	215
構文:	215
目的:	215
例:	215

getmgmtportroute 216

構文: 216

目的: 216

例: 216

getmuxconfig 217

構文: 217

目的: 217

例: 217

getpefconfig 218

構文: 218

目的: 218

例: 219

control 220

構文: 220

目的: 220

例: 220

action\_control 221

構文: 221

目的: 221

例: 221

startup\_delay 222

構文: 222

目的: 222

例: 222

alert\_startup\_delay 222

構文: 222

目的: 222

例: 222

event_filter_count	223
構文:	223
目的:	223
例:	223
event_filter	223
構文:	223
目的:	223
例:	224
event_filter_data1	224
構文:	224
目的:	224
例:	225
alert_policy_count	225
構文:	225
目的:	225
例:	225
alert_policy	226
構文:	226
目的:	226
例:	226
system_guid	227
構文:	227
目的:	227
例:	227
alert_string_count	227
構文:	227
目的:	227
例:	227

alert\_string\_key 228

構文: 228

目的: 228

例: 228

alert\_string 228

構文: 228

目的: 228

例: 229

oem\_filter\_count 229

構文: 229

目的: 229

例: 229

oem\_filter 230

構文: 230

目的: 230

例: 230

getsensoreventenable 231

構文: 231

例: 231

getthreshold|threshold 232

構文: 232

目的: 233

例: 233

help 234

構文: 234

目的: 234

例: 234



ipmc 237  
    構文: 237  
    目的: 237  
    例: 238

localaddress 239  
    構文: 239  
    目的: 239  
    例: 239

mgmtportstate 239  
    構文: 239  
    目的: 239  
    例: 240

minfanlevel 240  
    構文: 240  
    目的: 240  
    例: 241

muxstate 241  
    構文: 241  
    目的: 241  
    例: 242

networkelementid 242  
    構文: 242  
    目的: 242  
    例: 243

poll 243  
    構文: 243  
    目的: 243  
    例: 243

sel 244  
    構文: 244  
    目的: 244  
    例: 245

sendamc 246  
    構文: 246  
    目的: 246  
    例: 247

sendcmd 247  
    構文: 247  
    目的: 247  
    例: 248

sensor 248  
    構文: 248  
    目的: 248  
    例: 249

sensordata 252  
    構文: 252  
    目的: 252  
    例: 253

sensorread 254  
    構文: 254  
    目的: 254  
    例: 254

session 255  
    構文: 255  
    目的: 255  
    例: 256

setacousticlevel 256

構文: 256

目的: 256

例: 257

setbootdev 257

構文: 257

目的: 257

例: 258

setextracted 258

構文: 258

目的: 258

例: 258

setfanlevel 259

構文: 259

目的: 259

例: 259

setfanpolicy 260

構文: 260

目的: 260

例: 260

setfruledstate 261

構文: 261

目的: 261

例: 262

sethysteresis 263

構文: 263

目的: 263

例: 263

setipmbstate 264

構文: 264

目的: 264

例: 264

setlanconfig 265

構文: 265

目的: 265

auth\_enables 266

構文: 266

目的: 266

例: 267

ip 267

構文: 267

目的: 267

例: 267

subnet\_mask 268

構文: 268

目的: 268

例: 268

ipv4\_hdr\_param 268

構文: 268

目的: 268

例: 269

arp\_control 269

構文: 269

目的: 269

例: 269

arp_interval	270
構文:	270
目的:	270
例:	270
dft_gw_ip	270
構文:	270
目的:	270
例:	270
backup_gw_ip	271
構文:	271
目的:	271
例:	271
community	271
構文:	271
目的:	271
例:	271
destination_type	272
構文:	272
目的:	272
例:	272
destination_address	272
構文:	272
目的:	272
例:	273
setlocked	273
構文:	273
目的:	273
例:	274

setmgmtportroute 274

構文: 274

目的: 274

例: 274

setmuxconfig 275

構文: 275

目的: 275

例: 275

setpefconfig 276

構文: 276

目的: 276

control 278

構文: 278

目的: 278

例: 278

action\_control 279

構文: 279

目的: 279

例: 279

startup\_delay 279

構文: 279

目的: 279

例: 280

alert\_startup\_delay 280

構文: 280

目的: 280

例: 280

event\_filter 280  
    構文: 280  
    目的: 281  
    例: 281

event\_filter\_data1 282  
    構文: 282  
    目的: 282  
    例: 282

alert\_policy 283  
    構文: 283  
    目的: 283  
    例: 283

system\_guid 283  
    構文: 283  
    目的: 284  
    例: 284

alert\_string\_key 284  
    構文: 284  
    目的: 284  
    例: 285

alert\_string 285  
    構文: 285  
    目的: 285  
    例: 285

oem\_filter 285  
    構文: 285  
    目的: 285  
    例: 286

setpowerlevel 286  
    構文: 286  
    目的: 287  
    例: 287

setsensordata 288  
    構文: 288  
    目的: 288  
    例: 290

setsensoreventenable 291  
    構文: 291  
    目的: 291  
    例: 292

setthreshold 292  
    構文: 292  
    目的: 293  
    例: 293

setuserlabel 294  
    構文: 294  
    目的: 294  
    例: 295

shelf 295  
    構文: 295  
    目的: 296  
    シェルフ FRU 情報の表示 296  
    構文: 296  
    目的: 296  
    例: 299



使用可能な最大外部電流の変更	302
構文:	302
目的:	302
例:	303
期待される最小動作電圧の変更	304
構文:	304
目的:	304
例:	305
「シェルフマネージャーで制御されるアクティブ化」フラグの変更	306
構文:	306
目的:	307
例:	307
「シェルフマネージャーで制御される非アクティブ化」フラグの変更	311
構文:	311
目的:	311
例:	312
FRU の最大電力容量の変更	313
構文:	313
目的:	313
例:	314
次の電源投入前の遅延の変更	315
構文:	315
目的:	315
例:	315
FRU のアクティブ化準備の許可の変更	316
構文:	316
目的:	316
例:	316

FRU アクティブ化および電源記述子の順序変更 317

構文: 317

目的: 318

例: 318

シェルフ FRU 情報の更新 319

構文: 319

目的: 319

例: 320

例: 320

シェルフ FRU 情報ストレージデバイスの更新 321

構文: 321

目的: 321

例: 321

shelfaddress 321

構文: 321

目的: 321

例: 322

shmstatus 323

構文: 323

目的: 323

例: 323

showhost 324

構文: 324

目的: 324

showunhealthy 325

構文: 325

目的: 325

例: 325

switchover	325
構文:	325
目的:	325
例:	326
terminate	326
構文:	326
目的:	326
例:	326
user	327
構文:	327
目的:	327
ユーザー情報の表示	327
構文:	327
目的:	327
例:	328
新しいユーザーの追加	328
構文:	328
目的:	328
例:	329
ユーザーの削除	329
構文:	329
目的:	329
例:	330
ユーザーの有効化および無効化	330
構文:	330
目的:	330
例:	330

ユーザー名の変更 332

構文: 332

目的: 332

例: 332

ユーザーのパスワードの変更 333

構文: 333

目的: 333

例: 333

指定ユーザーおよび指定チャンネルのチャンネルアクセス設定の変更 334

構文: 334

例: 334

userlabel 335

構文: 335

目的: 335

例: 335

version 336

構文: 336

目的: 336

例: 336

B. シェルフマネージャーの IPMI 拡張コマンドとセンサー 337

IPMI 拡張コマンド 338

Notify Shelf Manager About an Extracted FRU 339

Initiate Shelf Manager Switchover 340

Reset IPMC Configuration Variables 341

Set AMC Time Out Parameter 341

Get AMC Time Out Parameter 342

Set Boot Page 343

Get Boot Page 344

Set Front Panel Reset Button State	345
Get Front Panel Reset Button State	346
Set IPMC Control Bits	347
Get IPMC Control Bits	348
Set Management Port	349
Get Management Port	350
Set Shelf FRU Record Data	351
Get Shelf FRU Record Data	355
Set SOL Fail Over Link Change Timeouts	357
Get SOL Fail Over Link Change Timeouts	358
Get Version	359
Set Thermal Trip	360
Get Thermal Trip	361
Subscribe for Event Notifications	362
用語集	363
索引	371



# 目次

---

- ☒ 1-1 Sun Netra CT900 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
- ☒ 1-2 ATCA シェルフの例 7
- ☒ 1-3 シェルフマネージャーのスイッチオーバー信号 9
- ☒ 2-1 シェルフアラームパネルコネクタ 16
- ☒ 3-1 IPMI FRU 情報の配置 105





# 表目次

---

表 1-1	Sun Netra CT サーバーのシステム管理者用ソフトウェア	2
表 1-2	Sun Netra CT900 サーバーボードのアクセス方法	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とインタフェース	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 1-5	シェルフの物理アドレスと FRU	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の構成パラメータ	36
表 3-1	シェルフマネージャーにより実装される IPMI コマンド	80
表 3-2	シェルフマネージャーの IPMI 拡張コマンド	87
表 3-3	シェルフマネージャーの CLI コマンドの概要	89
表 3-4	16M バイトフラッシュのフラッシュパーティション	125
表 3-5	64M バイトフラッシュのフラッシュパーティション	126
表 3-6	シェルフマネージャーの CLI コンソール関連のコマンド	146
表 A-1	getlanconfig の LAN 構成パラメータ	201
表 A-2	PEF 構成パラメータ	218
表 A-3	setlanconfig の LAN 構成パラメータ	265
表 A-4	getlanconfig の PEF 構成パラメータ	276
表 A-5	shelf コマンドのパラメータ	297
表 B-1	シェルフマネージャーの IPMI 拡張コマンド	338
表 B-2	構成を保持できる IPMC 変数	353



# はじめに

---

このマニュアルでは、Oracle Sun Netra CT900 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。シェルフマネージャーおよび IPMI のコマンドリファレンスに関する情報も含まれます。

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) について理解していることを前提としています。

---

## UNIX コマンドの使い方

このドキュメントには、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。このような情報については、次のマニュアルを参照してください。

- システムに付属のソフトウェアドキュメント
- 下記にある Oracle Solaris オペレーティングシステム (Oracle Solaris OS) のドキュメント:

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

---

# シェルプロンプトについて

シェル	プロンプト
C シェル	<i>machine-name%</i>
C シェル (スーパーユーザー)	<i>machine-name#</i>
Bourne シェルと Korn シェル	\$
Bourne シェルと Korn シェル (スーパーユーザー)	#
U-Boot シェル	shmm500
Linux シェル	#
シェルフマネージャークマンド行インタフェース (CLI)	CLI>

---

---

## 関連ドキュメント

Sun Netra CT900 サーバーのドキュメントを次の表に示します。『Important Safety Information for Sun Hardware Systems』を除き、以下に示すすべてのドキュメントは、次の Web サイトからオンラインで入手できます。

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

用途	タイトル	形式	場所
ご使用の手引き	『Sun Netra CT900 サーバーご使用の手引き』	印刷物 PDF	出荷用キット オンライン
最新情報	『Sun Netra CT900 Server Product Notes』	PDF HTML	オンライン
概要	『Sun Netra CT900 サーバー概要』	PDF HTML	オンライン
設置	『Sun Netra CT900 サーバー設置マニュアル』	PDF HTML	オンライン
保守	『Sun Netra CT900 Server Service Manual』	PDF HTML	オンライン

---

用途	タイトル	形式	場所
リファレンス	『Sun Netra CP3140 Switch Software Reference Manual』	PDF HTML	オンライン
安全性	『Sun Netra CT900 Server Safety and Compliance Manual』	PDF HTML	オンライン
安全性	『Important Safety Information for Sun Hardware Systems』 (印刷版のみ)	印刷物	出荷用キット

追加の情報について、Oracle Solaris OS、OpenBoot PROM ファームウェア、Sun Netra CP3010 ボード、Sun Netra CP3020 ボード、Sun Netra ATCA CP3220 ブレードサーバー、Sun Netra ATCA CP3060 ブレードサーバー、Sun Netra ATCA CP3260 ブレードサーバーの各製品のドキュメントを参照する必要がある場合があります。これらのドキュメントはオンラインで入手できます。

## ドキュメント、サポート、およびトレーニング

次の Web サイトでは追加リソースが提供されています。

- ドキュメント <http://www.oracle.com/technetwork/indexes/documentation/index.html>
- サポート <https://support.oracle.com>
- トレーニング <https://education.oracle.com>



# 第1章

---

## 概要

---

この章には、次のセクションが含まれています。

- 1 ページの「Sun Netra CT900 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」
- 

## Sun Netra CT900 サーバーのソフトウェア

Sun Netra CT900 サーバーのシェルフソフトウェアには、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
  - ファームウェア (U-Boot、IPMC)
- 

注 – The Advanced Telecom Computing Architecture® (ATCA) では、電気通信分野で使用されている表現に合わせて、シェルフという用語を採用しています。従来使用していたシャーシという用語と基本的に同じ意味です。

---

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 Sun Netra CT サーバーのシステム管理者用ソフトウェア

カテゴリ	名前	説明
シェルフ管理	Pigeon Point Shelf Manager	シェルフマネージャソフトウェアは、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。サーバーの管理用に、RMCP (Remote Management Control Protocol) と、IPMI への CLI 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 (シェルフ管理カード上)	Monterey Linux	Monterey Linux は、シェルフ管理カード (ShMM) にインストールされ、シェルフマネージャソフトウェアアプリケーションを実行します。
ファームウェア	U-Boot	Firmware on the シェルフ管理 cards that performs power-on self-test (POST) and controls booting of the シェルフ管理 card software.
	Intelligent Platform Management Controller (IPMC)	Sun Netra CP3x60 ボードなど、Sun がサポートするノードボード上の IPMI コントローラ経由の通信を可能にするシステム管理コントローラファームウェア。
オペレーティングシステム (Sun Netra ATCA ノードボード上)	Oracle Solaris オペレーティングシステム (Oracle Solaris OS)	Oracle Solaris OS は、Sun Netra CP3010、Sun Netra CP3x20、CP3x60 ノードボードなどの Sun Netra ATCA ノードボード上で動作します。Oracle Solaris 10 は、オプションで、Sun Netra ノードボードにプリインストールされます。Oracle Solaris 10 およびその他のバージョンの Oracle Solaris OS は、ユーザーがダウンロードしてインストールすることができます。
	Linux OS	一部のバージョンの Linux OS は、Sun Netra CP3x20 ボード上で動作します。Linux OS のサポートについては、ボードのドキュメントを参照してください。

Sun Netra CT900 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、アクティブなシェルフ管理カードからバックアップ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。アクティブな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および管理に使用されます。バックアップ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、背面に外部接続用の Ethernet ポートを備えています。



一部の Sun Netra ATCA ノードボードや背面切り替えモジュール (RTM) には、ディスクなどの周辺機器を接続できます。Sun Netra ATCA ノードボードは、ユーザーアプリケーションも実行します。Sun Netra CT 900 サーバーでは、各ノードボードが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みなされます。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、およびその他のシステム FRU (現場交換可能ユニット) で構成されます。

---

**注** – このマニュアル内の「ノードボード」という用語は、特に指定のないかぎり、Sun Netra CP3010 ボードなどの Sun Netra ATCA ノードボード (または、ブレードサーバー) を指します。

Sun Netra CT900 サーバーでは、PICMG<sup>®</sup> 3.x に準拠した、サードパーティー製の ATCA ノードボードを使用できます。これらのボードは、必ずしも Oracle Solaris OS を実行しているとはかぎらないため、Sun Netra CT900 サーバーのシステム管理ソフトウェアを実行しません。このため、Sun Netra ノードボードと同程度に管理することはできません。

---

表 1-2 は、各種ボードへのアクセス方法をまとめたものです。シェルフ管理カードは、同時に 22 のセッション (1 つの Tip 接続と、21 の Telnet 接続) をサポートします。

表 1-2 Sun Netra CT900 サーバーボードのアクセス方法

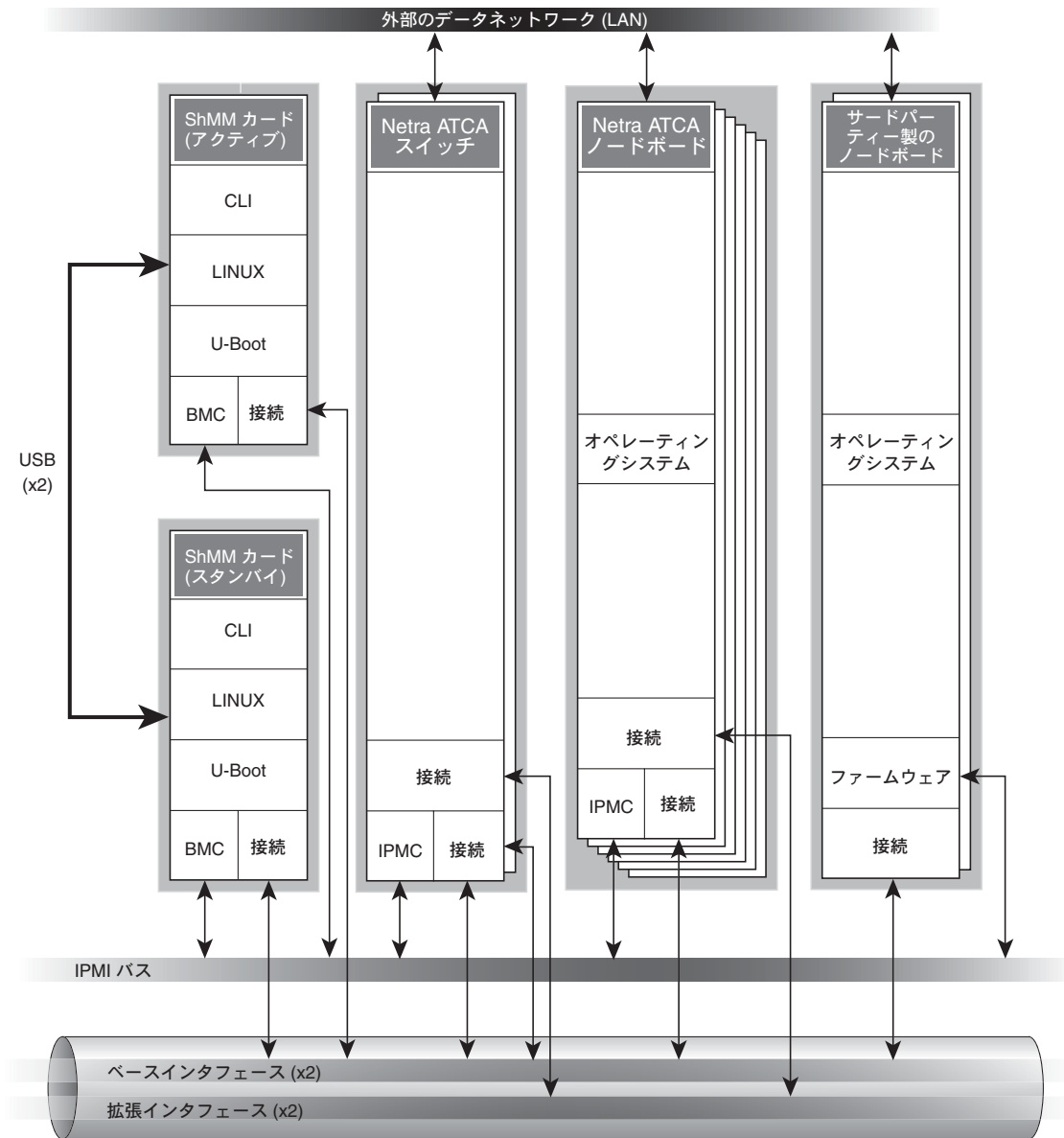
ボード	アクセス方法
シェルフアラームパネル (SAP)	フロントパネルには次のポートがあります。 <ul style="list-style-type: none"><li>• RJ-45 DTE コネクタ付きのシリアル (RS-232) ポート 2 つ。シリアル 1 ポートは、デフォルトのアクティブなカードである、上部のシェルフ管理カード (ShMM1) 用のコンソール接続です。シリアル 2 ポートは、デフォルトのバックアップカードである、下部のシェルフ管理カード (ShMM2) 用のコンソール接続です。</li><li>• Telco アラームコネクタ (DB-15)</li></ul>
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。 <p><b>注</b> – これらのポートに背面からアクセスするには、背面切り替えモジュール (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。</p>

表 1-2 Sun Netra CT900 サーバーボードのアクセス方法 (続き)

ボード	アクセス方法
ノードボード (Sun Netra ATCA ノードボード) (スロット 1 - 6 および 9 - 14)	<p>Sun Netra CP3010 ボードのフロントパネルには、次のポートがあります。</p> <ul style="list-style-type: none"> <li>• Tip 接続または 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> <p>注 - これらのポートに背面からアクセスするには、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。</p> <p>Sun Netra CP3020 および CP3060 ボードのフロントパネルには、次のポートがあります。</p> <ul style="list-style-type: none"> <li>• Tip 接続または 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> <p>詳細は、特定のノードボードに対応する Sun Netra ATCA CPU ボードのドキュメントを参照してください。</p> <p>console CLI コマンドを使用すると、ShMM からすべての ATCA ノードボードにアクセスできます。</p>
サードパーティー製のノードボード (スロット 1 - 6 と 9 - 14)	サードパーティー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、ノードボード、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 Sun Netra CT900 サーバーのソフトウェアインターフェースおよびハードウェアインターフェースの論理図



---

# 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が、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 の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 などのモジュラープラットフォーム用に設計された Intelligent Platform Management (IPM) の構成要素です。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牢な基盤を提供しますが、その十分なサポートには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。

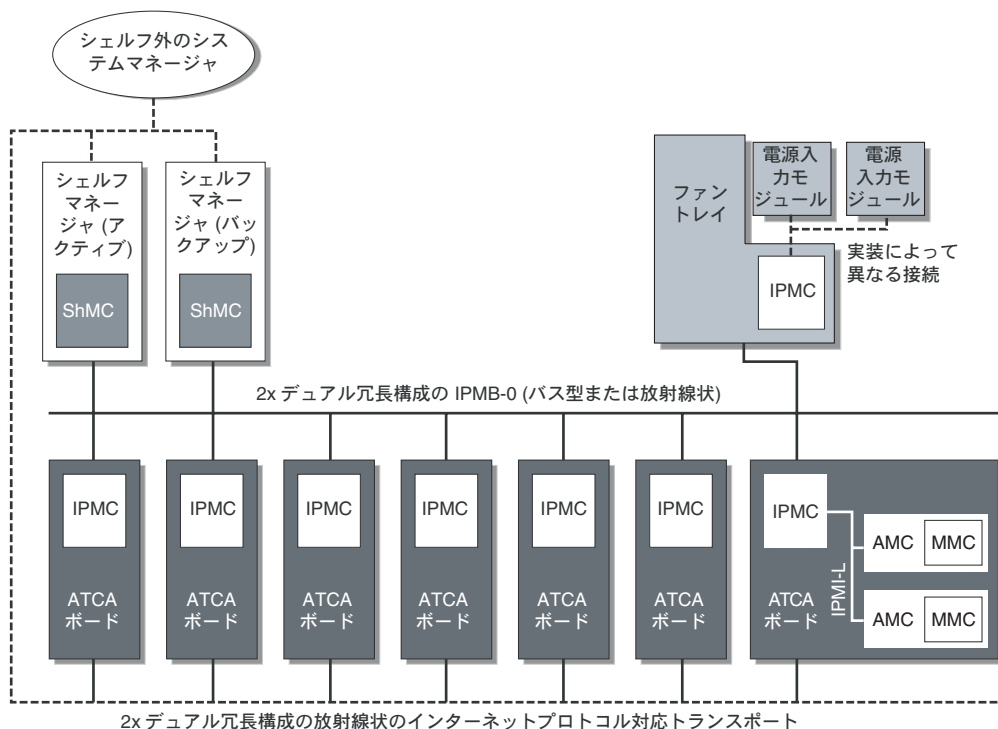
図 1-2 に、ATCA 仕様の観点から ATCA シェルフの 1 つの実例の論理要素を示します。

AdvancedTCA のシェルフマネージャーは、シェルフ内で IPM コントローラと通信します。各 IPM コントローラは、ボード、ファントレイ、電源入力モジュールなどの 1 つ以上の FRU (現場交換可能ユニット) をローカルに管理します。シェルフ内の管理通信は、主に、AdvancedTCA 内の IPMB-0 としてデュアル冗長に基づいて実装された Intelligent Platform Management Bus (IPMB) を介して実行されます。

PICMG Advanced Mezzanine Card (AdvancedMC または AMC) 仕様の AMC.0 では、AdvancedTCA の物理および管理アーキテクチャにスムーズに適合するように設計されたホットスワップ可能なメザニンフォームファクタを定義しています。

図 1-2 は、1 つの IPMC と、それぞれにモジュール管理コントローラ (MMC) を搭載した 2 つの取り付け済み AMC モジュールを含む AMC キャリアを示しています。キャリア上の管理通信は、IPMB-L ("L" はローカルを意味します) 上で行われます。

図 1-2 ATCA シェルフの例



総合的なシステムマネージャ (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と通信します。

図 1-2 は、3 つの管理レベルであるボード、シェルフ、およびシステムを示しています。次の節では、ATCA 準拠のシェルフマネージャ (ShMM) を実装する、シェルフマネージャのソフトウェアと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。
- 全体のシステムマネージャーが、システムマネージャーのインタフェースから管理および追跡に参加できるようにします。通常は Ethernet 経由で実装されます。

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。

- シェルフ内に実装された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 で指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警告を送信します。
- 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。
- 高可用性を実現する、冗長な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、定期的なストローブ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、バッテリー付き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、可用性を最大化するために、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、シェルフ内に装着された IPM コントローラと IPMB-0 を介して通信を行うのは、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

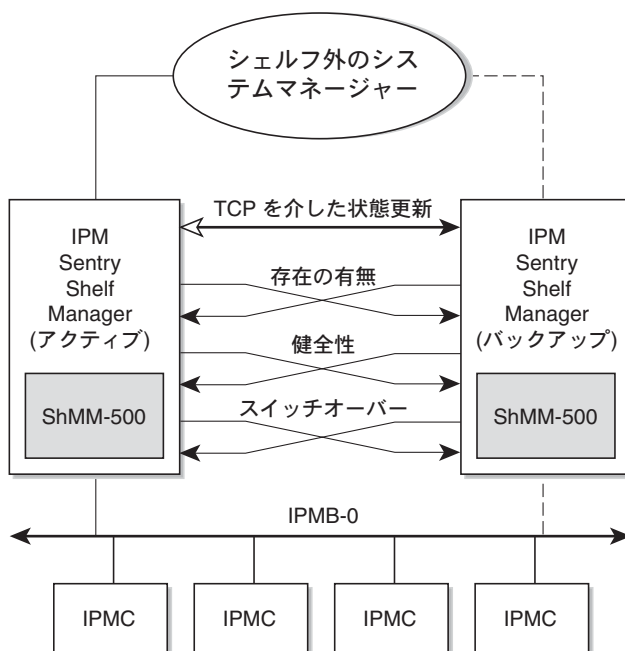


表 1-3 に、信号と説明を示します。

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とインタフェース

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情報を送信するために使用されます。どちらのシェルフ管理カードも、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

アクティブなシェルフマネージャーは、IPMB 上で ShMM デバイス (アドレス 20h) を公開し、IPMB と 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 ShMM 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メモリー (揮発性および非揮発性) 内に保持し、その状態をアクティブな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あります。次のように、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 – アクティブおよびバックアップシェルフマネージャーは、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、アクティブなシェルフマネージャーが動作していない、または正常でないかと判断して、アクティブなシェルフマネージャーの役割を強制的に引き継ぎます。



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、アクティブな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、この信号が停止したことは、ピアシェルフマネージャーをホストしているボードがシェルフから取り外されたことを示します。Remote Healthy 信号は、初期化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、この信号が停止したことは、リモートのシェルフマネージャーが正常でなくなった (通常、電源が切断されたかリセットされた) ことを示します。

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、アクティブな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、アクティブなシェルフ管理カードの電源が切断さ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 信号が停止する前に、TCP 接続が閉じることがあります。TCP 接続が閉じた理由を確認するために、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ます。この信号がまだアクティブな場合には、しばらく待機したあとでもう一度状態を取得します。Remote Healthy 信号が最終的に停止した場合、バックアップシェルフマネージャーはアクティブなシェルフマネージャーが動作していないと判断し、スイッチオーバーを開始します。

Remote Healthy 信号がアクティブな場合、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判断します。この場合、スイッチオーバーは開始されません。その代わりに、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な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。ピアシェルフマネージャーがすでにアクティブな場合には、シェルフマネージャー内の特別なロジックによって、シェルフマネージャーが起動時にアクティブ化を試行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、無限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。アクティブなシェルフマネージャーのウォッチドッグタイマーがトリガーされると、そのシェルフ管理カードがリセットされ、バックアップシェルフ管理カードで Remote Healthy 信号が停止するため、スイッチオーバーが開始されます。

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が再初期化を行い、キャッシュされている状態情報を有効にして、IPMB 上の IPM コントローラから必要な情報を収集します。このアクティブなシェルフマネージャーは、次に、IPMB 上の ShMC デバイス (アドレス 20h) を公開し、以前アクティブだったシェルフマネージャーとシステムマネージャーの間で RMCP およびその他のシェルフ外対話に使用されていた IP アドレスを引き継ぎます。RMCP セッション情報はアクティブなシェルフマネージャーからバックアップシェ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後も維持されます。RMCP を使用してい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、これまでアクティブだったシェルフマネージャーが消滅することも、それ自体がバックアップシェルフマネージャーとして再初期化されることもできます。バックアップシェ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シェ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

シェルフマネージャーのもう 1 つの主要サブシステムには、システム管理者インタフェースが実装されます。システム管理者とは、オペレーションセンター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。シェ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。それぞれが、異なるメカニズムによってシェ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、個別に実装されたシェルフ製品間の相互運用性を最大限に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 を介したシェ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。RMCP を使用してシェルフと通信するシステム管理者は、ATCA 準拠の任意のシェルフマネージャーと通信する必要があります。この下位レベルのインタフェースは、システム管理者がシェルフマネージャーをプロキシとして使用してシェルフ内の IPM コントローラに IPMI コマンドを実行できる機能など、シェルフの IPMI の側面へのアクセスを提供します。

RMCP は、LAN 経由で 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シェルフマネージャーに対して実行できる、包括的なテキストコマンドセットを提供します。

# システム管理作業

Sun Netra CT900 サーバシステムの管理には、通常、設置、構成、および管理の各作業が含まれます。

Oracle Solaris ユーザーアカウントの追加など、Sun Netra CT900 サーバ上での Oracle Solaris OS の管理は、ノードボードにログインして行います。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、構成および管理目的で使用する場合は、Sun Netra CT900 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

Sun Netra CT900 サーバを前から見た場合、物理スロットには左から右の順に番号が付いています。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 1-5 に、シェルフの物理アドレスと関連付けられている FRU 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	スイッチ 7	スイッチ 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	該当 なし	該当 なし
ハードウェア 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

表 1-5 シェルフの物理アドレスと FRU

	アドレス (16 進数)	FRU #
シェルフ/シャーシ	20	該当なし
PPS BMC	20	0
シェルフ EEPROM 1	20	1

表 1-5 シェルフの物理アドレスと FRU (続き)

	アドレス (16 進数)	FRU #
シェルフ EEPROM 2	20	2
ファントレイ 1	20	3
ファントレイ 2	20	4
ファントレイ 3	20	5
PEM 1	20	6
PEM 2	20	7
SAP	20	8

## 第2章

# システムの設定

---

Sun Netra CT900 サーバーの設定には、基本的にアクティブな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。アクティブなシェルフ管理カードの CLI を使用すると、ノードボード、スイッチボード、シェルフ管理カード (ShMM)、電源入力モジュール (PEM)、ファントレーなど、システムレベルの設定や管理を行うことができます。シェルフ管理カードの CLI インタフェースは、ローカルでもリモートでも使用できます。

この章には、次のセクションが含まれています。

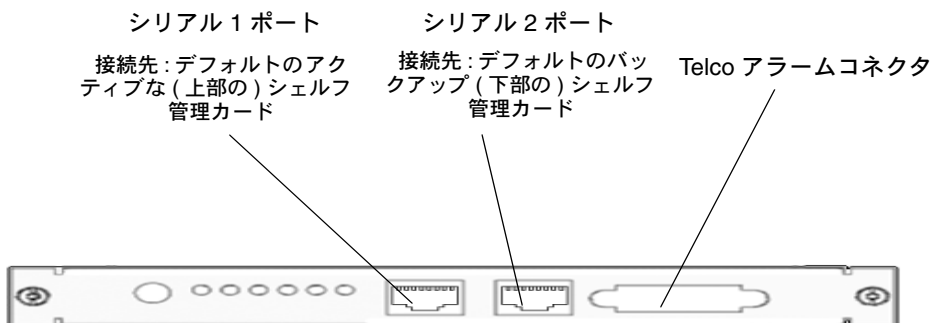
- 16 ページの「シェルフ管理カードへのアクセス」
- 17 ページの「U-Boot の設定」
- 25 ページの「シェルフ管理カードの Ethernet ポートの構成」
- 35 ページの「シェルフマネージャーの構成ファイルの設定」
- 64 ページの「日付と時刻の設定」
- 66 ページの「シェルフ管理カードでのユーザーアカウントの設定」
- 68 ページの「リモート root ログインの無効化」
- 70 ページの「シェルフマネージャーでの OpenHPI の構成」
- 76 ページの「シェルフマネージャーでの RADIUS の構成」

## 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には、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。アクティブな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 (ShMM) に接続する場合は、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、上部のシェルフ管理カード (ShMM1) に接続する際に使用します。ShMM1 はデフォルトのアクティブなシェルフ管理カードです。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には、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、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する必要があります。

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、アクティブな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る場合は、`clia switchover` コマンドを使用して、シェルフ管理カードをアクティブに変更するか、終了してアクティブなシェルフ管理カードにログオンします。詳細は、[323 ページの「shmstatus」](#) および [325 ページの「switchover」](#) を参照してください。

以降の各節では、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 の使用方法については、[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ー内の 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 の基本的な初期化を実行し、ユーザーが明示的に自動起動機能を無効にしていなかったかぎり (つまり、ファームウェアを保守用の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、フラッシュ内のカーネルと root ファイルシステムの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できますが、対象の運用環境に固有の設定が必要となります。ShMM の電源を投入すると、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、使用前に設定しておく必要がある一連の環境変数があります。表 2-1 で、デフォルトの環境変数について説明します。

表 2-1 デフォルトの U-Boot 環境変数

環境変数	説明
addmisc	quiet、reliable_upgrade、console の各設定を bootargs に付加します。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。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ます。この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	自動起動を行うために実行される U-Boot コマンド。デフォルトは次のとおりです。 run ramargs addmisc; bootm \$(kernel_start) \$(rfs_start)



表 2-1 デフォルトの U-Boot 環境変数 (続き)

環境変数	説明
bootdelay	自動起動の遅延値 (秒単位)。
bootfile	net および nfs 起動オプションで使用され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。
ethladdr	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	フラッシュファイルシステム (/etc および /var) を消去して、出荷時のデフォルト (y/n) に戻すよう、Linux に指示します。フラッシュの消去後、システム起動スクリプトによってこの変数の設定が n に戻されます。デフォルトは n です。
gatewayip	デフォルトゲートウェイの IP アドレス。この変数をカーネルコマンド行の一部として渡すことで、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できます。
hostname	ネットワークホスト名。デフォルトは shmm500 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設定するかどうかを指定します (y/n)。デフォルトは y です。
ipaddr	オンチップの一次 Ethernet インタフェースで使用される IP アドレス。rc_ifconfig 変数が y に設定されている場合は、ipdevice で指定されたネットワークインタフェースを自動的に設定するために、この変数が使用されます。なお、システム起動スクリプトによって、この変数の最下位ビットが ShMM キャリアのハードウェアアドレスの最下位ビットに設定されます。つまり、ハードウェアアドレスが偶数値の場合は IP アドレスの最下位ビットが 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 で同期された 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 で実行されます。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。デフォルトは 192.168.0.22 です。
ipladdr	USB インタフェースで使用される IP アドレス。この変数をカーネルコマンド行の一部として渡すことで、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できます。デフォルトは 192.168.1.2 です。
ip6f または 2A (Sun レガシー) ddr	二次 Ethernet インタフェースで使用される IP アドレス。この変数をカーネルコマンド行の一部として渡すことで、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できます。デフォルトは 192.168.2.1 です。

表 2-1 デフォルトの U-Boot 環境変数 (続き)

環境変数	説明
ipdevice	ipaddr に対応するデバイス。デフォルトは eth0 です。
ipldevice	ipladdr に対応するデバイス。デフォルトは usb0 です。
ip2device	ip6F または 2A (Sun レガシー) ddr に対応するデバイス。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。
logging	メッセージログファイルを ram と flash のどちら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。
monitor_daemons	この変数を y に設定すると、OpenHPI モニターが有効になります。デフォルトは n です。
net	この変数は、TFTP サーバーからカーネルと .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	ネットワークのネットマスク。デフォルトは 255.255.255.0 です。
password_reset	出荷時のデフォルトのパスワード (ユーザー root のパスワード sunct900) に戻すよう、Linux に指示します。デフォルトは n です。
post_normal	起動のたびに実行される POST テストのリストを指定します。この変数が設定されていない場合は、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	起動のたびにではなく、電源投入によるリセット後のみ実行される POST テストのリストを指定します。この変数が設定されていない場合は、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	起動時に進捗メッセージをシリアルコンソールに出力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 起動オプションで使用される .rfs イメージを指定します。デフォルトは sentry.rfs です。
ramsize	システムメモリーのサイズ (バイト単位)。デフォルトの設定は、ビルド時の構成ブロックでエンコードされる 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 スクリプトで 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 での IP アドレスの設定を許可) です。

表 2-1 デフォルトの U-Boot 環境変数 (続き)

環境変数	説明
rc2	呼び出す二次 RC スクリプトを指定します。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、 <code>/etc/rc.acb3</code> 、または所定のターゲットプラットフォームに適したその他のスクリプトです。
reliable_upgrade	信頼できるソフトウェアアップグレード処理を ShMM 上で有効にす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。現在、この変数を n に設定することはできません。この変数を n に設定すると、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。
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。デフォルトは 192.168.0.2 です。
serverip	TFTP サーバーの IP アドレス。デフォルトは 192.168.0.7 です。
start_rc2_daemons	起動後に <code>snmpd/boa</code> デーモンと <code>shelfman</code> デーモンを開始するかどうかを二次起動スクリプトに指示します。デフォルトは y です。
time_server	実行時の同期用のタイムサーバー。この変数を指定しない場合は、システムの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、 <code>ip1device</code> 変数に <code>usb0</code> を設定して、同期が正しく行われるようにする必要があります。
time_proto	ネットワークタイムサーバーから時刻を取得する際に使用するプロトコル。設定可能な値は、 <code>ntp</code> および <code>rdate</code> です。
timezone	CCC <i>n</i> 形式の現地時間。 <i>n</i> 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) で、CCC はタイムゾーンを示します。デフォルトは UTC です。DST タイムゾーンについては、66 ページの「夏時間のタイムゾーンの設定」を参照してください。

## 環境変数値の表示

環境変数の表示は、U-Boot シェルまたは Linux シェルで行うことができます。

### U-Boot シェルでの環境変数の表示

U-Boot シェル内から現在の環境変数値を表示するには、`printenv` コマンドを使用します。`printenv` に引数を指定しない場合は、すべての環境変数値が出力されます。

```
printenv [variable_name]
```

例:

```
shmm500 printenv bootdelay  
bootdelay=3
```

## Linux シェルでの環境変数の表示

Linux シェルから現在の環境変数値を表示するには、`getenv` コマンドを使用します。`getenv` に引数を指定しない場合は、すべての環境変数値が出力されます。

```
getenv [variable_name]
```

例:

```
# getenv bootdelay  
bootdelay=3
```

## 環境変数への値の割り当て

環境変数への値の割り当ては、U-Boot シェルまたは Linux シェルで、`setenv` コマンドを使用して行うことができます。U-Boot シェルで変数を設定した場合は、`saveenv` コマンドを使用して、フラッシュに変更を保存する必要があります。

環境変数に値を割り当てるには、次の形式を使用します。

```
setenv variable_name new_value
```

例:

```
setenv bootdelay 1
```

U-Boot シェルで変数を設定した場合は、次のように入力して変更を保存します。

```
saveenv
```

## シェルフマネージャーのデフォルトの環境変数

U-Boot をはじめて起動したときに、次のデフォルトの環境変数が定義されます。

```
baudrate=115200  
ipaddr=192.168.0.22  
serverip=192.168.0.7  
netmask=255.255.255.0  
bootfile=sentry.kernel
```

```
ramdisk=sentry.rfs
rootpath=/rootfs
ramargs=setenv bootargs root=/dev/ram rw
net=tftpboot 80400000 $(bootfile); tftpboot 81200000 $(ramdisk); run ramargs
addmisc; bootm 80400000 81200000
nfsargs=setenv bootargs root=/dev/nfs rw nfsroot=$(serverip):$(rootpath)
addip=setenv bootargs $(bootargs)
ip=$(ipaddr):$(serverip):$(gatewayip):$(netmask):$(hostname):$(ipdevice)
addmisc=setenv bootargs $(bootargs) $(quiet) console=$(console),$(baudrate)
reliable_upgrade=$(reliable_upgrade)
nfs=tftpboot 80800000 $(bootfile); run nfsargs addip addmisc; bootm
bootcmd=run ramargs addmisc; bootm $(kernel_start) $(rfs_start)
console=ttyS0
quiet=quiet
ipdevice=eth0
ipldevice=eth1
ipladdr=192.168.1.2
gatewayip=192.168.0.1
rmcpaddr=192.168.0.2
hostname=shmm500
flash_reset=n
password_reset=n
logging=ram
timezone=UTC
rc_ifconfig=n
start_rc2_daemons=y
reliable_upgrade=y
ethact=Au1X00 ETHERNET
stdin=serial
stdout=serial
stderr=serial
ethaddr=00:18:49:00:0c:42
eth1addr=00:18:49:00:0c:43
serial#=08009847
kernel_start=0xbe080000
rfs_start=0xbe440000
rc2=/etc/rc.acb3
bootdelay_old=3
post_normal=
bootdelay=3
```

これらの環境変数の一部は、ShMM が使用されるネットワークコンテキストに適した値で再設定する必要があります。

## U-Boot 変数の変更に関する推奨事項

一部の U-Boot 変数を変更するとシステムに問題が発生する可能性があるため、管理者は変更を行う前に次の推奨事項を確認してください。

変更可能な U-Boot 変数は次のとおりです。

```
baudrate
console
gatewayip
hostname
ipaddr
ip6F または 2A (Sun レガシー) ddr
password_reset
netmask
rmcpaddr
serverip
time_proto
time_server
timezone
```

次の変数は、上級ユーザーがデバッグのためにのみ変更するようにしてください。

```
bootfile
boot_delay
flash_reset
logging
net
nfs
post_normal
post_poweron
quiet
ramdisk
start_rc2_daemons
rc_ifconfig
```

次の変数を変更しないでください。

```
addmisc
bootargs
bootcmd
ethaddr
ethladdr
io_config
ipdevice
ipldevice
ipladdr
ip2device
kernel_start
ramargs
ramsize
rc2
reliable_upgrade
rfs_start
```

次の Ethernet 構成変数を変更しないでください。

```
rc_ifconfig = y
ipdevice = eth0
ipldevice = usb0
ipladdr = 192.168.1.2
ip2device = eth1
```

---

## 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ている 2 つの Ethernet ポートを使用します。RMCP は ATCA が必要とする唯一のシェルフ外インタフェースであるため、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、フルアクセス権を持つ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 ポートにその範囲内の一意の IP アドレス (192.168.0.2 など) を設定する必要があります。冗長 ShMM の設定では、1 つの ShMM (アクティブな ShMM) のみが RMCP Ethernet ポートで使用できる RMCP IP アドレスを持ちます。バックアップの 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 ShMM がアクティブな役割を引き継いだときにのみ有効になります。この方法により、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 アドレス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 IP アドレスが割り当てられません。そのため、ネットワーク経由で ShMM にアクセスすることができません。ただし、状況によっては、オペレーティングシステムの起動直後に 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、ネットワーク経由で ShMM にアクセスできるようにすると便利な場合があります。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めます。

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) に割り当ててのではなく、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、初期設定スクリプト /etc/rc 内で行われます。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。例:

```
setenv ipaddr 192.168.1.240
```



### 3. ShMM を再起動する前に、変更を保存します。

```
U-Boot 1.1.4 (Feb 26 2009 - 06:28:24)
CPU: Au1550 324 MHz, id: 0x02, rev: 0x00
Board: ShMM-500
S/N: 08002600
DRAM: (Samsung K4S511633F) 128 MB
Flash: 64 MB
In: serial
Out: serial
Err: serial
ADM1060: configuration version 80
Net: Au1X00 ETHERNET
Hit any key to stop autoboot: 3

shmm500 setenv rc_ifconig y
rc_ifconfig=y
shmm500
shmm500 setenv ipaddr 192.168.1.240
shmm500 printenv ipaddr
ipaddr=192.168.1.240
shmm500
shmm500 saveenv
Saving Environment to EEPROM...
shmm500
shmm500 boot
```

IP アドレス 192.168.1.240 が、ShMM の eth0 インタフェースとして表示されます。

4. シェルフマネージャの構成ファイル /etc/shelfman.conf 内の RMCP\_NET\_ADAPTER の値を eth0:1 に変更します。たとえば、RMCP\_NET\_ADAPTER = \$IPDEVICE を次のように変更します。

```
RMCP_NET_ADAPTER = eth0:1
```

---

**注** - この変更により、RMCP アドレス ([clia] getlanconfig 1 ip の出力) は、インタフェース eth0 ではなく、インタフェース eth0:1 として表示されます。

---

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つの各冗長 ShMM に割り当てられる実際の初期 IP アドレスは、`ipaddr` の値に基づいていますが、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。上記の例では、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することができます。

## RMCP アドレスの伝播

シェルフマネージャーのオプション機能では、RMCP IP アドレスと最下位ビットのみが異なる 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ができます。バックアップ ShMM のネットマスクとデフォルトゲートウェイは、アクティブな 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 の場合、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の構成ファイル (`/etc/shelfman.conf`) で、シェルフマネージャーの構成パラメータ `PROPAGATE_RMCP_ADDRESS` を `TRUE` に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に接続します。冗長シェルフマネージャ間の通信には、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。これらのインタフェースの名前は、`usb0` と `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、インタフェース `usb0` がピアシェルフマネージャー上でアクティブである場合（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われます。

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、シェルフマネージャーの構成ファイル `/etc/shelfman.conf` 内で、次のように 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 つだけ使用されている場合、`/etc/shelfman.conf` で指定された冗長 IP アドレスから 2 つの異なる IP アドレスが派生します。これらのアドレスは冗長接続の 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)。動作が正常に行われるようにするためには、同じシェルフマネージャー上の 2 つの終端 (`usb0` および `usb1`) がそれぞれ異なる論理ネットワークに属している必要があります。一方のシェルフマネージャー上の `usb0` と、もう一方のシェルフマネージャー上の `usb1` が、同じ論理ネットワークに属している必要があります。これらの考慮事項に基づき、`/etc/shelfman.conf` で指定された冗長 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で、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も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` に設定されます。冗長性を実現するために USB ネットワークインタフェースが使用されている場合も、`/etc/shelfman.conf` 内の該当するパラメータにこの値を使用することをお勧めします。

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、次の定義があるものとします。

```
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、IP アドレスの割り当て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、IP アドレスの割り当て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するように構成するには、次のネットワークパラメータを変更する必要があります。

- 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、起動されたアクティブな 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の 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、この接続は、115200 ボー、N/8/1 です。ShMM キャリアを再起動してスペースキーを押し、自動起動プロセスに割り込みます。次のように表示されます。

```
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 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: <any_key> #Hit any key now
shmm500
```

2. 現在のネットワーク設定を表示します。

```
shmm500 getenv rmcpaddr netmask gatewayip
rmcpaddr=192.168.0.44
netmask=255.255.255.0
gatewayip=192.168.0.1
shmm500
```

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。

```
shmm500 setenv rmcaddr 10.1.1.10
shmm500 setenv netmask 255.255.0.0
shmm500 setenv gatewayip 10.1.1.1
shmm500 saveenv
Un-Protected 1 sectors
Erasing sector 0 ... Erasing sector at 0x 800000
ok.
Saving Environment to EEPROM...done.
shmm500
```

4. ShMM を起動して完全に機能する状態にし、root ユーザーとしてログインします。

```
shmm500 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、フェイルオーバーの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。

シェルフマネージャーをはじめて起動した場合、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、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 で行った変更がその先へと伝播されます。

これ以外の場合は、次の手順を使用して、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必要があります。

## 6. アクティブな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、`cp1d` コマンドを使用して、アクティブなシェルフ管理カードに対してのみ行う必要があります。この操作により、冗長インタフェースを介して、バックアップもこのネットワーク構成変更で更新されます。アクティブな 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。

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

## 7. 現在の IP 設定を取得します。

```
# clia getlanconfig 1

Pigeon Point 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. 次のように、IP 設定を変更します。

```
# cli setlanconfig 1 ip 10.1.1.10

Pigeon Point Shelf Manager Command Line Interpreter
IP set successfully

# cli setlanconfig 1 subnet_mask 255.255.0.0

Pigeon Point Shelf Manager Command Line Interpreter
Subnet Mask set successfully

# cli setlanconfig 1 dft_gw_ip 10.1.1.1

Pigeon Point Shelf Manager Command Line Interpreter
Default Gateway Address set successfully
#
```

## 上部の ShMM と下部の ShMM への異なる IP アドレスの割り当て

デフォルトの構成では、アクティブな ShMM には RMCP IP アドレスを、バックアップの ShMM には RMCP IP1 アドレス (自動的に割り当てられる) を使用してアクセスできます。一部のネットワークでは、RMCP アドレスとは異なる IP アドレスを使用して ShMM を識別することが望ましい場合があります。その利点は、アクティブな ShMM にはその IP アドレスと RMCP アドレスのいずれかを使用してアクセスでき、バックアップの ShMM にはその IP アドレスを使用してアクセスできることにあります。

各 ShMM を 1 つの IP アドレスと、共有される 1 つの RMCP アドレス (つまり、アクティブな ShMM のみが RMCP アドレスを持つ) を使用して構成するには、次の操作を行います。

## 1. 上部の ShMM を構成します。

- a. 次のように、上部の ShMM で U-Boot 変数を変更します。詳細は、[22 ページ](#)の「環境変数への値の割り当て」を参照してください。

```
setenv ipaddr IP-addr-for-upper-ShMM
setenv ipdevice eth0                # Default DO NOT CHANGE
setenv ip1addr 192.168.1.2          # Default DO NOT CHANGE
setenv ip1device usb0               # Default DO NOT CHANGE
setenv gatewayip gateway-ip-address
setenv netmask netmask              # Usually 255.255.255.0
setenv rc_ifconfig y
setenv hostname hostname-of-lower-ShMM # Optional
saveenv                             # Optional - use in U-Boot
                                     # shell to save variables
```

- b. 上部の ShMM で、`/etc/shelfman.conf` ファイルを編集します。

- i. `PROPAGATE_RMCP_ADDRESS` 変数の値を `FALSE` に変更します。

```
PROPAGATE_RMCP_ADDRESS = FALSE
```

- ii. `RMCP_NET_ADAPTER` を `eth0:1` に変更します。

- c. アクティブな ShMM から `/etc/readhwaddr` を削除します。

## 2. 下部の ShMM を構成します。

- a. 次のように、下部の ShMM で U-Boot 変数を変更します。詳細は、[22 ページ](#)の「環境変数への値の割り当て」を参照してください。

```
setenv ipaddr IP-addr-for-lower-ShMM # Must be different from upper
                                       # ShMM
setenv ipdevice eth0                # Default DO NOT CHANGE
setenv ip1addr 192.168.1.2          # Default DO NOT CHANGE
setenv ip1device usb0               # Default DO NOT CHANGE
setenv gatewayip gateway-ip-address
setenv netmask netmask              # Usually 255.255.255.0
setenv rc_ifconfig y
setenv hostname hostname-lower-ShMM # Optional-different from
                                       # upper ShMM
saveenv                             # Optional - use in U-Boot
                                     # shell to save variables
```



- b. 下部の ShMM で、`/etc/shelfman.conf` ファイルを編集します。
  - i. `PROPAGATE_RMCP_ADDRESS` 変数の値を `FALSE` に変更します。

```
PROPAGATE_RMCP_ADDRESS = FALSE
```

- ii. `RMCP_NET_ADAPTER` を `eth0:1` に変更します。
  - c. アクティブな ShMM から `/etc/readhwaddr` を削除します。
- 3. 両方の ShMM を再起動したあとに、両方の ShMM で `ifconfig eth0` コマンドと `ifconfig eth0:1` コマンドを使用して、変更を検証することができます。  
バックアップの ShMM では、デバイス `eth0:1` が未定義になります。

---

## シェルフマネージャの構成ファイルの設定

シェルフマネージャの構成ファイル (`shelfman.conf`) は、`/etc` ディレクトリにあります。ファイル内の各行は、注釈行 (`#` で始まる) か、構成パラメータの割り当てを表す「名前 = 値」のペアになります。名前と値は、等号 (`=`) で区切られます。

構成パラメータの名前では、大文字と小文字が区別されません。各構成パラメータの型は、ブール型、数値、文字列、IP アドレスのいずれかになります。

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。
数値型	整数値 (符号付きの場合もある)。16 進表記法「 <code>0x...</code> 」もサポートされています。
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 進表記の IP アドレス ( <code>xxx.xxx.xxx.xxx</code> )。

`$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。例:

```
DEFAULT_RMCP_IP_ADDRESS = $IPADDR
```

シェルフマネージャがはじめて起動されたあとに、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。RMCP インタフェースまたは CLI シェルフ外インタフェースを使用すると、LAN 構成パラメータにアクセスしたり、LAN 構成パラメータを変更したりすることができます。また、LAN 構成パラメータは、シェルフマネージャの再起動時に、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 IP アドレスとゲートウェイに対して行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）よりも優先されます。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 に設定して、アドレスをシェルフマネージャ構成ファイルと 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の構成パラメータ

名前	タイプ	デフォルト	説明
2_X_SYSTEM	ブール型	なし	このパラメータを指定すると、現在のシステムが AdvancedTCA として明示的に指定されます (FALSE の場合)。指定しない場合 (TRUE)、システムのタイプが自動的に選択されます。システムタイプに対する誤ったハードウェア検出アルゴリズムを上書きする必要がないかぎり、このパラメータを指定することはお勧めしません。
ACTIVATE_LOCAL_WITHOUT_SHELF_FRU	ブール型	FALSE	TRUE に設定すると、シェルフ FRU 情報が見つからなくても、アクティブなシェルフマネージャー (物理シェルフマネージャーと論理シェルフマネージャーを表す) によって公開される両方の IPM コントローラがアクティブになります。物理シェルフマネージャーの IPM コントローラのペイロードの消費電力は、シェルフ内の対応するスロットの電力容量を超える可能性があるため、このオプションは注意して使用する必要があります。
ALARM_CUTOFF_TIMEOUT	数値型	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)。秒単位で指定します。
ALLOW_ALL_COMMANDS_FROM_IPMB	ブール型	FALSE	TRUE に設定すると、RMCP インタフェースで許可されたほとんどのコマンドが、IPMB-0 でも許可されます (セッション関連のコマンドは除く)。たとえば、「コールドリセット」コマンドやユーザー管理コマンドが IPMB-0 で許可されます。

表 2-2 シェルフマネージャーの構成パラメータ (続き)

名前	タイプ	デフォルト	説明
ALLOW_CHANGE_EVENT_RECEIVER	ブール型	TRUE	TRUE に設定すると、シェルフマネージャーのイベント受信者アドレスを、20h、LUN 0 以外のアドレスに設定することができます。FALSE に設定すると、シェルフマネージャーのイベント受信者アドレスを変更しようとしても、すべて拒否されます。
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。
ALLOW_RESET_STANDALONE	ブール型	FALSE	TRUE に設定すると、シェルフマネージャーに使用可能なバックアップがなくても、「コールドリセット」コマンドが受け付けられ、シェルフマネージャーが再起動されます。デフォルトでは、「コールドリセット」コマンドはデュアル冗長構成の場合にのみ受け付けられます。「コールドリセット」コマンドが受け付けられると、スイッチオーバーが実行されます。
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。
ATCA_TESTER_COMPATIBILITY	ブール型	FALSE	この変数を設定すると、シェルフマネージャーでイベント処理の最適化が無効になるため、シェルフマネージャーの動作は、Polaris ATCA Tester に対応します。
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、そのアドレスに送信される「メッセージの送信」要求に自動的に変換します。
CARRIER	文字列 (16)	PPS	ShMM が取り付けられている特定のキャリアボードの名前。
CARRIER_OPTIONS	文字列 (256)	SUNCT900	キャリア固有のオプション。サポート対象の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。

表 2-2 シェルフマネージャーの構成パラメータ (続き)

名前	タイプ	デフォルト	説明
COOLING_FAN_DECREASE_TIMEOUT	数値型	0	正常な状態で冷却アルゴリズムが動作しているときに、ファン速度を連続的に下げる際の最小タイムアウト値。この値は、COOLING_POLL_TIMEOUT の値の倍数にする必要があります。倍数でない場合は、次の倍数に切り上げられます。パラメータが省略された場合、または 0 に設定された場合、このタイムアウト値は COOLING_POLL_TIMEOUT と同じ値になります。
COOLING_FAN_INCREASE_TIMEOUT	数値型	0	マイナーアラート状態で冷却アルゴリズムが動作しているときに、ファン速度を連続的に上げる際の最小タイムアウト値。この値は、COOLING_POLL_TIMEOUT の値の倍数にする必要があります。倍数でない場合は、次の倍数に切り上げられます。パラメータが省略された場合、または 0 に設定された場合、このタイムアウト値は COOLING_POLL_TIMEOUT と同じ値になります。
COOLING_IGNORE_LOCAL_CONTROL	ブール型	FALSE	ファンに対してローカル制御機能を使用しないでください。ファンレベルはシェルフマネージャーによって明示的に管理されます。
COOLING_MANAGEMENT	文字列 (64)	"" (未定義)	指定する場合は、冷却管理を実装する共有ライブラリの名前を指定します。実際のライブラリ名は、libcooling_<xxx>.so になります。<xxx> は、この構成パラメータの値です。このライブラリは、シェルフマネージャーによって動的に読み込まれ、/var/bin または /lib に格納されます。
COOLING_POLL_TIMEOUT	数値型	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大時間 (秒単位)。
CPLD_ACTIVE_WORKAROUND	ブール型	該当なし	該当なし
CTCA_FRU_RESET_TIMEOUT	数値型	該当なし	該当なし
CTCA_HEALTHY_TIMEOUT	数値型	該当なし	該当なし
CTCA_INITIAL_FAN_LEVEL	数値型	該当なし	該当なし

表 2-2 シェルフマネージャーの構成パラメータ (続き)

名前	タイプ	デフォルト	説明
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、対応するパラメータが 0.0.0.0 に設定されている場合に、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 構成パラメータにゼロ以外のゲートウェイ IP アドレスが設定されている場合、シェルフマネージャーの構成ファイルで設定されている値は無視されます。
DEFAULT_GATEWAY_IP_ADDRESS2	IP アドレス	なし	チャンネル 2 の IPMI LAN 構成パラメータで、対応するパラメータが 0.0.0.0 に設定されている場合に、二次ネットワークインタフェース上のシェルフ外 (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 構成パラメータにゼロ以外のゲートウェイ IP アドレスが設定されている場合、シェルフマネージャーの構成ファイルで設定されている値は無視されます。デフォルトでは未定義です。このパラメータは、USE_SECOND_CHANNEL の値が TRUE の場合にのみ使用されます。
DEFAULT_RMCP_IP_ADDRESS	IP アドレス	なし	シェルフ外 (RMCP ベース) 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。この IP アドレスは、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、対応するパラメータが 0.0.0.0 に設定されている場合にのみ使用されます。LAN 構成パラメータやシェルフ FRU 情報にゼロ以外の IP アドレスが設定されている場合、シェルフマネージャーの構成ファイルで設定されている値は無視されます。
DEFAULT_RMCP_IP_ADDRESS2	IP アドレス	なし	二次ネットワークインタフェースでのシェルフ外 (RMCP ベース) 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。この IP アドレスは、チャンネル 2 の IPMI LAN 構成パラメータで、対応するパラメータが 0.0.0.0 に設定されている場合にのみ使用されます。LAN 構成パラメータにゼロ以外の IP アドレスが設定されている場合、シェルフマネージャーの構成ファイルで設定されている値は無視されます。

表 2-2 シェルフマネージャーの構成パラメータ (続き)

名前	タイプ	デフォルト	説明
DEFAULT_RMCP_NETMASK	IP アドレス	変数	RMCP 通信に使用されるネットワークアダプタのネットワークマスク。このマスクは、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、対応するパラメータが 0.0.0.0 に設定されている場合にのみ使用されます。デフォルト値は、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 アドレスのクラスによって異なります (パラメータ DEFAULT_RMCP_IP_ADDRESS を参照)。たとえば、クラス C の IP アドレスの場合、このパラメータは 255.255.255.0 に設定されます。
DEFAULT_RMCP_NETMASK2	IP アドレス	変数	RMCP 通信に使用される二次ネットワークアダプタのネットワークマスク。このマスクは、チャンネル 2 の IPMI LAN 構成パラメータで、対応するパラメータが 0.0.0.0 に設定されている場合にのみ使用されます。デフォルト値は、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 アドレスのクラスによって異なります (パラメータ DEFAULT_RMCP_IP_ADDRESS2 を参照)。たとえば、クラス C の IP アドレスの場合、このパラメータは 255.255.255.0 に設定されます。
DEFAULT_VLAN_ID	数値型	0	一次 LAN チャンネルに使用されるデフォルトの仮想 LAN ID。この値は、対応する LAN 構成パラメータのデフォルト値としてのみ使用されます。LAN 構成パラメータを定義して ShMM に保存すると、シェルフマネージャーの構成ファイルで設定されている値は無視されるようになります。
DEFAULT_VLAN_ID2	数値型	0	二次 LAN チャンネルに使用されるデフォルトの仮想 LAN ID。この値は、対応する LAN 構成パラメータのデフォルト値としてのみ使用されます。LAN 構成パラメータを定義して ShMM に保存すると、シェルフマネージャーの構成ファイルで設定されている値は無視されるようになります。

表 2-2 シェルフマネージャーの構成パラメータ (続き)

名前	タイプ	デフォルト	説明
DETECT_DEADLOCKS	ブール型	TRUE	この変数は、シェルフマネージャーの CLI および RMCP のサーバー機能のデッドロック検出を有効にします。検出は、CLI 要求および RMCP 要求を処理するスレッドによって定期的にストローブされる必要のある内部のウォッチドッグに基づいて行われます。いずれかのスレッドが内部のウォッチドッグのストローブに失敗すると、実際のウォッチドッグはストローブされず、最終的には ShMM がリセットされ、バックアップのシェルフマネージャーへのフェイルオーバーが開始されます。さらに、デッドロックが存在するかどうかを調べるために、ロックデータの構造が定期的に直接確認されます。
DEVICE_POLL_TIMEOUT	数値型	10 秒	シェルフマネージャーが「デバイス ID の取得」コマンドを使用して IPMB デバイスに連続ポーリングを行う際の間隔 (秒単位)。
DHCP_FOR_RMCP_ONLY	ブール型	FALSE	この変数が設定されると、DHCP メカニズムによって RMCP IP アドレスのみが割り当てられます。ShMM のプライベート IP アドレスは DHCP の影響を受けません。
DHCP_SERVER_ADDRESS	IP アドレス	なし	このパラメータは DHCP サーバーの IP アドレスです。変数 USE_DHCP が TRUE の場合にのみ適用されます。このパラメータが省略されたか 0.0.0.0 に設定されており、USE_DHCP 変数が TRUE の場合、シェルフマネージャーは、ブロードキャスト検出要求に応答する DHCP サーバーからのアドレス情報を受け入れます。

表 2-2 シェルフマネージャーの構成パラメータ (続き)

名前	タイプ	デフォルト	説明
ENABLE_DIRECT_SHELF_FRU_WRITE	ブール型	FALSE	この変数は、IPMI の「FRU データの書き込み」コマンドを使用して、シェルフマネージャーでシェルフ FRU 情報 (IPMB-0 の 20h にある IPM コントローラ上の FRU デバイス ID #254) への直接書き込みを許可するかどうかを制御します。デフォルトでは直接書き込みは禁止されています (PICMG 3.0 R2.0 ECN-002 により、シェルフ FRU 情報への書き込みについては、特別なロックングプロトコルを使用し、一度に 1 人のユーザーのみがアクティブになるように定められています)。この変数を TRUE に設定すると、ECN-002 以前の動作に準拠したシステムマネージャーのアプリケーションとの互換性を維持するために、直接書き込みが有効になります。
EXIT_IF_HEALTHY_LOST_IN_STANDALONE_MODE	ブール型	FALSE	この変数は、シェルフマネージャーがバックアップなしで実行されているときに、正常なローカルビットがないことが検出された場合の動作を定義します。この変数が TRUE の場合、シェルフマネージャーが終了し、ShMM とシェルフマネージャーが再起動します。この変数が FALSE の場合、シェルフマネージャーによって正常なビットが設定され、処理が続行します。
EXIT_IF_NO_SHELF_FRU	ブール型	FALSE	TRUE に設定すると、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します (通常、ShMM がリセットされます)。
EXTERNAL_EVENT_HANDLER	文字列 (255)	""	PEF を介してイベントのローカル処理を行うシェルフマネージャー上の実行可能ファイル (またはスクリプトファイル) のパスです。
FAN_FULL_SPEED_DELAY	数値型	0	シェルフマネージャーの起動後、またはスイッチオーバー後の遅延時間 (秒単位)。この間、冷却アルゴリズムでは現在のファントレイ数を確認しないため、既存のファントレイがアクティブになるのに十分な時間が与えられます。このパラメータは、シェルフ内のファントレイの実際の数に指定されている数よりも少ない場合に、冷却管理によってファン速度が最大にされるキャリアに適用されます。



表 2-2 シェルフマネージャーの構成パラメータ (続き)

名前	タイプ	デフォルト	説明
FAN_LEVEL_STEP_DOWN	数値型	1	正常な状態で冷却アルゴリズムが動作しているときに、ファン速度を下げる際のファンのステップ数。このパラメータは、ShMM キャリア固有の冷却アルゴリズムに置き換えられる場合があります。
FAN_LEVEL_STEP_UP	数値型	1	マイナーアラート状態で冷却アルゴリズムが動作しているときに、ファン速度を上げる際のファンのステップ数。このパラメータは、キャリア固有の冷却アルゴリズムに置き換えられる場合があります。
HPDL	ブール型	FALSE	シェルフマネージャーで Hardware Platform Description Language (HPDL) のサポートを有効にします。キャリアとシャーシの HPDL データおよび SDR は、FRU 情報またはファイルから取得され、プラットフォームの動作だけでなく、管理対象の FRU およびセンサーの数とタイプを定義するために使用されます。
HPDL_ON_SUBSIDIARY_FRUS	ブール型	FALSE	補助の FRU に格納されている HPDL 情報のサポートを有効にします。TRUE の場合、シェルフマネージャーは、補助の FRU の FRU 情報で、HPDL データと SDR を検索します。特定の FRU でデータが見つかった場合、キャリアまたはシャーシの HPDL データや SDR によるその FRU の定義が、それらのデータを使用して置き換えられます。
IGNORE_FAILED_DIRECTED_POWER_DOWN	ブール型	TRUE	このパラメータは、クリティカルアラートが発生した場合のボードの電源切断障害を無視し、シャーシ内のすべてのボードの電源を切断しないようにシェルフマネージャーに指示します。また、このパラメータは、メジャーアラートが発生した場合のボードの電力レベル低下障害を無視し、シャーシ内のすべてのボードの電力レベルを下げないようにシェルフマネージャーに指示します。このパラメータを FALSE に設定すると、クリティカルアラート冷却状態を引き起こしたボードの電源が切断されない場合、シャーシ内のすべてのボードの電源が切断されます。また、メジャーアラート冷却状態を引き起こしたボードの電力レベルが下がらない場合、シャーシ内のすべてのボードの電力レベルが下げられます。

表 2-2 シェルフマネージャの構成パラメータ (続き)

名前	タイプ	デフォルト	説明
INITIAL_FAN_LEVEL	数値型	5	シェルフマネージャがファントレーに適用する初期のファンレベル。通常、ファンレベルの値は 0 - 15 の範囲になります。0 は最小ファン速度、15 は最大ファン速度を示します。
INITIAL_SLOW_LINK_DELAY	数値型	100	シェルフマネージャが、シェルフマネージャとシステムマネージャの間の物理ネットワークリンク (RMCP リンク。構成パラメータ SWITCHOVER_ON_BROKEN_LINK の説明を参照) の整合性テストを開始するまでの初期遅延時間 (秒単位)。シェルフ電源の投入時の初期化に長時間かかる低速のネットワークリンクに対処するには、遅延時間をゼロ以外にします。
INNER_SEQUENCE_NUMBER IN_SEND_MSG_RESPONSE	ブール型	TRUE	この変数は、LAN から IPMB にブリッジされた「メッセージの送信」コマンドへの応答に使用されるシーケンス番号を制御します。TRUE の場合、「メッセージの送信」要求にカプセル化されたコマンドのシーケンス番号が使用されます。FALSE の場合、「メッセージの送信」要求自体のシーケンス番号が使用されます。最新バージョンの PICMG 3.0 仕様に記述されている説明によると、最初のバリエントが正しいこととなりますが、シェルフマネージャではこれまで 2 番目のバリエントが使用されていました。 従来の動作は、値を FALSE に設定することで復元することができます。
IPMB_ADDRESS	数値型	0	シェルフマネージャの IPMB アドレス。このアドレスで、ハードウェアアドレスを上書きします。値を 0 に設定すると、シェルフマネージャ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_LINK_ISOLATION_TIMEOUT	数値型	-1	放射線状のシェルフで、分離アルゴリズムによって IPMB リンクが無効化されている場合、この間隔 (秒単位) の後にリンクが自動的に有効化されます。-1 (デフォルト) は、「タイムアウトなし」を表します。
IPMB_RETRIES	数値型	3	要求に対する応答がない場合に、IPMB 要求の再送信を試みる回数。この回数に達すると、再送信を停止します。

表 2-2 シェルフマネージャーの構成パラメータ (続き)

名前	タイプ	デフォルト	説明
IPMB_RETRY_TIMEOUT	数値型	4 秒	シェルフマネージャーが、IPMB 要求を送信してから次に要求を再送信するまでに応答を待機する時間。
IPMB_RETRY_TIMEOUT_MSEC	数値型	0	再試行タイムアウトの値のミリ秒部分。再試行タイムアウト値が 1 秒よりも短い場合は、この構成変数に実際のタイムアウト値が含まれます。一方、構成変数 IPMB_RETRY_TIMEOUT の値は 0 になります。
IPMC_PRESERVE_ON_REVISION_CHANGE	ブール型	TRUE	この変数を TRUE に設定すると、ファームウェアや補助ファームウェアのバージョン情報だけが変更された場合、ファームウェアのアップグレード後もシェルフマネージャーで IPM コントローラの ID が保持されます。この変数を FALSE に設定すると、シェルフマネージャーの通常のポーリング中に「デバイス ID の取得」応答データが変更された場合、別の IPM コントローラが存在していることを示しているようにみなされます。
ISOLATE_MUX_ADDRESS	数値型	0x70	7 ビット I <sup>2</sup> C マルチプレクサのアドレス (シェルフマネージャーからマルチプレクサ (マスター専用の I <sup>2</sup> C バス上) へのアクセスを SHMM_GPIO8 を使用して制御しているプラットフォーム上のアドレス)。
ISOLATE_MUX_IGNORE_COUNT	数値型	10	このパラメータは、障害のあるバスを (シェルフマネージャーからマルチプレクサ (マスター専用の I <sup>2</sup> C バス上) へのアクセスを SHMM_GPIO8 を使用して制御しているプラットフォーム上で) 有効にしようとする前に、そのバスへのアクセスをこの回数だけスキップするように分離アルゴリズムに指示します。
ISOLATE_MUX_ON_GPIO8	ブール型	FALSE	シェルフマネージャーからマルチプレクサ (マスター専用の I <sup>2</sup> C バス上) へのアクセスを SHMM_GPIO8 を使用して制御しているプラットフォームでは、TRUE に設定する必要があります。
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。

表 2-2 シェルフマネージャーの構成パラメータ (続き)

名前	タイプ	デフォルト	説明
M7_TIMEOUT	数値型	-1 (秒) または 600 (ShMM バージョン 2.4.9-R3U2 以降)。	FRU が M7 状態を維持できる最大時間 (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、タイムアウトなしを表します。このパラメータを 0 に設定すると、FRU が M7 状態に移行しないようにすることができます。
MAX_ALERT_POLICIES	数値型	64	使用できる PEF アラートポリシーの最大数。
MAX_ALERT_STRINGS	数値型	64	使用できる PEF アラート文字列の最大数。
MAX_ATCA_FANLEVEL	数値型	8	NEBS 方針が有効な場合に使用する最大ファンレベルを指定します。
MAX_ATCA_TEMP	数値型	25	NEBS 方針を有効にする場合の吸気温度の上限を指定します。
MAX_CMD_SUBSCRIBER_IDLE_TIME	数値型	60	このパラメータでは、コマンド加入者がソケットから通知を読み取る際の最大タイムアウト値 (秒単位) を定義します。このタイムアウトを超えると、加入者は応答なしとみなされ、自動的に登録が解除されます。
MAX_DEFERRED_ALERTS	数値型	32	未処理の PEF アラートの最大数。
MAX_EVENT_FILTERS	数値型	64	使用できる PEF イベントフィルタの最大数。
MAX_EVENT_SUBSCRIBERS	数値型	64	シェルフマネージャー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	数値型	60 秒	イベントが届いてから、加入者がシェルフマネージャーからこのイベントを取得するまでの、イベント加入者に対する最大タイムアウト (秒単位)。このタイムアウトを超えると、加入者は応答なしとみなされ、自動的に登録が解除されます。
MAX_INCOMING_IPMB_REQUESTS	数値型	128	受信 IPMB 要求用の内部シェルフマネージャーキューのサイズ。受信 IPMB 要求は処理されるまでこのキューに格納されます。
MAX_NODE_BUSY_RETRANSMISSIONS	数値型	255	受信者が応答で常に完了コード Node Busy を返す場合の IPMB コマンドの最大再送回数。
MAX_OEM_FILTERS	数値型	16	OEM タイプのシステムイベントログ (SEL) エントリを処理するための特殊な PEF イベントフィルタの最大数。
MAX_PENDING_CMD_NOTIFICATIONS	数値型	32	このパラメータは、コマンド保留通知の最大数を指定します。

表 2-2 シェルフマネージャーの構成パラメータ (続き)

名前	タイプ	デフォルト	説明
MAX_PENDING_EVENT_NOTIFICATIONS	数値型	1024	アクティブな各加入者の未処理のイベント通知の最大数。
MAX_PENDING_IPMB_REQUESTS	数値型	192	応答を待機している保留中の IPMB 要求の最大数。
MAX_SEL_ENTRIES	数値型	1024	システムイベントログ (SEL) のエントリの最大数。
MAX_SESSIONS	数値型	32	同時 IPMI セッションの最大数。
MAX_USERS	数値型	32	IPMI ユーザーの最大数。
MICRO_TCA	ブール型	FALSE	TRUE の場合、シェルフマネージャーは MicroTCA シェルフマネージャーとして動作します (キャリアマネージャーとのやり取りには、2 つ目の RMCP チャネルが使用されます)。
MIN_FAN_LEVEL	数値型	1	最小ファンレベル。ファンレベルが自動的に制御されている場合、冷却管理コードでいずれかのファンのファンレベルをこの値よりも低い値にすることはできません。
MIN_SHELF_FRUS	数値型	2	シェルフマネージャーが正しく起動するために検出する必要のあるシェルフ内のシェルフ FRU の最小数。
NORMAL_STABLE_TIME	数値型	3600	通常モードのシェルフマネージャーで、動的に検出される最小ファンレベル (熱警告が生成されない最小ファンレベル) が保持される時間 (秒単位)。この時間が経過すると、冷却アルゴリズムは最小ファンレベルを下げて、可能であれば、シェルフの熱負荷も低下している場合にシェルフでファンレベルを下げるができるようにします。
NOTIFY_POLL_PERIOD_NSEC	数値型	1000000	このパラメータは、内部キュー内の新しい通知の通知スレッドにおける以降のチェックのための時間 (ナノ秒) を設定します。デフォルト値は 1000000 (1 ミリ秒) です。

表 2-2 シェルフマネージャーの構成パラメータ (続き)

名前	タイプ	デフォルト	説明
PET_FORMAT	数値型	0	Platform Event Filtering 機能のイベント処理によって開始される警告アクションとしてシェルフマネージャーから送信される Platform EvenTrap の形式を指定します。値は、次のように定義されています。 <ul style="list-style-type: none"> <li>• 0: IPMI Platform Event Trap Format v1.0 仕様で定義されているデフォルトの IPMI 形式。</li> <li>• 1: プレーンテキスト形式。イベントの詳細はすべて単一の変数でプレーンの ASCII テキストとして送信されます。</li> <li>• 2: 多変数形式。各イベントフィールドはそれぞれ個別の変数としてエンコードされます。</li> </ul>
PHYSICAL_SENSORS	ブール型	TRUE	ADM1026 および LM75 チップでホストされ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投入できます。
PROPAGATE_RMCP_ADDRESS	ブール型	FALSE	TRUE の場合、アクティブなシェルフマネージャーによって RMCP IP アドレスがバックアップのシェルフマネージャーに伝達されます。バックアップのシェルフマネージャーでは、その IP アドレスを使用して (ただし、最下位ビットを反転させて)、RMCP_NET_ADAPTER 変数で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冗長モードで実行します。2 枚のシェルフ管理カードが搭載されたシステムでは TRUE に設定する必要があります。
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 2 つ目のネットワークアダプタの名前 (この目的でデュアル USB ネットワークインタフェースが使用されている場合)。

表 2-2 シェルフマネージャーの構成パラメータ (続き)

名前	タイプ	デフォルト	説明
REDUNDANCY_NETMASK	数値型	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 の場合) では、ネットマスクは IP アドレスのクラスに基づいて自動的に決定されます。
REDUNDANCY_PORT	数値型	1040	シェルフマネージャーの冗長インスタンス間のやり取りに使用される 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。このアドレスでは、実際には最下位ビットのみ異なる対の IP アドレスを指定します。これらのアドレスは、ハードウェアアドレスに従って冗長なシェルフマネージャーに割り当てられます。
RESERVATION_RETRIES	数値型	10	シェルフマネージャーで「デバイス SDR の予約」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	クロス接続リンクがハードウェアでサポートされる場合に、RMCP ベースの通信に使用される回復したネットワークアダプタの名前。
RMCP_WITHOUT_SHELF_FRU	ブール型	FALSE	この設定パラメータが定義されて TRUE に設定されている場合に限り、シェルフ FRU がいないときに RMCP を使用できます。
SDR_READ_RETRIES	数値型	3	シェルフマネージャーで「デバイス SDR の読み取り」コマンドを再試行する最大回数。
SEL_HIGH_WATERMARK	数値型	0	SEL の自動消去を制御するアルゴリズムの高位境界値。SEL 内の空きエントリの実際の割合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、SEL から古い記録を古い順に削除するスレッドがシェルフマネージャーによって開始されます。
SEL_LOW_WATERMARK	数値型	0	SEL の自動消去を制御するアルゴリズムの低位境界値。SEL から古い記録を消去するスレッドが開始されると、SEL におけるエントリの割合がこの値を下回るまで記録が消去されます。
SENSOR_POLL_INTERVAL	数値型	1	シェルフマネージャーによるローカルのシェルフマネージャーセンサーの連続するポーリングの間隔 (秒単位)。

表 2-2 シェルフマネージャーの構成パラメータ (続き)

名前	タイプ	デフォルト	説明
SESSION_SEQUENCE_WINDOW	数値型	128	受入可能な RMCP シーケンス番号のウィンドウ。このウィンドウが広くなれば、転送中にドロップされる RMCP パケットに対するシェルフマネージャーの許容度が高くなります。受信したパケットと以前のパケットのシーケンス番号の差がこのウィンドウサイズを超えると、RMCP セッションは破壊されたセッションとしてシェルフマネージャーにより閉じます。
SHELF_FRU_IN_EEPROM	ブール型	TRUE	TRUE の場合、シェルフ FRU 情報はキャリア固有の方法でバックプレーン上の EEPROM から取得されます。FALSE の場合、シェルフ FRU 情報はフラッシュファイルシステム上のファイルから取得されます。
SHELF_FRU_IPMB_SOURCE1	数値型	0	定義されている場合 (0 以外の値の場合)、シェルフ内のシェルフ FRU 情報の中で最初に指定されたソースの IPMB アドレスを指定します (シェルフ FRU は FRU 1 にあります)。この値が定義されている場合、IPMB 上のシェルフ FRU の検索は指定されているソースのみに制限されます。
SHELF_FRU_IPMB_SOURCE2	数値型	0	定義されている場合 (0 以外の値の場合)、シェルフ内のシェルフ FRU 情報の中で 2 番目に指定されたソースの IPMB アドレスを指定します (シェルフ FRU は FRU 1 にあります)。この値が定義されている場合、IPMB 上のシェルフ FRU の検索は指定されているソースのみに制限されます。
SHELF_FRU_TIMEOUT	数値型	5 秒	初期化時に、シェルフマネージャーがシェルフ FRU 情報デバイスの検出を待機する間隔。
SHORT_SEND_MSG_RESPONSE	ブール型	TRUE	シェルフマネージャーから提供される「メッセージの送信」応答のタイプを決定します。PICMG 3.0 ECR で必要であるか (TRUE の場合)、シェルフマネージャーの前のバージョンと互換します (FALSE の場合)。
SWAPPED_CROSS_CONNECTS	ブール型	FALSE	奇数のハードウェアアドレスを持つ ShMM でクロス接続に使用されるネットワークアダプタの名前をスワップします。



表 2-2 シェルフマネージャーの構成パラメータ (続き)

名前	タイプ	デフォルト	説明
SWITCHOVER_ON_HANDLE_OPEN	ブール型	FALSE	TRUE の場合、シェルフマネージャーのスイッチオーバー関連の動作は、次のようにホットスワップハンドルの状態の影響を受けます。 <ul style="list-style-type: none"> <li>• ホットスワップハンドルが開いているためにアクティブなシェルフマネージャーが M1 状態に移行すると、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が開始されます。</li> <li>• アクティブなシェルフマネージャーが M5 状態に移行し、使用できるバックアップシェルフマネージャーがない場合、アクティブなシェルフマネージャーは停止されず、無制限に M5 の状態のままになります。</li> </ul>
SWITCHOVER_TIMEOUT_ON_BROKEN_LINK	数値型	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間の物理ネットワークリンク (RMCP リンク) が壊れた場合にシェルフマネージャーでスイッチオーバーを開始するかどうかとそのタイミングに影響します。少なくともこのパラメータで指定した時間 (秒単位) リンクが壊れたままであれば、スイッチオーバーが行われます。タイムアウト期間内にリンクが復元されると、スイッチオーバーは実行されません。このパラメータの値が -1 の場合、壊れた RMCP リンクで自動スイッチオーバーは実行されません。
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。
SYSTEM_MANAGER_TRUNCATES_SEL	ブール型	FALSE	TRUE の場合、SEL を自動的に切り捨てるシェルフマネージャーのアルゴリズムが無効になります。システムマネージャーは、シェルフマネージャーで「イベントログ使用不可」タイプのセンサーの SEL 状態の値を監視し、「SEL エントリの削除」コマンドをシェルフマネージャーに送信して SEL からイベントを削除し、SEL を切り捨てます。
TACHOMETER_THRESHOLD_UPDATE_DELAY	数値型	15	動的なファン回転速度計しきい値がサポートされるシェルフでのファンレベルの設定とファン回転速度計しきい値の更新間の遅延を制御します。

表 2-2 シェルフマネージャの構成パラメータ (続き)

名前	タイプ	デフォルト	説明
TASKLET_RETRIES	数値型	3	最終的に中断されるまで各シェルフマネージャのタスクレット (アクティブ化、非アクティブ化、情報の取得) が再試行される回数。
TURBO_MODE_MIN_FAN_FAILURES	数値型	1	このパラメータは、デフォルトの HPDL 冷却管理アルゴリズムで残りのファンが TURBO モードになる (フルスピードで実行される) 回転速度計低速障害 (メジャーしきい値または重大なしきい値を超えた場合) の数を指定します。シェルフにゾーン冷却が実装されている場合、指定された数の回転速度計障害がゾーンごとにカウントされます。また、TURBO モードの設定は、対応するゾーンの冷却に関与するファントレーにのみ影響します。
UNCONDITIONAL_SDR_REREAD_ON_VERSION_CHANGE	ブール型	FALSE	値が TRUE の場合、シェルフマネージャが IPM コントローラからバージョンの変更イベントを受信すると、「デバイス SDR 情報の取得」応答のセンサー生成変更インジケータが変更されない場合でも、そのコントローラから SDR が無条件に再読み取りされます。これは、この点において完全に IPMI に準拠していない ATCA ボードのために行われます。この変数のデフォルト値は FALSE です。
USE_DHCP	ブール型	FALSE	DHCP サーバーに対して、シェルフマネージャに対する RMCP にアクセス可能な IP アドレスとプライベート IP アドレスの割り当てを要求します。設定パラメータ PREFERRED_DHCP_SERVER を使用すると、優先 DHCP サーバーの IP アドレスを指定できます。
USE_SECOND_CHANNEL	ブール型	FALSE	このパラメータは、ShMM 上の 2 つのネットワークインタフェースが RMCP 通信に使用される場合にのみ適用されます。TRUE の場合、ShMM 上のネットワークインタフェースが並列モードで使用されます。FALSE の場合、冗長モードで使用されます。
VERBOSITY	数値型	7	シェルフマネージャの詳細レベル。デフォルトの詳細レベルは 7 で、エラーメッセージ、警告メッセージ、および情報メッセージです。詳細は、63 ページの「 <a href="#">詳細レベルの説明</a> 」を参照してください。

表 2-2 シェルフマネージャーの構成パラメータ (続き)

名前	タイプ	デフォルト	説明
VERBOSITY_CONSOLE	数値型	該当なし	シェルフマネージャーのコンソール出力の詳細レベル。
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 に設定すると、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 で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 が最初に起動したときに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では、U-Boot で設定された次の環境変数がインポートされます。

\$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
\$IPLADDR	デフォルトの冗長 IP アドレス
\$IPLDEVICE	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ゲートウェイ

環境変数 \$CARRIER および \$CARRIER\_OPTIONS は、二次 RC スクリプトで設定されます。このキャリア固有の起動スクリプトの名前は、U-Boot 環境変数 rc2 で定義されます。シェルフマネージャーは、必要に応じて出荷時のデフォルトパラメータ値にリセットできます (122 ページの「ファイルシステムの再初期化」を参照)。

デフォルトの構成ファイルのコピーを例 2-1 に示します。

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
#
# CARRIER_OPTIONS: This parameter specifies the carrier-specific options.
# Default is an empty string.
```

例 2-1 デフォルトの shelfman.conf ファイル (続き)

```
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. デフォルト
# interval is 600 seconds.
#
ALARM_CUTOFF_TIMEOUT = 600

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

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

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

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

例 2-1 デフォルトの shelfman.conf ファイル (続き)

```
# 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

# MAX_OEM_FILTERS: This parameter specifies the number of available entries
```

例 2-1 デフォルトの shelfman.conf ファイル (続き)

```
# 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
# 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
```

例 2-1 デフォルトの shelfman.conf ファイル (続き)

```
# 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.
#
RESERVATION_RETRIES = 10
```

例 2-1 デフォルトの shelfman.conf ファイル (続き)

```
# 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 EEPROMs 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. ポート
# default is TRUE.
#
VERIFY_SHELF_FRU_CHECKSUM = TRUE
```



例 2-1 デフォルトの shelfman.conf ファイル (続き)

```
# 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

# RMCP_NET_ADAPTER: This parameter specifies the name of network adapter
# used for RMCP-based communications. Default is eth0:1 if it does not
```

例 2-1 デフォルトの shelfman.conf ファイル (続き)

```
# 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.
# With a -1 value, no automatic switchovers on RMCP link faults will occur.
# The default value is 10 second.
```

例 2-1 デフォルトの shelfman.conf ファイル (続き)

```
#
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
# msecs which is used to holds the BD_SEL# line low in order to reset a
# CompactPCI board. Default is 500 milliseconds.
#
# CTCA_FRU_RESET_TIMEOUT = 500
```

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

```
# 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
```

## キャリア固有の構成ファイル

共通の構成ファイル `/etc/shelfman.conf` が読み取られたあとに、キャリア固有の構成ファイル `/etc/shelfman.conf.carrier-name` がシェルフマネージャーによって読み取られます。`carrier-name` は、関連のシェルフで使用される ShMM キャリアの名前で、小文字で指定します。キャリア固有の構成ファイルの設定は、共通の構成ファイルの同じ変数の設定よりも優先されます。

このメカニズムにより、キャリア固有の基準で共通設定を再定義することができます。通常、キャリア固有のファイルでは少数の重大な構成変数のみ定義されます。たとえば、`MIN_FAN_LEVEL` パラメータの適切な値は、シェルフアーキテクチャーとそのアーキテクチャーに実装されるファン機能によって決まることがあります。このメカニズムでは、このようなシェルフ固有の制約を適用できます。

このメカニズムの結果、キャリア固有の構成ファイルで指定される設定パラメータの有効な値を変更するためには、変更をその構成ファイルで行う必要があります。共通の構成ファイルでこのような変数を変更しても、何の効果もありません。



---

**注意** – `shelfman.conf.acb` ファイルの変数は変更しないでください。これらの設定には、キャリア固有の設定が含まれ、`/etc/shelfman.conf` ファイルの値よりも優先されます。これらのシステム固有の変数を変更すると、システムが不安定になる可能性があります。

2つの例外は、`SWITCHOVER_TIMEOUT_ON_BROKEN_LINK` 変数と `PROPAGATE_RMCP_ADDRESS` 変数です。R3U1 よりも前のリリースでこれらの変数を有効にするには、`shelfman.conf.acb` ファイルで変更する必要があります。リリース R3U1 以降では、これらの変数は `shelfman.conf.acb` ファイルから削除されているため、`/etc/shelfman.conf` ファイルで変更する必要があります。

---

## 詳細レベルの説明

詳細レベルでは、設定パラメータ `CONSOLE_LOGGING_ENABLED` および `SYSLOG_LOGGING_ENABLED` の設定方法に応じて、追加の出力をコンソールまたは Syslog に送信できます。`VERBOSITY` 設定パラメータは、16 進数のビットマスクで、各ビットは特定のタイプのメッセージの出力を有効にします。

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 (推奨しません)
0x20	詳細トレースメッセージ
0x40	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ついて表示されるメッセージ (推奨しません)
0x80	内部ロックの取得と解放に関する詳細メッセージ (推奨しません)

デフォルトのデバッグレベルは 7 で、エラーメッセージ、警告メッセージ、および情報メッセージを出力できます。

## 日付と時刻の設定

システムを初めて起動するときに、クロックは設定されないため、初期化する必要があります。初期設定で、クロックは、1970年1月1日に設定されています。日付には、シリアルコンソールからアクセスできます。

```
# 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 *MMDDHHmmCCYY.ss* です。各表記の意味は次のとおりです。

<i>MM</i>	月
<i>DD</i>	日
<i>HH</i>	時間 (24 時間表記を使用)
<i>mm</i>	分
<i>CC</i>	世紀
<i>YY</i>	年
<i>.ss</i>	秒

例:

```
# date 092916282007.10
Sat Sep 29 16:28:00 UTC 2007
```

日付を固定するには、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。

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

このエラーは無視してかまいません。これは元の日付が初期化されていない状態であるために表示されます。

## タイムサーバーからの日付と時刻の取得

シェルフ管理カードに、リアルタイムクロック (RTC) バッテリーが搭載されていない場合は、システムの起動時にタイムサーバーからシステムの日付と時刻を取得し、その後は定期的に同期することができます。選択したタイムサーバーは、`rdate` ユーティリティに必要とされるように TCP を介して RFC 868 をサポートする必要があります。この機能を有効にするには、U-Boot 変数 `time_server` を定義する必要があります。また、オプションで追加の変数 `timezone` を定義できます。

`time_server` 変数には、シェルフマネージャーが起動後にシステムの時刻をクエリーするタイムサーバーの IP アドレスが格納されます。この変数は、環境変数 `TIMESERVER` として Linux レベルに伝達されます。この変数を設定すると、起動スクリプト `/etc/netconfig` でスクリプト `/etc/timesync` がデーモンとして実行されます。デーモンは永久ループで実行され、タイムサーバーに対してデフォルト間隔の 300 秒でクエリーが実行されます。この間隔を変更するには、スクリプト `/etc/timesync` を編集し、変数 `INTERVAL` の値を変更します。

---

**注** - `time_server` 変数を指定する場合は、正しく同期するために、`ip1device` 変数を `usb0` に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、そのあとにグリニッジ標準時 (GMT) からのオフセットが格納されます。オフセットは、グリニッジの西側のタイムゾーンの場合は正の値になり、グリニッジの東側のタイムゾーンの場合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である `UTC0` は協定世界時 (UTC) で、グリニッジ時刻と一致します。

タイムサーバーから送信される時刻は 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や、正しく設定されていない場合、タイムゾーンから取得される時刻は正しく解釈されません。タイムゾーンの 3 文字の名前はシェルフマネージャーで使用されませんが、Linux のタイムゾーンを設定するために伝達されます (たとえば、タイムゾーン名 `XXX0` を使用すると、`date` コマンドでは、`Thu Sep 9 21:24:24 XXX 2004` のような出力が生成されます)。

米国の東部時間における `timezone` 定義の例を次に示します。

```
timezone = EST5
```

この例の数字 5 はタイムゾーンが GMT の 5 時間西であることを示します。EST は任意の 3 文字で置き換えることができます。これらの文字を使用して、Linux の `date` コマンド出力でタイムゾーンを指定します。

## 夏時間のタイムゾーンの設定

timezone 変数を夏時間用に設定するには、U-Boot の timezone 変数を夏時間に合わせて設定する必要があります。timezone 変数は、Linux レベルに環境変数 TZ として伝達されて、TZ Linux 変数の形式と似たような形式を使用します。

### ▼ 夏時間のタイムゾーンを設定する

1. U-Boot プロンプトで任意のキーを押して自動起動プロセスを停止します。

```
U-Boot 1.x.x (Nov 11 2008 - 11:32:08)
:
:
Hit any key to stop autoboot: 0
```

2. timezone 変数を設定します (PST/PDT など)。

```
shmm500 setenv timezone PST8PDT,M3.2.0/2,M11.1.0/2
```

3. 新しいタイムゾーン値を保存します。

```
shmm500 saveenv
```

4. ShMM をリセットします。

```
shmm500 reset
```

---

## シェルフ管理カードでのユーザーアカウントの設定

RMCP アクセスのためのユーザーアカウントは、シェルフマネージャーの CLI を使用して設定されます。ユーザー情報は、アクティブなシェルフ管理カードに入力され、すぐにスタンバイシェルフ管理カードでミラー化されるか、共有されます。シェルフ管理カードは、パスワードを使用する 32 のアカウントをサポートします。



## ▼ RMCP アクセスのためのユーザーアカウントを追加する

1. アクティブなシェルフ管理カードにログインします。
2. ユーザーを追加します。

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

各変数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 コマンドの最初のバイト (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 - コールバックに制限される

*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)

次の例は、ユーザー名を root、特権レベルを管理者、パスワードを PICMG guru とするユーザー 9 を追加する方法を示しています。

```
# clia user add 9 "root" 0x40 4 "PICMG guru"  
Pigeon Point Shelf Manager Command Line Interpreter  
User 9 added successfully  
#  
# clia user  
Pigeon Point 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"  
#
```

アクセス権および `clia user` コマンドの詳細は、[327 ページの「user」](#) を参照してください。

## ユーザー名の制限

ユーザー名フィールドの最大文字数は 16 文字です。このフィールドには、1 文字以上の小文字の英字を含める必要があり、最初の文字は英字にする必要があります。

`username` に指定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

パスワードは 16 文字までに制限され、16 文字を超えた場合は切り詰められます。

---

## リモート root ログインの無効化

telnet、ssh、および ftp を使用して ShMM に root としてリモートからログインするときに、root パスワードがネットワーク上に表示される可能性があります。これらのリモート root ログインを無効にすると、root パスワードを保護するために必要なセキュリティを強化できます。

リモート root ログインを無効にするには、アクティブな ShMM とスタンバイの ShMM の両方で telnet、ssh、および ftp 構成ファイルから root ユーザー ID を削除し、root 以外のユーザーを追加して、新しいユーザーに su - 機能を付与します。この作業を行うには、次の手順を実行します。

1. アクティブな ShMM に root としてログインします。
2. SSH のリモート root ログインを無効にします。

/etc/inetd.conf ファイルを編集し、ssh で始まる行の終わりの /bin/sshd -i コマンドに -g オプションを追加します。例:

```
ssh stream tcp      nowait root    /bin/tcpd /bin/sshd -i -g
```

3. ftp の root ログインを無効にします。

/etc/ftppaccess ファイルを編集し、行の始まりに # (ポンド記号) を挿入して、allow-uid %0 パラメータをコメントアウトし、deny-uid %0 パラメータをファイルの新しい行として追加します。例:

```
#allow-uid %0
deny-uid %0
```

4. telnet の root ログインを無効にします。

/etc/securetty ファイルを編集し、ttyp で始まるすべてのエントリを削除するか、コメントアウトします。console および ttyS0 行は削除またはコメントアウトしないでください。これらの行をコメントアウトまたは削除すると、シリアルコンソールでの root アクセスも拒否されます。

5. 新しい root 以外のユーザーを追加します。

```
#adduser username
password: password
#
```

6. 新しいユーザー ID に su - 機能を付与します。

a. /etc/busybox.conf ファイルを作成し、例に示す 2 行を追加します。

```
# cat /etc/busybox.conf
[SUID]
su = ssx root.0 # su can be performed from non-root
passwd = ssx root.0 # non-root user can change his password
#
```

b. ファイルの所有権とグループを root に変更します。例:

```
# chown 0.0 /etc/busybox.conf
# chmod 600 /etc/busybox.conf
```

7. スタンバイ ShMM で同じ手順を繰り返し、再起動します。

---

## 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) のオープンソース実装である OpenHPI のサポートが含まれます。HPI は、通常シャーシおよびラックベースのサーバーでコンピュータハードウェアを管理するためのインタフェースを提供します。HPI へは、SNMP MIB を使用して、OpenHPI SNMP サブエージェントを通じてアクセスします。OpenHPI および SNMP の詳細は、『Sun Netra CT900 サーバーソフトウェア開発者ガイド』を参照してください。

システム管理者が注意する必要があるのは、次の 2 つの構成ファイルです。

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

### /etc/openhpi.conf ファイル

R3U3 よりも前の ATCA リリースを使用している場合、OpenHPI 構成ファイル /etc/openhpi.conf を更新して ShMM の正しい IP アドレスを指定する必要があります。構成ファイルを更新したあとに、ShMM をリセットとして変更を実装する必要があります。

---

**注** - /etc/openhpi.conf ファイルの IP アドレスには変数を使用しないでください。したがって、\$IPADDR および localhost は使用できません。

---

R3U3 以降のリリースでは、/etc/openhpi.conf で RMCP アドレスを設定する必要はありません。ShMM ソフトウェアでは、ミッドプレーン FRU から RMCP アドレスを含む Shelfman IP 接続記録を自動的に読み取り、その RMCP アドレスで openhpid を開始します。

## ▼ /etc/openhpi.conf ファイルを変更する

1. R3U3 よりも前の ATCA リリースを使用している場合は、/etc/openhpi.conf ファイルを編集し、libpigeonpoint addr パラメータの値を ShMM の IP アドレスに変更します。

/etc/openhpi.conf ファイルの一部を次に示します。

```
handler libpigeonpoint {
    entity_root = "{SYSTEM_CHASSIS,1}"
    name = "lan" # RMCP
    addr = "10.12.235.18" # Host name or IP address
    port = "623" # RMCP port
    auth_type = "none" # none, md5 or straight
    auth_level = "admin" # operator or admin
    logflags = "" # "" means logging off; also use "file stdout"
    logfile = "openhpi" # log file name prefix; ${logfile}.log
    logfile_max = "50" # maximum log file size in kilobytes
    UseCachedSdrs = "no" # set it to "yes" to use Cached Device SDR repository
}
```

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。

例:

```
# reboot
```

## /etc/snmpd.conf ファイル

SNMP エージェント構成ファイルの /etc/snmpd.conf では、SNMP エージェントの動作が定義され、アクセス制御とトラップ設定に関する設定が含まれています。アクセス制御、SNMPv3 の構成、およびトラップの設定について、以降の節で説明します。詳細は、『Sun Netra CT900 サーバーソフトウェア開発者ガイド』を参照してください。

### アクセス制御

SNMP エージェントは、RFC 2575 で定義されている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。このため、構成ファイルで次のキーワードが認識されます。

- com2sec
- group

- access
- view

また、次の使いやすいラッパー設定も認識されます。

- rocommunity
- rwcommunity
- rouser
- rwuser

この節では、さまざまなタイプとレベルのアクセスを受け入れることができるように snmpd プログラムを構成する方法を定義します。

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

VACM アクセス構成テーブルで SNMPv3 USM ユーザーを作成します。group、access、および view 設定を組み合わせる方が効率的で強力ですが、上記のラッパー設定の方が簡単に使用できます。

ユーザーが使用する必要のある最小レベルの認証とプライバシーは、最初のトークンで指定されます (デフォルトでは auth)。OID (オブジェクト識別子) パラメータでは、ユーザーのアクセスを、指定された OID よりも下に制限します。

```
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と読み取り/書き込みコミュニティを作成します。これらは、より複雑で強力な com2sec、group、access、および view 設定行のクイックラッパーです。上記の設定では、グループが作成されず、テーブルが大きくなる可能性があるため、これらの設定ほど効率はよくありません。これらの設定は、複雑な環境ではお勧めできません。これらの設定は、環境が比較的単純であるか、パフォーマンスへの負の影響を抑えられる場合に使用してください。

source トークンの形式については、次の com2sec 設定の節で説明します。OID トークンでは、そのコミュニティに対するアクセスを、指定された OID よりも下に制限します。

```
com2sec name source community
```

source/community の組から、セキュリティ 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/ビットとして指定できます。受信パケットと最初に一致した 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プします。 *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* では、 *context* を受信 PDU のコンテキストに対して照合する方法として、 *exact* または *prefix* のいずれかを指定します。 *read*、 *write*、 および *notify* では、 対応するアクセスに使用するビューを指定します。 *v1* または *v2c* アクセスの場合、 *level* は、 *noauth* で、 *context* は空になります。

*view name type subtree [mask]*

名前の付いた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、 ピリオド (.) または コロン (:) で区切られた 16 進オクテットのリストです。 *mask* を指定しない場合、 デフォルトで *ff* が使用されます。 *mask* を使用すると、 テーブル内の 1 つの行へのアクセスを比較的簡単に制御できます。 たとえば、 次のように ISP として各顧客にそれぞれ独自のインタフェースへのアクセスを付与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、ユーザーは行のフィールドを変更できます。

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 の構成

engineID *string*

engineID で、SNMPv3 メッセージに応答できるように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では、engineID は *string* から設定されます。engineID のデフォルト値は、マシンのホスト名で最初に見つかった IP アドレスに設定されます。

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

MD5 および SHA は、使用する認証の種類ですが、SHA を使用するためには OpenSSL がインストールされたパッケージを作成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プロトコルは、DES のみです。*privpassphrase* を指定しないと、*authpassphrase* と同じであると見なされます。

---

注 - 作成されたユーザーは、上記で説明した VACM アクセス制御テーブルにも追加されない限り無効です。

---



---

注 - パスフレーズの最小文字数は 8 文字です。

---

## トラップの設定と宛先の通知

`trapcommunity string`

トラップの送信時に使用するデフォルトコミュニティの *string* を定義します。このコマンドは、このコミュニティ文字列で使用する 3 つのコマンドのいずれかの前 (直前) に使用する必要があります。

`trapsink host [community [port]]`

`trap2sink host [community [port]]`

`informsink host [community [port]]`

トラップを受信する (または `informsink` では通知を行う) ホストを定義します。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。有効な場合は、認証の失敗に関するトラップも送信します。複数の `trapsink`、`trap2sink`、および `informsink` 行を指定して複数の宛先を指定できます。SNMPv2 トラップを送信する場合は `trap2sink` を使用し、情報通知を送信する場合は `informsink` を使用します。*community* が指定されていない場合、前の `trapcommunity` 設定の文字列が使用されます。*port* が指定されていない場合、よく知られた SNMP トラップポート (162) が使用されます。

`trapsess [snmpcmdargs] host`

任意のタイプのトラップの宛先を SNMP の任意のバージョンで指定できる、より汎用的なトラップ構成トークンです。このトークンを使用するには、`v2c` または `v3` のバージョン番号も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、必要に応じて設定を追加、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。

## OpenHPI デーモンモニターの有効化

デーモンモニターを有効にすると、OpenHPI プロセスが監視されます。また、何らかの理由でデーモンが中止または終了した場合、次のアクションが実行されます。

- システムでアクティブな ShMM とバックアップ ShMM が実行されている場合は、次のようになります。
  - アクティブな ShMM でデーモンが中止または終了すると、スイッチオーバーが発生します。
  - バックアップ ShMM でデーモンが中止または終了すると、再起動が行われます。
- システムで 1 つの ShMM しか実行していない場合は、その ShMM が再起動されます。

デフォルトでは、この機能は無効になっています。ShMM に R3UI の更新プログラムをインストールしたあと、次のようにモニターを有効にできます。

1. U-Boot 変数 `monitor_daemons` を `y` に設定します。

詳細は、[17 ページの「U-Boot の設定」](#)を参照してください。

```
# setenv monitor_daemons y
# reboot
```

---

注 – U-Boot シェルで変数を設定した場合は、`saveenv` コマンドを入力してから、`reboot` コマンドを入力して変更を保存します。

---

2. バックアップ ShMM に対して同じコマンドを繰り返します。

## シェルフマネージャーでの RADIUS の構成

Remote Authentication Dial-In User Service (RADIUS) は、リモートアクセスサーバーが中央のサーバーと通信してダイヤルインユーザーを認証し、要求されたシステムまたはサービスへのアクセスを承認できるようにするためのクライアント/サーバープロトコルおよびソフトウェアです。このプロトコルは、RFC2865 で扱われています。

シェルフマネージャバージョン 2.4.x (およびそれ以降) は、RADIUS クライアントのサポートを提供します。システム管理者が行う必要があるのは、シェルフマネージャで `/etc/raddb/server` ファイルを編集し、サーバー情報を指定することだけです。`/etc/raddb/server` ファイルの例を次に示します。

```
#<radius_server_ip_address> <radius_secret> <wait_period>
127.0.0.1      secret      1
other-server  other-secret  3
```

各表記の意味は次のとおりです。

- *radius\_server\_ip\_address* は、RADIUS サーバーの IP アドレスです。
- *radius\_secret* は、RADIUS サーバーと RADIUS クライアントが共有するシークレットテキスト文字列です。サーバーとクライアントが通信するためには、どちらも同じテキスト文字列で構成する必要があります。
- *wait\_period* は、秒単位のタイムアウトで、サーバーで応答が失敗したと判断するまでモジュールが待機する秒数を制御します。デフォルトのタイムアウトは、3 秒です。

複数の RADIUS サーバーが指定されている場合、各サーバーが順番に試行されます。成功または失敗を最初に返したサーバーによって、モジュールは成功または失敗を返します。サーバーが応答に失敗した場合にのみ、そのサーバーがスキップされて、次のサーバーが使用されます。

デフォルトでは、スーパーユーザー (root) が、console、telnet、または ssh を使用して、出荷時のデフォルトパスワードまたは RADIUS パスワードでシェルマネージャにログインできます。

シェルマネージャで、RADIUS パスワードのみ受け入れることができるようにする、またはローカルの root パスワードのみ受け入れることができるようにするには、標準の PAM 構成ファイルを変更する必要があります。これらのファイルは、シェルマネージャの /etc/pam.d/login および /etc/pam.d/sshd です (構成については、標準の PAM (ポートアプリケーションマッピング) 規則を参照)。



## 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、RMCP インタフェースを使用して Ethernet 経由で行います。

この章が次の節を含めている:

- [80 ページの「IPMI LAN インタフェース」](#)
- [88 ページの「シェルフマネージャーのコマンド行インタフェース」](#)
- [96 ページの「システムの監視」](#)
- [120 ページの「シェルフマネージャーの再初期化」](#)
- [123 ページの「シェルフ管理カードの再プログラミング」](#)
- [146 ページの「ノードボードコンソールへの接続」](#)
- [149 ページの「手動によるノードボードの正常な停止」](#)

# IPMI LAN インタフェース

IPMI LAN インタフェースは ATCA 仕様への準拠に必要であり、RMCP (Remote Management Control Protocol) を介したシェルフマネージャーとの IPMI メッセージングをサポートしています。RMCP を使用してシェルフと通信すると、システム管理者は ATCA 準拠の任意のシェルフマネージャーと通信できます。この下位レベルのインタフェースは、システム管理者がシェルフマネージャーをプロキシとして使用してシェルフ内の IPM コントローラに IPMI コマンドを実行できる機能など、シェルフの IPMI の側面へのアクセスを提供します。

## サポートされている IPMI コマンド

IPMI の標準コマンドは、PICMG 3.0 の ATCA 仕様に記載されています。表 3-1 に、シェルフマネージャーにより実装される IPMI コマンドを示します。セキュリティ上の考慮事項により、RMCP インタフェースを介して受け取るか、IPMB-0 を介して受け取るかによって特定のコマンドがどのように処理されるかは異なります。

表 3-1 シェルフマネージャーにより実装される IPMI コマンド

コマンド	NetFn	CMD	RMCP インタフェースからの着信	IPM コントローラからの着信
Get Device ID	App	01h	サポートされる	サポートされる
Cold Reset	App	02h	サポートされる	サポートされる
Warm Reset	App	03h	サポート対象外	サポートされない
Get Self Test Results	App	04h	サポートされる	サポートされる
Manufacturing Test On	App	05h	サポート対象外	サポートされない
Set ACPI Power State	App	06h	サポートされる	サポートされる
Get ACPI Power State	App	07h	サポートされる	サポートされる
Get Device GUID	App	08h	サポートされる	サポートされる
Reset Watchdog Timer	App	22h	サポートされる	サポートされる
Set Watchdog Timer	App	24h	サポートされる	サポートされる
Get Watchdog Timer	App	25h	サポートされる	サポートされる
Set BMC Global Enables	App	2Eh	サポートされる	サポートされる
Get BMC Global Enables	App	2Fh	サポートされる	サポートされる
Clear Message Flags	App	30h	サポートされる	サポートされる
Get Message Flags	App	31h	サポートされる	サポートされる

表 3-1 シェルフマネージャーにより実装される IPMI コマンド (続き)

コマンド	NetFn	CMD	RMCP インタフェースからの着信	IPM コントローラからの着信
Enable Message Channel Receive	App	32h	サポート対象外	サポートされない
Get Message	App	33h	サポート対象外	サポートされない
Send Message	App	34h	サポートされる	サポートされる
Read Event Message Buffer	App	35h	サポート対象外	サポートされない
Get BT Interface Capabilities	App	36h	サポート対象外	サポートされない
Get System GUID	App	37h	サポートされる	サポートされる
Get Channel Authentication Capabilities	App	38h	サポートされる	サポートされる(*)
Get Session Challenge	App	39h	サポートされる	サポートされない
Activate Session	App	3Ah	サポートされる	サポートされない
Set Session Privilege Level	App	3Bh	サポートされる	サポートされない
Close Session	App	3Ch	サポートされる	サポートされない
Get Session Info	App	3Dh	サポートされる	サポートされる(*)
Get AuthCode	App	3Fh	サポートされる	サポートされる(*)
Set Channel Access	App	40h	サポートされる	サポートされる(*)
Get Channel Access	App	41h	サポートされる	サポートされる(*)
Get Channel Info	App	42h	サポートされる	サポートされる(*)
Set User Access	App	43h	サポートされる	サポートされる(*)
Get User Access	App	44h	サポートされる	サポートされる(*)
Set User Name	App	45h	サポートされる	サポートされる(*)
Get User Name	App	46h	サポートされる	サポートされる(*)
Set User Password	App	47h	サポートされる	サポートされる(*)
Activate Payload	App	48h	サポート対象外	サポートされない
Deactivate Payload	App	49h	サポート対象外	サポートされない
Get Payload Activation Status	App	4Ah	サポート対象外	サポートされない
Get Payload Instance Info	App	4Bh	サポート対象外	サポートされない
Set User Payload Access	App	4Ch	サポート対象外	サポートされない
Get User Payload Access	App	4Dh	サポート対象外	サポートされない
Get Channel Payload Support	App	4Eh	サポート対象外	サポートされない
Get Channel Payload Version	App	4Fh	サポート対象外	サポートされない
Get Channel OEM Payload Info	App	50h	サポート対象外	サポートされない

表 3-1 シェルフマネージャーにより実装される IPMI コマンド (続き)

コマンド	NetFn	CMD	RMCP インタフェースからの着信	IPM コントローラからの着信
Master Write-Read	App	52h	サポート対象外	サポートされない
Get Channel Cipher Suites	App	54h	サポート対象外	サポートされない
Suspend/Resume Payload Encryption	App	55h	サポート対象外	サポートされない
Set Channel Security Keys	App	56h	サポート対象外	サポートされない
Get System Interface Capabilities	App	57h	サポート対象外	サポートされない
Get Chassis Capabilities	シャーシ	00h	サポートされる	サポートされる
Get Chassis Status	シャーシ	01h	サポートされる	サポートされる
Chassis Control	シャーシ	02h	サポートされる	サポートされる
Chassis Reset	シャーシ	03h	サポート対象外	サポートされない
Chassis Identify	シャーシ	04h	サポート対象外	サポートされない
Set Chassis Capabilities	シャーシ	05h	サポートされる	サポートされる
Set Power Restore Policy	シャーシ	06h	サポート対象外	サポートされない
Get System Restart Cause	シャーシ	07h	サポート対象外	サポートされない
Set System Boot Options	シャーシ	08h	サポート対象外	サポートされない
Get System Boot Options	シャーシ	09h	サポート対象外	サポートされない
Set Front Panel Button Enables	シャーシ	0Ah	サポート対象外	サポートされない
Set Power Cycle Interval	シャーシ	0Bh	サポート対象外	サポートされない
Get POH Counter	シャーシ	0Fh	サポート対象外	サポートされない
Set LAN Configuration Parameters	トランスポート	01h	サポートされる	サポートされる(*)
Get LAN Configuration Parameters	トランスポート	02h	サポートされる	サポートされる
Suspend BMC ARPs	トランスポート	03h	サポートされる	サポートされる(*)
Get IP/UDP/RMCP statistics	トランスポート	04h	サポート対象外	サポートされない
Set Serial/Modem Configuration	トランスポート	10h	サポート対象外	サポートされない
Get Serial/Modem Configuration	トランスポート	11h	サポート対象外	サポートされない
Set Serial/Modem Mux	トランスポート	12h	サポート対象外	サポートされない



表 3-1 シェルフマネージャーにより実装される IPMI コマンド (続き)

コマンド	NetFn	CMD	RMCP インタフェースからの着信	IPM コントローラからの着信
Get TAP Response Codes	トランスポート	13h	サポート対象外	サポートされない
Set PPP UDP Proxy Transmit Data	トランスポート	14h	サポート対象外	サポートされない
Get PPP UDP Proxy Transmit Data	トランスポート	15h	サポート対象外	サポートされない
Send PPP UDP Proxy Packet	トランスポート	16h	サポート対象外	サポートされない
Get PPP UDP Proxy Receive Data	トランスポート	17h	サポート対象外	サポートされない
Serial/Modem Connection Active	トランスポート	18h	サポート対象外	サポートされない
Callback	トランスポート	19h	サポート対象外	サポートされない
Set User Callback Options	トランスポート	1Ah	サポートされる	サポートされる(*)
Get User Callback Options	トランスポート	1Bh	サポートされる	サポートされる(*)
SOL Activating	トランスポート	20h	サポート対象外	サポートされない
Set SOL Configuration Parameters	トランスポート	21h	サポート対象外	サポートされない
Get SOL Configuration Parameters	トランスポート	22h	サポート対象外	サポートされない
Get FRU Inventory Area Info	ストレージ	10h	サポートされる	サポートされる
Read FRU Data	ストレージ	11h	サポートされる	サポートされる
Write FRU Data	ストレージ	12h	サポートされる	サポートされる
Get SDR Repository Info	ストレージ	20h	サポートされる	サポートされる
Get SDR Repository Allocation Info	ストレージ	21h	サポート対象外	サポートされない
Reserve SDR Repository	ストレージ	22h	サポートされる	サポートされる
Get SDR	ストレージ	23h	サポートされる	サポートされる
Add SDR	ストレージ	24h	サポートされる	サポートされる
Partial Add SDR	ストレージ	25h	サポートされる	サポートされる
Delete SDR	ストレージ	26h	サポートされる	サポートされる

表 3-1 シェルフマネージャーにより実装される IPMI コマンド (続き)

コマンド	NetFn	CMD	RMCP インタフェースからの着信	IPM コントローラからの着信
Clear SDR Repository	ストレージ	27h	サポートされる	サポートされる
Get SDR Repository Time	ストレージ	28h	サポートされる	サポートされる
Set SDR Repository Time	ストレージ	29h	サポートされる	サポートされる
Enter SDR Repository Update Mode	ストレージ	6F または 2A (Sun レガシー) h	サポート対象外	サポートされない
Exit SDR Repository Update Mode	ストレージ	2Bh	サポート対象外	サポートされない
Run Initialization Agent	ストレージ	2Ch	サポート対象外	サポートされない
Get SEL Info	ストレージ	40h	サポートされる	サポートされる
Get SEL Allocation Info	ストレージ	41h	サポートされる	サポートされる
Reserve SEL	ストレージ	42h	サポートされる	サポートされる
Get SEL Entry	ストレージ	43h	サポートされる	サポートされる
Add SEL Entry	ストレージ	44h	サポートされる	サポートされる
Partial Add SEL Entry	ストレージ	45h	サポートされる	サポートされる
Delete SEL Entry	ストレージ	46h	サポートされる	サポートされる
Clear SEL	ストレージ	47hh	サポートされる	サポートされる
Get SEL Time	ストレージ	48h	サポートされる	サポートされる
Set SEL Time	ストレージ	49h	サポートされる	サポートされる
Get Auxiliary Log Status	ストレージ	5Ah	サポート対象外	サポートされない
Set Auxiliary Log Status	ストレージ	5Bh	サポート対象外	サポートされない
Set Event Receiver	S/E	00h	サポートされる	サポートされる
Get Event Receiver	S/E	01h	サポートされる	サポートされる
Event Message	S/E	02h	サポートされる	サポートされる
Get PEF Capabilities	S/E	10h	サポートされる	サポートされる
Arm PEF Postpone Timer	S/E	11h	サポートされる	サポートされる
Set PEF Configuration Parameters	S/E	12h	サポートされる	サポートされる
Get PEF Configuration Parameters	S/E	13h	サポートされる	サポートされる
Set Last Processed Event ID	S/E	14h	サポートされる	サポートされる
Get Last Processed Event ID	S/E	15h	サポートされる	サポートされる
Alert Immediate	S/E	16h	サポートされる	サポートされる
PET Acknowledge	S/E	17h	サポートされる	サポートされる

表 3-1 シェルフマネージャーにより実装される IPMI コマンド (続き)

コマンド	NetFn	CMD	RMCP インタフェースからの着信	IPM コントローラからの着信
Get Device SDR Info	S/E	20h	サポートされる	サポートされる
Get Device SDR	S/E	21h	サポートされる	サポートされる
Reserve Device SDR Repository	S/E	22h	サポートされる	サポートされる
Get Sensor Reading Factors	S/E	23h	サポートされる	サポートされる
Set Sensor Hysteresis	S/E	24h	サポートされる	サポートされる
Get Sensor Hysteresis	S/E	25h	サポートされる	サポートされる
Set Sensor Threshold	S/E	26h	サポートされる	サポートされる
Get Sensor Threshold	S/E	27h	サポートされる	サポートされる
Set Sensor Event Enable	S/E	28h	サポートされる	サポートされる
Get Sensor Event Enable	S/E	29h	サポートされる	サポートされる
Re-arm Sensor Events	S/E	6F または 2A (Sun レガシー) h	サポートされる	サポートされる
Get Sensor Event Status	S/E	2Bh	サポートされる	サポートされる
Get Sensor Reading	S/E	2Dh	サポートされる	サポートされる
Set Sensor Type	S/E	2Eh	サポートされる	サポートされる
Get Sensor Type	S/E	2Fh	サポートされる	サポートされる
Get PICMG Properties	PICMG	00h	サポートされる	サポートされる
Get Address Info	PICMG	01h	サポートされる	サポートされる
Get Shelf Address Info	PICMG	02h	サポートされる	サポートされる
Set Shelf Address Info	PICMG	03h	サポートされる	サポートされる
FRU Control	PICMG	04h	サポートされる	サポートされる
Get FRU LED Properties	PICMG	05h	サポートされる	サポートされる
Get LED Color Capabilities	PICMG	06h	サポートされる	サポートされる
Set FRU LED State	PICMG	07h	サポートされる	サポートされる
Get FRU LED State	PICMG	08h	サポートされる	サポートされる
Set IPMB State	PICMG	09h	サポートされる	サポートされる
Set FRU Activation Policy	PICMG	0Ah	サポートされる	サポートされる
Get FRU Activation Policy	PICMG	0Bh	サポートされる	サポートされる
Set FRU Activation	PICMG	0Ch	サポートされる	サポートされる
Get Device Locator Record ID	PICMG	0Dh	サポートされる	サポートされる

表 3-1 シェルフマネージャーにより実装される IPMI コマンド (続き)

コマンド	NetFn	CMD	RMCP インタフェースからの着信	IPM コントローラからの着信
Set Port State	PICMG	0Eh	サポートされる	サポートされる
Get Port State	PICMG	0Fh	サポートされる	サポートされる
Compute Power Properties	PICMG	10h	サポートされる	サポートされる
Set Power Level	PICMG	11h	サポートされる	サポートされる
Get Power Level	PICMG	12h	サポートされる	サポートされる
Renegotiate Power	PICMG	13h	サポート対象外	サポートされる
Get Fan Speed Properties	PICMG	14h	サポートされる	サポートされる
Set Fan Level	PICMG	15h	サポートされる	サポートされる
Get Fan Level	PICMG	16h	サポートされる	サポートされる
Bused Resource	PICMG	17h	サポート対象外	サポートされる
Get IPMB Link Info	PICMG	18h	サポートされる	サポートされる
Get Shelf Power Allocation	PICMG	19h	サポートされる	サポートされる
Get Shelf Manager IPMB Address	PICMG	1Bh	サポートされる	サポートされる
Set Fan Policy	PICMG	1Ch	サポート対象外	サポートされない
Get Fan Policy	PICMG	1Dh	サポート対象外	サポートされない
FRU Control Capabilities	PICMG	1Eh	サポートされる	サポートされる
FRU Inventory Device Lock Control	PICMG	1Fh	サポートされる	サポートされる
FRU Inventory Device Write	PICMG	20h	サポートされる	サポートされる
Get Shelf Manager IP Addresses	PICMG	21h	サポートされる	サポートされる

注 - (\*) が付記されているコマンドは、構成パラメータ `ALLOW_ALL_COMMANDS_FROM_IPMB` が `TRUE` に設定されている場合に、IPMB-0 側のみでサポートされます。

## 「自己診断テスト結果の取得」 IPMI コマンド

「自己診断テスト結果の取得」コマンドでは、ShMM の電源が投入されているとき、U-Boot ユーティリティーによって実行された POST テストの結果が返されます。すべてのテストに合格すると、状態コード 0x55 が返されます。いずれかのテストに失敗すると、デバイス固有の障害コード 0x59 が返され、3 番目のバイトに次のビットマスクが含まれます。

[7:5]	保留
[4]	1b = Ethernet テストに失敗しました
[3]	1b = UART テストに失敗しました
[2]	1b = U-Boot CRC テストに失敗しました
[1]	1b = I <sup>2</sup> C テストに失敗しました
[0]	1b = メモリーテストに失敗しました

## IPMI の OEM 拡張コマンド

シェルフマネージャーでは、表 3-1 に示す仕様で定義されている IPMI コマンド以外にも、いくつかの OEM 定義の IPMI コマンドをシステムマネージャーの利便性のために実装しています。シェルフマネージャーの IPMI 拡張コマンドを表 3-2 に示します。説明は付録 B に記載されています。

表 3-2 シェルフマネージャーの IPMI 拡張コマンド

コマンド	NetFn	オペレーション コード
Get Shelf FRU Record Data	3Eh	1h
Set Shelf FRU Record Data	2Eh	05h
Notify Shelf Manager About an Extracted FRU	2Eh	2h
Initiate Shelf Manager Switchover	3Eh	3h
Subscribe for Event Notifications	2Eh	4h

---

# シェルフマネージャのコマンド行インタフェース

シェルフマネージャのコマンド行インタフェース (CLI) を使用すると、テキストコマンドを使用して、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自体と通信できます。この CLI は IPMI ベースのコマンドセットで、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通じて 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 の配置、現在のセンサーの値、しきい値の設定、最近のイベント、シェルフ全体の状態など、シェルフの現在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オンします。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。最初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 が維持する 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が動作している必要があります。

例:

```
# clia ipmc 20

Pigeon Point 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 は対話型モードになります。このモードでは、プログラムは端末に繰り返し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取り、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す。例:

```
# clia

Pigeon Point 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
#
```

## シェルフマネージャの CLI コマンド

シェルフマネージャの CLI には、次のコマンドが実装されています。コマンドごとにサブセクションが設けられ、コマンド名のアルファベット順に[付録 A の 153 ページの「シェルフマネージャの CLI コマンド」](#)で詳細に説明されています。

表 3-3 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 アドレス FRU デバイス ID	指定した FRU をアクティブにします。
airfilterreplaced	dd.mm.yyyy (省略可能)	エアフィルタを交換する日付を設定します。
alarm	アラームの種類 クリア 情報	Telco アラームをアクティブにするか、クリアします。アラーム情報も表示します。
amcportstate	IPMB アドレス FRU デバイス ID または AMC 番号 (省略可能)	指定した AMC の AMC ポート状態情報を表示します。AMC 番号を指定しないと、指定した IPM コントローラでアクティブなすべての AMC の AMC ポート状態情報が報告されます。
board	スロット番号 (省略可能)	ボードの情報を表示します。

表 3-3 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
boardreset	スロット番号	指定した 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)	指定した操作を、バス型の E-キーイング管理リソースで実行します。
console	スロット番号	指定し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 アドレス FRU デバイス ID	指定した 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ー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		対話型モードのインタプリタを終了します。
fans	IPMB アドレス (省略可能) FRU デバイス ID (省略可能)	ファンの情報を表示します。
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	Sun Netra CP3x60 ノードボード上のシステムファームウェアを、指定し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 アドレス (省略可能) FRU デバイス ID (省略可能) FRU の種類 (省略可能)	シェルフ内の単体またはグループの FRU に関する情報を表示します。FRU は種類または親 IPM コントローラで選択します。
frucontrol	IPMB アドレス FRU デバイス ID コマンド オプション	指定した FRU に FRU 制御コマンドを送信します。
frudata	IPMB アドレス (省略可能) FRU デバイス ID (省略可能) ブロック/バイトオフセット (省略可能) データ (省略可能)	指定した FRU 上の FRU 情報への raw アクセスを提供します。
frudatar	IPMB アドレス FRU デバイス ID ファイル名	指定した FRU の FRU データ領域を読み取り、指定したファイルにデータを保存します。
frudataw	IPMB アドレス FRU デバイス ID ファイル名	指定した FRU の FRU データ領域の指定したファイルに FRU データを書き込みます。
fruinfo	IPMB アドレス FRU デバイス ID	ユーザーが理解しやすい FRU 情報を出力します。



表 3-3 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
getacousticlevel	ETSI NEBS-A NEBS-U	システムの騒音レベルとファン速度を表示します。
getbootdev	IPMB アドレス FRU デバイス ID または AMC アドレス	システムの起動デバイスのパラメータを表示します。
getfanlevel	IPMB アドレス (省略可能) FRU デバイス ID (省略可能)	指定した FRU によって制御されるファンの現在のレベルを表示します。
getfanpolicy	IPMB アドレス (省略可能) FRU デバイス ID (省略可能) <サイトの種類> (省略可能) <サイト番号> (省略可能)	ファントレー制御モードの情報または指定したファントレーの FRU カバレッジの情報を取得します。このコマンドでは、(「ファンポリシーの設定」コマンドに基づいて) シェルフマネージャーによってサイトの自律制御が有効または無効にされたかどうか、および (ファンジオグラフィレコードに基づいて) FRU のサイトがファンの有効範囲にあるかどうかの 2 つの異なるデータが返されます。
getfruledstate	IPMB アドレス (省略可能) FRU 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 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)	指定し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 アドレス IPMB リンク番号 (省略可能)	ターゲットアドレスでの IPMB-0 の現在の状態を表示します。リンク番号を指定していて、ターゲット IPMC が IPMB スイッチである場合、特定のリンクの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) 設定セレクト (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。
getmgmtportroute	スロット番号	管理ポートのルーティング構成を表示します。
getmuxconfig	スロット番号 (省略可能)	シェルフのミッドプレーンレコードからマルチプレクサ (MUX) の構成情報を取得します。
getpefconfig	パラメータ名または番号 (省略可能) 設定セレクト (省略可能)	PEF 構成パラメータを取得して表示します。

表 3-3 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
getsensoreventenable	IPMB 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)	指定し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。
getthreshold   threshold	IPMB 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)	特定のセンサーのしきい値情報を表示します。
help		サポートされるコマンドの一覧を表示します。
ipmc	IPMB アドレス (省略可能)	シェルフ内の 1 つ以上の IPM コントローラの情報を表示します。
localaddress		現在のシェルフマネージャの IPMB アドレスを取得します。
minfanlevel	ファンレベル (省略可能)	最小ファンレベルを表示または設定します。
mgmtportstate	スロット	IPMC の OEM コマンド「Ethernet ポートアクセスの取得」を使用して、管理ポートの状態またはポートのルーティング構成について IPMC に対してクエリーします。
muxstate	スロット番号	現在のマルチプレクサ (MUX) の状態とポートのルーティング構成について IPMC に対してクエリーします。
networkelementid	ネットワーク要素の識別子 (省略可能)	ネットワーク要素の識別子の取得または設定を許可します。
poll		IPMB-0 上の IPM コントローラのポーリングを開始します。
sel	IPMB アドレス (省略可能) 項目の数 (省略可能)	ターゲット IPM コントローラで維持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。
sendamc	IPMB アドレス AMC アドレスまたは FRU ID NetFn コマンドコード Byte1 (省略可能) ... (省略可能) ByteN (省略可能)	対応する IPM コントローラの背後に透過的に存在する Advanced Management Controller (AMC) に、任意の IPMI コマンドを透過的に送信します。

表 3-3 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
sendcmd	IPMB アドレス ネットワーク関数 コマンドコード Byte1 (省略可能) ... (省略可能) ByteN (省略可能)	ターゲット IPMC に任意の IPMI コマンドを透過的に送信します。
sensor	IPMB 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)	単体またはグループのセンサーに関する情報を表示します。センサーは、IPM コントローラアドレス、番号、または名前を選択します。
sensordata	IPMB 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)	特定のセンサーの値情報を表示します。
sensorread	IPMB アドレス センサー番号	特定のセンサーの raw 値情報を表示します (センサーを説明するセンサーデータレコードをすべて無視)。
session		アクティブな RMCP セッションの情報を表示します。
setacousticlevel	ETSI NEBS-A NEBS-U	システムの騒音レベルとファン速度を設定します。
setbooddev	IPMB アドレス FRU デバイス ID または AMC アドレス 起動デバイスのパラメータ	システムの起動デバイスのパラメータを設定します。
setextracted	IPMB アドレス FRU デバイス ID	指定した FRU がシェルフから物理的に抽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 アドレス FRU デバイス ID level	指定した FRU によって制御されるファンの新しいレベルを設定します。
setfanpolicy	IPMB アドレス FRU デバイス ID 実行されるアクション: ENABLE または DISABLE タイムアウト (省略可能) サイトの種類 (省略可能) サイト番号 (省略可能)	シェルフ FRU にファンジオグラフィックレコードがある場合、これに加えて冷却管理のためにファントレーを有効または無効にします。

表 3-3 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
setfruledstate	IPMB アドレス FRU デバイス ID LED ID または ALL LED の操作 LED の色 (省略可能)	特定の LED または指定した FRU のすべての LED の状態を設定します。
sethysteresis	IPMB 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) ヒステリシス値	指定したセンサーの新しいヒステリシス値を設定します。
setipmbstate	IPMB アドレス IPMB バス名 (A または B) IPMB リンク番号 (省略可能) 実行されるアクション	ターゲット IPM コントローラで IPMB-A または IPMB-B (または特定の IPMB リンク) を無効または有効にします。
setlanconfig	チャンネル パラメータ名または番号 その他のパラメータ	指定したチャンネルで LAN 構成パラメータ値を設定します。
setlocked	IPMB アドレス FRU デバイス ID state	指定した FRU のロックビットを指定した状態 (0 - ロック解除、1 - ロック) に設定します。
setmgmtportroute	スロット番号 1 または 0	フロントまたは背面アクセスを含む、管理ポートルートを設定します。
setmuxconfig	スロット番号 ポート 1: ルート値 ポート 2: ルート値	シェルフのミッドプレーンレコードに、特定のスロットのマルチプレクサ (MUX) 構成情報を設定します。MUX の構成設定は、ブレードの起動時に有効になります。
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 アドレス FRU デバイス ID 電力レベルまたは OFF Copy	指定した 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。
setsensordata	IPMB アドレス (省略可能) センサー名 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) event_data (省略可能)	指定したセンサーの読み取り、表明または非表明マスク、イベントデータバイトを変更します。

表 3-3 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
setsensoreventenable	IPMB 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 アドレス センサー名 センサー番号 しきい値の種類 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。
setuserlabel	シェルフ名 スロット番号名	ユーザーが割り当てたシェルフとボードの名前を構成します。ポート名はスロット番号に割り当てられます。
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では、シェルフ属性の設定、および特定の領域に関する追加情報の取得が可能です。
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 (有効またはバックアップ) を表示します。
showhost	スロット番号	特定の Sun Netra CP3x60 ノード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。

表 3-3 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
userlabel	シェルフ スロット スロット番号 (省略可能)	ユーザーが割り当てたシェルフとボードの名前を表示します。ポート名はスロット番号に割り当てられます。
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、簡略モードおよび冗長モードの実行形式をサポートしています。これらの形式では提供す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。冗長モードを選択するには、コマンド行でコマンドの直後、位置引数の前にオプション `-v` を指定します。バックアップシェルフマネージャーで実行されるコマンドは、バックアップシェルフマネージャーのローカルにあるオブジェクト (センサー、FRU、IPM コントローラなど) にしかアクセスできません。

特定のコマンドの実行対象が、アクティブなシェルフマネージャーであるか、バックアップのシェルフマネージャーであるかユーザーが判断できるよう、バックアップシェルフマネージャーで CLI コマンドが実行されると、「機能を制限し、バックアップのシェルフマネージャーで実行されています」というメッセージが発行されます。

## システムの監視

シェルフマネージャーの CLI には、システムを監視しシステムの状態を表示するコマンドが多数あります。この節では、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、[89 ページの「シェルフマネージャーの CLI コマンド」](#)または[付録 A 153 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。

次のタスク例では、使用するコマンドとその出力を示します。

- [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](#)
- [ボードの詳細情報を表示する](#)
- [ボード上のセンサーを一覧表示する](#)

- 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 を一覧表示する
- 特定の IPM コントローラの情報を表示する
- IPM コントローラの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 3、4、5、7 および 8 にのみボードがあります。

```
# clia board
Pigeon Point Shelf Manager Command Line Interpreter

Physical Slot # 3
92: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x01
    PICMG Version 2.2
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)

92: 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"

Physical Slot # 4
8e: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x01
    PICMG Version 2.1
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)

8e: 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-3060"

8e: FRU # 1
    Entity: (0xc1, 0x61)
    Hot Swap State: M4 (Active), Previous: M7 (Communication Lost),
Last State Change Cause: Normal State Change (0x0)
    Device ID String: "SB AMC-HD-A-80X"

Physical Slot # 5
8a: 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)

8a: 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-3020"

Physical Slot # 7

82: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x01

PICMG Version 2.1

Hot Swap State: M4 (Active), Previous: M3 (Activation In Process), Last State Change Cause: Normal State Change (0x0)

82: 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: "ATS1160"

82: FRU # 1

Entity: (0xc0, 0x60)

Hot Swap State: M4 (Active), Previous: M3 (Activation In Process), Last State Change Cause: Normal State Change (0x0)

Device ID String: "TM1460A RTM"

Physical Slot # 8

84: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x01

PICMG Version 2.1

Hot Swap State: M4 (Active), Previous: M3 (Activation In Process), Last State Change Cause: Normal State Change (0x0)

84: 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: "CP3140H-BEG"

84: FRU # 1

Entity: (0xc0, 0x60)

Hot Swap State: M4 (Active), Previous: M3 (Activation In Process), Last State Change Cause: Normal State Change (0x0)

Device ID String: "XCP3040H-RTC"

#



- ボードの詳細情報を表示する

この例では、物理スロット 14 のボードの詳細情報を示します。

```
# clia board -v 4
Pigeon Point Shelf Manager Command Line Interpreter

Physical Slot # 4
8e: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x01
    PICMG Version 2.1
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID: 0x00, Revision: 0, Firmware: 0.210, IPMI ver 1.5
    Manufacturer ID: 00006F or 2A (Sun legacy), Product ID: 0bf4,
Auxiliary Rev: 00000000
    Device ID String: "NetraCP-3060"
    Global Initialization: 0xc, Power State Notification: 0xc,
Device Capabilities: 0x29
    Controller provides Device SDRs
    Supported features: 0x29
        "Sensor Device" "FRU Inventory Device" "IPMB Event
Generator"
    8e: Base Interface (0x00), Channel: 1
        Link: Enabled Ports: 1
        Peer Addr: 0x82, Link Type: PICMG 3.0 Base Interface
10/100/1000 BASE-T, Ext: 0 (10/100/1000 BASE-T)
    8e: Base Interface (0x00), Channel: 2
        Link: Enabled Ports: 1
        Peer Addr: 0x84, Link Type: PICMG 3.0 Base Interface
10/100/1000 BASE-T, Ext: 0 (10/100/1000 BASE-T)
    8e: Fabric Interface (0x01), Channel: 1
        Link: Enabled Ports: 1
        Peer Addr: 0x82, Link Type: PICMG 3.1 Ethernet Fabric
Interface, Ext: 0
    8e: Fabric Interface (0x01), Channel: 2
        Link: Enabled Ports: 1
        Peer Addr: 0x84, Link Type: PICMG 3.1 Ethernet Fabric
Interface, Ext: 0

8e: 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-3060"
    Site Type: 0x00, Site Number: 04
```

```
Current Power Level: 0x01, Maximum Power Level: 0x01, Current  
Power Allocation: 174.0 Watts
```

```
8e: FRU # 1
```

```
Entity: (0xc1, 0x61)
```

```
Hot Swap State: M4 (Active), Previous: M7 (Communication Lost),  
Last State Change Cause: Normal State Change (0x0)
```

```
Device Type: "FRU Inventory Device behind management  
controller" (0x10), Modifier 0x0
```

```
Device ID String: "SB AMC-HD-A-80X"
```

```
Current Power Level: 0x01, Maximum Power Level: 0x01, Current  
Power Allocation: 21.1 Watts
```

```
#
```

- **ボード上のセンサーを一覧表示する**

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を示します。

```
# clia sensor 92
```

```
Pigeon Point Shelf Manager Command Line Interpreter
```

```
92: LUN: 0, Sensor # 0 ("FRU 0 Hot Swap")
```

```
Type: Discrete (0x6f), "Hot Swap" (0xf0)
```

```
Belongs to entity: (0xa0, 96) [FRU # 0]
```

```
92: LUN: 0, Sensor # 2 ("IPMB Physical")
```

```
Type: Discrete (0x6f), "IPMB Link" (0xf1)
```

```
Belongs to entity: (0xa0, 96) [FRU # 0]
```

```
92: LUN: 0, Sensor # 4 ("CPU1 Temp")
```

```
Type: Threshold (0x01), "Temperature" (0x01)
```

```
Belongs to entity: (0x3, 96) [FRU # 0]
```

```
92: LUN: 0, Sensor # 5 ("CPU2 Temp")
```

```
Type: Threshold (0x01), "Temperature" (0x01)
```

```
Belongs to entity: (0x3, 96) [FRU # 0]
```

```
92: LUN: 0, Sensor # 6 ("Inlet Temp")
```

```
Type: Threshold (0x01), "Temperature" (0x01)
```

```
Belongs to entity: (0x3, 96) [FRU # 0]
```

```
92: LUN: 0, Sensor # 7 (" +12.0V")
```

```
Type: Threshold (0x01), "Voltage" (0x02)
```

```
Belongs to entity: (0x14, 96) [FRU # 0]
```

```
92: LUN: 0, Sensor # 8 (" -12.0V")
```

```
Type: Threshold (0x01), "Voltage" (0x02)
Belongs to entity: (0x14, 96) [FRU # 0]

92: LUN: 0, Sensor # 9 ("+5.0V VCC")
Type: Threshold (0x01), "Voltage" (0x02)
Belongs to entity: (0x14, 96) [FRU # 0]

92: LUN: 0, Sensor # 10 ("+3.3V Main")
Type: Threshold (0x01), "Voltage" (0x02)
Belongs to entity: (0x14, 96) [FRU # 0]

92: LUN: 0, Sensor # 11 ("+3.3V StandBy")
Type: Threshold (0x01), "Voltage" (0x02)
Belongs to entity: (0x14, 96) [FRU # 0]

92: LUN: 0, Sensor # 12 ("VBAT")
Type: Threshold (0x01), "Voltage" (0x02)
Belongs to entity: (0x14, 96) [FRU # 0]

92: LUN: 0, Sensor # 13 ("VDD Core0")
Type: Threshold (0x01), "Voltage" (0x02)
Belongs to entity: (0x14, 96) [FRU # 0]

92: LUN: 0, Sensor # 14 ("VDD Core1")
Type: Threshold (0x01), "Voltage" (0x02)
Belongs to entity: (0x14, 96) [FRU # 0]

92: LUN: 0, Sensor # 15 ("VTT 1.25V")
Type: Threshold (0x01), "Voltage" (0x02)
Belongs to entity: (0x14, 96) [FRU # 0]

92: LUN: 0, Sensor # 16 ("VDD 1.2V")
Type: Threshold (0x01), "Voltage" (0x02)
Belongs to entity: (0x14, 96) [FRU # 0]

Belongs to entity: (0x14, 96) [FRU # 0]

92: LUN: 0, Sensor # 17 ("VCC TM 2.5V")
Type: Threshold (0x01), "Voltage" (0x02)
Belongs to entity: (0x14, 96) [FRU # 0]

92: LUN: 0, Sensor # 18 ("VDD +2.5V")
Type: Threshold (0x01), "Voltage" (0x02)
Belongs to entity: (0x14, 96) [FRU # 0]
```

```

92: LUN: 0, Sensor # 19 ("VDD +1.5V")
    Type: Threshold (0x01), "Voltage" (0x02)
    Belongs to entity: (0x14, 96) [FRU # 0]

92: LUN: 0, Sensor # 20 ("System Event")
    Type: Discrete (0x6f), "System Event" (0x12)
    Belongs to entity: (0xa0, 96) [FRU # 0]

92: LUN: 0, Sensor # 3 ("BMC Watchdog")
    Type: Discrete (0x6f), "Watchdog 2" (0x23)
    Belongs to entity: (0x3, 96) [FRU # 0]

92: LUN: 0, Sensor # 21 ("RTM Presence")
    Type: Discrete (0x6f), "Entity Presence" (0x25)
    Belongs to entity: (0xa0, 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) の情報を示します。

```

# clia sensor 92 9
Pigeon Point Shelf Manager Command Line Interpreter

92: LUN: 0, Sensor # 9 ("+5.0V VCC")
    Type: Threshold (0x01), "Voltage" (0x02)
    Belongs to entity: (0x14, 96) [FRU # 0]

```

- サーバー内のすべての IPMC を一覧表示する

この例では、ipmc コマンドの標準的な出力を示します。

```

# clia ipmc
Pigeon Point 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 コントローラの情報を表示する

この例では、アドレス 9C の IPM コントローラの基本情報を示します。

```

# clia ipmc 9c
Pigeon Point 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 コントローラの詳細情報を表示する

この例では、アドレス 9C の IPM コントローラの詳細情報を示します。

```

# clia ipmc -v 9c
Pigeon Point 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: "PPS 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` コマンドを実行すると、システム内のすべての FRU の情報を表示できます。または FRU のアドレス、および必要に応じて FRU ID を指定すると、特定の FRU の情報を表示できます。シャーシの-slot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
Pigeon Point Shelf Manager Command Line Interpreter

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"
```

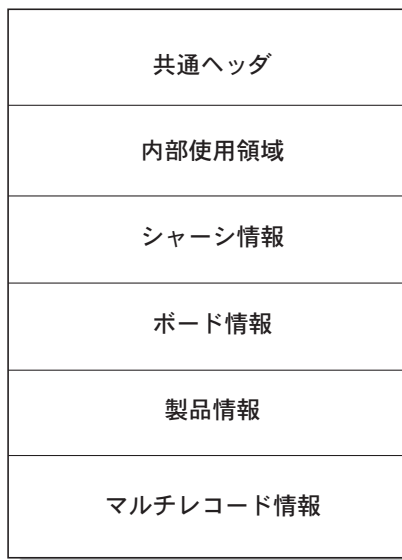
## IPMI FRU 情報の配置

[図 3-1](#) に、IPMI FRU 情報の構成を示します。各パーティションには、特定のタイプのデータがあります。

- 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

Sun Netra CT900 サーバーのスロット 7 および 8 にあるスイッチボードには、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

次のタスク例では、使用するコマンドとその出力を示します。

- 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
- アドレス 8e にあるノードボードの FRU #0 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
Pigeon Point 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: "PPS 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: (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 # 4
Entity: (0x1e, 0x1)
Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
Device ID String: "Fan Tray 1"

20: FRU # 5
Entity: (0x1e, 0x2)
Device ID String: "Fan Tray 1" Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
Device ID String: "Fan Tray 2"

20: FRU # 6
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 # 7
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"

20: FRU # 8
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"

82: FRU # 0
Entity: (0xa0, 0x60)
Hot Swap State: M4 (Active), Previous: M4 (Active, Last State
Change Cause: Normal State Change (0x0)
Device ID String: "NetraCP-3140"

84: FRU # 0
Entity: (0xa0, 0x60)
Hot Swap State: M4 (Active), Previous: M4 (Active, Last State
Change Cause: Normal State Change (0x0)
Device ID String: "NetraCP-3140"

8e: FRU # 0
Entity: (0xa0, 0x60)
Hot Swap State: M4 (Active), Previous: M4 (Active), Last State
Change Cause: Normal State Change (0x0)
Device ID String: "NetraCP-3010"

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-3020"

```

- 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 9a
Pigeon Point Shelf Manager Command Line Interpreter

9a: FRU # 0
Entity: (0ad0, 0x60)
Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
Device ID String: "NetraCP-3020"

#

```

- 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 のシャーシ上の 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
Pigeon Point 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 Defined Record
Version = 2
Manufacturer ID = 0x303833

```

- アドレス 8e にあるノードボードの FRU #0 に関する詳細情報を表示する

この例では、物理アドレス 8e にあるノードボード上の FRU デバイス ID 0 に関して、詳細な FRU 情報が表示されています。

```

# clia fruinfo 8e 0
Pigeon Point Shelf Manager Command Line Interpreter

8e: FRU # 0, FRU Info
Common Header:      Format Version = 1

Board Info Area:
Version            = 1
Language Code      = 25
Mfg Date/Time      = Mar 20 00:00:00 2006 (5372640 minutes
since 1996)
Board Manufacturer = Sun Microsystems, Inc.
Board Product Name = NetraCP-3060
Board Serial Number = 00000000000000000001
Board Part Number  = 00000000000005017313

```

```
FRU Programmer File ID = fru-info.inf

Product Info Area:
  Version = 1
  Language Code = 25
  Manufacturer Name = Sun Microsystems, Inc.
  Product Name = NetraCP-3060
  Product Part / Model# = 0000000000005017313
  Product Version = Rev 1.00TEST
  Product Serial Number = 0000000000000000001
  Asset Tag =
  FRU Programmer File ID = fru-info.inf

Multi Record Area:
  PICMG Board Point-to-Point Connectivity Record (ID=0x14)
    Version = 0

  AMC Carrier Information Table Record (ID=0x1a)
    Version = 0

  AMC Carrier Activation and Current Management Record (ID=0x17)
    Version = 0

  AMC Carrier Point-to-Point Connectivity Record (ID=0x18)
    Version = 0

  AMC Carrier Point-to-Point Connectivity Record (ID=0x19)
    Version = 0

  AMC Carrier Point-to-Point Connectivity Record (ID=0x19)
    Version = 0
```

● 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
Pigeon Point 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
Pigeon Point 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
Pigeon Point 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、[296 ページの「シェルフ FRU 情報の表示」](#)を参照してください。

## 例

次のタスク例では、使用するコマンドとその出力を示します。

- [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
Pigeon Point Shelf Manager Command Line Interpreter

Cooling state: "Normal"

# clia shelf -v cooling_state
Pigeon Point 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
Pigeon Point Shelf Manager Command Line Interpreter

Fans state: "Normal"

# clia shelf -v fans_state
Pigeon Point 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
Pigeon Point 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
Pigeon Point 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 Watts
  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
Pigeon Point 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ーの再初期化

この節では、シェルフ管理カードの 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。



**注意** – U-boot を再初期化すると、いくつかの重要な変数がリセットされるため、バックアップ ShMM が何度も再起動する可能性があります。これを防ぐには、再初期化前に既存の変数のコピーを作成し、再初期化後にそれらを適用します。

復元する必要がある変数のリストについては、[24 ページの「U-Boot 変数の変更に関する推奨事項」](#)で、上級ユーザーがデバッグのためにのみ変更する変数のリストと、変更しない変数のリストを参照してください。

## ▼ 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。

```
shmm500 eeprom write 80400000 0 1000  
EEPROM @0x50 write: addr 80400000 off 0000 count 4096 ... done
```

2. 次のように、シェルフ管理カードをリセットします。

```
shmm500 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  
  
YOU MUST HIT ANY KEY NOW TO BREAK THE BOOT PROCESS
```

3. デフォルトの Sun 環境を設定します。

```
shmm500 setenv rc2 /etc/rc.acb3
```

4. 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。

saveenv コマンドは、次のように 2 回実行する必要があります。

```
shmm500 saveenv  
shmm500 saveenv
```

5. 次のコマンドを使用して、シェルフ管理カードを再起動します。

```
shmm500 boot
```

## 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ー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。

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

`flash_reset` 変数は、フラッシュメモリー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 `sunct900` です。ユーザーは、シェルフマネージャ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。

```
# setenv password_reset y
# 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。次のグループがあります。

- 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 `cp1dtool` 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か、アップグレードセッションで問題が発生した場合にこれを判断できる十分に正常なソフトウェアコピーから起動することが保証されます。アップグレードセッションで問題が発生したと判断された場合は、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が実行されます。

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 ([127 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に必要なカスタム処理を、信頼できるアップグレードの適切な時点で確実に実行することができます。この節の以降の部分では、これらのフックを使用して実装された Pigeon Point Shelf Manager ファームウェアの信頼できるアップグレードのプロビジョニング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。

この節は、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 3-4 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 3-4 16M バイトフラッシュのフラッシュパーティション

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)	コンテンツ
0	0.5	/dev/mtdchar10, /dev/mtdblock10	/var/upgrade	/var/upgrade JFFS2 ファイルシステムの後半
0.5 +(FLASH_SIZE -16)/2	1.5	/dev/mtdchar5, /dev/mtdblock5	マウントされま せん	もう一方の /var JFFS2 ファイルシステム
FLASH_SIZE/2 - 62	1	/dev/mtdchar6, /dev/mtdblock6	マウントされま せん	もう一方の /etc JFFS2 ファイルシステム
FLASH_SIZE/2 - 53	1	/dev/mtdchar7	マウントされま せん	もう一方の Linux カーネ ルイメージ
FLASH_SIZE/2 - 44	0.25	/dev/mtdchar8	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
FLASH_SIZE/2 - 3.754.25	3.75	/dev/mtdchar9	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
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

表 3-5 に、シェルフ管理カードで維持管理される 64M バイトフラッシュデバイスのフラッシュパーティションの概要を示します。

表 3-5 64M バイトフラッシュのフラッシュパーティション

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)	コンテンツ
0	0.5	/dev/mtdchar10, /dev/mtdblock10	/var/upgrade	/var/upgrade JFFS2 ファイルシステムの後半
0.5	1	/dev/mtdchar7	マウントされません	「もう一方の」 Linux カーネルイメージ
1.5	1	/dev/mtdchar6, /dev/mtdblock6	マウントされません	「もう一方の」 /etc JFFS2 ファイルシステム
2.5	1.75	/dev/mtdchar5, /dev/mtdblock5	マウントされません	「もう一方の」 /var JFFS2 ファイルシステム
4.25	15.75	/dev/mtdchar9	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
20	8	/dev/mtdchar12, /dev/mtdblock12	マウントされません	app1_jffs アプリケーション固有 JFFS2 パーティションの後半
28	0.25	/dev/mtdchar8	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
28.25	3.75	/dev/mtdchar11, /dev/mtdblock11	マウントされません	app_jffs アプリケーション固有 JFFS2 パーティションの後半
32	0.5	/dev/mtdchar10, /dev/mtdblock10	/var/upgrade	/var/upgrade JFFS2 ファイルシステムの前半
32.5	1	/dev/mtdchar2	マウントされません	Linux カーネル
33.5	1	/dev/mtdchar1, /dev/mtdblock1	/etc	/etc JFFS2 ファイルシステム
34.5	1.75	/dev/mtdchar0, /dev/mtdblock0	/var	/var JFFS2 ファイルシステム
36.25	15.75	/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
52	8	/dev/mtdchar12, /dev/mtdblock12	マウントされません	app1_jffs アプリケーション固有 JFFS2 パーティションの前半
60	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
60.25	3.75	/dev/mtdchar11, /dev/mtdblock11	マウントされません	app_jffs アプリケーション固有 JFFS2 パーティションの後半

## /var/upgrade ファイルシステム

123 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 (127 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かにかかわらず、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 (128 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 から Linux カーネルにカーネルパラメータ文字列で渡される U-Boot 環境変数の `reliable_upgrade` (124 ページの「フラッシュのパーティション分割」を参照) が `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のサポートにより、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で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 (127 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)、新しいアップグレード処理セッションの状態で上書きされます。

`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/step4vshn 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 マウント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、`user` 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が与えられます。この書き込み権は、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。これに加えて、シェルフマネージャの起動時に U-Boot、Linux カーネル、およびルートファイルイメージを含むパーティションがすべて読み取り専用になるため、重要な起動イメージがアプリケーションによって誤って消去されないことが保証されます。

ユーティリティーは、指定したすべてのイメージをフラッシュメモリー内の所定の場所にインストールしたあと、「フック」スクリプトを呼び出します。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) を有効にすることにより、アップグレードイメージがフラッシュにインストールされてから、信頼できるアップグレード処理のハードウェア機構がアップグレード処理によって開始されるまでの間に、アプリケーションに必要なカスタム処理を実行できるようにします。

フックスクリプト (`/etc/upgrade/step4hshm`) は、シェルフマネージャに付属しています。このスクリプトは次のアクションを実行します。

- シェルフマネージャを終了し、シェルフを再起動せずにバックアップ 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は停止します。
- 暫定的な `/etc` および `/var` フラッシュパーティションをマウントし、これらのパーティション上のファイルをすべて消去します。
- オプションで、`/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。
- オプションで、現在の非揮発性シェルフマネージャ情報を `/var/nvdata` ディレクトリから暫定 `/var` ファイルシステムにコピーします。または、オプションで、`/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。
- 一時的に (次の起動まで) 起動遅延を 0 に設定します。この目的は、次の起動の時間を最小限に抑え、信頼できるアップグレードのウォッチドッグタイマーの有効期間が早期に期限切れになるのを防止することです。

このスクリプトは、サブシェルとして呼び出され、引数で指定された文字列または引数なし (空文字列) のいずれか 1 つのパラメータを渡されます。渡されるパラメータにより、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処理を決定する処理モードが定義されます。スクリプトは次の引数値を受け取り、対応するアクションを実行します。

- パラメータなし - スクリプトは暫定 /etc ディレクトリと暫定 /var ディレクトリの両方を消去し、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処理モードです。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 ディレクトリと暫定 /var ディレクトリの両方を消去します。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と構成は保持されません。
- etc\_copy - スクリプトは暫定 /etc ディレクトリと暫定 /var ディレクトリの両方を消去し、/etc の内容と非揮発性情報を /var/nvdata から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 ディレクトリと暫定 /var ディレクトリの両方を消去し、/etc ディレクトリと /var ディレクトリの内容全体を暫定パーティションにコピーします。この場合、構成だけでなく、/var/bin に配置されている実行可能ファイルもコピーされ、RFS イメージから取得した同じ名前の実行可能ファイルが上書きされます。この処理モードは、アップグレード後も保持する必要がある特別な実行可能ファイル (特別なバージョンのシェルフマネージャーやその他のユーティリティなど) が /var/bin ディレクトリに格納されている場合に便利です。

スクリプトは、成功した場合には 0 を、失敗した場合には 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、タイムアウト時間が 12.8 秒に設定されたアップグレード WDT を起動します。このタイムアウト時間は、リセット後に起動するソフトウェアが、しばらくの間アップグレード WDT をストローブすることなく、`rupgrade_tool -c` (アップグレード WDT がアクティブになっている場合にストローブします) を呼び出せるポイントまで進むのに十分な時間であるとみなされます。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したあとに、信頼できるアップグレード処理を開始します。`rupgrade_tool -c` は /etc/rc スクリプトから呼び出されます。以下で説明するように、`rupgrade_tool -c` によって検出された特定の状況は、アップグレード処理の失敗を示し、修正措置を必要とします。これには、ShMM のソフトリセットを必要とする状況が含まれます。これは、`rupgrade_tool -c` の呼び出しが戻らず、ShMM のリセットを引き起こす場合があることを意味します。リセットが行われると、ShMM は永続フラッシュにインストールされたソフトウェアに戻されます。

アップグレード WDT がアクティブになっていて、`rupgrade -c` の呼び出しよりも前のステップで起動した場合は、ShMM がすでに永続フラッシュ内のソフトウェアに戻ったことを意味します。この場合、ユーティリティーはアップグレード WDT を無効にし、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 がアクティブになっているにもかかわらず、まだ起動していない場合は、ShMM が暫定フラッシュから正常に起動した (この時点まで) ことを意味します。ユーティリティーはアップグレード 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 が非アクティブになっているにもかかわらず、`/var/upgrade/status` ファイルの内容が、アップグレード処理が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ステップで電源が再投入されたことにより、ShMM が再起動したことを意味しています。この場合、ユーティリティーは、アップグレード WDT がアクティブになっていて起動している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 が非アクティブになっていて、`/var/upgrade/status` が存在しないか、または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、ユーティリティーは進行中のアップグレード処理がないことを示すリターンコード 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初期化を正常に完了したあとに、シェルフマネージャーの内部で行われます。シェルフマネージャーが自動的に起動しない場合は、`/etc/rc` スクリプトの最後に呼び出しが行われます。

`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とアップグレード WDT の無効化を開始します。

ユーティリティーは、処理を完了する前に、アップグレード処理が正常に完了したことを示すレコードで `/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成功した場合には値 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それぞれ返します。

`-f` 指定子が指定されている場合、`rupgrade_tool -w` は終了する前に `/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証するときにシェルレベルのインタフェースとして使用します。

`rupgrade_tool -S` は値 0 を返します。

- -u

成功したアップグレードセッションを取り消して、前の永続フラッシュデバイスに戻ります。

`rupgrade_tool -u` は ShMM を再起動します。

- -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ーの使用シナリオ

ShMM のアップグレードを実行する場合は、信頼できるアップグレードユーティリティーを次の順序で使用します。

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行くことも、`telnet`、`rsh`、`ssh`、またはこれらと同等のユーティリティーを使用してネットワーク経由でリモートから行くこともできます。
2. ユーザーは `rupgrade_tool -s` によって ShMM が再起動されるまで待機します。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、再起動の状態は U-Boot ファームウェアおよび Linux によってシリアルコンソールに出力されるメッセージから明らかです。ShMM への接続がリモートの場合、再起動の状態はローカル接続と比べて不明瞭です。たとえば、Telnet 接続は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みなすか、アップグレードセッションの完了に必要な時間が経過するまで待機してから、`rupgrade_tool -w` (このコマンドも上記のリモートシェルツールのいずれかを使用してリモートから呼び出します) を呼び出してアップグレードセッションの状態を確認できます。待機する時間は、アップグレードイメージのサイズ、シェルフ管理カードにイメージをコピーするのに使用するプロトコル、およびイメージ検証スクリプトによって実行されるアクションによって異なります。
3. シェルフ管理カードで、起動スクリプト `/etc/rc` が無条件で `rupgrade_tool -c` を呼び出します。この呼び出しから進行中のアップグレードがないことを示す値 1 が返された場合、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た場合、起動スクリプトは通常モードの起動シーケンスを開始します。ただし、アップグレードセッションが進行中であることを示す値 0 が返された場合、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の検証を開始します。この検証にアップグレード WDT タイムアウト時間よりも長い時間がかかる場合は、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、最後にシェルフマネージャーを起動して最終的な検証を実行します。ウォッチドッグタイマーの間隔が 12.8 秒に設定されているため、`/etc/rc` スクリプトでの `rupgrade_tool -c` の呼び出しから WDT のストローブまでの時間、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時間はともに 12.8 秒以下でなければなりません。

4. 初期化中、シェルフマネージャーは、ピア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る前に、アップグレード WDT をもう一度ストローブします。ネットワーク接続の確立には、最大で 6 秒かかることがあります。その後、初期化が正常に完了した (これは新しい構成が妥当であることを示します) 後に、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. ユーザーがオプションで `rupgrade_tool -w` を呼び出し、アップグレードセッションの状態を確認します。上記で説明したように、このオプションはリモートアップグレードセッションの場合に特に役立ちます。リモートアップグレードセッションでは、ローカルアップグレードの場合とは異なり、シリアルコンソールに出力されるメッセージでアップグレードの進捗状況を直接確認することができないからです。

信頼できるアップグレードが完了したあと、何らかの理由で新しいイメージを受け入れることができない場合、ユーザーは元のイメージに戻すことができます。これを行うには、`rupgrade_tool -u` を呼び出します。

必要な場合は、リモートネットワークホスト上で実行できる簡単なスクリプトを作成することで、上記のシーケンスを簡単に自動化できます。または、オペレータが信頼できるアップグレードを手動で (シリアルコンソールからローカルに、またはネットワーク経由でリモートから) 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:

この例では、3 つのコンポーネントすべて (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示し、`/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ます。イメージはすべてローカルの `/tmp` から取得します (これらのイメージがすでに何らかの方法でこの場所にコピーされていることを示します)。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` からそれぞれ取得します。アップグレード処理はシリアルコンソールから開始します。アップグレード処理の各ステップに関する追加情報を提供するために、コンソールログの所々にコメントが挿入されています。

まず、コマンドプロンプトから `rupgrade_tool` を起動します。これらのパラメータは、3 つのフラッシュイメージをすべて更新し、それとともにシェルフマネージャーの非揮発性データと構成ファイルを保持す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ーは、検証スクリプトを呼び出し、/tmp 内に現在格納されているイメージを確認しようとします。指定されたファイル指示子のうちのいずれかが /tmp 内に見つからなかった場合、ユーティリティーは処理を停止し、次のようなメッセージを生成します。

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

ユーティリティーは、暫定フラッシュ内の指定した場所にイメージをコピーします。

```
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呼び出されます。このスクリプトは、シェルフマネージャーを停止し、暫定フラッシュに非揮発性情報をコピーします。

```
/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によって ShMM がリセットされます。これにより、U-boot が暫定フラッシュから起動します。

```
* 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: AulX00 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` を呼び出し、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。このツールは 0 を返し、アップグレードが進行中であることを確認します。この結果を受けて、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であるため、rc スクリプト内でウォッチドッグタイマーがもう一度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] Pigeon Point 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=6F or 2A (Sun
legacy)b0d2e4
<*> 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を完了します。

```
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:

この例では、RFS イメージのみの信頼できるアップグレードを示し、`/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 サーバーから取得します。FTP サーバー上の RFS イメージのパスは `/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します。

---

注 - RFC イメージのみが明示的に更新されるため、U-Boot およびカーネルイメージは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自動的にコピーされます。

---

ローカルシステムはネットワーク経由で FTP サーバーにアクセスできる必要があります (つまり、ローカルシステムのネットワークアダプタが構成済みで稼動していて、ShMM から FTP サーバーへのルートが存在している必要があります)。この例では、ShMM にネットワークアドレス 192.168.1.174 が構成されています (ShMM は FTP サーバーと同じネットワークにあります)。

```
# 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 のパラメータは、RFS のみがアップグレードされること、およびコピープロトコルは FTP で、パスワードを指定せずにユーザー admin で指定した IP アドレスとファイルに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
```

ユーザーは FTP サイトのパスワードを要求されます。この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.
```

次のステップでは、特別なスクリプト `step4vsh` が呼び出されます。このスクリプトは、U-Boot およびカーネルイメージを永続フラッシュから暫定フラッシュにコピーします。その後、アップグレードユーティリティーが暫定フラッシュ内の指定の位置に RFS イメージをコピーします。

```
rupgrade_tool: invoking scripts (step4v*) [--r=sentry.mips.rfs --
proto=ftp:192.168.1.253:/tftpboot/ru-mips:admin --hook=etc_copy]
/etc/upgrade/step4vsh: Erasing /dev/mtdchar7...Done
etc/upgrade/step4vsh: Copying Kernel from /dev/mtdchar2 to
/dev/mtdchar7...Done
/etc/upgrade/step4vsh: Erasing /dev/mtdchar8...Done
/etc/upgrade/step4vsh: Copying U-Boot from /dev/mtdchar3 to
/dev/mtdchar8...Done
rupgrade_tool: copying sentry.mips.rfs from /tmp to /dev/mtdchar9
using 'cp' protocol
```

`step4hsh` フックスクリプトが呼び出されます。このスクリプトは、シェルマネージャーを停止し、非揮発性データを保持します。その後、ユーティリティーはアップグレード WDT を起動し、再起動します。

```
rupgrade_tool: invoking scripts (step4h*) [etc_copy]
/etc/upgrade/step4hsh: Stopping Shelf Manager...
/etc/upgrade/step4hsh: Erasing /var and /etc, copying
/var/nvdata..
/etc/upgrade/step4hsh: Operation: copy /etc and /var/nvdata.
/etc/upgrade/step4hsh: 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、一定の非活動時間が経過したあとに Telnet セッションが終了します。数秒後、ターゲットに再接続して `rupgrade_tool -w` を呼び出し、信頼できるアップグレードの状態を確認できるようになります。

```
# 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:

この例では、信頼できるアップグレードの失敗を示します。暫定フラッシュからの起動後、信頼できるアップグレードが完了する前に、電源が切断されます。電源が再投入されたあとに、永続フラッシュへのロールバックが実行されます。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します。3つのイメージはすべて /tmp 内にすでに存在しているものとします。

```
# 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...

Pigeon Point 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によって ShMM がリセットされ、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 (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、電源が再投入されます。電力損失のために暫定フラッシュの割り当てが失われ、システムが永続フラッシュに戻ります。

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

CPU: Aul550 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: AulX00 ETHERNET
Hit any key to stop autoboot: 0
## Booting image at bfb00000 ...
Image Name: MIPS Linux-2.4.26
Created: 2005-04-11 10:35:08 UTC
Image Type: MIPS Linux Kernel Image (gzip compressed)
Data Size: 843129 Bytes = 823.4 kB
Load Address: 80100000
Entry Point: 802bc040
Verifying Checksum ... OK
Uncompressing Kernel Image ... OK
## Loading Ramdisk Image at bfc40000 ...
Image Name: sentry RFS Ramdisk Image
Created: 2005-04-11 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 スクリプトの次のステップは、`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。これにより、信頼できるアップグレードが失敗したことが判明します。「restoring ADM1060 EEPROM to RAM」というメッセージは、ShMM システムの監視デバイス (ADM1060) を指しています。ADM1060 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をサポートするいくつかのハードウェアの側面を実装します。このメッセージは、起動プロセスに影響を与える主な変数が、信頼できるアップグレードを試行する前の状態に復元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
```

---

## ノードボードコンソールへの接続

シェルフマネージャには、アクティブなシェルフ管理カード (ShMM) からノードボードに接続してコンソールを開く機能があります。この機能は一般に「ネットコンソール」と呼ばれます。アクティブな ShMM に Ethernet ポート経由でログインすることから始まります。この機能に ShMM のシリアルポートを使用することはお勧めしません。

---

**注** – ネットコンソール機能を使用するには、シェルフ管理カードがアクティブなシェルフ管理カードである必要があります。また、サーバーのミッドプレーンのスロット 7 にスイッチカードが取り付けられている必要もあります。

---

ノードボードとのコンソールセッションが確立したら、`passwd` などのシステム管理コマンドを実行したり、状態とエラーメッセージを読み取ったり、その特定のスロットのボードを停止したりできます。

---

**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、コンソール出力は ShMM 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出力されます。これはケーブルの接続時に ShMM のコンソールセッションがアクティブになっていた場合でも同様です。

---

## シェルフマネージャとノードボード間でのコンソールセッションの確立

コンソールを使用できるようにシステムを構成したら、アクティブな ShMM にログインしてスロットのコンソールを開くことができます。Sun Netra CT システムでは、ノードボードスロットごとに 1 つのコンソールセッションを開くことができます。表 3-6 に、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る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。

表 3-6 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 [-f] slot-number</code>	コンソールモードを開始し、 <i>slot-number</i> のノードボードに接続します。 <i>slot-number</i> は、ノードボードが挿入されているスロットの番号です。 <code>-f</code> オプションを指定すると、新しいコンソールセッションが開始する前に既存のコンソールセッションが終了します。これにより、強制的に新しいコンソールセッションになります。
<code>~q</code> または <code>~.</code>	現在のコンソールセッションから切断します。

---



## ▼ 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

### 1. ShMM にログインします。

ShMM には、SAP 上のシリアルポート接続または Ethernet ポート接続に接続された端末からログインできます。

### 2. ShMM がアクティブな ShMM かどうかを確認します。

ログインしたら、次の作業に進む前に、`clia shmstatus` コマンドを使用して、アクティブな 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、SAP 上のほかのシリアルポートに接続するか、`clia switchover` コマンドを使用して ShMM を Active に変更します (詳細は、[153 ページの「シェルフマネージャーの 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-number
```

`slot-number` は 1 - 6 および 9 - 14 です。たとえば、スロット 4 のボードへのコンソールを開くには、次のように入力します。

```
# clia console 4
```

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

- `console login%` (Oracle Solaris レベル)
- `#` (Oracle Solaris レベルで、前のユーザーがスーパーユーザーとしてログインし、コンソールから切断する前にログアウトしなかった場合)
- `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 シェルからログアウトします。
2. 次に示すように、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断します。

```
prompt ~q
```

コンソールから切断しても、リモートホストから自動的にログアウトすることはありません。リモートホストからログアウトしない限り、そのボードに次に接続するコンソールユーザーに対して前のセッションのシェルプロンプトが表示されます。

---

注 - 作業を終えたら、必ずコンソールセッションからログアウトしてください。

---

## コンソールのエスケープ文字の変更

コンソールセッションを切断するには、通常、`~.` または `~q` エスケープシーケンスを使用します。チルダ (`~`) はコンソールのデフォルトのエスケープ文字です。このエスケープ文字は変更できます。これを行うには、`/etc/netconsole.conf` ファイルの `Escape_Char` 変数の値を変更します。たとえば、`Escape_Char="#"` と指定すると、エスケープ文字がポンド記号 (`#`) に変更され、コンソールを切断するエスケープシーケンスが `#`、または `#q` に変更されます。変更を有効にするには、アクティブなコンソールセッションをすべて再起動する必要があります。

`Escape_Char` 変数が存在しないか未定義の場合、エスケープ文字はデフォルトで `~` になります。

## コンソールセッション中のボードのユーザーラベルの表示

管理者は、ShMM CLI の `setuserlabel` コマンド ([294 ページの「setuserlabel」](#) および [335 ページの「userlabel」](#) を参照) を使用して、シェルプロンプトとスロットに名前 (またはユーザーラベル) を割り当てることができます。ユーザーラベルは `shelf-name: slot-name` という形式で表示されます。

ユーザーは、コンソールセッション中にエスケープ文字と `?` または `L` を入力することで、セッションのユーザーラベルを表示できます。デフォルトのエスケープ文字 `~` (チルダ) を使用する場合、エスケープシーケンスは `~?` または `~L` になります。

例:

```
# clia console 6
console login
. . . . .
# ~?
Connected to "ATCA02:CP3020-06" console.
#
```

---

## 手動によるノードボードの正常な停止

Sun Netra CP3010、CP3020、および CP3060 ノードボードでは、ノードボードまたは FRU が誤って取り外されるのを防止するために、手動による正常な停止の手順が用意されています。正常な停止とは、ノードボードのペイロードで実行され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段階的に休止 (停止) することです。

---

**注** – Sun Netra CP3220、CP3250、および CP3260 ノードボードでは、無効化が有効になっている場合、ボード上のホットスワップラッチが開くと、自動的に正常な停止が実行されます。

---

これらのいずれかの Sun 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には、最初にノードボード上で実行されているアプリケーションとオペレーティングシステムを正常に停止する必要があります。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上でコンソールセッションを開始し、サイトの停止手順に従って正常な停止を実行することができます。

停止が完了したら、FRU (またはノードボード) の無効化を有効にし、ボード上のホットスワップラッチを開きます。ノードボードの交換または再取り付けが完了したら、FRU (またはノードボード) の無効化を無効にし、ホットスワップラッチを開いたときに異常な方法での停止が行われないようにします。

このプロセスに関連する詳細な手順を次に示します。

## ▼ ノードボードを停止する

この手順を実行するには、ノードボード上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場合、青色のホットスワップ LED が点滅し続け、ボードがホットスワップ準備完了 (青色の LED の常時点灯で示されます) 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ます。

---

1. アクティブな 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。  
[147 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とオペレーティングシステムを停止します。  
サイトの正常な停止の順に従います。停止が完了したら、コンソールセッションを閉じます ([コンソールセッションを終了する](#)を参照)。
4. 次のシェルフマネージャー 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された非アクティブ化を有効にします。

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

例:

```
# clia shelf deactivation 0x41 0xfe 0
Pigeon Point Shelf Manager Command Line Interpreter

Updating Shelf FRU Info, address: 0x41, FRU ID # 254
  Cached information updated
  Wrote Information to the Shelf FRU

#
```

5. 次のシェルフマネージャー CLI コマンドを使用して、シェルフマネージャーの制御された非アクティブ化が有効になっていることを確認します。

```
# clia shelf pm
```

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

例:

```
# clia shelf pm
Pigeon Point 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. ノードボード上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 が常時点灯しているときに、ノードボードの取り外しまたは交換を行います。



---

**注意** – モジュールを取り扱うときには、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. ノードボードを交換または再取り付けしたら、次のシェルフマネージャー 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された非アクティブ化を無効にします。

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

例:

```
# clia shelf deactivation 0x41 0xfe 1
Pigeon Point 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 コマンドについて説明し、使用可能な各コマンドの構文と使用法を示します。CLI では、AdvancedTCA シェルフコンテキストがサポートされます。

便宜上、IPMB アドレスと数値の FRU 識別子に基づく参照表記法の代わりに、次の方法でシェルフコンポーネントの主要な種類を参照できるようになっています。

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

---

**注** - shm 1 および shm 2 を使用すると、シェルフ FRU のアドレステーブルに記載されている冗長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 はハードウェアアドレスの数値が小さいシェルフマネージャに関連し、shm 2 はハードウェアアドレスの数値が大きいシェルフマネージャに関連します。

---

冗長構成では、以下に示すコマンドの一部がバックアップシェルフマネージャでサポートされません。バックアップシェルフマネージャでは、次のコマンドのみが認識されます。

- debuglevel
- localaddress
- shmstatus
- switchover

ほとんどの情報コマンドでは、簡略モードおよび冗長モードの実行形式をサポートしています。これらの形式では提供す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。冗長モードを選択するには、コマンド行でコマンドの直後、位置引数の前にオプション `-v` を指定します。

以下のコマンド構文では、省略可能な要素は角括弧 ([, ]) で囲まれ、コマンド行の変数要素 (IPMB アドレスや FRU デバイス ID など) は斜体で表示されます。パラメータの選択肢は垂直バー (|) で区切られます。

TELCO アラーム (マイナー、メジャー、およびクリティカル) の読み取りおよび設定のために SNMP がサポートされます。

---

## activate

### 構文:

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

### 目的:

このコマンドは、指定した FRU に IPMI コマンド「FRU のアクティブ化の設定 (FRU のアクティブ化)」を送信します。FRU は、所有している IPM コントローラの IPMB アドレスと FRU デバイス ID を使用して指定します。FRU デバイス ID 0 は、PICMG 3.0 コンテキストの IPM コントローラ本体を指定します。

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記載されていない FRU、または「シェルフマネージャーによって制御されるアクティブ化」属性が 0 (DISABLED) に設定されている FRU に対して有効です。これらの FRU は、シェルフマネージャーによって自動的にアクティブ化されず、M2 (アクティブ化要求) 状態のままになります。その他の FRU は、M2 状態になると自動的にシェルフマネージャーによってアクティブ化されます。M2 状態になっていない FRU をアクティブ化しようとしても、何も行われません。

### 例:

アドレス 9C にある IPM コントローラ本体をアクティブ化します。

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



---

# alarm

## 構文:

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

## 目的:

このコマンドは、Telco アラーム出力へのアクセスを提供します。minor、major、および critical の各パラメータを使用すると、対応するアラーム出力を設定できます。これらのアクションは累積的です。つまり、コマンド `clia alarm minor` のあとに `clia alarm major` を実行すると、マイナーアラームとメジャーアラームの両方が設定されます。アクション `clear` は、マイナーアラーム出力とメジャーアラーム出力をクリアします。クリティカルアラーム出力はクリアできません。アクション `info` は、シェルフで最後に発生したアラームに関する情報を表示します。

パラメータを指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
Pigeon Point Shelf Manager Command Line Interpreter

    alarm mask: 0x00
# clia alarm major
Pigeon Point Shelf Manager Command Line Interpreter

Returned completion code: 0
# clia alarm
Pigeon Point Shelf Manager Command Line Interpreter

    alarm mask: 0x02
    Major Alarm
# clia alarm clear
Pigeon Point Shelf Manager Command Line Interpreter

Returned completion code: 0
# clia alarm
Pigeon Point Shelf Manager Command Line Interpreter

alarm mask: 0x00
# clia alarm info
```

```
Pigeon Point Shelf Manager Command Line Interpreter
```

```
Last saved alarm information:  
Alarm mask: 0x02  
Alarm date/time: Wed May 10 10:54:04 2007  
Alarm source: Remote request  
Alarm reason: On-demand setting alarms mask: 0x02  
#
```

---

## airfilterreplaced

### 構文:

```
airfilterreplaced [dd.mm.yyyy]
```

### 目的:

シェルフ FRU 情報の Pigeon Point 定義のマルチレコードでは、ファントレーのエアフィルタの交換日を管理できます。このマルチレコードには、エアフィルタの交換に関する次の情報が含まれます。

- 「エアフィルタの交換日」 – エアフィルタが最後に交換された日付を示します。形式は *dd.mm.yyyy* です。
- 「エアフィルタの交換予定日」 – エアフィルタの再交換が必要になる日付 (フィルタの有効期限が切れる日付) を示します。形式は *dd.mm.yyyy* です。

オペレータは、エアフィルタを交換したあと、シェルフマネージャー CLI ツールを使用して、シェルフ FRU 情報の上記のデータを変更する必要があります。

コマンドパラメータとして指定する日付 *dd.mm.yyyy* は、フィルタの有効期限が切れる日付 (エアフィルタの再交換が必要になる日付) を示します。この日付が省略された場合、デフォルトの有効期限は現在の日付から 6 か月後です。

このコマンドを実行すると、シェルフマネージャーによってシェルフ FRU 情報が次のように更新されます。「エアフィルタの交換日」フィールドには現在のカレンダー日付が格納され、「エアフィルタの交換予定日」フィールドには有効期限が切れる日付 (コマンドで指定した日付か、デフォルトの 6 か月後の日付) が格納されます。

---

**注** – このコマンドは、キャリア固有の特別なサポートを必要とし、すべての ShMM キャリアに実装されているわけではありません。このコマンドが現在のキャリアに実装されていない場合、コマンドを使用するとエラーメッセージが表示されます。

---

例:

エアフィルタの次の交換日をシェルフ FRU 情報に設定します。

```
# cli airfilterreplaced 25.10.2007
Pigeon Point Shelf Manager Command Line Interpreter

Command executed successfully
#
```

---

## amcportstate

構文:

```
amcportstate [-v] IPMB-address [fru-id | amc amc-number]
amcportstate [-v] board n [fru-id | amc amc-number]
amcportstate [-v] shm n [fru-id | amc amc-number]
```

目的:

このコマンドは、特定の AMC の AMC ポート状態情報を表示します。*fru-id* または *amc-number* が省略された場合、指定した IPM コントローラでアクティブなすべての AMC の AMC ポート状態情報が報告されます。

例:

```
# cli amcportstate 98

Pigeon Point Shelf Manager Command Line Interpreter

98: FRU # 1 (AMC # 5)
  Channel 0:
    Link 1 configuration:
      lane mask 03, type 07, type extension 02, grouping ID
00, status 0 (Disabled)
    Link 2 configuration:
      lane mask 01, type 07, type extension 02, grouping ID
00, status 1 (Enabled)
    Link 3 configuration:
      lane mask 02, type 07, type extension 02, grouping ID
00, status 0 (Disabled)

98: FRU # 2 (AMC # 6)
```

```
Channel 0:
  Link 1 configuration:
    lane mask 03, type 07, type extension 02, grouping ID
00, status 0 (Disabled)
  Link 2 configuration:
    lane mask 01, type 07, type extension 02, grouping ID
00, status 1 (Enabled)
  Link 3 configuration:
    lane mask 02, type 07, type extension 02, grouping ID
00, status 0 (Disabled)
```

```
# clia amcportstate 9c 2
```

```
Pigeon Point Shelf Manager Command Line Interpreter
```

```
9C: FRU # 2 (AMC # 6)
```

```
Channel 0:
  Link 1 configuration:
    lane mask 0f, type 05, type extension 01, grouping ID
00, status
1 (Enabled)
  Channel 1:
    Link 1 configuration:
      lane mask 0f, type 05, type extension 01, grouping ID
00, status
1 (Enabled)
  Channel 2:
    Link 1 configuration:
      lane mask 01, type f0, type extension 00, grouping ID
00, status
1 (Enabled)
```

```
# clia amcportstate 88 amc 6
```

```
Pigeon Point Shelf Manager Command Line Interpreter
```

```
88: FRU # 2 (AMC # 6)
```

```
Channel 0:
  Link 1 configuration:
    lane mask 01, type 07, type extension 01, grouping ID
00, status 1 (Enabled)
  Channel 1:
    Link 1 configuration:
      lane mask 01, type 07, type extension 01, grouping ID
00, status 0 (Disabled)
```

---

# board

## 構文:

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

## 目的:

このコマンドと `boardreset` コマンドは、ATCA ボードと連携し、IPM コントローラ アドレスと FRU デバイス ID の代わりに物理スロット番号を引数として受け取るという点でほかのコマンドとは異なります。

`board` コマンドは、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 に関する情報を表示します。表示される項目のリストは、[176 ページの「fru」](#) と [237 ページの「ipmc」](#) に記載されています。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 進数で指定する必要があります。PICMG 3.0 システムの場合、物理アドレスと IPMB アドレスの対応付けはシェルフ 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ていない場合は、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情報を取得します (この例では、物理スロット 4、5、および 8 にのみボードがあります)。

```
# clia board
Pigeon Point Shelf Manager Command Line Interpreter

Physical Slot # 4
8e: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x01
    PICMG Version 2.1
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)

8e: 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-3060"

8e: FRU # 1
    Entity: (0xc1, 0x61)
    Hot Swap State: M4 (Active), Previous: M7 (Communication Lost),
Last State Change Cause: Normal State Change (0x0)
    Device ID String: "SB AMC-HD-A-80X"

Physical Slot # 5
8a: 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)

8a: 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-3020"

Physical Slot # 8
84: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x01
    PICMG Version 2.1
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)

84: 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: "CP3140H-BEG"
```

```
84: FRU # 1
```

```
Entity: (0xc0, 0x60)
Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
Device ID String: "XCP3040H-RTC"
```

```
#
```

物理スロット 4 のボードに関する詳細情報を取得します。

```
# clia board -v 4
```

```
Pigeon Point Shelf Manager Command Line Interpreter
```

```
Physical Slot # 4
```

```
8e: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x01
```

```
PICMG Version 2.1
```

```
Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
```

```
Device ID: 0x00, Revision: 0, Firmware: 0.210, IPMI ver 1.5
```

```
Manufacturer ID: 00006F or 2A (Sun legacy), Product ID: 0bf4,
Auxiliary Rev: 00000000
```

```
Device ID String: "NetraCP-3060"
```

```
Global Initialization: 0xc, Power State Notification: 0xc,
Device Capabilities: 0x29
```

```
Controller provides Device SDRs
```

```
Supported features: 0x29
```

```
"Sensor Device" "FRU Inventory Device" "IPMB Event
Generator"
```

```
8e: Base Interface (0x00), Channel: 1
```

```
Link: Enabled Ports: 1
```

```
Peer Addr: 0x82, Link Type: PICMG 3.0 Base Interface
10/100/1000 BASE-T, Ext: 0 (10/100/1000 BASE-T)
```

```
8e: Base Interface (0x00), Channel: 2
```

```
Link: Enabled Ports: 1
```

```
Peer Addr: 0x84, Link Type: PICMG 3.0 Base Interface
10/100/1000 BASE-T, Ext: 0 (10/100/1000 BASE-T)
```

```
8e: Fabric Interface (0x01), Channel: 1
```

```
Link: Enabled Ports: 1
```

```
Peer Addr: 0x82, Link Type: PICMG 3.1 Ethernet Fabric
Interface, Ext: 0
```

```
8e: Fabric Interface (0x01), Channel: 2
```

```
Link: Enabled Ports: 1
```

```
Peer Addr: 0x84, Link Type: PICMG 3.1 Ethernet Fabric
Interface, Ext: 0

8e: 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-3060"
  Site Type: 0x00, Site Number: 04
  Current Power Level: 0x01, Maximum Power Level: 0x01, Current
Power Allocation: 174.0 Watts

8e: FRU # 1
  Entity: (0xc1, 0x61)
  Hot Swap State: M4 (Active), Previous: M7 (Communication Lost),
Last State Change Cause: Communication Lost due to local failure
(0x5)
  Device Type: "FRU Inventory Device behind management
controller" (0x10), Modifier 0x0
  Device ID String: "SB AMC-HD-A-80X"
  Current Power Level: 0x01, Maximum Power Level: 0x01, Current
Power Allocation: 21.1 Watts
#
```

---

## boardreset

### 構文:

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

### 目的:

このコマンドは、指定した物理スロットのボードに IPMI コマンド「FRU 制御 (コールドリセット)」を送信して、ボードをリセットします。

物理アドレスは 10 進数で指定する必要があります。PICMG 3.0 システムの場合、物理アドレスと IPMB アドレスの対応付けはシェルフ 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ていない場合は、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
10	8C
11	90
12	94
13	98
14	9C

例:

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

```
# clia boardreset 14
Pigeon Point 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ます。

- `info [resource]`
- `release resource`

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

#### 目的:

このコマンドは、バス型 E-キーイング管理リソースの現在の状態に関する情報を表示し、その状態を変更できるようにします。

サブコマンドはすべて、リソース ID をパラメータの 1 つとして受け入れます。リソース ID は、0 から始まるリソース番号か短いリソース名です。次のリソース名とリソース番号が定義されています。

番号	短い名前	説明
0	mtb1	メタリックテストバスのペア 1
1	mtb2	メタリックテストバスのペア 2
2	clk1	同期クロックのグループ 1
3	clk2	同期クロックのグループ 2
4	clk3	同期クロックのグループ 3

以降の節では、`busres` コマンドのいくつかの主な用途に対応する構文について説明します。

## バス型 E-キーイング管理リソースの状態の表示

#### 構文:

```
busres info [resource]
```

#### 目的:

このコマンドは、指定したリソース、またはリソース ID を指定し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` はリソース ID です。サポートされているリソース ID のリストは、[163 ページの「busres」](#)に記載されています。

例:

メタリックテストバスのペア 2 の状態に関する情報を取得します。

```
# clia busres info mtb2
Pigeon Point Shelf Manager Command Line Interpreter

Metalic Test Bus pair 2 (ID 1): Owned by IPMC 0x82, Locked
#
```

## 指定したリソースの解放

構文:

```
busres release | force resource
```

目的:

このコマンドは、リソースの現在の所有者に「バス型に接続された資源の制御」要求を送信し、リソースを解放するように指示します。コマンド構文が `busres release resource` の場合は、「バス型に接続された資源の制御 (解放)」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、バス型に接続された資源の制御 (強制)」コマンドが送信されます。これらの ATCA コマンドの詳細は、PICMG 3.0 R1.0 仕様の 3.7.3.4 節を参照してください。

パラメータ `resource` はリソース ID です。サポートされているリソース ID のリストは、[163 ページの「busres」](#) に記載されています。

例:

現在の所有者にメタリックテストバスのペア 2 を解放するように強制します。

```
# clia busres force mtb2
Pigeon Point Shelf Manager Command Line Interpreter

Force operation succeeded
#
```

## 指定したリソースのロック/ロック解除

### 構文:

```
busres lock | unlock resource
```

### 目的:

このコマンドは、指定したリソースをロック (`busres lock resource`) またはロック解除 (`busres unlock resource`) します。リソースがロックされている場合、別の IPM コントローラが「バス型に接続された資源の制御 (要求)」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で応答します。リソースがロック解除されている場合、別の IPM コントローラが「バス型に接続された資源の制御 (要求)」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で応答し、「バス型に接続された資源の制御 (解放)」を現在の所有者に送信します。現在の所有者がリソースを解放した場合、次の要求でこのリソースが要求者に与えられます。

ロックできるのは IPM コントローラが所有するリソースだけです。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` はリソース ID です。サポートされているリソース ID のリストは、[163 ページの「busres」](#)に記載されています。

### 例:

同期クロックのグループ 3 をロックします。

```
# clia busres lock clk3
Pigeon Point Shelf Manager Command Line Interpreter

Lock operation succeeded
#
```

# 「バス型に接続された資源の制御 (クエリー)」 コマンドの送信

## 構文:

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

## 目的:

このコマンドは、指定した IPM コントローラに「バス型に接続された資源の制御 (クエリー)」要求を送信します。IPM コントローラがコマンド行で指定されていない場合は、リソースの現在の所有者に要求が送信されます。応答を受信すると、noupdate フラグが指定されていない限り、該当する変更がリソーステーブルで行われます (たとえば、現在の所有者と考えられている IPM コントローラが制御なし状態で応答した場合、それを反映するようにテーブルが変更されます)。このフラグがコマンド行で渡された場合、受信した情報に基づいてリソーステーブルが変更されることはありません。

パラメータ *resource* はリソース ID です。サポートされているリソース ID のリストは、[163 ページの「busres」](#)に記載されています。

パラメータ *target* には、要求の送信先の 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、クエリー要求に対する応答として受信した情報はリソーステーブルの更新に使用できません。

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の情報は提供されません。

## 例:

メタリックテストバスのペア 1 に関するクエリーをアドレス 0x82 の IPM コントローラに送信します。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
Pigeon Point 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
```

目的:



---

**注意** – これは経験豊富なユーザー向けのコマンドです。そのため、独自の判断で使用しないでください。

---

このコマンドは、指定したリソースの所有者をリソーステーブルに直接設定します。コマンドの実行前にリソースの所有者が異なっていた場合でも、「バス型に接続された資源の制御」コマンドは送信されません。これは、テストおよび復旧の目的にのみ使用する下位レベルのコマンドです。

パラメータ *resource* はリソース ID です。サポートされているリソース ID のリストは、[163 ページの「busres」](#)に記載されています。

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。リソースがどの IPM コントローラによっても所有されないことを指定する場合は、0 を IPMB アドレスとして使用します。

例:

ボード 1 をメタリックテストバスのペア 1 の新しい所有者として設定します。

```
# clia busres setowner mtb1 board 1
Pigeon Point Shelf Manager Command Line Interpreter

New owner is set successfully
#
```

## 「バス型に接続された資源の制御 (バスによって制限されない)」 コマンドの送信

構文:

```
busres sendbusfree resource target
```

目的:



---

**注意** – これは経験豊富なユーザー向けのコマンドです。そのため、独自の判断で使用しないでください。

---

このコマンドは、指定した IPM コントローラに「バス型に接続された資源の制御 (バスによって制限されない)」要求を送信します。別の IPM コントローラがリソースを所有している場合でも、要求の送信前にリソーステーブルに対する操作は実行されません。ただし、リソーステーブルはこの要求に対する応答に基づいて変更されます。つまり、IPM コントローラがリソースの所有を受け入れた場合、その IPM コントローラは新しい所有者としてリソーステーブルに設定されます。これは、テストおよび復旧の目的にのみ使用する下位レベルのコマンドです。

パラメータ *resource* はリソース ID です。サポートされているリソース ID のリストは、[163 ページの「busres」](#)に記載されています。

パラメータ *target* には、要求の送信先の 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 コントローラによっても所有されないことを指定する場合は、0 を IPMB アドレスとして使用します。

例:

メタリックテストバスのペア 1 に対する「バスを使用しない」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
Pigeon Point Shelf Manager Command Line Interpreter

IPMC rejected ownership of the resource
#
```

---

# console

## 構文:

`console slot-number [-f]`

## 目的:

このコマンドは、指定し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確立できます。有効なスロット番号は1-6および9-14です。

-f オプションを指定すると、新しいコンソールセッションが開始する前に既存のコンソールセッションが終了します。これにより、強制的に新しいコンソールセッションになります。

---

**注** - コンソール機能を使用するには、シェルフ管理がアクティブなシェルフ管理カードである必要があります。また、サーバーのミッドプレーンのスロット7にスイッチカードが取り付けられている必要もあります。

---

ノードボードとのコンソールセッションが確立したら、`passwd`などのシステム管理コマンドを実行したり、状態とエラーメッセージを読み取ったり、その特定のスロットのボードを停止したりできます。

---

**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、コンソール出力はShMM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出力されます。これはケーブルの接続時にShMMのコンソールセッションがアクティブになっていた場合でも同様です。

---

現在のコンソールセッションから切断するには、`~q`または`~.`(チルダとドット)を入力します。

## 例:

物理スロット4のノードボード上の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
```

## 目的:

このコマンドは、指定した FRU に IPMI コマンド「FRU のアクティブ化の設定 (FRU の非アクティブ化)」を送信します。FRU は、所有している IPM コントローラの IPMB アドレスと FRU デバイス ID を使用して指定します。FRU デバイス ID 0 は、PICMG 3.0 コンテキストの IPM コントローラ本体を指定します。

## 例:

アドレス 9C にある IPM コントローラ本体を非アクティブ化します。

```
# cli deactivate 9c 0
Pigeon Point Shelf Manager Command Line Interpreter

Command issued via IPMB, status = 0 (0x0)
    Command executed successfully
#
```

---

# debuglevel

## 構文:

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

## 目的:

このコマンドは、シェルフマネージャーの現在のデバッグレベルを表示します。または、新しい値が指定されている場合は、デバッグレベルを新しい値に設定します。

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 進数です。マスク内に設定された各ビットは、特定のタイプのデバッグ出力を有効にします。

- 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 – 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ついて表示されるメッセージ
- 0x0080 – 内部ロックの取得と解放に関する詳細メッセージ

シェルフマネージャリリース 2.4.4 以降では、システムログへのシェルフマネージャ出力とコンソールへの出力に個別のデバッグレベルを設定できます。これにより、たとえば、システムログへのシェルフマネージャ出力の通常の詳細レベルを維持しながら、システムコンソールを重大なエラーメッセージ専用予約しておくことができます。

このコマンドをパラメータなしで実行した場合、システムログとコンソールの両方の現在のデバッグレベル値が表示されます。両方のレベル値が同じである場合は、単一行の出力のみが生成されます。

このコマンドを 1 つのパラメータ *new-value* とともに実行した場合、指定したデバッグレベルがシステムログ出力とコンソール出力の両方に設定されます。

このコマンドを 2 つのパラメータとともに呼び出す場合は、最初のパラメータでシステムログ出力のデバッグレベルを指定し、2 番目のパラメータでコンソール出力のデバッグレベルを指定します。

シェルフマネージャのデフォルトのデバッグレベルは、システムログおよびコンソールともに 0x0007 です。ただし、この値は、シェルフマネージャの構成ファイルでシステムログとコンソールについて個別に上書きしたり、シェルフマネージャの起動時にコマンド行で *-v* オプションを使用してシステムログとコンソールの両方について上書きしたりできます。

このコマンドは、バックアップシェルフマネージャでも実行できます。

例:

現在のデバッグレベルを取得し、両方を 0x001F に設定します。

```
# clia debuglevel
Pigeon Point Shelf Manager Command Line Interpreter

Debug Mask is 0x0007
```

```
# clia debuglevel 1f
Pigeon Point Shelf Manager Command Line Interpreter

Debug Mask is 0x001f

# clia debuglevel
Pigeon Point Shelf Manager Command Line Interpreter
#
Debug Mask is 0x001f
```

システムログのデバッグレベルマスクを 0x0007 (情報) に設定し、コンソールマスクを 0x0003 (エラーおよび警告のみ) に設定します。

```
# clia debuglevel 7 3
Pigeon Point Shelf Manager Command Line Interpreter

Debug Mask is 0x0007
Console Debug Mask is set to 0x0003

# clia debuglevel
Pigeon Point Shelf Manager Command Line Interpreter

Debug Mask is 0x0007
Console Debug Mask is 0x0003
#
```

---

## exit | quit

### 構文:

```
exit
quit
```

### 目的:

コマンド `exit` または `quit` は、CLI の対話型モード (`clia` をパラメータなしで実行すると開始します) を終了します。

### 例:

```
# exit
#
```

---

# fans

## 構文:

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

## 目的:

このコマンドは、指定したファン FRU に関する情報を表示します。FRU デバイス ID が省略された場合は、指定したアドレスの IPM コントローラによって制御されるすべてのファン FRU の情報を表示します。IPMB アドレスも省略されている場合は、シェルフマネージャーが認識するすべてのファン FRU の情報を表示します。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
- 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 (上書きおよびローカルコントロールレベル (両方が使用可能な場合))

## 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
Pigeon Point Shelf Manager Command Line Interpreter

20: FRU # 3
  Current Level: 9
  Minimum Speed Level: 0, Maximum Speed Level: 15
20: FRU # 4
  Current Level: 9
  Minimum Speed Level: 0, Maximum Speed Level: 15
20: FRU # 5
  Current Level: 9
  Minimum Speed Level: 0, Maximum Speed Level: 15
#
```

# flashupdate

## 構文:

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

## 目的:

このコマンドは、Sun Netra CP3x60 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します。このコマンドは、Sun Netra CT900 サーバーに取り付けられた Sun Netra CP3x60 ボードのみに有効です。更新される Sun Netra CP3x60 ボードのシステムファームウェアには、ILOM、ALOM-CMT/ILOM、Hypervisor、OBP、Post、および VBSC が含まれます。

---

注 – Sun Netra CP3x60 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、次のことを把握しておく必要があります。

- 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
- ファームウェアイメージを格納するパス

*slot-number* には Sun Netra CP3x60 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:

Sun Netra CP3x60 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。このプロセスの完了には、数分かかる場合があります。正常に完了したら、`boardreset` コマンドを使用し、ボードをリセットします。

```
# 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
```

## 目的:

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 が省略された場合は、指定したアドレスの IPM コントローラによって制御されるすべての FRU の情報を表示します。IPMB アドレスも省略されている場合は、シェルフマネージャーが認識するすべての FRU の情報を表示します。

また、サイトの種類で FRU を選択することもできます。サイトの種類は、16 進数でコマンドパラメータに指定します。FRU とサイトの種類の関連付けは、シェルフ FRU 情報に格納されています。サイトの種類は、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 ShMM
- 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
- ほかのすべての値は予約済みです

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
- サイトの種類および番号 (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 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)。この情報は、FRU センサーデータレコード (SDR) から取得され、IPMI 仕様の 37.12 節に準拠しています
- 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、現在の割り当て消費電力 (ワット単位)

例:

アドレス 9C のすべての FRU の標準の情報を取得します。

```
# clia fru 9c 0
Pigeon Point 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: "PPS Sentry 6"
#
```

アドレス 9C のすべての FRU の詳細情報を取得します。

```
# clia fru -v 9c 0
Pigeon Point 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: "PPS 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 の詳細情報を取得します。

```
# clia fru -v 20 id=1
Pigeon Point Shelf Manager Command Line Interpreter

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 Type: "FRU Inventory Device behind management
controller" (0x10), Modifier 0x0
    Device ID String: "Shelf EEPROM 1"
    Current Power Level: 0x01, Maximum Power Level: 0x01, Current
Power Consumption: 5.0 Watts
#
```

---

## frucontrol

### 構文:

```
frucontrol IPMB-address fru-id command options

frucontrol board n command options
frucontrol shm n command options
frucontrol fan_tray n command options
```

### 目的:

このコマンドは、指定した FRU に FRU 制御コマンドを送信し、指定した処理を FRU ペイロード上で実行します。FRU は、所有している IPM コントローラの IPMB アドレスと FRU デバイス ID を使用して指定します。FRU デバイス ID 0 は、PICMG 3.0 コンテキストの IPM コントローラ本体を指定します。

*command* パラメータには、使用する FRU 制御コマンドを指定します。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の実行
- `warm_reset` (省略形は `wr`) – FRU ペイロードのウォームリセットの実行



- `diagnostic_interrupt` (省略形は `di`) – 診断の中断を実行
- `graceful_reboot` (省略形は `gr`) – FRU ペイロードの正常な再起動と、タイマー値およびタイムアウトアクションパラメータの設定

`options` パラメータは、`graceful_reboot` コマンドのタイマーオプションの設定に使用します。オプションは 16 進で、次に示す順序で指定します。

`timer_use timer_action pre-time_out countdown_LSB countdown_MSB`

詳細は、『ATCA Payload Graceful Shutdown Design』マニュアルの 8.3.1.1 節を参照してください。

注 – `warm_reset`、`diagnostic_interrupt`、および `graceful_reboot` は省略可能で、ブレードに依存します。また、Sun Netra CP3xxx ブレードでは、`warm_reset` と `diagnostic_interrupt` はサポートされていません。

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
Pigeon Point Shelf Manager Command Line Interpreter

    FRU Control: Controller 0x9c, FRU ID # 0, command 0x00, status
0(0x0)
    Command executed successfully
#
```

スロット 13 のボードに対してコールドリセットコマンドを実行します。

```
# clia frucontrol board 13 cr
Pigeon Point Shelf Manager Command Line Interpreter

    FRU Control: Controller 0x98, FRU ID # 0, command 0x00, status
0(0x0)
    Command executed successfully
#
```

IPMB アドレス 92 の FRU 0 に対して `graceful_reboot` コマンドを実行します。

```
# clia frucontrol 92 0 gr 03 01 0 b0 04
Pigeon Point Shelf Manager Command Line Interpreter
```

```
FRU control: Controller 0x92, FRU ID # 0, command 0x00, status
0 (0x0)
    Command executed successfully
#
```

スロット 13 のボードに対して graceful\_reboot コマンドを実行します。

```
# clia frucontrol board 13 gr 03 01 0 b0 04
Pigeon Point Shelf Manager Command Line Interpreter

    FRU control: Controller 0x98, FRU ID # 0, command 0x02, 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* は、次のものに置き換えることができます。

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

### 目的:

このコマンドでは、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 情報の読み書きに使用できます。

読み取り形式では、この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、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 (182 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 情報ストレージの内容を、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 (182 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
Pigeon Point 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
Pigeon Point 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
Pigeon Point Shelf Manager Command Line Interpreter

Writing 2 bytes to IPM 0x20, FRU # 1, offset: 1, status = 0(0x0)
#
# clia frudata 20 1 0
```

```

Pigeon Point 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
#
# cli fruinfo 20 1 1 0 1
Pigeon Point Shelf Manager Command Line Interpreter

Writing 2 bytes to IPM 0x20, FRU # 1, offset: 1, status = 0(0x0)
#
# cli fruinfo 20 1 0
Pigeon Point 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 byte 1 [byte2 ... [byte 16] ...]

```

*addr fru\_id* は、次のものに置き換えることができます。

```

board n
shm n
fan_tray n

```

### 目的:

このコマンドは、指定した 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 (つまり、FRU 情報を指定した 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、指定した FRU に対する IPMI コマンド「FRU インベントリ領域情報の取得」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、名前を付け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
Pigeon Point 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* は、次のものに置き換えることができます。

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

目的:

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した 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、ファイルから指定した FRU に書き込みます。

```
# clia frudataw 20 2 /var/tmp/20.2.bin
Pigeon Point 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* は、次のものに置き換えることができます。

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

目的:

このコマンドは、ユーザーに理解しやすい形式で FRU 情報を表示します。

例:

特定の FRU の FRU 情報を表示します。

```
# clia fruinfo 20 0
Pigeon Point Shelf Manager Command Line Interpreter

20: FRU # 0, FRU Info
  Failure status: 203 (0xcb)
  Requested data not present
#

# clia fruinfo 20 1
Pigeon Point 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
#
# clia fruinfo -v -x 20 1

Pigeon Point Shelf Manager Command Line Interpreter

20: FRU # 1, FRU Info
Common Header:      Format Version = 1
  01 00 01 05  0E 18 00 D3

Chassis Info Area:
  Version      = 1
  Chassis Type          = (1)
  Chassis Part Number   = 0x55 0xAA
  Chassis Serial Number = 5I:5
  Custom Chassis Info   =
  01 04 01 02  55 AA 83 55  AA 55 C1 00  00 00 00 00
  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 61

Board Info Area:
  Version      = 1
  Language Code          = 25
  Mfg Date/Time         = Jun 16 15:37:00 2011 (8129737 minutes
since 1996)
  Board Manufacturer    = Pigeon Point Systems
  Board Product Name    = Shelf Manager
  Board Serial Number   = PPS0000000
  Board Part Number     = A
  FRU Programmer File ID =
  Custom Board Info     =
  01 09 19 C9  0C 7C D4 50  69 67 65 6F  6E 20 50 6F
  69 6E 74 20  53 79 73 74  65 6D 73 D6  53 68 65 6C
  66 20 4D 61  6E 61 67 65  72 20 20 20  20 20 20 20
  20 20 CA 50  50 53 30 30  30 30 30 30  30 C2 41 20
  C0 C1 00 00  00 00 00 A0

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 =
Custom Product Info   =
01 0A 19 D4 50 69 67 65 6F 6E 20 50 6F 69 6E 74
20 53 79 73 74 65 6D 73 D6 53 68 65 6C 66 20 4D
61 6E 61 67 65 72 20 20 20 20 20 20 20 20 20 C6
30 30 30 30 30 30 C9 52 65 76 2E 20 31 2E 30 30
CA 50 50 53 30 30 30 30 30 30 30 C0 C0 C1 00 6A

Multi Record Area:
Record Type           = Management Access Record
Version = 2
Sub-Record Type: Component Name (0x05)
Sub-Record Data:     = ShMM
03 02 05 A6 50 05 53 68 4D 4D

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
Hw Addr: 4f, Site # 15, Type: "AdvancedTCA Board" 00
Hw Addr: 50, Site # 16, Type: "AdvancedTCA Board" 00
C0 02 4B 44 AF 5A 31 00 10 01 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 10
41 01 00 42 02 00 43 03 00 44 04 00 45 05 00 46
06 00 47 07 00 48 08 00 49 09 00 4A 0A 00 4B 0B
00 4C 0C 00 4D 0D 00 4E 0E 00 4F 0F 00 50 10 00

```

```

PICMG Backplane Point-to-Point Connectivity Record (ID=0x04)
  Version = 0
  P2P Slot Descriptor:
    Channel Type           = 0x0B PICMG@3.0 Base Interface
    LocalSlot/HW Address = 0x41
    Channel Count         = 0x0F
    Channel Descriptor    = LocalChannel 2, RemoteChannel 2,
RemoteSlot 0x42
    Channel Descriptor    = LocalChannel 3, RemoteChannel 1,
RemoteSlot 0x43
    Channel Descriptor    = LocalChannel 4, RemoteChannel 1,
RemoteSlot 0x44
    Channel Descriptor    = LocalChannel 5, RemoteChannel 1,
RemoteSlot 0x45
    Channel Descriptor    = LocalChannel 6, RemoteChannel 1,
RemoteSlot 0x46
    Channel Descriptor    = LocalChannel 7, RemoteChannel 1,
RemoteSlot 0x47
    Channel Descriptor    = LocalChannel 8, RemoteChannel 1,
RemoteSlot 0x48
    Channel Descriptor    = LocalChannel 9, RemoteChannel 1,
RemoteSlot 0x49
    Channel Descriptor    = LocalChannel 10, RemoteChannel 1,
RemoteSlot 0x4A
    Channel Descriptor    = LocalChannel 11, RemoteChannel 1,
RemoteSlot 0x4B
    Channel Descriptor    = LocalChannel 12, RemoteChannel 1,
RemoteSlot 0x4C
    Channel Descriptor    = LocalChannel 13, RemoteChannel 1,
RemoteSlot 0x4D
    Channel Descriptor    = LocalChannel 14, RemoteChannel 1,
RemoteSlot 0x4E
    Channel Descriptor    = LocalChannel 15, RemoteChannel 1,
RemoteSlot 0x4F
    Channel Descriptor    = LocalChannel 16, RemoteChannel 1,
RemoteSlot 0x50
  P2P Slot Descriptor:
    Channel Type           = 0x0B PICMG@3.0 Base Interface
    LocalSlot/HW Address = 0x42
    Channel Count         = 0x0F
    Channel Descriptor    = LocalChannel 2, RemoteChannel 2,
RemoteSlot 0x41
    Channel Descriptor    = LocalChannel 3, RemoteChannel 2,
RemoteSlot 0x43

```

```

Channel Descriptor = LocalChannel 4, RemoteChannel 2,
RemoteSlot 0x44
Channel Descriptor = LocalChannel 5, RemoteChannel 2,
RemoteSlot 0x45
Channel Descriptor = LocalChannel 6, RemoteChannel 2,
RemoteSlot 0x46
Channel Descriptor = LocalChannel 7, RemoteChannel 2,
RemoteSlot 0x47
Channel Descriptor = LocalChannel 8, RemoteChannel 2,
RemoteSlot 0x48
Channel Descriptor = LocalChannel 9, RemoteChannel 2,
RemoteSlot 0x49
Channel Descriptor = LocalChannel 10, RemoteChannel 2,
RemoteSlot 0x4A
Channel Descriptor = LocalChannel 11, RemoteChannel 2,
RemoteSlot 0x4B
Channel Descriptor = LocalChannel 12, RemoteChannel 2,
RemoteSlot 0x4C
Channel Descriptor = LocalChannel 13, RemoteChannel 2,
RemoteSlot 0x4D
Channel Descriptor = LocalChannel 14, RemoteChannel 2,
RemoteSlot 0x4E
Channel Descriptor = LocalChannel 15, RemoteChannel 2,
RemoteSlot 0x4F
Channel Descriptor = LocalChannel 16, RemoteChannel 2,
RemoteSlot 0x50
C0 02 65 2B AE 5A 31 00 04 00 0B 41 0F 42 42 00
43 61 00 44 81 00 45 A1 00 46 C1 00 47 E1 00 48
01 01 49 21 01 4A 41 01 4B 61 01 4C 81 01 4D A1
01 4E 01 01 4F E1 01 50 01 02 0B 42 0F 41 42 00
43 62 00 44 82 00 45 A2 00 46 C2 00 47 E2 00 48
02 01 49 22 01 4A 42 01 4B 62 01 4C 82 01 4D A2
01 4E C2 01 4F E2 01 50 02 02

PICMG Shelf Power Distribution Record (ID=0x11)
Version = 0
Feed count: 1
Feed:
Maximum External Available Current: 50.0 Amps
Maximum Internal Current: Not specified
Minimum Expected Operating Voltage: -40.5 Volts
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
C0 02 2C A7 6B 5A 31 00 11 00 01 F4 01 FF FF 51
10 41 FE 42 FE 43 FE 44 FE 45 FE 46 FE 47 FE 48
FE 49 FE 4A FE 4B FE 4C FE 4D FE 4E FE 4F FE 50
FE
```

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
```

```
C0 82 57 81 E6 5A 31 00 12 00 0A 10 41 FE 96 00  
40 42 FE 96 00 40 43 FE 96 00 40 44 FE 96 00 40  
45 FE 96 00 40 46 FE 96 00 40 47 FE 96 00 40 48  
FE 96 00 40 49 FE 96 00 40 4A FE 96 00 40 4B FE  
96 00 40 4C FE 96 00 40 4D FE 96 00 40 4E FE 96  
00 40 4F FE 96 00 40 50 FE 96 00 40
```

```
#
```

---

## getacousticlevel

### 構文:

```
getacousticlevel ETSI NEBS-A NEBS-U
```

### 目的:

このコマンドは、システムの騒音レベルとファン速度を表示します。このコマンドはサーバーシャーシに使用できます。これによりユーザーは、システムの現在の騒音レベルを取得できます。騒音レベルを設定すると、変数が変更され、デフォルトの冷却アルゴリズムに従ってファン速度が設定されます。この変更は動的に行われ、持続的でShMMの再起動は必要ありません。「setacousticlevel」も参照してください。

---

**注** – これらのコマンドは、SUNCT900 冷却アルゴリズムを使用している場合のみ使用できます。

---

**例:**

システムの騒音レベルとファン速度を取得します。

```
# clia getacousticlevel ETSI
Pigeon Point Shelf Manager Command Line Interpreter

System acoustic level - ETSI
fan speed              - 5#
```

---

## getbootdev

**構文:**

```
getbootdev IPMB-0-address [FRU-device-ID | IPMB-L-address]
```

**目的:**

このコマンドは、指定した IPM コントローラのシステム起動パラメータを表示します。AdvancedMC アクセスを対象にしない場合、2 番目のパラメータは 0 に設定するか省略します。2 番目のパラメータが 70h を超える場合、AMC アドレスの IPMB-L アドレスが使用されます。これ以外の場合、2 番目のパラメータは FRU ID として扱われ、AMC アドレスマッピングで IPMB-L アドレスに変換されます。

**例:**

IPMB アドレス 82h にある IPM コントローラのシステム起動オプションを取得します。

```
# clia getbootdev 82
Pigeon Point Shelf Manager Command Line Interpreter

Get boot device option: status = 0x0 (0)
Response data (raw): 01 05 00 00 00 00 00
Decoded:
    Parameter version: 1
    Parameter valid = TRUE
    Boot option selector: 5
    Boot flags valid = FALSE
    Boot device selector: 0 (No override)

#
```

キャリアの IPMB-0 アドレスが 90h で MMC の IPMB-L アドレス が 72h である AMC のシステム起動オプションを取得します。

```
# clia getbootdev 90 72
Pigeon Point Shelf Manager Command Line Interpreter

Get boot device option: status = 0x0 (0)
Response data (raw): 01 05 80 04 00 00 00
Decoded:
    Parameter version: 1
    Parameter valid = TRUE
    Boot option selector: 5
    Boot flags valid = TRUE
    Boot device selector: 1 (Force PXE)

#
```

---

## getfanlevel

### 構文:

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

### 目的:

このコマンドは、コマンドパラメータで指定した FRU によって制御され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
Pigeon Point Shelf Manager Command Line Interpreter

20: FRU # 2 Override Fan Level: 1, Local Fan Level: 255
#
```



---

# getfanpolicy

## 構文:

```
getfanpolicy [fan-tray-addr [fan-tray-fru_id]] [-s addr|site_type [fru_id |  
site_number]]
```

## 目的:

ファントレー制御モードの情報または指定したファントレーの FRU カバレッジの情報を取得します。このコマンドでは、(「ファンポリシーの設定」コマンドに基づいて) シェルフマネージャーによってサイトの自律制御が有効または無効にされたかどうか、および (ファンジオグラフィックレコードに基づいて) FRU のサイトがファンの有効範囲にあるかどうかの 2 つの異なるデータが返されることに注意してください。

パラメータ *fan-tray-addr* および *fan-tray-fru\_id* では、ファントレーを指定します。コマンドがパラメータを受け入れない場合があり、その場合にはすべてのファントレーおよび FRU の情報が返されます。

数値引数を 16 進数として処理させる場合は、プレフィックス "0x" を使用しないとエラーが返されます。

フラグ -s は、ファントレーの対象サイトを定義するパラメータの前に付けます。

*site\_type* パラメータには、Board、PEM、ShelfFRU、ShelfManager、FanTray、FanFilterTray、Alarm、Mezzanine、PMC、RTM のいずれかの値を指定できます。

## 例:

IPMB アドレス 20h、FRU ID 3 にあるファントレーのファンポリシーを取得します。

```
# clia getfanpolicy 0x20 3  
Pigeon Point Shelf Manager Command Line Interpreter  
  
Fan Tray: 0x20, FRU Id # 3  
  Policy Type: Any Site  
  Policy Timeout: 20 seconds  
  Policy Applied: Tue Oct 17 02:32:06 2006  
#
```

IPMB アドレス 20h、FRU ID 1 のサイトに取り付けられた、IPMB アドレス 20h、FRU ID 3 のファンのファンポリシーを取得します。

```
# clia getfanpolicy 0x20 3 -s 0x20 1
Pigeon Point Shelf Manager Command Line Interpreter

Fan Tray: 0x20, FRU Id # 3
  Policy Type: Per Site
  Policy Timeout: 20 seconds
  Policy Applied: Tue Oct 17 02:39:06 2006
  Site Type: Dedicated ShMC, Site Number: 1
  Site Covered: TRUE
#
```

---

## getfruledstate

### 構文:

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

### 目的:

このコマンドは、現在の FRU の LED の状態を、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。

特定の LED または指定した FRU のすべての LED の情報を表示できます。対象の LED の IPMB アドレスおよび FRU ID は省略することもできます。FRU ID を省略すると、指定した IPM コントローラのすべての FRU 上のすべての LED の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 の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
Pigeon Point 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 状態の詳細情報を表示します。

```
# clia getfruLEDstate -v FC
Pigeon Point 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
Pigeon Point 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
Pigeon Point 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]]
```

## 目的:

このコマンドは、指定し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。

このコマンドを使用すると、対象のコントローラが複数の LUN 上にあるセンサーをサポートしている場合に、センサー番号に論理ユニット番号 (LUN) を付けて修飾することができます。LUN が省略された場合、指定し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、3 のいずれかを指定できます。(LUN 2 は予約されています。)センサー名は通常コントローラ内で一意であることが前提であるため、センサー名は LUN 番号で修飾しません。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の情報が表示されます。IPMB-address が省略された場合、指定した 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
Pigeon Point 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] (in radial IPMB-0 environment)
getipmbstate IPMB-address (in bused IPMB-0 environment)
```

## 目的:

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。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 の両方の情報が出力されます。

このコマンドは、バス型環境と放射線状環境では動作が異なります。バス型環境、または対象の IPMC が IPMB ハブ以外の場合の放射線状環境では、引数 *link* は使用しません。対象の IPM コントローラの IPMB-A および IPMB-B の状態情報が表示されます。

放射線状環境で、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* が省略された場合、コマンドは放射線状のすべての IPMB リンクの状態情報を出力します。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) の情報を出力します。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。

いずれの場合も、IPMB-A および IPMB-B の両方の状態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
Pigeon Point 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)
```

放射線状環境のシェルフマネージャーのリンク 8 の現在の状態を表示します。

```
# clia getipmbstate 20 8
Pigeon Point 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]]
```

### 目的:

このコマンドは、指定したチャンネルで、指定した LAN 構成パラメータの値を表示します。構成パラメータ名または番号を指定しない場合、指定したチャン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 コマンドで使用できる LAN 構成パラメータの名前と番号を示します。

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 チャン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 の各特権レベルの認証タイプ有効化フラグを含む 5 つの 8 ビットの値。
ip	3	LAN チャンネルに割り当てられた IP アドレスを含む、小数点付き 10 進数表記の文字列値 (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 チャンネルに割り当てられた : 記号で区切られた 6 つの 16 進バイトで構成される MAC アドレスを含む文字列値 (例: 00:A0:24:C6:18:2F)。

表 A-1 getlanconfig 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、小数点付き 10 進数表記の文字列値 (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 3 つの 8 ビットの値。 <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]) およびサービスの種類 (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 チャンネル上でアドレス解決プロトコル (ARP) の動作を制御する 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 の送信を有効にする</li> </ul>
arp_interval	11	Gratuitous ARP の秒単位での間隔。固定小数点形式で示し、小数部が含まれる場合もある。
dft_gw_ip	12	デフォルトゲートウェイの IP アドレスを含む、小数点付き 10 進数表記の文字列値。
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 6 つの 16 進バイトで構成される文字列値。
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、小数点付き 10 進数表記の文字列値。
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 6 つの 16 進バイトで構成される文字列値。
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 警告の宛先の最大数。



表 A-1 getlanconfig の LAN 構成パラメータ (続き)

destination_type	18	<p>指定された設定セクタで識別される宛先の種類。設定セクタを指定しない場合、すべての宛先の種類が表示されます。各宛先の種類のエントリには、次のフィールドが含まれます。</p> <ul style="list-style-type: none"> <li>• 宛先の種類 (0 - 7)</li> <li>• 警告確認応答フラグ</li> <li>• 警告確認応答タイムアウト/再試行間隔の秒数 (1 - 256)</li> <li>• 再試行回数 (0 - 7)</li> </ul>
destination_address	19	<p>指定した設定セクタに関連付けられた宛先アドレス。設定セクタを指定しない場合、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。</p> <ul style="list-style-type: none"> <li>• ゲートウェイセクタ: 0 - デフォルトを使用、1 - バックアップを使用</li> <li>• IP アドレス (小数点付き 10 進形式の文字列)</li> <li>• MAC アドレス (コロン [:] で区切られた、6 つの 16 進バイト値からなる文字列)</li> </ul>

**例:**

以降の節では、使用でき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
Pigeon Point Shelf Manager Command Line Interpreter

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
```

### 目的:

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。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 に設定します。

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
Pigeon Point 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
```

### 目的:

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) のそれぞれに対し、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の種類を示します。パラメータ値は 5 つのバイトのシーケンスで表され、各バイトがそれぞれの特権レベルに対応します。これは、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 に設定します。

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  
Pigeon Point 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
```

### 目的:

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、小数点付き 10 進数表記で表示します。

### 例:

```
# clia getlanconfig 1 ip  
Pigeon Point Shelf Manager Command Line Interpreter  
  
IP Address: 172.16.2.203
```

## ip\_source

### 構文:

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

### 目的:

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。値は次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  
Pigeon Point Shelf Manager Command Line Interpreter  
  
IP Address Source: Static Address (Manually Configured) (0x01)  
#
```

mac

構文:

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

目的:

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  
Pigeon Point Shelf Manager Command Line Interpreter  
  
MAC Address: 90:91:91:91:91:91  
#
```

subnet\_mask

構文:

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

目的:

このコマンドは、チャンネルで使用される現在の IP サブネットマスクを、小数点付き 10 進数表記で表示します。

例:

```
# clia getlanconfig 1 subnet_mask
Pigeon Point Shelf Manager Command Line Interpreter

Subnet Mask: 255.255.255.0
#
```

## ipv4\_hdr\_param

構文:

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

目的:

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 19.2 節に準拠しています。

例:

```
# clia getlanconfig 1 ipv4_hdr_param
Pigeon Point Shelf Manager Command Line Interpreter

IPv4 Header Parameters: 0x40:0x40:0x10
#
```

## pri\_rmcp\_port

構文:

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

目的:

このコマンドは、チャンネルで使用される現在の 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
Pigeon Point Shelf Manager Command Line Interpreter

Primary RMCP Port Number: 0x026f
#
```

## sec\_rmcp\_port

構文:

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

目的:

このコマンドは、チャンネルで使用される現在の 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
Pigeon Point Shelf Manager Command Line Interpreter

Primary RMCP Port Number: 0x0298
#
```

## arp\_control

構文:

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

## 目的:

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 に設定します。

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 arp_control
Pigeon Point Shelf Manager Command Line Interpreter

BMC-generated ARP Control: 02
    Enable BMC-generated Gratuitous Response
#
```

## arp\_interval

### 構文:

```
getlanconfig channel destination_count
getlanconfig channel 11
```

### 目的:

このコマンドは、チャンネルで使用される現在の 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
Pigeon Point Shelf Manager Command Line Interpreter

Gratuitous ARP Interval: 2.0 seconds
#
```



## dft\_gw\_ip

### 構文:

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

### 目的:

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、小数点付き 10 進数表記で表示します。

### 例:

```
# clia getlanconfig 1 dft_gw_ip  
Pigeon Point 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
```

### 目的:

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、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 dft_gw_mac  
Pigeon Point Shelf Manager Command Line Interpreter  
  
Default Gateway MAC Address: N/A  
#
```

## backup\_gw\_ip

### 構文:

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

### 目的:

このコマンドは、チャンネルで使用されるバックアップゲートウェイの IP アドレスを、小数点付き 10 進数表記で表示します。

### 例:

```
# clia getlanconfig 1 backup_gw_ip  
Pigeon Point 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
```

### 目的:

このコマンドは、チャンネルで使用されるバックアップゲートウェイの 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 backup_gw_mac  
Pigeon Point Shelf Manager Command Line Interpreter  
  
Backup Gateway MAC Address: N/A  
#
```

## community

### 構文:

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

### 目的:

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  
Pigeon Point Shelf Manager Command Line Interpreter  
  
Community String: "public"  
#
```

## destination\_count

### 構文:

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

### 目的:

このコマンドは、チャンネルで使用できる警告の宛先の最大数を表示します。これは、シェルフマネージャーの構成パラメータであり、shelfman 構成ファイルを通じてのみ変更できます。

### 例:

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

## destination\_type

### 構文:

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

### 目的:

このコマンドは、*set-selector* と同等の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。宛先について、次の情報が示されます。

- 宛先セクタ
- 警告の宛先の種類 (PET トラップまたは OEM 宛先、警告の確認応答を行うかどうか)
- 警告確認応答タイムアウト
- 再試行の回数

設定セクタが省略された場合、すべてのアクティブな宛先とその番号が表示されます。

### 例:

```
# clia getlanconfig 1 destination_type 2  
Pigeon Point 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  
Pigeon Point 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]
```

## 目的:

このコマンドは、*set-selector* と同等の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。宛先について、次の情報が示されます。

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

設定セクタが省略された場合、すべてのアクティブな宛先アドレスとその番号が表示されます。

## 例:

```
# cli getlanconfig 1 destination_address 2  
Pigeon Point 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  
#
```

# getmgmtportroute

## 構文:

```
getmgmtportroute [slot-number]
```

## 目的:

このコマンドは、管理ポートのルーティング構成を表示します。この情報は、ミッドプレーンに OEM レコードとして保存されます。

指定したスロットの管理ポートルーティング情報を表示するには、*slot-number* オプションを使用します。スロット番号が省略された場合、すべてのスロットの情報が返されます。このデータはミッドプレーンのみから取得されます。

構成は、ブレードの存在にかかわらず、永続的であり各スロットに対して指定されます。

- 管理ルート構成の設定については、[274 ページ](#)の「[setmgmtportroute](#)」コマンドを参照してください。
- 管理ポートの状態やルーティング構成を判定するための IPMC のクエリーについては、[239 ページ](#)の「[mgmtportstate](#)」コマンドを参照してください。
- 管理ポート (フロントまたは背面アクセス) の構成については、[349 ページ](#)の「[Set Management Port](#)」を参照してください。

## 例:

指定したスロットの管理ポートルーティング情報を表示します。

```
# cli getmgmtportroute 4
Pigeon Point Shelf Manager Command Line Interpreter

MGMT port Routing Information from Midplane FRU
-----
Slot   MGMT
-----
4      Front
```

# getmuxconfig

## 構文:

```
getmuxconfig [slot-number]
```

## 目的:

このコマンドは、指定したスロットのシェルフのミッドプレーン OEM マルチレコードに保存されているマルチプレクサ (MUX) 構成情報を表示します。slot-number を指定しない場合、すべてのスロットのルーティング情報が表示されます。ブレードに MUX コントローラがない場合、IPMC は ShMM によって送信された MUX ルーティング情報を無視します。MUX コントローラは、NIU ポートをゾーン 2 (バックプレーン) またはゾーン 3 (ARTM) に経路指定するために使用されます。指定したスロットの MUX 構成情報を表示するには、slot-number オプションを使用します。

シェルフマネージャーは、ストレージ (ミッドプレーン FRU) から読み取った構成情報を IPMC に提供し、ブレードをアクティブ化する前に、IPMC は MUX をプログラムする必要があります。詳細は、コマンド [setmuxconfig](#) および [muxstate](#) を参照してください。

## 例:

物理スロット 5 のノードボード上の MUX に対するポートルーティング情報を表示します。

```
# clia getmuxconfig 5
Pigeon Point Shelf Manager Command Line Interpreter

10GbE NIU-XAUI Routing Information from Midplane FRU
-----
Slot   XAUI-1           XAUI-2
-----
5      Zone3 (ARTM)     Zone2 (Back Plane)
```

# getpefconfig

## 構文:

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

## 目的:

このコマンドは、指定した PEF 構成パラメータの値を表示します。構成パラメータ名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と番号を示します。

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 (PEF の有効化、PEF 起動遅延の有効化など) を表す 8 ビットの値。
action_control	2	PEF アクショングローバル制御フラグ (リセットの有効化、電源切断の有効化など) を表す 8 ビットの値。
startup_delay	3	システムの電源投入とリセット後に PEF を遅延させる時間 (秒単位)。
alert_startup_delay	4	システムの電源投入とリセット後に警告を遅延させる時間 (秒単位)。
event_filter_count	5	イベントフィルタの最大数。
event_filter	6	指定した設定セクタで識別されるイベントフィルタテーブルエントリ。設定セクタを指定しない場合、すべてのアクティブなイベントフィルタが表示されます。
event_filter_data1	7	指定した設定セクタで識別されるイベントフィルタテーブルエントリの最初のバイト。設定セクタを指定しない場合、すべてのアクティブなイベントフィルタが表示されます。
alert_policy_count	8	警告ポリシーの最大数。
alert_policy	9	指定した設定セクタで識別される警告ポリシーテーブルエントリ。設定セクタを指定しない場合、すべてのアクティブな警告ポリシーが表示されます。



表 A-2 PEF 構成パラメータ (続き)

system_guid	10	PET トラップの GUID フィールドを設定するために使用される GUID。
alert_string_count	11	警告文字列の最大数。
alert_string_key	12	指定した設定セクタで識別される警告文字列キー。設定セクタを指定しない場合、すべての警告文字列キーが表示されます。
alert_string	13	指定した設定セクタで識別される警告文字列。設定セクタを指定しない場合、すべての警告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	指定した設定セクタで識別される OEM フィルタテーブルエントリ。設定セクタを指定しない場合、すべてのアクティブなイベントフィルタが表示されます。

例:

PEM パラメータテーブル全体を取得および表示します。

```
# clia getpefconfig
Pigeon Point 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
#
```

以降の節では、使用でき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
```

### 目的:

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あり、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 の有効化
- 0x02 – PEF アクションに対するイベントメッセージの生成の有効化
- 0x04 – システムの電源投入とリセット時の PEF 起動遅延の有効化
- 0x08 – PEF 警告起動遅延の有効化

その他のビットは予約されているため、0 に設定します。

### 例:

```
# clia getpefconfig control
Pigeon Point 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
```

## 目的:

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あり、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 – 警告アクションの有効化
- 0x02 – 電源切断アクションの有効化
- 0x04 – リセットアクションの有効化
- 0x08 – 電源再投入アクションの有効化
- 0x10 – OEM アクションの有効化
- 0x20 – 診断の中断の有効化

その他のビットは予約されているため、0 に設定します。

## 例:

```
# clia getpefconfig action_control  
Pigeon Point 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
```

### 目的:

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あり、起動時に PEF 機能が遅延する秒数を表します。

### 例:

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

## alert\_startup\_delay

### 構文:

```
getpefconfig startup_delay  
getpefconfig 4
```

### 目的:

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あり、起動時に警告機能が遅延する秒数を表します。

### 例:

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

## event\_filter\_count

### 構文:

```
getpefconfig event_filter_count  
getpefconfig 5
```

### 目的:

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、シェルフマネージャーの構成パラメータであり、`shelfman` 構成ファイルを通じてのみ変更できます。

### 例:

```
# cli getpefconfig event_filter_count  
Pigeon Point Shelf Manager Command Line Interpreter  
  
PEF Number of Event Filters: 64  
#
```

## event\_filter

### 構文:

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

### 目的:

このコマンドは、`set-selector` と同等の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ついて次の情報が表示されます。

- フィルタの構成: フィルタがソフトウェア構成か製造元事前設定か
- イベントフィルタアクションマスク
- 警告ポリシー番号
- イベントの重要度
- 照合するイベント発信元アドレス (255 = 任意のアドレス)
- 照合するソースチャンネル/LUN (255 = 任意のソースチャンネル/LUN に一致)
- 照合するセンサーの種類

- 照合するセンサー番号
- 照合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、3 の AND、コンペア 1 (CMP1)、およびコンペア 2 (CMP2) マスク

設定セレクトが省略された場合、すべてのアクティブなイベントフィルタテーブルエントリとその番号が表示されます。

例:

```
# clia getpefconfig event_filter 2
Pigeon Point 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]
```

目的:

このコマンドは、*set-selector* と同等の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最初の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。バイト内のビットの意味は次のとおりです。

- 0x80 – このフィルタは有効です。
- 0x40 – このフィルタは製造元で事前に設定され、ソフトウェアでは変更できません。

その他のビットは予約されているため、0 になります。

設定セレクタが省略された場合、アクティブな各イベントフィルタテーブルエントリの最初のバイトと対応するフィルタ番号が表示されます。

例:

```
# clia getpefconfig event_filter_data1 2
Pigeon Point 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
```

目的:

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、警告ポリシーテーブルのサイズです。これは、シェルフマネージャーの構成パラメータであり、`shelfman` 構成ファイルを通じてのみ変更できます。

例:

```
# clia getpefconfig alert_policy_count
Pigeon Point Shelf Manager Command Line Interpreter

    Alert Policies Count: 64
#
```

# alert\_policy

## 構文:

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

## 目的:

このコマンドは、*set-selector* と同等のインデックスを持つ警告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。各警告ポリシーについて次の情報が表示されます。

- ポリシー番号
- ポリシータイプ (前の宛先に送信された警告について)
- 宛先チャンネル番号
- 宛先セレクタ
- 警告文字列キー

設定セレクタが省略された場合、すべてのアクティブな警告ポリシーテーブルエントリーとその番号が表示されます。

## 例:

```
# clia getpefconfig alert_policy 2  
Pigeon Point 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
```

### 目的:

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 で警告の宛先に送信される GUID を表します。この GUID は個別の GUID として定義することも、システム GUID と同等のものとして定義することもできます (「システム GUID IPMI の取得」コマンドで取得できます)。

### 例:

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

## alert\_string\_count

### 構文:

```
getpefconfig alert_string_count  
getpefconfig 11
```

### 目的:

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、警告文字列テーブルのサイズであり、同時に使用できる警告文字列の最大数です。この値は、シェルフマネージャーの構成パラメータであり、`shelfman` 構成ファイルを通じてのみ変更できます。

### 例:

```
# clia getpefconfig alert_string_count  
Pigeon Point Shelf Manager Command Line Interpreter  
  
Alert Strings Count: 64  
#
```

## alert\_string\_key

### 構文:

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

### 目的:

このコマンドは、*set-selector* と同等のインデックスを持つ警告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 を使用して、揮発性警告文字列を指定できます。各キーは、警告を生成する目的で、イベントフィルタを警告文字列に関連付けます。各警告文字列キーについて次の情報が表示されます。

- 警告文字列キー番号
- 関連付けられているイベントフィルタ番号
- 関連付けられている警告文字列番号

設定セレクタが省略された場合、すべてのアクティブな警告文字列キーテーブルエントリとその番号が表示されます。

### 例:

```
# clia getpefconfig alert_string_key 2
Pigeon Point 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]
```

### 目的:

このコマンドは、*set-selector* と同等のインデックスを持つ警告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 を使用して、揮発性警告文字列を指定できます。このコマンドは、一度に文字列全体を表示します。

設定セレクトが省略された場合、定義されているすべての警告文字列とその番号が表示されます。

例:

```
# clia getpefconfig alert_string 2
Pigeon Point Shelf Manager Command Line Interpreter

Alert Strings:
    0x02: "This is the alert string"
#
```

## oem\_filter\_count

構文:

```
getpefconfig oem_filter_count
getpefconfig 96
```

目的:

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。これは、シェルフマネージャの構成パラメータであり、`shelfman` 構成ファイルを通じてのみ変更できます。

OEM フィルタテーブルは、Pigeon Point Systems が定義した IPMI 仕様の OEM 拡張です。これを使用すると、プラットフォームイベントに加えて、PEF を、OEM のタイムスタンプありまたはタイムスタンプなしの SEL エントリ (レコードタイプ範囲 C0h-FFh) に適用できます。

例:

```
# clia getpefconfig oem_filter_count
Pigeon Point Shelf Manager Command Line Interpreter

PEF Number of OEM Filters: 16
#
```

## oem\_filter

### 構文:

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

### 目的:

OEM フィルタテーブルは、Pigeon Point Systems が定義した IPMI 仕様の OEM 拡張です。これを使用すると、プラットフォームイベントに加えて、PEF を、OEM のタイムスタンプありまたはタイムスタンプなしの SEL エントリ (レコードタイプ範囲 C0h-FFh) に適用できます。

OEM フィルタテーブルの各エントリは、この OEM フィルタを適用するレコードタイプの範囲 (OEM レコードタイプの範囲内で) と、レコードタイプが一致するレコードが SEL に格納されたときに呼び出される警告ポリシー番号を定義します。

このコマンドは、*set-selector* と同等の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 フィルタについて次の情報が表示されます。

- バイト 1: SEL レコードタイプ範囲下限
- バイト 2: SEL レコードタイプ範囲上限
- バイト 3: バイト 1 およびバイト 2 で指定された範囲と一致するレコードタイプが含まれる SEL エントリに対して呼び出される警告ポリシー番号

設定セレクトが省略された場合、すべてのアクティブな OEM フィルタテーブルエントリとその番号が表示されます。

### 例:

```
# clia getpefconfig oem_filter  
Pigeon Point 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、指定したセンサーの現在のイベント有効化マスク値を表示します。

このコマンドを使用すると、ターゲットコントローラが複数の LUN 上にあるセンサーをサポートしている場合に、センサー番号に論理ユニット番号 (LUN) を付けて修飾することができます。LUN が省略された場合、すべての LUN の指定したセンサー番号を持つセンサーに関する情報が表示されます。*lun* には、値 0、1、3 のいずれかを指定できます。(LUN 2 は予約されています。)

センサー名は通常コントローラ内で一意であることが前提であるため、センサー名は LUN 番号で修飾しません。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の情報が表示されます。

このコマンドは、指定し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。各センサーの次の属性も示されます。

- 所有している 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 からのデバイス ID 文字列)、およびセンサーにアクセス可能な LUN
- センサーの種類

## 例:

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

```
# clia getsensoreventenable -v fe "Local Temp"
Pigeon Point 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 と番号を指定します。

```
# clia getsensoreventenable -v fe 0:3
Pigeon Point 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]]
```

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

## 目的:

このコマンドは、指定したセンサーのサポートされてい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。各センサーの次の属性も示されます。

- 所有している 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 からのデバイス ID 文字列)、およびセンサーにアクセス可能な LUN
- センサーの種類とイベント/読み取りタイプコード

このコマンドを使用すると、ターゲットコントローラが複数の LUN 上にあるセンサーをサポートしている場合に、センサー番号に論理ユニット番号 (LUN) を付けて修飾することができます。LUN が省略された場合、すべての LUN の指定したセンサー番号を持つセンサーに関する情報が表示されます。lun には、値 0、1、3 のいずれかを指定できます。(LUN 2 は予約されています。)

センサー名は通常コントローラ内で一意であることが前提であるため、センサー名は LUN 番号で修飾しません。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の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 に関するしきい値を取得します。

```
# clia getthreshold -v fe "Local Temp"
Pigeon Point 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 と番号を指定します。

```
# clia getthreshold -v fe 0:3
Pigeon Point 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]]
```

### 目的:

このコマンドは、サポートされているコマンドとその構文のヘルプ情報を表示します。

### 例:

コマンドとその構文のリストを取得します。

```
# clia help
Pigeon Point Shelf Manager Command Line Interpreter

Command Line Interface command set:
Parameters are case insensitive
In general:
    IPMB address is hexadecimal ALWAYS.
    All other numbers may be either decimal and hexadecimal (0x notation
required for hexadecimal numbers)
    -v turns on verbose output

activate <addr> <fru_id>
alarm <alarm status/action>
```



```

board [slot_number]
boardreset <slot number>
busres force <res>
busres info [<res>]
busres lock <res>
busres query [-v] <res> [<target> [nouupdate]]
busres release <res>
busres sendbusfree <res> <target>
busres setowner <res> <target>
busres unlock <res>
deactivate <addr> <fru_id>
debuglevel [<mask> [<console 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>
getbootdev <addr> [<fru-id> | <amc-addr>]
getfanlevel <addr> <fru_id>
getfanpolicy [<addr> [<fru_id>]] [-s <addr>|site_type
    [<fru_id>|site_number]]
getfruledstate [-v] [<addr> [<fru_id> [<LedId>|ALL]]]
gethysteresis [ <addr> [ [ lun: ]<sensor id> | <sensor name> ] ]
getipmbstate <addr> [<link>]
getlanconfig <channel number> <parameter number> | <parameter name>
getmgmtportroute [slot]
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
mgmtportstate <slot>
minfanlevel [<min fan level>]
networkelementid [<id>]
poll
quit
sel [clear] [ <addr> [ <number of items> [<number of first item> ] ] ]
sel info [<addr>]
sendamc <addr> <amc> <netfn> <command> [<parameters ...>]
sendcmd <addr> <netfn> <command> [<parameters ...>]
sensor [ <addr> [ [ lun: ]<sensor id> | <sensor name> ] ]
sensordata [ <addr> [ [ lun: ]<sensor id> | <sensor name> ] ]

```

```

sensorread <addr> [ lun: ]<sensor id>
session
setbootdev <addr> <fru-id | amc-addr> <boot-device>
setextracted <addr> <fru_id>
setfanlevel <addr> <fru_id> <state>
setfanpolicy <addr> <fru_id> <ENABLE|DISABLE [timeout]>
        [-s <addr>|site_type <fru_id>|site_number]
setipmbstate <addr> A|B [<link>] 0|1
setlanconfig <channel number> <parameter number> | parameter name
        <parameters ...>
setlocked <addr> <fru_id> <value>
setmgmtportroute <slot> < 1 / 0 >
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
setfruledstate <addr> <fru_id> <LedId>|ALL <LedOp|tail> [LedColor]
setpowerlevel <addr> <fru_id> [<pwr_lvl>|OFF] [Copy]
shelf <parameters>
shelfaddress ["<shelf address>"]
shmstatus
showunhealthy
switchover
terminate [-reboot]
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
console [slot_number]
userlabel [ shelf | slot ] <slot number>
setuserlabel [ shelf | slot ] [ <shelf name> | <slot number> ] <slot name>
flashupdate <slot number> -s <server IP address> -f <path name>
showhost <slot number> [version]
amcportstate [-v] <ipmc> <fru_id>

```

特定のコマンドのヘルプを取得します。

```
# clia help shelf pwrreorder
Pigeon Point 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
```

### 目的:

このコマンドは、指定したアドレスにある IPM コントローラに関する情報を表示します。*IPMB-address* が省略された場合は、シェルフマネージャーが認識しているすべての 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。

- コントローラの IPMB アドレス (2 桁の 16 進数)。
- IPM コントローラのエンティティ ID とエンティティインスタンス。
- IPM コントローラの最大 FRU デバイス ID。
- PICMG 拡張バージョン。このバージョンは、PICMG 3.0 準拠の IPM コントローラでは 2.0 になります。

IPM コントローラの FRU デバイス 0 (IPM コントローラ自身を表します) の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 動作中
- M5 - 停止要求

- M6 – 停止中
- M7 – 通信不可

冗長モードでは、IPM コントローラに関する次の追加情報が表示されます。

- 「デバイス ID の取得」 IPMI コマンドによって返される情報 (製造元 ID、製品 ID、デバイス ID、デバイスファームウェアバージョン、サポートされる IPMI バージョンを含む)
- コントローラ SDR のデバイス ID 文字列
- コントローラ SDR からの電源状態通知属性 (16 進数)
- コントローラ SDR からのグローバル初期化属性 (16 進数)
- コントローラ SDR からのデバイス機能属性 (16 進数)
- 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のマスクと各ビットのテキストによる説明
- E-キーイングの対象となるポートのリストとその状態 (有効/無効)

例:

アドレス 9C にある IPM コントローラに関する情報を取得します。

```
# clia ipmc 9c
Pigeon Point 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 にある 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
Pigeon Point 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: "PPS 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
```

### 目的:

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、その汎用 BMC アドレス 0x20 ではなく、そのハードウェアアドレスに基づいて表示します。これらのアドレスは、冗長なシェルフマネージャー間で異なります (これに対し、BMC アドレスは冗長なシェルフマネージャー間で共有されます)。

### 例:

```
# cli localaddress
Pigeon Point Shelf Manager Command Line Interpreter

Local IPMB Address = 0xFC
#
```

---

## mgmtportstate

### 構文:

```
mgmtportstate [slot-number]
```

### 目的:

このコマンドは、現在のポート構成を確認し、変更が必要かどうかを判断するのに便利です。このコマンドは、IPMC の OEM コマンド「Ethernet ポートアクセスの取得」を使用して、管理ポートの状態またはポートのルーティング構成を IPMC に対してクエリーします。

- 管理ルート構成の設定については、274 ページの「[setmgmtportroute](#)」コマンドを参照してください。
- 管理ポート (フロントまたは背面アクセス) の構成については、349 ページの「[Set Management Port](#)」を参照してください。

Sun Netra CT900 サーバシャーシの代わりにサードパーティー製のシャーシを使用している場合は、使用しているシステムマネージャーまたはシェルフマネージャーソフトウェアを変更し、電力割当量のネゴシエーションと電力の印加を開始する前に、「Ethernet ポートアクセスの設定」OEM コマンドを IPMC に送信するようにします。このコマンドは、M2 状態のあと、M4 状態の前に送信する必要があります。

例:

スロットの管理ポートの状態を表示します。

```
# clia mgmtportstate 3
Pigeon Point Shelf Manager Command Line Interpreter

Current slot MGMT port Routing Information
-----
Slot      MGMT
-----
3         Rear (ARTM)
```

---

## minfanlevel

構文:

```
minfanlevel [level]
```

目的:

このコマンドは、最小ファンレベルを表示または設定します。通常の状態では、温度状態が正常である間、冷却管理アルゴリズムによりシステム内のファンのレベルが徐々に下げられます。しかし、冷却管理アルゴリズムは、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最小レベルよりもファンレベルを下げることはありません。

最小ファンレベルのデフォルト値は 1 です。最小ファンレベルを大きな値に設定しても、コマンド `clia setfanlevel` または ATCA コマンド `SetFanLevel` を RMCP に対して実行してファンレベルをそれよりも下げることができなくなるわけではありません。最小ファンレベルは、冷却管理機能によるファンレベルの自動管理のみに影響します。

このコマンドでパラメータを指定しない場合、現在の最小ファンレベルが表示されます。

このコマンドで整数のパラメータを指定すると、最小ファンレベルがパラメータの値に設定されます。

例:

```
# clia minfanlevel 3
Pigeon Point Shelf Manager Command Line Interpreter

Minimal Fan Level is set to 3

# clia minfanlevel
Pigeon Point Shelf Manager Command Line Interpreter

Minimal Fan Level is 3
#
```

---

## muxstate

構文:

`muxstate slot-number`

目的:

このコマンドは、指定したスロット内のブレードの、現在のマルチプレクサ (MUX) の状態とポートのルーティング構成を IPMC に対してクエリーします。このコマンドは、Oracle Solaris OS が動作しているホストのみで有効です。

シェルフマネージャーは、ブレードをアクティブ化する前に IPMC がホスト MUX 構成をプログラムできるように、シェルフのミッドプレーン FRU に格納されているスロットの MUX 構成情報を IPMC に提供します。

---

**注** – ShMM MUX 構成は、Oracle Solaris ホストの NIU ドライバ構成に一致している必要があります。さもないと、構成に失敗します。

---

MUX 構成を ShMM に保存するために `setmuxconfig` コマンドが使用され、保存されている MUX 構成を表示するために `getmuxconfig` コマンドが使用されます。詳細は、コマンド `getmuxconfig` および `setmuxconfig` を参照してください。

例:

物理スロット 4 のノードボード上の MUX に対するポートルーティング情報を表示します。

```
# cli muxstate 4
Pigeon Point Shelf Manager Command Line Interpreter

Current slot 10GbE NIU-XAUI Routing Information
-----
Slot    XAUI-1          XAUI-2
-----
4       Zone3 (ARTM)    Zone2 (Back Plane)
```

---

## networkelementid

構文:

```
networkelementid ["id"]
```

目的:

このコマンドはキャリア固有であり、必ずしもすべての ShMM キャリアでサポートされているわけではありません。

このコマンドは、このパラメータが現在のキャリアでサポートされている場合、ネットワーク要素の識別子を表示または設定します。ネットワーク要素の識別子を設定するには、スーパーユーザー (UID 0) 特権が必要です。

コマンド行パラメータ ID として指定されたネットワーク要素の識別子は、特定のキャリアで定義されている形式である必要があります。

コマンド行でパラメータを指定しない場合、現在のネットワーク要素の識別子が表示されます。



例:

```
# clia networkelementid
Pigeon Point Shelf Manager Command Line Interpreter

Network Element ID: "0123456789A"
#
# clia networkelementid "01234567890"
Pigeon Point Shelf Manager Command Line Interpreter

Network Element ID is set successfully to "01234567890"
#
```

---

## poll

構文:

poll

目的:

このコマンドは、「デバイス ID の取得」コマンドをすべての IPMB アドレスに送信することで、IPMB-0 上の IPM コントローラの再検出を開始します。

このコマンドは、主に、IPM コントローラのホットスワップ状態マシンのサポートがオプションで、IPMB 上の新しい IPM コントローラがシェルフマネージャーによってすぐに認識されない可能性がある、PICMG 2.x シェルフで有効です。コマンド poll を実行すると、シェルフマネージャーが新しい IPM コントローラを認識します。

AdvancedTCA シェルフでは、このコマンドは不要です。これは、新しい IPM コントローラが、最初にホットスワップイベントを送信するときに、シェルフマネージャーによって自動的に認識されるためです。ただし、IPMB-0 生成再検出サイクルが必要な場合に、このコマンドを AdvancedTCA シェルフで使用できます。

例:

```
# clia poll
Pigeon Point Shelf Manager Command Line Interpreter

IPMB polling thread started
#
```

---

# sel

## 構文:

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

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

## 目的:

このコマンドは、指定した IPM コントローラ (デフォルトでは IPMB アドレス 20h にあるコントローラ) 上のシステムイベントログ (SEL) を表示します。オプションのパラメータ *record-count* では、SEL 内のレコードを、レコード番号 *starting-entry* から始めて何個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、表示する最初の SEL レコードの、SEL の先頭からの相対エントリ番号です。*record-count* と *starting-entry* の範囲は、1 から SEL 内の全レコード数までです。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 レコードについて、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、単語 Event が表示されるイベントだけがサポートされています)
- タイムスタンプ (タイムスタンプがあるレコードの場合)
- 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
- イベントを生成したセンサーの種類と番号
- イベント/読み取りタイプコード
- raw 形式と処理形式 (使用可能な場合) の、3 バイトのイベントデータ

コマンド `sel clear` は、指定した IPM コントローラ (デフォルトでは IPMB アドレス 20h にあるコントローラ) 上の SEL をクリアします。

`-v` オプションを指定すると、SEL エントリの出力がよりわかりやすくなります。

例:

シェルフマネージャー上で SEL を読み取ります。

```
# clia sel info
Pigeon Point 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
Pigeon Point 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
0x006F or 2A (Sun legacy): 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
```

エントリ # 15 (0x0f) から 5 個の SEL エントリを取得します。

```
# clia sel 20 5 15
Pigeon Point 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 をクリアします。

```
# clia sel clear
Pigeon Point Shelf Manager Command Line Interpreter

SEL clear: issued successfully
      SEL clearing completed
# clia sel
Pigeon Point Shelf Manager Command Line Interpreter

SEL is empty
#
```

---

## sendamc

### 構文:

```
sendamc addr amc-address netfn command-code [parameter1 ... parameterN]
```

### 目的:

このコマンドを使用すると、対応する IPM コントローラの背後に存在する Advanced Management Controller (AMC) に、透過的な方法で IPMI コマンドを送信できます。このコマンドのすべてのパラメータは、00h - FFh の範囲の 16 進数です。プレフィックス "0x" は不要です。ターゲットコントローラは *AMC-address* パラメータで指定します。このパラメータが 70h よりも大きい場合は、IPMB-L 上の実際の AMC アドレスです。70h よりも小さい場合は、対応する AMC を表す FRU デバイス ID です。コマンドの NetFn コードは *netfn* パラメータで指定します。コマンドのコードは、*command-code* パラメータで指定します。コマンドの要求データバイトは、*parameter1*、*parameter2* などで表します。

このコマンドは IPMI コマンドの完了コードを報告し、応答データは 16 進数のバイトとして表示されます。

例:

「デバイス ID の取得」コマンドを AMC (IPMB アドレス 84h、FRU ID 1) に送信します。コマンドの NetFn は 06h であり、コマンドのコードは 01h です。このコマンドには要求データは必要ないため、*parameter1*、*parameter2* などは指定しません。

```
# clia sendamc 84 1 6 1
Pigeon Point Shelf Manager Command Line Interpreter

Completion code: 0x0 (0)
Response data: 34 80 01 20 51 29 0A 40 00 EF BE
#
```

「デバイス ID の取得」コマンドを AMC (IPMB アドレス 84h、AMC アドレス 72h) に送信します。コマンドの NetFn は 06h であり、コマンドのコードは 01h です。このコマンドには要求データは必要ないため、*parameter1*、*parameter2* などは指定しません。

```
# clia sendamc 84 72 6 1
Pigeon Point Shelf Manager Command Line Interpreter

Completion code: 0x0 (0)
Response data: 34 80 01 20 51 29 0A 40 00 EF BE
#
```

---

## sendcmd

構文:

```
sendcmd IPMB-address netfn command-code [parameter1 ... parameterN]
```

目的:

このコマンドを使用すると、透過的な方法で IPM コントローラに IPMI コマンドを送信できます。このコマンドのすべてのパラメータは、00h - FFh の範囲の 16 進数です。プレフィックス "0x" は不要です。ターゲットコントローラは *IPMB-address* パラメータで指定します。コマンドの NetFn コードは *netfn* パラメータで指定します。コマンドのコードは、*command-code* パラメータで指定します。コマンドの要求データバイトは、*parameter1*、*parameter2*、などで表します。

このコマンドは IPMI コマンドの完了コードを報告し、応答データは 16 進数のバイトとして表示されます。

例:

「デバイス ID の取得」コマンドをシェルフマネージャ (IPMB アドレス 20h) に送信します。コマンドの NetFn は 06h であり、コマンドのコードは 01h です。このコマンドには要求データは必要ないため、*parameter1*, *parameter2* などは指定しません。

```
# clia sendamc 84 1 6 1
Pigeon Point Shelf Manager Command Line Interpreter

Completion code: 0x0 (0)
Response data: 00 80 02 30 51 BF 0A 40 00 00 00
#
```

---

## 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]]
```

目的:

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。ターゲットセンサーは、その IPM コントローラの IPMB アドレスと、センサー番号またはセンサー名 (二重引用符で囲まれた、センサー SDR からのデバイス ID 文字列) で指定します。センサー名もセンサー番号も指定しない場合は、指定した IPM コントローラにあるすべてのセンサーに関する情報が表示されます。パラメータを何も指定しない場合は、既知のすべてのセンサーに関する情報が表示されます。

このコマンドを使用すると、ターゲットコントローラが複数の LUN 上にあるセンサーをサポートしている場合に、センサー番号に論理ユニット番号 (LUN) を付けて修飾することができます。LUN が省略された場合、すべての LUN の指定したセンサー番号を持つセンサーに関する情報が表示されます。*lun* には、値 0、1、3 のいずれかを指定できます (LUN 2 は予約されています)。

センサー名は通常コントローラ内で一意であることが前提であるため、センサー名は LUN 番号で修飾しません。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の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。

- 所有している 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 からのデバイス ID 文字列)、およびセンサーにアクセス可能な LUN
- センサーの種類とイベント/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)

冗長モードの場合のみ、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 の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)

冗長モードでは、しきい値ベースのセンサーの場合のみ、次の情報が表示されます。

- センサーユニット: ベースおよび変更
- ユニットの割合、修飾子、レート
- アナログ形式とフラグ
- 線形化パラメータ、M、B、K1、K2 係数
- 許容範囲と精度係数
- 定格、通常最大、通常最小、最大、最小の値
- 上限: 非クリティカル、クリティカル、回復不能
- 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: 正および負

例:

IPM コントローラ 20 上のセンサー FAN TRAY 0 に関する標準情報を取得します。

```
# clia sensor 20 "Fan Tray 0"
Pigeon Point Shelf Manager Command Line Interpreter

20: LUN: 0, Sensor # 12 ("FAN Tray 0")
    Type: Discrete (0x6f), "Entity Presence" (0x25)
    Belongs to entity: (0x1e, 96) [FRU # 3]
#
```

IPM コントローラ 8a 上のセンサー 5 に関する詳細情報を取得します。

```
# clia sensor -v 8a 5
Pigeon Point Shelf Manager Command Line Interpreter

8a: LUN: 0, Sensor # 5 ("Board Temp")
  Type: Threshold (0x01), "Temperature" (0x01)
  Belongs to entity: (0x3, 96) [FRU # 0]
  Assertion Mask: 0x0a80
    Upper Non-Critical Going High
    Upper Critical Going High
    Upper Non-Recoverable Going High
  Deassertion Mask: 0x7a80
    Upper Non-Critical Going High
    Upper Critical Going High
    Upper Non-Recoverable Going High
  Settable / Readable Mask: 0x3838
    Upper Non-Critical Threshold is Comparison Returned
    Upper Critical Threshold is Comparison Returned
    Upper Non-Recoverable Threshold Comparison is Returned
    Upper Non-Critical Threshold is Readable
    Upper Critical Threshold is Readable
    Upper Non-Recoverable Threshold is Readable
    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 = 100, B = 0, K1 = 0, K2 = -2
  Tolerance = 0, Accuracy = 0, Accuracy EXP = 0
  Analog Flags: 0x0
  Nominal: 60 (0x3c), Normal max: 100 (0x64), Normal min: -40
  (0xd8)
  Sensor max: 120 (0x78), Sensor min: -40 (0xd8)
  Upper Thresholds:
    Non-Critical: 48 (0x30) Critical: 60 (0x3c) Non-Recoverable:
  68 (0x44)
  Lower Thresholds:
    N/A
  Hysteresis:
    Positive: 0 (0x00), Negative 0 (0x00)

#
```



上記と同じですが、センサーの LUN を明示的に指定しています。

```
# clia sensor -v 8a 0:5
Pigeon Point Shelf Manager Command Line Interpreter

8a: LUN: 0, Sensor # 5 ("Board Temp")
  Type: Threshold (0x01), "Temperature" (0x01)
  Belongs to entity: (0x3, 96) [FRU # 0]
  Assertion Mask: 0x0a80
    Upper Non-Critical Going High
    Upper Critical Going High
    Upper Non-Recoverable Going High
  Deassertion Mask: 0x7a80
    Upper Non-Critical Going High
    Upper Critical Going High
    Upper Non-Recoverable Going High
  Settable / Readable Mask: 0x3838
    Upper Non-Critical Threshold is Comparison Returned
    Upper Critical Threshold is Comparison Returned
    Upper Non-Recoverable Threshold Comparison is Returned
    Upper Non-Critical Threshold is Readable
    Upper Critical Threshold is Readable
    Upper Non-Recoverable Threshold is Readable
    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 = 100, B = 0, K1 = 0, K2 = -2
  Tolerance = 0, Accuracy = 0, Accuracy EXP = 0
  Analog Flags: 0x0
  Nominal: 60 (0x3c), Normal max: 100 (0x64), Normal min: -40
  (0xd8)
  Sensor max: 120 (0x78), Sensor min: -40 (0xd8)
  Upper Thresholds:
    Non-Critical: 48 (0x30) Critical: 60 (0x3c) Non-Recoverable:
  68 (0x44)
  Lower Thresholds:
    N/A
  Hysteresis:
    Positive: 0 (0x00), Negative 0 (0x00)

#
```

---

# 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]]
```

## 目的:

このコマンドは、指定したセンサーの実効値 (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 (ディスクリットセンサーの場合) を示します。ターゲットセンサーは、その IPM コントローラの IPMB アドレスと、センサー番号またはセンサー名 (二重引用符で囲まれた、センサー SDR からのデバイス ID 文字列) で指定します。センサー名もセンサー番号も指定しない場合は、指定した IPM コントローラ上にあるすべてのセンサーの値が表示されます。パラメータを何も指定しない場合は、既知のすべてのセンサーの値が表示されます。

このコマンドを使用すると、ターゲットコントローラが複数の LUN 上にあるセンサーをサポートしている場合に、センサー番号に論理ユニット番号 (LUN) を付けて修飾することができます。LUN が省略された場合、すべての LUN の指定したセンサー番号を持つセンサーに関する情報が表示されます。*lun* には、値 0、1、3 のいずれかを指定できます (LUN 2 は予約されています)。

センサー名は通常コントローラ内で一意であることが前提であるため、センサー名は LUN 番号で修飾しません。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の情報が表示されます。

各センサーについて、次の情報が示されます。

- 所有している 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 からのデバイス ID 文字列)、およびセンサーにアクセス可能な LUN
- センサーの種類とイベント/読み取りタイプコード
- しきい値ベースのセンサーの場合はセンサー値、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のマスク (raw 形式)
- しきい値超過状態 (16 進形式、デコーディングあり)

値または表明された状態は、raw 形式と処理形式の両方で示されます。処理形式では、アナログ値が M、B、R に従って変換され、単位名とともに示されます (例: 27 degrees)。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/読み取りコードが 2 の場合、表明された状態 0 は Transition to Idle として示されます)。

例:

IPM コントローラ FE 上の温度センサー Local Temp に関するセンサーデータ値を取得します。

```
# clia sensordata FE "Local Temp"
Pigeon Point 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 上のディスクリート (ホットスワップ) センサー (#0) に関するセンサーデータ値を取得します。

```
# clia sensordata 9c 0
Pigeon Point 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
```

同じセンサーに関するセンサーデータ値を取得しますが、LUN で明示的に修飾しています。

```
# clia sensordata 9c 0:0
Pigeon Point 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
```

## 目的:

このコマンドは、指定したセンサーの raw 値を示します。コマンド `sensorread` と `sensordata` の唯一の違いは、コマンド `sensorread` はターゲット IPM コントローラの有無やセンサー番号の妥当性を確認せずに、単に IPMB 経由で「センサー読み取りの取得」要求を直接送信するという点です。このコマンドはセンサーの 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。

このコマンドを使用すると、ターゲットコントローラが複数の LUN 上にあるセンサーをサポートしている場合に、センサー番号に論理ユニット番号 (LUN) を付けて修飾することができます。LUN が省略された場合は、LUN 0 が使用されます。`lun` には、値 0、1、3 のいずれかを指定できます。(LUN 2 は予約されています。)

各センサーについて、次の情報が示されます。

- 所有している 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 からのデバイス ID 文字列)、およびセンサーにアクセス可能な LUN
- センサーの種類とイベント/読み取りタイプコード
- しきい値ベースのセンサーの場合はセンサー値、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のマスク (raw 形式)

## 例:

IPM コントローラ FC 上のセンサー 4 に関するセンサーデータ値を取得します。`sensorread` コマンドは未処理のセンサー値のみを示すことに注意してください。また、明示的な LUN を使用したコマンド例にも注意してください。

```
# clia sensordata fc 4
Pigeon Point 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
Pigeon Point 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
Pigeon Point 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
```

### 目的:

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示します。この情報には、次の項目が含まれています。

- 実現可能な最大セッション数と、現在アクティブなセッションの数
- 現在アクティブなセッションごとに、次の情報が示されます

- セッションハンドル
- セッションのアクティブ化時に使用されたユーザー ID とユーザー名
- セッションの最大特権レベル
- IPMI チャネル番号と種類
- (LAN セッションの場合) ピア IP アドレスとポート番号

例:

```
# clia session
Pigeon Point 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
#
```

---

## setacousticlevel

構文:

```
setacousticlevel ETSI NEBS-A NEBS-U
```

目的:

このコマンドは、システムの騒音レベルを設定します。このコマンドはサーバーシャーシがある場合に使用でき、SUNCT900 冷却アルゴリズムを使用している場合にのみ適用できます。このコマンドを使用すると、カスタム構成要件に応じて、デフォルトのファン速度をさまざまな騒音レベルに設定することができます。騒音レベルを設定すると、変数が変更され、デフォルトの冷却アルゴリズムに従ってファン速度が設定されます。この変更は動的に行われ、持続的で ShMM の再起動は必要ありません。

「getacousticlevel」も参照してください。

例:

システムの騒音レベルとファン速度を設定します。

```
# clia setacousticlevel NEBS-A
Pigeon Point Shelf Manager Command Line Interpreter

Changed the system acoustic level to 'NEBS-A'
Fan speed changed to '7'.
```

---

## setbootdev

構文:

```
getbootdev IPMB-0-address FRU-device-ID | IPMB-L-address boot-device
```

目的:

このコマンドは、指定した IPM コントローラのシステム起動パラメータを設定します。AdvancedMC アクセスを対象にしない場合は、このコマンドの 2 番目のパラメータを 0 に設定してください。2 番目のパラメータが 70h を超えると、AMC アドレスの IPMB-L アドレスとして扱われます。これ以外の場合、2 番目のパラメータは FRU ID として扱われ、AMC アドレスマッピングで IPMB-L アドレスに変換されます。

*boot-device* パラメータの値は次のとおりです。

- 1 または pxe (起動前の実行環境)
- 2 または disk (デフォルトのハードドライブ)
- 3 または safe (デフォルトのハードドライブ、セーフモード)
- 4 または diag (デフォルトの診断パーティション)
- 5 または cd (デフォルトの CD/DVD)
- 14 または bios (BIOS)
- 15 または floppy (フロッピー/一次リムーバブルメディア)

例:

IPMB アドレス 82h にある IPM コントローラのシステム起動オプションを pxe (起動前実行環境) として設定します。

```
# clia setbootdev 82 0 1
Pigeon Point Shelf Manager Command Line Interpreter

Set boot device option: status = 0x0 (0)
Boot device set to 1 (Force PXE)
Response data (raw): A2
#
```

---

## setextracted

構文:

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

目的:

このコマンドは、指定した FRU がシェルフから物理的に抽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。指定した FRU の状態が M7 の場合は、シェルフマネージャーによって状態が M0 (FRU が物理的に存在しない) に設定されます。

例:

```
# clia setextracted 9c 0
Pigeon Point 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
```

## 目的:

このコマンドは、コマンドパラメータで指定した FRU によって制御されるファンに対し、新しいレベルを設定します。最小値は 1、最大値は 15 です。

このコマンドに `all` 修飾子を付けた場合は、シェルフ内の既知のすべてのファンが同じレベルに設定されます。

## 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。

```
# cli setfanlevel 20 2 5  
Pigeon Point Shelf Manager Command Line Interpreter  
  
20: FRU # 2 Set Fan Level to: 5  
#
```

シェルフ内の既知のすべてのファンのファンレベルを 4 に設定します。

```
# cli setfanlevel all 4  
Pigeon Point Shelf Manager Command Line Interpreter  
  
72: FRU # 0 Set Fan Level to: 4  
76: FRU # 0 Set Fan Level to: 4  
#
```

---

# setfanpolicy

## 構文:

```
setfanpolicy fan-tray-addr fan-tray-fru_id ENABLE | DISABLE [timeout]
[-s addr|site_type [fru_id | site_number]
```

## 目的:

このコマンドは、シェルフ FRU にファンジオグラフィーレコードがある場合、これに加えて冷却管理のためにファントレーを有効または無効にします。

パラメータ *fan-tray-addr* および *fan-tray-fru\_id* では、ファントレーを指定します。ファントレーに DISABLE ポリシーを指定した場合は、追加パラメータ *timeout* でこのポリシーの有効期間を指定できます。*timeout* パラメータは秒単位で処理された後、PICMG 3.0 仕様に従って 5 秒単位で丸められます。*timeout* パラメータの値は 21 分 (1260 秒) を超えないようにしてください。*timeout* の最小値は 5 秒です。*timeout* 変数を指定しない場合、DISABLE ポリシーは無制限とみなされます。

フラグ *-s* は、ファントレーの対象サイトを定義するパラメータの前に付けます。

*site\_type* パラメータには、Board, PEM, ShelfFRU, ShelfManager, FanTray, FanFilterTray, Alarm, Mezzanine, PMC, RTMのいずれかの値を指定できます。

数値引数を 16 進数として処理させる場合は、プレフィックス "0x" を使用しないとエラーが返されます。

## 例:

ファンポリシーを 60 秒間無効にします。ファントレーは IPMB アドレス 20h、FRU ID 3 にあります。このファントレーの対象サイトは IPMB アドレス 12h、FRU ID 0 にあります。

```
# clia setfanpolicy 0x20 3 DISABLE 60 -s 0x12 0
Pigeon Point Shelf Manager Command Line Interpreter

Fan policy updated successfully
#
```

ファンポリシーを無期限で無効にします。ファントレーは IPMB アドレス 20h、FRU ID 3 にあります。このファントレーの対象サイトは、サイトの種類「PICMG ボード」およびサイト番号 7 で定義されます。

```
# clia setfanpolicy 0x20 3 DISABLE -s board 7
Pigeon Point Shelf Manager Command Line Interpreter

Fan policy updated successfully
#
```

IPMB アドレス 20h、FRU ID 3 にあるファントレーと、このファンのすべての対象サイトに対し、ファンポリシーを有効にします。

```
# clia setfanpolicy 0x20 3 ENABLE
Pigeon Point Shelf Manager Command Line Interpreter

Fan policy updated successfully
#
```

---

## 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*

### 目的:

このコマンドを使用すると、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* は、IPM コントローラの IPMB アドレスです。2 番目の引数 *fru-id* は、FRU デバイス ID です。3 番目の引数には、LED ID (数値) または ALL を指定できます。後者の場合は、指定した操作がすべての LED に適用されます。

引数 *LedOp* 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 をオンにします。
- OFF – LED をオフにします。
- LOCAL – LED のローカル制御に戻します。

- BLINK – LED を *onTime* で指定したミリ秒間オンにした後、*offTime* で指定したミリ秒間オフにする操作を繰り返すことで、LED を点滅させます。
- TEST – ランプテストを *onTime* で指定したミリ秒間実行します。

TEST 操作の場合は、*onTime* の値を 12800 ミリ秒 (12.8 秒) 未満にしてください。  
BLINK 操作の場合は、*onTime* と *offTime* の両方の値を 10 - 2500 ミリ秒 の範囲内にしてください。

オプションのパラメータ *LedColor* は、英字名または10進数値で色を指定します。色を表す英字名は、以下のように PICMG 3.0 仕様に従って 10 進数値に対応しています (このパラメータを指定しない場合は、デフォルトの LED 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 にある IPM コントローラの FRU #0 の LED #1 をオフにします。

```
# clia setfruledstate 20 0 1 OFF
Pigeon Point Shelf Manager Command Line Interpreter

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

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

```
# clia setfruledstate 20 0 1 LOCAL
Pigeon Point Shelf Manager Command Line Interpreter

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

IPMB アドレス 20h にある IPM コントローラの FRU #0 の LED #1 で、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。オンの期間は 100 ミリ秒、オフの期間は 200 ミリ秒です。

```
# clia setfruledstate 20 0 0 BLINK 100 200
Pigeon Point 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
```

## 目的:

このコマンドは、指定したセンサーに対し、指定したヒステリシスの値を設定します。センサーはしきい値ベースのセンサーであることが必要です。このセンサーが指定されたしきい値のヒステリシスをサポートしており、ヒステリシスが設定可能である必要があります。

このコマンドを使用すると、ターゲットコントローラが複数の LUN 上にあるセンサーをサポートしている場合に、センサー番号に論理ユニット番号 (LUN) を付けて修飾することができます。このコマンドは、pos 引数がある場合は正のヒステリシスを設定し、neg 引数がある場合は負のヒステリシスを設定します。

## 例:

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

```
# cli sethysteresis FC 2 pos 10
Pigeon Point 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)
```

## 目的:

このコマンドは、ターゲット IPM コントローラ上の IPMB リンクを有効または無効にします。2 番目の引数は、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。最後の引数は、実行する操作を定義します (1 - リンクを有効化、0 - リンクを無効化)。

このコマンドは、バス型環境と放射線状環境では動作が異なります。バス型環境と、IPMB ハブ以外のターゲット IPM コントローラを使用した放射線状環境では、引数 *link* は使用しません。放射線状環境で IPMB ハブコントローラを使用する場合、引数 *link* はオプションです。

*link* がある場合は、このコマンドにより特定の放射線状 IPMB リンク (1 - 95) が有効または無効になります。*link* がない場合は、このコマンドにより放射線状システム内の IPMB ハブ上にあるリンクがすべて有効または無効になります。

## 例:

IPMB アドレス 92h にある 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  
Pigeon Point Shelf Manager Command Line Interpreter  
  
Command executed successfully
```

シェルフマネージャー (IPMB ハブ) 上にある放射線状の IPMB リンク 3、バス B を有効にします。

```
# clia setipmbstate 20 B 3 1  
Pigeon Point Shelf Manager Command Line Interpreter  
  
Command executed successfully
```

# setlanconfig

## 構文:

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

## 目的:

このコマンドは、指定したチャンネルで、指定した 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を明示的に指定してください。

表 A-3 に、setlanconfig コマンドで使用できる LAN 構成パラメータの名前と番号を示します。

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 の各特権レベルの認証タイプ有効化フラグを含む 5 つの 8 ビットの値。
ip	3	LAN チャンネルに割り当てられた IP アドレスを含む、小数点付き 10 進数表記の文字列値。
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、小数点付き 10 進数表記の文字列値。
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 3 つの 8 ビットの値。 <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]) およびサービスの種類 (ビット [4:1])</li></ul>
arp_control	10	LAN チャンネル上の ARP 動作を制御する 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 の送信を有効にする</li></ul>
arp_interval	11	固定小数点形式の Gratuitous ARP 間隔 (ここで、整数部は秒、小数部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、小数点付き 10 進数表記の文字列値。
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、小数点付き 10 進数表記の文字列値。

表 A-3 setlanconfig の LAN 構成パラメータ (続き)

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	指定された設定セクタで識別される宛先の種類。このパラメータには、設定セクタを指定する必要があります。各宛先の種類のエントリには、次のフィールドが含まれます。 <ul style="list-style-type: none"> <li>宛先の種類 (0 - 7)</li> <li>警告確認応答フラグ</li> <li>警告確認応答タイムアウト/再試行間隔の秒数 (1 - 256)</li> <li>再試行回数 (0 - 7)</li> </ul>
destination_address	19	指定した設定セクタに関連付けられた宛先アドレス。このパラメータには、設定セクタを指定する必要があります。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>ゲートウェイセクタ: 0 - デフォルトを使用、1 - バックアップを使用</li> <li>IP アドレス (小数点付き 10 進形式の文字列)</li> <li>MAC アドレス (コロン [:] 記号で区切られた、6 つの 16 進バイト値からなる文字列)</li> </ul>

## auth\_enables

### 構文:

```
setlanconfig channel auth_enables value1 value2 value3 value4 value5
setlanconfig channel 2 value1 value2 value3 value4 value5
```

### 目的:

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) のそれぞれに対し、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の種類を示します。パラメータ値は 5 つのバイトのシーケンスで表され、各バイトがそれぞれの特権レベルに対応します。これは、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* - *value5* では、これらのバイトの値を16 進数形式で表す必要があります。シェルフマネージャーでは現在、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。このため、これらの特権レベルに対応する *value1* パラメータと *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
Pigeon Point Shelf Manager Command Line Interpreter

Authentication Type Enables set successfully
#
```

## ip

構文:

```
setlanconfig channel ip value
setlanconfig channel 3 value
```

目的:

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。値には、IP アドレスを小数点付き 10 進数表記で指定してください。

例:

```
# clia setlanconfig 1 ip 172.16.2.203
Pigeon Point Shelf Manager Command Line Interpreter

IP Address set successfully
#
```

## subnet\_mask

### 構文:

```
setlanconfig channel subnet_mask value  
setlanconfig channel 6 value
```

### 目的:

このコマンドは、チャンネルで使用される現在の IP サブネットマスクを設定します。値には、サブネットマスクを小数点付き 10 進数表記で指定してください。

### 例:

```
# clia setlanconfig 1 subnet_mask 255.255.255.0  
Pigeon Point 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
```

### 目的:

このコマンドは、シェルフマネージャーの IP 4 ヘッダーパラメータを設定します。値は、コロンで区切られた 16 進表記の 3 つのシングルバイト値で表されます (*value1*、*value2*、*value3*)。バイトの内容は、IPMI 1.5 仕様の 19.2 節に準拠しており、次の属性が含まれます。

- 生存期間 – バイト 1
- IP ヘッダーフラグ (ビット [7:5]) – バイト 2
- 優先順位 (ビット [7:5]) およびサービスの種類 (ビット [4:1]) – バイト 3

例:

```
# clia setlanconfig 1 ipv4_hdr_param 37 E0 11  
Pigeon Point Shelf Manager Command Line Interpreter  
  
IPv4 Header Parameters set successfully  
#
```

## arp\_control

構文:

```
setlanconfig channel arp_control value  
setlanconfig channel 10 value
```

目的:

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。

- 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 に設定します。

例:

```
# clia setlanconfig 1 arp_control 3  
Pigeon Point Shelf Manager Command Line Interpreter  
  
BMC-generated ARP control set successfully  
#
```

## arp\_interval

### 構文:

```
setlanconfig channel arp_interval value  
setlanconfig channel 11 value
```

### 目的:

このコマンドは、チャンネルで使用される現在の ARP 間隔を設定します。値には、固定小数点形式の秒/ミリ秒数を指定してください (小数部が付いてもかまいません)。このパラメータは IPMI で定義されているため、500 ミリ秒で割り切れる最長時間間隔に切り詰められます。

### 例:

```
# clia setlanconfig 1 arp_interval 3.5  
Pigeon Point 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
```

### 目的:

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 アドレスを設定します。値には、IP アドレスを小数点付き 10 進数表記で指定してください。

### 例:

```
# clia setlanconfig 1 dft_gw_ip 172.16.2.100  
Pigeon Point 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
```

### 目的:

このコマンドは、チャンネルで使用され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。値には、IP アドレスを小数点付き 10 進数表記で指定してください。

### 例:

```
# clia setlanconfig 1 backup_gw_ip 172.16.2.100  
Pigeon Point Shelf Manager Command Line Interpreter  
  
Backup Gateway Address set successfully  
#
```

## community

### 構文:

```
setlanconfig channel community value  
setlanconfig channel 16 value
```

### 目的:

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設定します。値には、二重引用符で囲んだ文字列を指定してください。

### 例:

```
# clia setlanconfig 1 community "Community"  
Pigeon Point 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
```

### 目的:

このコマンドは、*set-selector* と同等の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。*value1*、*value2*、*value3* の各値は、IPMI 仕様の 19.2 節に従って新しい宛先に関する情報を示します。提供される情報は次のとおりです。

- 警告の宛先の種類 (PET トラップまたは OEM 宛先、警告の確認応答を行うかどうか)
- 警告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
Pigeon Point 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
```

### 目的:

このコマンドは、*set-selector* と同等の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で、必要な情報を指定します。

- *gateway-sel* – 使用するゲートウェイ。デフォルトゲートウェイの場合は 0、バックアップゲートウェイの場合は 1
- *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
Pigeon Point Shelf Manager Command Line Interpreter

Destination Addresses set successfully
#
```

---

## setlocked

構文:

```
setlocked IPMB-address fru-id 0 | 1
setlocked board n 0 | 1
setlocked shm n 0 | 1
setlocked fan_tray n 0 | 1
```

目的:

このコマンドでは、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 (ロック解除の場合は 0、ロックの場合は 1 を指定)。FRU は、所有している IPM コントローラの IPMB アドレスと FRU デバイス ID を使用して指定します。FRU デバイス ID 0 は、PICMG 3.0 コンテキストの IPM コントローラ本体を指定します。

ロックビットは、PICMG 3.0 の仕様に従って、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ている場合、この移行は許可されません。シェルフマネージャーが「停止」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。

このコマンドは、バックアップシェルフマネージャーでも実行できます。その場合、ロックビットは、バックアップシェルフマネージャーにローカルな FRU に対して指定された状態にのみ設定できます。

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
Pigeon Point Shelf Manager Command Line Interpreter

Lock set successfully to 0x0
#
```

---

## setmgmtportroute

構文:

```
setmgmtportroute [slot-number] [1/0]
```

目的:

このコマンドでは、管理ポートルートを設定します。この情報は、ミッドプレーンに OEM レコードとして保存されます。

構成は、ブレードの存在にかかわらず、永続的であり各スロットに対して指定されます。

- 現在の管理ルート構成については、[216 ページ](#)の「[getmgmtportroute](#)」コマンドを参照してください。
- 管理ポートの状態やルーティング構成を判定するための IPMC のクエリーについては、[239 ページ](#)の「[mgmtportstate](#)」コマンドを参照してください。
- 管理ポート (フロントまたは背面アクセス) の構成については、[349 ページ](#)の「[Set Management Port](#)」を参照してください。

例:

スロットの管理ポートルーティングの状態を表示します。

```
# clia setmgmtportroute 4 0
Pigeon Point Shelf Manager Command Line Interpreter

MGMT port Routing Information from Midplane FRU
-----
Slot   MGMT
-----
  4    Rear (ARTM)
```



---

# setmuxconfig

## 構文:

```
setmuxconfig slot-number [port:zone] [port:zone]
```

## 目的:

このコマンドは、指定したスロットのシェルフのミッドプレーン OEM マルチレコードに保存されているマルチプレクサ (MUX) 構成 (またはポートルーティング) の情報を更新します。このコマンドを使用すると、ボードのスワップ後も MUX 構成を維持することができます。

port パラメータには、10 GbE NIU ポートを指定し、zone パラメータには、そのポートのルーティング先を指定します。有効な port 値は、XAUI-1 および XAUI-2 (省略形は x1 および x2) です。有効な zone 値は zone2 および zone3 (省略形は z2 および z3) です。zone2 はバックプレーンで、zone3 は拡張背面切り替えモジュール (ARTM) です。

シェルフマネージャーは、ブレードをアクティブ化する前に IPMC がホスト MUX 構成をプログラムできるように、シェルフのミッドプレーン FRU に格納されているスロットの MUX 構成情報を IPMC に提供します。詳細は、コマンド [getmuxconfig](#) および [muxstate](#) を参照してください。

## 例:

物理スロット 5 のノードボード上の MUX に対するポートルーティング情報を設定します。

```
# cli setmuxconfig 5 x1:z3 xaui2:Zone2
Pigeon Point Shelf Manager Command Line Interpreter

Slot 5 : Successfully modified the mux routing.
```

# setpefconfig

## 構文:

```
setpefconfig parameter-name additional-parameters  
setpefconfig parameter-number additional-parameters
```

## 目的:

このコマンドは、指定した 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できる PEF 構成パラメータの名前と番号を示します。

表 A-4 getlanconfig 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 (PEF の有効化、PEF 起動遅延の有効化など) を表す 8 ビットの値
action_control	2	PEF のアクショングローバル制御フラグ (リセットの有効化、電源切断の有効化など) を表す 8 ビットの値
startup_delay	3	システムの電源投入とリセット後に PEF を遅延させる時間 (秒単位)
alert_startup_delay	4	システムの電源投入とリセット後に警告を遅延させる時間 (秒単位)

表 A-4 getlanconfig の PEF 構成パラメータ (続き)

event_filter	6	<p>指定した設定セクタで識別されるイベントフィルタテーブルエントリ。次に示す、IPMI 仕様バージョン 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>• フィルタの構成</li> <li>• イベントフィルタアクション</li> <li>• 警告ポリシー番号</li> <li>• イベントの重要度</li> <li>• ジェネレータ ID バイト 1</li> <li>• ジェネレータ ID バイト 2</li> <li>• センサーの種類</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>指定した設定セクタで識別されるイベントフィルタテーブルエントリの最初のバイト</p>
alert_policy	9	<p>指定した設定セクタで識別される警告ポリシーテーブルエントリ。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>• チャンネル番号 (4 ビットの値)</li> <li>• 宛先セクタ (4 ビットの値)</li> <li>• 警告文字列セット/セクタ</li> </ul>
system-guid	10	<p>PET トラップの GUID フィールドを設定するために使用される GUID</p>
alert_string_key	12	<p>指定した設定セクタで識別される警告文字列キー。8 ビットの 2 つの値 (イベントフィルタ番号と警告文字列のセット) で構成されます</p>

表 A-4 getlanconfig の PEF 構成パラメータ (続き)

alert_string	13	指定した設定セクタで識別される警告文字列。
oem_filter	97	指定した設定セクタで識別される OEM フィルタテーブルエントリ。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 レコードタイプ範囲下限</li> <li>• バイト 2: SEL レコードタイプ範囲上限</li> <li>• バイト 3: 上記の範囲と一致するレコードタイプが含まれる SEL エントリに対して呼び出される警告ポリシー番号</li> </ul>

## control

### 構文:

```
setpefconfig control value
setpefconfig 1 value
```

### 目的:

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あり、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 の有効化
- 0x02 – PEF アクションに対するイベントメッセージの生成の有効化
- 0x04 – システムの電源投入とリセット時の PEF 起動遅延の有効化
- 0x08 – PEF 警告起動遅延の有効化

その他のビットは予約されているため、0 に設定します。値は 16 進数で入力してください。

### 例:

```
# cli setpefconfig control 7
Pigeon Point Shelf Manager Command Line Interpreter

PEF control set successfully
#
```

## action\_control

### 構文:

```
setpefconfig action_control value  
setpefconfig 2 value
```

### 目的:

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あり、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 – 警告アクションの有効化
- 0x02 – 電源切断アクションの有効化
- 0x04 – リセットアクションの有効化
- 0x08 – 電源再投入アクションの有効化
- 0x10 – OEM アクションの有効化
- 0x20 – 診断の中断の有効化

その他のビットは予約されているため、0 に設定します。値は 16 進数で入力してください。

### 例:

```
# clia setpefconfig action_control 3f  
Pigeon Point Shelf Manager Command Line Interpreter  
  
PEF action control set successfully  
#
```

## startup\_delay

### 構文:

```
setpefconfig startup_delay value  
setpefconfig 3 value
```

### 目的:

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あり、起動時に PEF 機能が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
Pigeon Point Shelf Manager Command Line Interpreter

PEF startup delay set successfully
#
```

## alert\_startup\_delay

構文:

```
setpefconfig startup_delay value
setpefconfig 4 value
```

目的:

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あり、起動時に警告機能が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
Pigeon Point 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
```

## 目的:

このコマンドは、*set-selector* と同等の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。新しい要素の内容は、IPMI仕様バージョン 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 (*value1 - value19*) で指定されます。

- フィルタの構成
- イベントフィルタアクション
- 警告ポリシー番号
- イベントの重要度
- ジェネレータ ID バイト 1
- ジェネレータ ID バイト 2
- センサーの種類
- 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、警告アクションをトリガーします。警告は、警告ポリシー 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  
Pigeon Point 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
```

### 目的:

このコマンドは、*set-selector* と同等の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最初の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 進数で指定してください。バイト内のビットの意味は次のとおりです。

- 0x80 – このフィルタは有効です
- 0x40 – このフィルタは製造元で事前に設定され、ソフトウェアでは変更できません

その他のビットは予約されているため、0 になります。

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  
Pigeon Point 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  
Pigeon Point 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
```

### 目的:

このコマンドは、指定した設定セレクタで識別される警告ポリシーテーブルエントリを設定します。新しい要素の内容は、IPMI 1.5 のテーブル 15-4 の定義に従ってエンコードされた、次の 5 つの 16 進数の数値 (*value1* - *value5*) で指定されます。

- 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)
- 宛先セレクタ (4 ビットの値)
- 警告文字列セット/セレクタ

### 例:

次の例では、次の属性を持つ警告ポリシーテーブルエントリ 2 を設定します。

- ポリシー番号 = 5
- 有効なポリシー = 常にこの宛先に警告を送信する
- 宛先チャンネル = 1
- 宛先セレクタ = 1
- 警告文字列セレクタ = すべてのイベントに対して文字列 1 を使用する

```
# cli setpefconfig alert_policy 2 5 8 1 1 1  
Pigeon Point 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
```

## 目的:

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 で警告の宛先に送信される GUID を表します。この GUID は、個別の GUID として定義することも、システム GUID と同じ GUID として定義することもできます。

`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「システム GUID の取得」コマンドの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
Pigeon Point Shelf Manager Command Line Interpreter

GUID set successfully
#
# clia setpefconfig system_guid none
Pigeon Point 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
```

### 目的:

このコマンドは、*set-selector* と同等のインデックスを持つ警告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。設定セレクタ 0 を使用して、揮発性警告文字列を指定できます。各キーは、警告を生成する目的でイベントフィルタを警告文字列と関連付けます。各キーは、イベントフィルタ番号と警告文字列番号で構成されます。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
Pigeon Point 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>
```

目的:

このコマンドは、*set-selector* と同等のインデックスを持つ警告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 を使用して、揮発性警告文字列を指定できます。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."
Pigeon Point 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
```

目的:

OEM フィルタテーブルは、Pigeon Point Systems が定義した IPMI 仕様の OEM 拡張です。これを使用すると、プラットフォームイベントに加えて、PEF を、OEM のタイムスタンプありまたはタイムスタンプなしの SEL エントリ (レコードタイプ範囲 C0h-FFh) に適用できます。

OEM フィルタテーブルの各エントリは、この OEM フィルタを適用するレコードタイプの範囲 (OEM レコードタイプの範囲内で) と、レコードタイプが一致するレコードが SEL に格納されたときに呼び出される警告ポリシー番号を定義します。

このコマンドは、OEM フィルタテーブルエントリを設定します。エントリ番号は、指定した設定セクタで識別されます。エントリは、次の 3 つの数値で構成されます。

- バイト 1: SEL レコードタイプ範囲下限
- バイト 2: SEL レコードタイプ範囲上限
- バイト 3: 上記の範囲と一致するレコードタイプが含まれる SEL エントリに対して呼び出される警告ポリシー番号

例:

```
# clia getpefconfig oem_filter
Pigeon Point 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
Pigeon Point Shelf Manager Command Line Interpreter

OEM filter set successfully
#
# clia getpefconfig oem_filter
Pigeon Point 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
#
```

---

## setpowerlevel

構文:

```
setpowerlevel IPMB-address fruid [pwr_lvl |OFF] [COPY]
```

*IPMB-address* の代わりに、次を使用することもできます。

```
board n
shm n
```

## 目的:

このコマンドでは、FRU の電力レベルを制御できます。その結果、IPMB-0 で、指定した IPM コントローラに対する「電力レベルの設定」コマンドが、シェルフマネージャーによって実行されます。シェルフマネージャーコアは電力レベルの管理と対応する電力割当量の追跡を行うため、このコマンドを使用する場合 (特に 0 以外の電力レベルを指定する場合) には、細心の注意が必要です。このコマンドのユーザーは、AdvancedTCA 仕様で定義されている AdvancedTCA 電源管理アーキテクチャを熟知している必要があります。

ターゲットの FRU は、IPM コントローラの IPMB アドレス、および FRU デバイス ID (*fruid*) で指定されます。また、ボード番号や専用のシェルフマネージャー番号を使用して、ターゲットの FRU を指定することもできます。

3 つ目の引数 *pwr\_lvl* は電力レベルです。指定できる電力レベルは 0h - 14 h (使用可能な場合) です。電力レベル 0 はキーワード OFF と同等です。この場合、可能であれば、指定した FRU の電源が切断されます。

電力レベルを指定しない場合、FRU の現在の電力レベルは変更されません。これは、電力レベルの値として 0xFF を指定する場合と同等です。指定した場合、電力レベルは、指定した FRU で以前に調整した消費電力値のいずれかを選択するインデックスになります。各消費電力値は、FRU での使用が承認されている最大消費電力 (ワット単位) に一致します。

電源が投入された AdvancedTCA FRU には常に、FRU (実際には FRU を表す IPM コントローラ) とシェルフマネージャーの間に設定された最大 20 (14h) の消費電力のセットがあります。*pwr\_lvl* 引数では、FRU で使用が承認される最大電力としてこのセットの消費電力のいずれかを選択します。

その後、別の変更が行われるまで、その FRU は承認された電力量を超える電力を消費することができません。現在および最大の電力レベル、および現在の電力レベルに関連付けられている承認消費電力 (ワット単位) は、`clia fru -v` コマンドでどの FRU でも使用できます。

オプションパラメータ `COPY` では、必要な電力レベルを現在の電力レベルにコピーするかどうかを指定します (背景については、AdvancedTCA 仕様を参照)。このパラメータを指定しない場合、必要な電力レベルは現在の電力レベルにコピーされません。

## 例:

```
# clia setpowerlevel 84 0 OFF
Pigeon Point Shelf Manager Command Line Interpreter

Operation completed with status = 0x0
#
```

# setsensordata

## 構文:

```
setsensordata <IPMB-address> <sensor-name> [reading [-r] <value>]  
[assertion_<mask>] [deassertion_<mask>] [ event_data <b1> <b2> <b3> |  
event_data_no_offset <b1> <b2> <b3>]
```

```
setsensordata <IPMB-address> [<lun>:]<sensor-number> [reading [-r] <value>]  
[assertion_<mask>] [deassertion_<mask>] [ event_data <b1> <b2> <b3> |  
event_data_no_offset <b1> <b2> <b3>]
```

-d オプションを使用すると、状態マスクで特定の状態が設定されたディスクリートセンサーのみが表示されます。

## 目的:

このコマンドは、指定したセンサーの読み取り、表明/非表明マスク、イベントデータバイトを変更します。センサーは、IPMI 仕様 v 2.0 の補足 E372 の定義に従って、設定可能として定義する必要があります (対応する SDR のセンサー初期化バイトでビット [7] を 1 に設定する必要があります)。

センサーは、所有している 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。また、ボード番号や専用のシェルフマネージャー番号を使用して、ターゲットの IPM コントローラを指定することもできます。

このコマンドを使用すると、ターゲットコントローラが複数の LUN 上にあるセンサーをサポートしている場合に、センサー番号に論理ユニット番号 (LUN) を付けて修飾することができます。<lun> には、値 0、1、3 を指定できます。(LUN 2 は予約されています。)LUN が省略された場合、最小の LUN の指定されたセンサー番号を持つセンサーにコマンドが適用されます (たとえば、コマンドで LUN 修飾子を明示的に指定せずにセンサー 3 を指定し、ターゲットコントローラで LUN 1 のセンサー 3 を公開し、LUN 3 の別のセンサー 3 を公開すると、LUN 1 のセンサー 3 にコマンドが適用されます)。

このコマンドのパラメータは、IPMI 仕様 v 2.0 の補足 E372 で定義されている IPMI コマンド「センサーの読み取りとイベントステータスの設定」の規則に従います。

節 reading [-r] <value> では、センサーの新しい読み取りを指定します。オプション -r を指定すると、<value> が生の値として解釈されます。このオプションを指定しない場合は、処理済みの値として解釈され、SDR データ (線形化、M、B など) を使用して生の値に変換されます。

節 assertion <mask> および deassertion <mask> では、センサーの現在の表明/非表明の条件マスクを 16 ビット値で指定します。各ビットの意味は次のように定義されています。

しきい値ベースのセンサーの場合:

- [15:12] - 予約済み、0000 にする必要がある
- [11] - 1b = 回復不能な上限上昇の表明/非表明条件
- [10] - 1b = 回復不能な上限低下の表明/非表明条件
- [9] - 1b = クリティカル上限上昇の表明/非表明条件
- [8] - 1b = クリティカル上限低下の表明/非表明条件
- [7] - 1b = 非クリティカル上限上昇の表明/非表明条件
- [6] - 1b = 非クリティカル上限低下の表明/非表明条件
- [5] - 1b = 回復不能な下限上昇の表明/非表明条件
- [4] - 1b = 回復不能な下限低下の表明/非表明条件
- [3] - 1b = クリティカル下限上昇の表明/非表明条件
- [2] - 1b = クリティカル下限低下の表明/非表明条件
- [1] - 1b = 非クリティカル下限上昇の表明/非表明条件
- [0] - 1b = 非クリティカル下限低下の表明/非表明条件

ディスクリットセンサーの場合:

- [15] - 予約済み、0 にする必要がある
- [14] - 1b = 状態 14 の表明/非表明イベントの発生
- [13] - 1b = 状態 13 の表明/非表明イベントの発生
- [12] - 1b = 状態 12 の表明/非表明イベントの発生
- [11] - 1b = 状態 11 の表明/非表明イベントの発生
- [10] - 1b = 状態 10 の表明/非表明イベントの発生
- [9] - 1b = 状態 9 の表明/非表明イベントの発生
- [8] - 1b = 状態 8 の表明/非表明イベントの発生
- [7] - 1b = 状態 7 の表明/非表明イベントの発生
- [6] - 1b = 状態 6 の表明/非表明イベントの発生
- [5] - 1b = 状態 5 の表明/非表明イベントの発生
- [4] - 1b = 状態 4 の表明/非表明イベントの発生
- [3] - 1b = 状態 3 の表明/非表明イベントの発生
- [2] - 1b = 状態 2 の表明/非表明イベントの発生
- [1] - 1b = 状態 1 の表明/非表明イベントの発生
- [0] - 1b = 状態 0 の表明/非表明イベントの発生

節 `event_data <b1> <b2> <b3>` および `event_data_no_offset <b1> <b2> <b3>` を使用して、センサーがイベントを生成するときに IPMI Platform Event 要求で送信されるイベントデータバイトを指定します。バリエーション `event_data` を使用すると、イベントオフセット (イベントデータバイト 1 の下位ニブル) が `<b1>` から取得されます。バリエーション `event_data_no_offset` を使用すると、イベントの発生時にイベントオフセットが自動的に生成され、バイト `<b1>` の下位ニブルが無視されます。

このコマンドは、バックアップシェルフマネージャーでも実行できます。その場合、コマンドは、バックアップシェルフマネージャーにローカルなセンサーに対してのみ適用されます。

次の例では、設定可能なセンサー Eth0 Front の種類が Entity Presence になっており、キャリアボードのフロントパネルの Ethernet 接続状態を監視および制御しています。コマンド `setsensordata` が実行され、センサーの状態が Entity Present から Entity Absent に変わります。Entity Absent はフロントパネルの Ethernet 接続をオフにします。

#### 例:

Eth0 Front の状態が Entity Present であることを確認します。

```
# clia sensordata 10 10
Pigeon Point Shelf Manager Command Line Interpreter
10: LUN: 0, Sensor # 10 ("Eth0 Front")
Type: Discrete (0x6f), "Entity Presence" (0x25)
Belongs to entity (0xf0, 0x60): FRU # 0
Status: 0xc0
All event messages enabled from this sensor
Sensor scanning enabled
Initial update completed
Sensor reading: 0x00
Current State Mask 0x0001
Entity Present
```

イベントを Present から Absent に変更します (表明イベント)。

```
# clia setsensordata 10 10 assertion 2
Pigeon Point Shelf Manager Command Line Interpreter
Sensor data set successfully
```

Eth0 Front の状態が Entity Absent であることを確認します。

```
# clia sensordata 10 10
Pigeon Point Shelf Manager Command Line Interpreter
10: LUN: 0, Sensor # 10 ("Eth0 Front")
Type: Discrete (0x6f), "Entity Presence" (0x25)
Belongs to entity (0xf0, 0x60): FRU # 0
```



```
Status: 0xc0
All event messages enabled from this sensor
Sensor scanning enabled
Initial update completed
Sensor reading: 0x00
Current State Mask 0x0002
Entity Absent
```

---

## 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、次を使用することもできます。

```
board n
shm n
```

### 目的:

このコマンドは、指定し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、所有している 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。また、ボード番号や専用のシェルフマネージャー番号を使用して、ターゲットの IPM コントローラを指定することもできます。

このコマンドを使用すると、ターゲットコントローラが複数の LUN 上にあるセンサーをサポートしている場合に、センサー番号に論理ユニット番号 (LUN) を付けて修飾することができます。LUN には、値 0、1、3 のいずれかを指定できます。(LUN 2 は予約されています。) LUN が省略された場合、最小の LUN の指定されたセンサー番号を持つセンサーにコマンドが適用されます (たとえば、コマンドで LUN 修飾子を明示的に指定せずにセンサー 3 を指定し、ターゲットコントローラで LUN 1 のセンサー 3 を公開し、LUN 3 の別のセンサー 3 を公開すると、LUN 1 のセンサー 3 にコマンドが適用されます)。

例:

IPM コントローラ FE の温度センサー Local Temp 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
Pigeon Point Shelf Manager Command Line Interpreter

Event enable mask set successfully
#
# clia getsensoreventenable -v fe "Local Temp"
Pigeon Point 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
#
```

同じセンサーで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
Pigeon Point 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、次を使用することもできます。

```
board n
shm n
```

## 目的:

このコマンドは、指定したセンサーの指定したしきい値の現在の値を変更します。センサーは、所有している 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。ターゲットのセンサーはしきい値ベースのセンサーであることが必要です。パラメータ *threshold-type* には、次の記号値のいずれかを指定できます。

- 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、ターゲット値が処理形式で指定されます (電圧センサーの場合はボルト、温度センサーの場合は摂氏で表されます)。オプション *-r* は、代わりに生の値 (通常はセンサー固有の規則に従って変換されたバイトサイズの数値) が使用されることを示します。

このコマンドを使用すると、ターゲットコントローラが複数の LUN 上にあるセンサーをサポートしている場合に、センサー番号に論理ユニット番号 (LUN) を付けて修飾することができます。 *lun* には、値 0、1、3 のいずれかを指定できます。(LUN 2 は予約されています。) LUN が省略された場合、最小の LUN の指定されたセンサー番号を持つセンサーにコマンドが適用されます (たとえば、コマンドで LUN 修飾子を明示的に指定せずにセンサー 3 を指定し、ターゲットコントローラで LUN 1 のセンサー 3 を公開し、LUN 3 の別のセンサー 3 を公開すると、LUN 1 のセンサー 3 にコマンドが適用されます)。

## 例:

IPM コントローラ 9C の温度センサー *emulated temp* の非クリティカル上限しきい値を 99 °C に設定します。

```
# clia threshold 9c 2
Pigeon Point 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
Pigeon Point Shelf Manager Command Line Interpreter

Threshold set successfully
#
# clia threshold 9c 0:2
Pigeon Point 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
#
```

---

## setuserlabel

### 構文:

```
setuserlabel shelf shelf-name
setuserlabel slot slot-number slot-name
```

### 目的:

このコマンドは、Sun Netra CT900 サーバーのシェルフおよびスロットにユーザーが割り当てた名前を設定します。スロット番号は、1 - 14 の間で指定してください。ユーザーが割り当てたラベルは、[userlabel](#) コマンドを使用して表示されます。

ユーザーが割り当てたシェルフおよびスロットの名前は、コンソールセッション中にシェルフおよびボードを一意に識別するために使用できます。詳細は、[148 ページの「コンソールセッション中のボードのユーザーラベルの表示」](#)を参照してください。

例:

```
# clia setuserlabel shelf ATCA02

Pigeon Point Shelf Manager Command Line Interpreter

Shelf label set successfully
#
# clia setuserlabel slot 6 CP3020-06

Pigeon Point Shelf Manager Command Line Interpreter

Slot#6 label set successfully
#
```

---

## shelf

構文:

`shelf subcommand`

次のサブコマンドがサポートされます。

- `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

#### 目的:

コマンド `shelf` は、主なシェルフ FRU 情報と、そのシェルフについて選択した現在の動作データを表示します。シェルフ FRU 情報の一部のフィールドは変更可能です。このコマンドによって表示または変更される情報の種類は、コマンドパラメータで指定します。

以降の節では、`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]
```

#### 目的:

この `shelf` コマンド構文は、主なシェルフ FRU 情報と、そのシェルフについて選択した現在の動作データを表示します。このコマンドによって表示される情報の種類は、コマンドパラメータで指定します。表 A-5 に、`shelf` コマンドで使用できるパラメータを示します。

表 A-5 shelf コマンドのパラメータ

コマンドパラメータ	提供される情報
cooling_state (cs に省略可能)	<p>シェルフの現在の冷却状態を示します。</p> <ul style="list-style-type: none"> <li>• 通常 - すべての温度センサーが通常の動作温度を示しています。</li> <li>• マイナーアラート - 少なくとも 1 つの温度センサーがマイナーアラート状態になっています。メジャーアラートまたはクリティカルアラート状態のセンサーはありません。</li> <li>• メジャーアラート - 少なくとも 1 つの温度センサーがメジャーアラート状態になっています。クリティカルアラート状態のセンサーはありません。</li> <li>• クリティカルアラート - 少なくとも 1 つの温度センサーがクリティカルアラート状態になっています。</li> </ul>
fans_state (fs に省略可能)	<p>シェルフ内のファン回転速度計の現在の状態を示します。</p> <ul style="list-style-type: none"> <li>• 通常 - すべてのファン回転速度計センサーが通常の動作速度を示しています。</li> <li>• マイナーアラート - 少なくとも 1 つのファン回転速度計センサーがマイナーアラート状態になっています。メジャーアラートまたはクリティカルアラート状態のセンサーはありません。</li> <li>• メジャーアラート - 少なくとも 1 つのファン回転速度計センサーがメジャーアラート状態になっています。クリティカルアラート状態のセンサーはありません。</li> <li>• クリティカルアラート - 少なくとも 1 つのファン回転速度計センサーがクリティカルアラート状態になっています。</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を表示します。次の情報が提供されます。</p> <ul style="list-style-type: none"> <li>• シェルフアドレス (シェルフの種類に従って表示)</li> <li>• アドレステーブルエントリのリスト (エントリごとに、ハードウェアアドレス、サイト番号、サイトの種類を表示)</li> </ul>

表 A-5 shelf コマンドのパラメータ (続き)

power_distribution (pd に省略可能)	<p>各電源フィードについて、次の情報が表示されます (大半は、シェルフ FRU 情報のシェルフ配電レコードから得られます)。</p> <ul style="list-style-type: none"> <li>• 使用可能な最大外部電流</li> <li>• 最大内部電流</li> <li>• 期待される最小動作電圧</li> <li>• 使用可能な実電力</li> <li>• 現在使用されている電力</li> <li>• フィードに接続されている FRU のリスト (FRU ごとに、ハードウェアアドレスと FRU デバイス ID を表示)</li> </ul>
power_management (pm に省略可能)	<ul style="list-style-type: none"> <li>• シェルフ FRU 情報内のシェルフ電源管理レコード。このレコードには、FRU 電源記述子のリストが含まれます。記述子ごとに、次の情報が表示されます</li> <li>• ハードウェアアドレス</li> <li>• FRU デバイス ID</li> <li>• FRU の最大電力容量</li> <li>• シェルフマネージャーで制御されるアクティブ化</li> <li>• 次の電源投入前の遅延</li> </ul>
pci_connectivity (pcic に省略可能)	<p>シェルフ FRU 情報内のシェルフ PCI 接続レコード。次の情報が提供されます。</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>• 位置アドレス</li> <li>• インタフェース番号</li> <li>• 使用可能なシステムスロット</li> </ul>
ha_connectivity (ha に省略可能)	<p>シェルフ FRU 情報内のシェルフ HA 接続レコード。次の情報が提供されます。</p> <ul style="list-style-type: none"> <li>• 放射線状接続のサポート</li> </ul>
h110_connectivity (h110c に省略可能)	<p>シェルフ FRU 情報内のシェルフ H110 接続レコード。次の情報が提供されます。</p> <ul style="list-style-type: none"> <li>• 位置アドレス</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 情報内のシェルフ二地点間接続レコード。次の情報が提供されます。</p> <ul style="list-style-type: none"> <li>• チャンネルの種類</li> <li>• チャンネル数</li> <li>• スロット/ハードウェアアドレス</li> <li>• チャンネル記述子</li> </ul>



コマンドパラメータ `cooling_state` および `fans_state` の場合は、詳細オプション `-v` を使用できます。このオプションは、コマンドパラメータの前に入力してください (例: `clia shelf -v cooling_state`)。このオプションを使用した場合は、センサー (温度またはファン回転速度計) のリストに続き、現在の状態が表示されます。各センサーは、組 (*IPMB-address, sensor\_number*) 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
Pigeon Point Shelf Manager Command Line Interpreter

Cooling state: "Normal"
#
```

シェルフのファン回転速度計の状態を取得します (詳細)。

```
# clia shelf -v fans_state
Pigeon Point 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
Pigeon Point 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
```

```
Pigeon Point 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
```

```
Pigeon Point 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
#
```

## 使用可能な最大外部電流の変更

### 構文:

```
shelf maxcurrent [feed] current
```

### 目的:

このコマンドは、指定したフィード番号に対して使用可能な最大外部電流を設定し、シェルフ内の既知のシェルフ FRU 情報インスタンスをすべて更新します。*feed* パラメータが省略された場合は、シェルフ FRU 情報内の最初のフィード (フィード 0) に対して値が設定されます。

パラメータ *feed* は、シェルフ FRU 情報内での当該フィードの記述順序に基づいた、0 から始まるフィード番号です。

パラメータ *current* は、必要な電流値 (アンペア単位) です。

例:

フィード 0 に使用可能な最大外部電流を 50 アンペアから 99 アンペアに変更します。

```
# clia shelf pd
Pigeon Point 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
Pigeon Point Shelf Manager Command Line Interpreter

Updating Shelf FRU Info
Cached information updated

# clia shelf pd
Pigeon Point 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
```

#

## 期待される最小動作電圧の変更

### 構文:

```
shelf minvoltage [feed] voltage
```

### 目的:

このコマンドは、指定したフィード番号に対して期待される最小動作電圧を設定し、シェルフ内の既知のシェルフ FRU 情報インスタンスをすべて更新します。*feed* パラメータが省略された場合は、シェルフ FRU 情報内の最初のフィード (フィード 0) に対して値が設定されます。

パラメータ *feed* は、シェルフ FRU 情報内での当該フィードの記述順序に基づいた、0 から始まるフィード番号です。

パラメータ *voltage* は、必要な値です。

例:

フィード 0 に対して期待される最小動作電圧を変更します。

```
# clia shelf pd
Pigeon Point 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
Pigeon Point Shelf Manager Command Line Interpreter

  Updating Shelf FRU Info
  Cached information updated

# clia shelf pd
Pigeon Point 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で制御されるアクティブ化」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]
```



## 目的:

このコマンドは、指定した IPM コントローラの指定した FRU について、「シェルフマネージャーで制御されるアクティブ化」フィールドを表示または変更します。このコマンドは、シェルフのアクティブ化および電源管理レコード内の既存のエントリに対してのみ、「シェルフマネージャーで制御されるアクティブ化」フラグを変更します。また、シェルフマネージャーで使用されるシェルフ FRU 情報のキャッシュバージョンも更新されます。これにより、シェルフマネージャーを再起動しなくても、「シェルフマネージャーで制御されるアクティブ化」フィールドの新しい値がただちに有効になります。

パラメータ *hardware-addr* は、16 進形式で 7 ビットのハードウェアアドレスです。

パラメータ *fru-id* は、16 進形式の FRU ID です。0xFE は、このハードウェアアドレスにあるすべての 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で制御されるアクティブ化を有効にします。

```
# clia shelf pm
Pigeon Point 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
```

```
Pigeon Point 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
```

```
Pigeon Point 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で制御される非アクティブ化」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]
```

### 目的:

このコマンドは、指定した IPM コントローラの指定した FRU について、「シェルフマネージャーで制御される非アクティブ化」フィールドを表示または変更します。このコマンドは、シェルフのアクティブ化および電源管理レコード内の既存のエントリに対してのみ、「シェルフマネージャーで制御される非アクティブ化」フラグを変更します。また、シェルフマネージャーで使用されるシェルフ FRU 情報のキャッシュバージョンも更新されます。これにより、シェルフマネージャーを再起動しなくても、「シェルフマネージャーで制御される非アクティブ化」フィールドの新しい値がただちに有効になります。

パラメータ *hardware-addr* は、16 進形式で 7 ビットのハードウェアアドレスです。

パラメータ *fru-id* は、16 進形式の FRU ID です。0xFE は、このハードウェアアドレスにあるすべての FRU を意味します。

値が 0 の場合は、シェルフマネージャーで制御される非アクティブ化が有効になり、シェルフマネージャーがボードを非アクティブ化できるようになります。値が 1 の場合は、シェルフマネージャーで制御される非アクティブ化が無効になり、シェルフマネージャーがボードを自動的に非アクティブ化できなくなります。シェルフマネージャーで制御される非アクティブ化が無効な場合は、システム管理者がボードの非アクティブ化を行います。

例:

ハードウェアアドレス 0x42 (IPMB アドレス 0x84) の IPM コントローラに対し、シェルフマネージャーで制御される非アクティブ化を有効にします。

```
# clia shelf pm
Pigeon Point 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:
200 Watts
  Shelf Manager Controlled Auto-Activation: Disabled
  Shelf Manager Controlled Auto-Deactivation: Enabled
  Delay Before Next Power On: 0.0 seconds

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

# clia shelf deactivation 42 0xfe 0
Pigeon Point 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
Pigeon Point 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:
200 Watts
    Shelf Manager Controlled Auto-Activation: Disabled
    Shelf Manager Controlled Auto-Deactivation: Enabled
    Delay Before Next Power On: 0.0 seconds

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

#
```

## FRU の最大電力容量の変更

### 構文:

```
shelf pwrcapability hardware-addr fru-id value
shelf pwrcapability board n value
shelf pwrcapability fan_tray n value
```

### 目的:

このコマンドは、指定した IPM コントローラの指定した FRU について、「FRU の最大電力容量」フィールドを変更します。

---

注 – 「FRU の最大電力容量」フィールドの設定値は、ご使用のシェルフ環境で安全な値を超えないようにしてください。

---

このコマンドは、シェルフのアクティブ化および電源管理レコード内の既存のエントリに対してのみ、このフィールドを変更します。また、シェルフマネージャーで使用されるシェルフ FRU 情報のキャッシュバージョンも更新されます。これにより、シェルフマネージャーを再起動しなくても、「FRU の最大電力容量」フィールドの新しい値がただちに有効になります。

- パラメータ *hardware-addr* は、16 進形式で 7 ビットのハードウェアアドレスです。
- パラメータ *fru-id* は、16 進形式の FRU ID です。0xFE は、このハードウェアアドレスにあるすべての FRU を意味します。
- パラメータ *value* は、このフィールドの新しい値 (ワット単位) です。取り得る値の範囲は 0 - 65535 です。

例:

ハードウェアアドレス 0x42 (IPMB アドレス 0x84) の IPM コントローラに対し、FRU の最大電力容量を 150 ワットに設定します。

```
# clia shelf pm
Pigeon Point 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
Pigeon Point Shelf Manager Command Line Interpreter

Updating Shelf FRU Info
    Cached information updated
#
# clia shelf pm
Pigeon Point 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
#
```



## 次の電源投入前の遅延の変更

### 構文:

```
shelf pwrdelay hardware-addr fru-id value
shelf pwrdelay board n value
shelf pwrdelay fan_tray n value
```

### 目的:

このコマンドは、指定した IPM コントローラの指定した FRU について、「次の電源投入前の遅延」フィールドを変更します。このコマンドは、シェルフのアクティブ化および電源管理レコード内の既存のエントリに対してのみ、このフィールドを変更します。また、シェルフマネージャーで使用されるシェルフ FRU 情報のキャッシュバージョンも更新されます。これにより、シェルフマネージャーを再起動しなくても、「次の電源投入前の遅延」フィールドの新しい値がただちに有効になります。

- パラメータ *hardware-addr* は、16 進形式で 7 ビットのハードウェアアドレスです。
- パラメータ *fru-id* は、16 進形式の FRU ID です。0xFE は、このハードウェアアドレスにあるすべての FRU を意味します。
- パラメータ *value* は、このフィールドの新しい値 (1/10 秒単位) です。取り得る値の範囲は 0 - 63 です。

### 例:

ハードウェアアドレス 0x42 (IPMB アドレス 0x84) の IPM コントローラに対し、次の電源投入前の遅延を 5 秒に設定します。

```
# clia shelf pm
Pigeon Point 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
Pigeon Point Shelf Manager Command Line Interpreter
```

```
Updating Shelf FRU Info
  Cached information updated
#
# clia shelf pm
Pigeon Point 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 のアクティブ化準備の許可の変更

### 構文:

```
shelf allowance value
```

### 目的:

このコマンドは、「FRU のアクティブ化準備の許可」パラメータを変更します。

パラメータ *value* は、このパラメータの新しい値 (秒単位) です。取り得る値の範囲は 0 - 255 です。

### 例:

FRU のアクティブ化準備の許可を 5 秒に設定します。

```
# clia shelf pm
Pigeon Point 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
Pigeon Point Shelf Manager Command Line Interpreter

Updating Shelf FRU Info
#
# clia shelf pm
Pigeon Point 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 アクティブ化および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* は、次のように置き換えることができます。

```
board n
fan_tray n
```

## 目的:

このコマンドは、シェルフ FRU 情報内の FRU アクティブ化および電源記述子の順序を変更します。このコマンドで順序変更できるのは、既存の記述子のみです。現在の実装についても、1つのシェルフのアクティブ化および電源管理レコード内の記述子しか順序変更できません。また、シェルフマネージャーで使用されるシェルフ FRU 情報のキャッシュバージョンも更新されます。これにより、シェルフマネージャーを再起動しなくても、記述子の新しい順序がただちに有効になります。

- パラメータ *hardware-addr-1* は、移動する必要がある記述子を表す、16進形式で7ビットのハードウェアアドレスです。
- パラメータ *fru-id-1* は、移動する必要がある記述子を表す、16進形式の FRU ID です。0xFE は、このハードウェアアドレスにあるすべての FRU を意味します。
- パラメータ *hardware-addr-2* は、*hardware-addr-1/fru-id-1* 記述子を置く位置の前またはあとにある記述子を表す、16進形式で7ビットのハードウェアアドレスです。
- パラメータ *fru-id-2* は、*hardware-addr-1/fru-id-1* 記述子を置く位置の前またはあとにある記述子を表す、16進形式の FRU ID 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置きます。

```
# clia shelf pm
Pigeon Point 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
Pigeon Point Shelf Manager Command Line Interpreter

Updating Shelf FRU Info
  Cached information updated

#
```

```
# clia shelf pm
Pigeon Point 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 情報の更新

### 構文:

```
shelf info_refresh
```

### 目的:

このコマンドを実行すると、シェルフ内のシェルフ FRU 情報について以前に検出されたソースが再度読み取られ、有効なシェルフ FRU 情報が含まれているソースが再評価されます。有効なシェルフ FRU 情報が確認されていることを前提として、すべてのシェルフ FRU 情報ストレージデバイスと、キャッシュされるシェルフ 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 (3.6.4 節) 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時に使用可能なシェルフ 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 情報が含まれた FRU 情報デバイスが 2 つ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 情報ソースを判断します。この選択は、各ストレージデバイスに含まれるデータを検証し、その内容を比較することで行われます。正しく選択が行われると、シェルフマネージャーはキャッシュされるシェルフ 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 情報ソースの更新に使用され、シェルフ FRU 情報の唯一のソースとして扱われます。したがって、シェルフ FRU 情報関連の操作はすべてマスターコピーを処理することになり、マスターコピーの変更は増分更新として、すべてのシェルフ FRU 情報ソースデバイスに自動的に伝播されます。

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、変更内容はシェルフマネージャーの再起動後に初めて完全に有効になります。

例:

更新成功: 2 つのシェルフ FRU 情報ソースが一致しています。

```
# clia shelf info refresh
Pigeon Point 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

#
```

例:

更新失敗: 両方のデータソースに不一致または無効なデータが含まれています。

```
# clia shelf info refresh
Pigeon Point 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
```

## 目的:

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、両方の EEPROM とマスターコピー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
Pigeon Point Shelf Manager Command Line Interpreter

Starting the Shelf FRU Info source device update
#
```

---

# shelfaddress

## 構文:

```
shelfaddress [up-to-20-characters-of-the-shelf-address]
shelfaddress -x byte1 ..... byteN
```

## 目的:

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。

オプション `-x` を指定しない場合は、最大 20 文字の ASCII 文字を含む文字列を二重引用符で囲んで新しいシェルフアドレスを指定します。

オプション `-x` を指定する場合は、空白文字で区切られた 16 進形式のバイト列として新しいシェルフアドレスを指定します。2 桁の 16 進数で表されるバイトを 20 個まで指定できます (プレフィックス `0x` は省略可能です)。

シェルフアドレスは、テキストとバイナリデータを区別する型インジケータとともに格納されます。テキストとして指定した場合にはテキストとして格納され、16 進数で指定した場合にはバイナリデータとして格納されます。

このコマンドを使用してシェルフアドレスを表示すると、型インジケータに応じてテキストまたは 16 進形式のバイト列として表示されます。

例:

```
# clia shelfaddress
Pigeon Point Shelf Manager Command Line Interpreter

Shelf Address Info: "1234"
#
# clia shelfaddress "NEW SHELF ADDRESS"
Pigeon Point Shelf Manager Command Line Interpreter

Shelf Address Info set successfully
#
# clia shelfaddress
Pigeon Point Shelf Manager Command Line Interpreter

Shelf Address Info: "NEW SHELF ADDRESS"
#
# clia shelfaddress -x 01 02 03 04 05
Pigeon Point Shelf Manager Command Line Interpreter

Shelf Address Info set successfully
#
# clia shelfaddress
Pigeon Point Shelf Manager Command Line Interpreter

Shelf Address Info: " 0x01 0x02 0x03 0x04 0x05 "
#
```



---

# shmstatus

## 構文:

```
shmstatus
```

## 目的:

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。冗長モードでは、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
Pigeon Point 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 [version]
```

## 目的:

このコマンドは、Sun Netra ノード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、Sun Netra CT900 サーバーに取り付けられた Sun Netra CP3x60 ノードボードのみに有効です。

パラメータ *slot-number* には、Sun Netra CP3x60 ボードのスロット番号を指定します。*version* オプションを使用すると、完全なバージョン情報が表示されます。

## 例:

スロット 2 の Sun 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
Pigeon Point Shelf Manager Command Line Interpreter

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
```

## 目的:

このコマンドは、問題があると認識される FRU のリストを表示します。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、リストに含まれます。

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のリストを表示します。

```
# clia showunhealthy
Pigeon Point Shelf Manager Command Line Interpreter
There are no unhealthy components in the shelf.
#
```

---

# switchover

## 構文:

```
switchover
```

```
switchover -force
```

## 目的:

このコマンドは、シェルフマネージャーの冗長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。

アクティブな ShMM が動かなくなった場合のために、バックアップシェルフマネージャーからの強制的スイッチオーバー (CLI コマンド `switchover -force`) がサポートされています。この場合、シェルフマネージャー間のネゴシエーションは行われず、バックアップシェルフマネージャーがただちにハードウェアの制御を引き継ぎます。

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
```

目的:

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ます。

- add
- delete
- enable
- name
- passwd
- channel

### 目的:

`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、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。

以降の節では、`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]
```

### 目的:

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
Pigeon Point Shelf Manager Command Line Interpreter

1: ""
    Channels 0-15 Privilege level: "Administrator"
    Flags: "IPMI Messaging"
#
# clia user -v
Pigeon Point 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
```

目的:

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 - コールバックに制限される

- *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
Pigeon Point Shelf Manager Command Line Interpreter

1: ""
    Channels 0-15 Privilege level: "Administrator"
    Flags: "IPMI Messaging"
#
# clia user add 9 "root" 0x40 4 "PICMG guru"
Pigeon Point Shelf Manager Command Line Interpreter

User 9 added successfully
#
# clia user
Pigeon Point 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
```

目的:

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。

```
# cli user delete 10
Pigeon Point Shelf Manager Command Line Interpreter

User 10 deleted successful
#
# cli user
Pigeon Point 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
```

目的:

このコマンドは、*userid* に指定され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。

```
# cli user
Pigeon Point 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
Pigeon Point Shelf Manager Command Line Interpreter

User 9 disabled successfully
#
# clia user -v
Pigeon Point 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
Pigeon Point Shelf Manager Command Line Interpreter

User 9 enabled successfully
#
# clia user
Pigeon Point 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
```

### 目的:

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。

```
# cli user
Pigeon Point Shelf Manager Command Line Interpreter

1: ""
    Channels 0-15 Privilege level: "Administrator"
    Flags: "IPMI Messaging"
9: "root"
    Channels 0-15 Privilege level: "Administrator"
    Flags: "IPMI Messaging"
#
# cli user name 9 newby
Pigeon Point Shelf Manager Command Line Interpreter

User 9, name changed successfully
#
# cli user
Pigeon Point 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
```

目的:

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
Pigeon Point 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
Pigeon Point Shelf Manager Command Line Interpreter

User 9, password changed successfully
#
# clia user
Pigeon Point 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ンネルのチャンネルアクセス設定の変更

構文:

```
user channel userid channel-number channel-access-flags privilege-level
```

目的: このコマンドは、指定されたチャンネルおよびユーザーのチャン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* – チャンネル番号
- *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 – コールバックに制限される
- *privilege-level* – ユーザーの特権レベル

例:

チャンネル 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
Pigeon Point Shelf Manager Command Line Interpreter

9: "newby"
    Channels 0-15 Privilege level: "Administrator"
    Flags: "IPMI Messaging"

#
# clia user channel 9 5 0x60 2
Pigeon Point Shelf Manager Command Line Interpreter

User 9, channel 5 access updated successfully
#
# clia user 9
Pigeon Point 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"
```

```
#
```

---

## userlabel

### 構文:

```
userlabel shelf | slot [slot-number]
```

### 目的:

このコマンドは、サーバーのシェルフおよびスロットにユーザーが割り当てた名前を表示します。スロット番号は、1 - 14 の間で指定してください。ラベルの設定には [setuserlabel](#) コマンドを使用します。

ユーザーが割り当てたシェルフおよびスロットの名前は、コンソールセッション中にシェルフおよびボードを一意に識別するために使用できます。詳細は、[148 ページの「コンソールセッション中のボードのユーザーラベルの表示」](#)を参照してください。

### 例:

```
# clia userlabel shelf  
  
Pigeon Point Shelf Manager Command Line Interpreter  
  
Shelf: "ATCA02"  
#  
# clia userlabel slot  
  
Pigeon Point Shelf Manager Command Line Interpreter  
  
Slot# 1: "CP3260-01"  
Slot# 2: "CP3220-02"  
Slot# 3: ""  
Slot# 4: ""  
Slot# 5: "CP3260-05"  
Slot# 6: "CP3020-06"  
Slot# 7: "Switch-07"  
Slot# 8: "Switch-08"  
Slot# 9: ""  
Slot# 10: "CP3260-10"
```

```
Slot# 11:  ""
Slot# 12:  "CP3220-12"
Slot# 13:  ""
Slot# 14:  ""
#
# clia userlabel slot 6

Pigeon Point Shelf Manager Command Line Interpreter

Slot# 6:   "CP3020-06"
#
```

---

## version

### 構文:

```
version
```

### 目的:

このコマンドは、シェルフマネージャーソフトウェアのバージョン情報を表示します。

### 例:

```
# clia version
Pigeon Point Shelf Manager Command Line Interpreter

Pigeon Point Shelf Manager ver. 2.4.2
Pigeon Point is a trademark of Pigeon Point Systems.
Copyright (c) 2002-2007 Pigeon Point Systems
All rights reserved
Build date/time: Jan 30 2007 16:39:37
Carrier: PPS; Subtype: 0; Subversion: 0
#
```

# シェルフマネージャの IPMI 拡張コマンドとセンサー

---

シェルフマネージャは、システムマネージャの便宜のために、OEM 定義の IPMI コマンドとセンサーをいくつか実装しています。

最初のメカニズムセットでは、PEF ベースのメカニズムに Telco アラーム制御の代替機能が提供されます。このメカニズムは、1つのセンサーと2つの OEM 拡張コマンドで構成されています。コマンドは、RMCP インタフェースを介してシステムマネージャで実行することが可能であり、これらのコマンドによって次のような機能が実装されます。

- 指定された Telco アラームの設定と解除を行う (設定と解除の両方の処理が同じコマンド内で原子的に実行されます)。
- Telco アラームセンサーの数を取得する。このセンサーを使用して、Telco アラームの現在の状態を IPMI に準拠した方法で読み取ることができます。

一部の ShMM キャリアでは、システムマネージャやその他の外部アプリケーションからアクセスできる汎用のデジタル出力ピンがサポートされています。IPMI の仕様では、こうしたエンティティに対する明示的なサポートがされていないため、システムマネージャがこのようなデジタル出力に対応できるように、第二の拡張コマンドセットが提供されています。

コマンドは、RMCP インタフェースを介してシステムマネージャで実行することが可能であり、これらのコマンドによって次のような機能が提供されます。

- 使用可能なデジタル出力のプロパティと数をクエリーする
- デジタル出力の現在の状態を取得する
- デジタル出力の設定と解除を行う

現在、これらのコマンドは、シェルフマネージャの一部の FRU 上の選択されたキャリアに実装されています (IPMB アドレス 20h)。ただし、今後はその他の IPMI コントローラや FRU にも実装される可能性があります。

最後の拡張セットでは、追加機能が提供されます。現在、次のような機能が含まれています。

- シェルフ FRU 情報のマルチレコードの内容を読み取る
- 抽出された FRU についてシェルフマネージャーに通知する
- シェルフマネージャーのスイッチオーバーを開始する
- イベント通知を購読する

このセットのコマンドは、OEM グループのコマンド (ネットワーク関数コード 2Eh) として実装され、要求と応答の最初の 3 バイトは、Pigeon Point Systems IANA: (0Ah, 40h, 00h) である必要があります。

## IPMI 拡張コマンド

Pigeon Point Systems (PSS) の拡張 IPMI コマンドの一覧を表 B-1 に示し、以降の各節でその内容について説明します。

表 B-1 シェルフマネージャーの IPMI 拡張コマンド

コマンド	NetFn	オペレーションコード
Notify Shelf Manager About an Extracted FRU	2Eh	2h
Initiate Shelf Manager Switchover	2Eh	3h
Reset IPMC Configuration Variables	0x2E	0x9A
Set AMC Time Out Parameter	0x2E	0xF1
Get AMC Time Out Parameter	0x2E	0xF0
Set Boot Page	0x2E	0x81
Get Boot Page	0x2E	0x82
Set Front Panel Reset Button State	0x2E	0x83
Get Front Panel Reset Button State	0x2E	0x84
Set IPMC Control Bits	0x2E	0xE9
Get IPMC Control Bits	0x2E	0xE8
Set Management Port	0x2E	0x9B
Get Management Port	0x2E	0x9C
Get Shelf FRU Record Data	2Eh	1h
Set Shelf FRU Record Data	2Eh	05h
Set SOL Fail Over Link Change Timeouts	0x2E	0xE7



表 B-1 シェルフマネージャーの IPMI 拡張コマンド (続き)

コマンド	NetFn	オペレーションコード
Get SOL Fail Over Link Change Timeouts	0x2E	0xE6
Get Version	0x2E	0x80
Set Thermal Trip	0x2E	E5
Get Thermal Trip	0x2E	0xE4
Subscribe for Event Notifications	2Eh	4h

## Notify Shelf Manager About an Extracted FRU

この拡張コマンドは、論理シェルフマネージャー (20h) に送信され、指定された FRU (通常は通信が失われた状態を示す M7 状態にある) がすでにシェルフ内ではなく、M0 状態に移行可能であり、そのすべてのリソースを解放できることをシェルフマネージャーに通知します。FRU デバイス ID が 0 の場合、コマンドは、指定された IPM コントローラで表現されるすべての FRU に適用されます。このコマンドは、CLI コマンド `setextracted` と同様の効力があります。

シェルフマネージャーは、Forced Mode が指定されている場合を除き、対象となる FRU が実際に Communication Lost (M7) 状態であることを確認します。それ以外の場合は、完了コード D5 (現在の状態ではサポートされていないコマンド) が返されます。

コマンド	NetFn	オペレーションコード
Notify Shelf Manager About an Extracted FRU	2Eh	2h

### 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	1	PPS IANA の下位バイト。値 0Ah が使用されます。
	2	PPS IANA の中位バイト。値 40h が使用されます。
	3	PPS IANA の上位バイト。値 00h が使用されます。
	4	IPMB アドレス。対象となる IPM コントローラの IPMB アドレスを示します。
	5	対象となる FRU デバイス ID。このコマンドの対象である FRU デバイス ID を示します。

タイプ	バイト	データフィールド
	6	フラグ。次に示すオプションのビットフィールドです。 [7:1] 予約済み。0 に設定されます。 [0] 強制モード。「強制モード」が使用される場合、このビットは 1b に設定されます。このモードの場合、シェルフマネージャーは、対象の FRU が実際に M7 状態にあるかどうかを確認しません。
応答データ	1	完了コード: [00] OK [C1] コマンドがサポートされていない [CC] 要求内に無効なデータがある (その他の完了コードについては、IPMI の仕様を参照)
	2	PPS IANA の下位バイト。値 0Ah が使用されます。
	3	PPS IANA の中位バイト。値 40h が使用されます。
	4	PPS IANA の上位バイト。値 00h が使用されます。

## Initiate Shelf Manager Switchover

この Pigeon Point の拡張コマンドは、シェルフマネージャーの論理アドレス (20h)、または、アクティブとバックアップのいずれかのシェルフマネージャーの物理アドレスを対象にすることができます。このコマンドは、アクティブからバックアップへのシェルフマネージャーのスイッチオーバーを開始します。スイッチオーバーを実行できない場合 (使用可能なバックアップシェルフマネージャーが存在しない場合など)、完了コード D5 (現在の状態ではサポートされていないコマンド) が返されます。

コマンド	NetFn	オペレーションコード
Initiate Shelf Manager Switchover	2Eh	3h

### 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	1	PPS IANA の下位バイト。値 0Ah が使用されます。
	2	PPS IANA の中位バイト。値 40h が使用されます。
	3	PPS IANA の上位バイト。値 00h が使用されます。

タイプ	バイト	データフィールド
	4	フラグ。アクションを指定するビットフィールド。 [0]: 1b = 現在のセッションでイベント通知を購読します。 0b = 購読を解除します。 [7:1] 予約済み。0 に設定されます。
応答データ	1	完了コード: [00] OK [C1] コマンドがサポートされていない [CC] 要求内に無効なデータがある (その他の完了コードについては、IPMI の仕様を参照)
	2	SPPS IANA の下位バイト。値 0Ah が使用されます。
	3	PPS IANA の中位バイト。値 40h が使用されます。
	4	PPS IANA の上位バイト。値 00h が使用されます。

## Reset IPMC Configuration Variables

IPMC 構成変数は、ファームウェアのアップグレード、およびダウングレードされるバージョンがこの機能をサポートしているかぎりにはダウングレードを通して保持されます。

IPMC には、SEEPROM (NV ストレージ) に格納されるパラメータのグループがあります。すべての変数が 1 つのデータ構造に含まれています。これらの各変数が関数を実行します。また、コマンドを介して各変数を設定したり読み取ることができます。これらの変数は、IPMC のリセット後に格納されます。ただし、以前は、新しいファームウェアやアップグレードされたファームウェアがインストールされると、これらの変数はデフォルト値にリセットされていました。

## Set AMC Time Out Parameter

ShMM、ペイロード、またはデバッグインタフェースから次のコマンドを使用して、AMC のタイムアウト値を設定します。

コマンド	NetFn	オペレーションコード
Set AMC Time Out Parameter	0x2E	0xF1

## 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	1	00
	2	00
	3	6F または 2A (Sun レガシー)
	4	遅延 LSB
	5	遅延 MSB
応答データ	1	完了コード: [00] OK [C1] コマンドがサポートされていない [CC] 要求内に無効なデータがある (その他の完了コードについては、IPMI の仕様を参照)
	2	00
	3	00
	4	6F または 2A (Sun レガシー)

## Get AMC Time Out Parameter

ShMM、ペイロード、またはデバッグインタフェースから次のコマンドを使用して、AMC のデフォルトのタイムアウト値を読み取ります。

コマンド	NetFn	オペレーションコード
Get AMC Time Out Parameter	0x2E	0xF0

## 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	1	00
	2	00
	3	6F または 2A (Sun レガシー)

タイプ	バイト	データフィールド
応答データ	1	完了コード: [00] OK [C1] コマンドがサポートされていない [CC] 要求内に無効なデータがある [CB] パラメータが設定されていない場合。  (その他の完了コードについては、IPMI の仕様を参照)
	2	00
	3	00
	4	6F または 2A (Sun レガシー)
	5	遅延 LSB
	6	遅延 MSB

## Set Boot Page

ShMM、ペイロード、またはデバッグインタフェースから次のコマンドを使用して、BIOS ブートページを設定します。ブートページのデフォルト値は 0 です。ビット 7-1 は 0 に設定してください。ユーザーが設定した値は SEEPROM に格納されます。次のブート時に、ブートページと同じ値が使用されます。

コマンド	NetFn	オペレーションコード
Set Boot Page	0x2E	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	1	00
	2	00
	3	6F または 2A (Sun レガシー)
応答データ	1	完了コード: [00] OK [C1] コマンドがサポートされていない [CC] 要求内に無効なデータがある [CB] パラメータが設定されていない場合。  (その他の完了コードについては、IPMI の仕様を参照)
	2	00

タイプ	バイト	データフィールド
	3	00
	4	6F または 2A (Sun レガシー)
	5	ブートページの値。0 = ページ 0、1 = ページ 1。

## Get Boot Page

ShMM、ペイロード、またはデバッグインタフェースから次のコマンドを使用して、BIOS ブートページを読み取ります。

コマンド	NetFn	オペレーションコード
Get Boot Page	0x2E	0x81

### 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	1	00
	2	00
	3	6F または 2A (Sun レガシー)
	4	ブートページ: 0 または 1
応答データ	1	完了コード: [00] OK [C1] コマンドがサポートされていない [CC] 要求内に無効なデータがある (その他の完了コードについては、IPMI の仕様を参照)
	2	00
	3	00
	4	6F または 2A (Sun レガシー)

## Set Front Panel Reset Button State

次のコマンドを使用して、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します。CPLD の電源投入時のデフォルトは 10 です。

コマンド	NetFn	オペレーションコード
Set Front Panel Reset Button State	0x2E	0x83

### 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	1	00
	2	00
	3	6F または 2A (Sun レガシー)
	4	フロントパネルの設定 <ul style="list-style-type: none"><li>• ビット 0 および 1 = フロントパネルのリセット状態:<ul style="list-style-type: none"><li>-00 = IPMC のリセットとシステムのハードリセット</li><li>-01 = システムに対する NMI</li><li>-10 = システムのハードリセット</li><li>-11 = フロントパネルのリセットボタンの無効化</li></ul></li><li>• ビット 2 - 7 = 0</li></ul>
応答データ	1	完了コード: <ul style="list-style-type: none"><li>[00] OK</li><li>[C1] コマンドがサポートされていない</li><li>[CC] 要求内に無効なデータがある</li></ul> (その他の完了コードについては、IPMI の仕様を参照)
	2	00
	3	00
	4	6F または 2A (Sun レガシー)

## Get Front Panel Reset Button State

次のコマンドを使用して、フロントパネルのリセットボタンの処理の設定を読み取ります。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
Get Front Panel Reset Button State	0x2E	0x84

### 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	1	00
	2	00
	3	6F または 2A (Sun レガシー)
応答データ	1	完了コード: [00] OK [C1] コマンドがサポートされていない [CC] 要求内に無効なデータがある (その他の完了コードについては、IPMI の仕様を参照)
	2	00
	3	00
	4	6F または 2A (Sun レガシー)
	5	フロントパネルの設定 <ul style="list-style-type: none"><li>ビット 0 および 1 = フロントパネルのリセット状態:<ul style="list-style-type: none"><li>-00 = IPMC のリセットとシステムのハードリセット</li><li>-01 = CPU に対する XIR</li><li>-10 = システムのハードリセット (CPU に対する POR)</li><li>-11 = フロントパネルのリセットボタンの無効化</li></ul></li><li>ビット 2 - 7 = 0</li></ul>



## Set IPMC Control Bits

次のコマンドを使用して、ブレードサーバーの LED と AMC のシャットダウン動作の構成を設定します。

コマンド	NetFn	オペレーションコード
Set IPMC Control Bits	0xE9	0xE9

### 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	1	00
	2	00
	3	6F または 2A (Sun レガシー)
	4	制御バイト。 <ul style="list-style-type: none"><li>ビット 0 = LED 2 (緑色の LED) の制御ビット:<ul style="list-style-type: none"><li>-1 = IPMC は緑色の LED を制御します。</li><li>-0 = IPMC は緑色の LED を制御しません。</li></ul></li><li>ビット 1 = LED 1 (オレンジ色/赤色の LED) の制御ビット。<ul style="list-style-type: none"><li>-1 = IPMC は LED1 のデフォルトの動作を制御します。</li><li>-0 = IPMC は LED1 を制御しません。</li></ul></li><li>ビット 2 = AMC のラッチの制御ビット。<ul style="list-style-type: none"><li>-1 = IPMC は、ラッチが開いた直後に AMC のシャットダウンを開始します。</li><li>-0 = IPMC は、ラッチが開いても AMC のシャットダウンを開始しません。</li></ul></li><li>ビット 3-7 = 予約済み。1 が書き込まれます (「注」を参照)。</li></ul>
応答データ	1	完了コード: [00] OK [C1] コマンドがサポートされていない [CC] 要求内に無効なデータがある (その他の完了コードについては、IPMI の仕様を参照)
	2	00
	3	00
	4	6F または 2A (Sun レガシー)

---

注 - 予約済みのいずれかのビット (3 - 7) に 0 が書き込まれようとする、IPMC は完了コード 0xCC でコマンドを拒否します。

---

## Get IPMC Control Bits

次のコマンドを使用して、ブレードサーバーの OOS LED 動作の現在の構成を表示します。

コマンド	NetFn	オペレーションコード
Get IPMC Control Bits	0x2E	0xE8

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	1	00
	2	00
	3	6F または 2A (Sun レガシー)
応答データ	1	完了コード: [00] OK [C1] コマンドがサポートされていない [CC] 要求内に無効なデータがある (その他の完了コードについては、IPMI の仕様を参照)
	2	00
	3	00
	4	6F または 2A (Sun レガシー)
	5	IPMC 制御ビット: <ul style="list-style-type: none"><li>• ビット 0: 緑色の LED の制御ビット。</li><li>• ビット 1: LED1 の制御ビット。</li><li>• ビット 2: AMC ラッチ制御ビット:</li><li>• ビット 3 - 7: 将来使用するために予約されています。</li></ul>

# Set Management Port

次のコマンドを使用して、フロントパネルまたは背面パネルへの管理ポートのアクセスを構成します。

コマンド	NetFn	オペレーションコード
Set Management Port	0x2E	0x9B

## 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	1	00
	2	00
	3	6F
	4	制御バイト • ビット 0 = ポートアクセスの設定: -0 = 背面パネルへのルート -1 = フロントパネルへのルート (デフォルト) • ビット 1-7 = 予約済み。0 が書き込まれます。
応答データ	1	完了コード: [00] OK [C1] コマンドがサポートされていない [CC] 要求内に無効なデータがある (その他の完了コードについては、IPMI の仕様を参照)
	2	00
	3	00
	4	6F

# Get Management Port

次のコマンドを使用して、管理ポート構成の現在の設定 (背面パネルまたはフロントパネルへのアクセス) を表示します。

コマンド	NetFn	オペレーションコード
Get Management Port	0x2E	0x9C

## 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	1	00
	2	00
	3	6F
応答データ	1	完了コード: [00] OK [C1] コマンドがサポートされていない [CC] 要求内に無効なデータがある (その他の完了コードについては、IPMI の仕様を参照)
	2	00
	3	00
	4	6F
	5	IPMC 制御ビット: <ul style="list-style-type: none"><li>ビット 0: 0 = 背面パネルにルーティングされたポート。 1 = フロントパネルにルーティングされたポート (デフォルト)。</li><li>ビット 1 - 7: 将来使用するために予約されています。</li></ul>

## Set Shelf FRU Record Data

この Pigeon Point 拡張コマンドを使用して、シェルフ FRU 情報内のマルチレコードに指定範囲のバイトを書き込むことができます。このコマンドは、アドレステーブルレコード (PICMG レコード ID = 10h)、シェルフ配電レコード (PICMG レコード ID = 11h) など、シェルフ FRU 情報内の PICMG 定義または OEM 定義のすべてのレコードに適用することができます。

書き込まれるレコードのタイプは、製造元の IANA と製造元固有のレコードタイプによって指定されます。0 ベースのレコード番号は、同じタイプのレコードの異なるインスタンスを区別するために使用することができます (シェルフ FRU 情報内に、同じタイプのレコードが複数存在する場合があります)。

コマンドを実装すると、チェックサムが正しい状態を維持できるように、対象となるレコードのチェックサムが自動的に更新されます。

このコマンドによって書き込まれるバイト数は、IPMB パケットのサイズによって制限されるため、15 バイト以下にする必要があります。この数はパラメータ「Byte Count」で指定し、コマンドの実際のデータバイト数と同じにする必要があります。コマンドの実際のデータバイト数が「Byte Count」パラメータの値と一致しない場合は、コマンドが拒否されます。

コマンド	NetFn	オペレーションコード	IANA:
Set Shelf FRU Record Data	2Eh	05h	0400Ah (PPS に割り当てられる)

## 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	1	PPS IANA の下位バイト。
	2	PPS IANA の中位バイト。
	3	PPS IANA の上位バイト。
	4	レコードの製造元 IANA の下位バイト。
	5	レコードの製造元 IANA の中位バイト。
	6	レコードの製造元 IANA の上位バイト。
	7	レコードタイプ。
	8	レコード番号。このフィールドは、アクセスされるレコードの数を指定します。レコード番号は 0 から始まります。
	9	オフセット。このフィールドは、レコードの先頭からのオフセットをバイト数で指定します。
	10	バイト数。このフィールドは、書き込まれるバイト数を指定します。
	11:N+10	
応答データ	1	完了コード: [00] OK [C1] コマンドがサポートされていない [CC] 要求内に無効なデータがある (その他の完了コードについては、IPMI の仕様を参照)
	2	PPS IANA の下位バイト。
	3	PPS IANA の中位バイト。
	4	PPS IANA の上位バイト。
	5	書き込まれる数。レコードに書き込まれるバイト数を示します。
	6:N+5	データ。この可変長フィールドには、レコードから取得したデータが格納されます。N は読み取りバイト数で指定されます。

たとえば、シェルフ FRU 情報内の最初のアドレステーブルレコードの最初の 9 バイトを設定するには、次のパラメータを使用します。

```
Record Manufacturer IANA Low Byte = 5Ah (PICMG)
Record Manufacturer IANA Middle Byte = 31h (PICMG)
Record Manufacturer IANA High Byte = 00h (PICMG)
Record type = 10h (Address Table)
Record Number = 00h (first record)
Offset = 00h
Byte Count = 09
Data = 41h 08h 00h 12h 09h 00 43h 0Ah 00(9 bytes)
```

この機能は、次の変数のために構成したカスタム値を保持します。

**表 B-2** 構成を保持できる IPMC 変数

変数	説明
log_level	デバッグメッセージのログレベルを制御します。
Payload timeout	ペイロード応答のタイムアウトを制御します。IPMC では、ペイロード警告の送信後、OS がこの時間内に応答することが見込まれます。正常な再起動機能と正常な停止機能で使用されます。
Payload shutdown timeout	正常な再起動/停止の仕様に示されているように、ペイロード停止のタイムアウトを制御します。これは、IPMC による OS の停止またはペイロードのリセットが行われる前に、OS がアプリケーション停止に費やす時間を示します。
Verbosity	メッセージのデバッグモードの詳細を制御します。特定のインタフェースとの間で送受信されるメッセージを選択し、有効または無効にすることができます。デバッグにのみ使用。
AMC power up timeout	ペイロードが M4 状態に移行する前に、AMC/ARTM が起動に費やす時間を制御します。
IPMC Control	IPMC の制御ビットを反映します。現在は、ビット 0 とビット 1 の 2 つのビットが、ボード上の LED2 と LED1 を制御しています。
Set SOL Fail Over Link Change Timeouts	一次リンクに障害が発生したときに IPMC が二次 Serial-over-LAN (SOL) リンクに切り替わるまで待機する時間、および、一次チャネルリンクが復旧した場合に一次チャネルに戻るまで待機する時間を制御します。
Get SOL Fail Over Link Change Timeouts	一次リンクに障害が発生したときに、IPMC が二次 Serial-over-LAN (SOL) リンクに切り替わるまで待機する時間を取得します。

これらの変数をデフォルトにリセットする場合は、「IPMC 構成変数のリセット」コマンドを使用します。

コマンド	NetFn	オペレーションコード
Reset IPMC Configuration Variables	0x2E	0x9A

#### 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	1	00
	2	00
	3	6F または 2A (Sun レガシー)
	4	設定されるパラメータ。 00 = すべてのパラメータをデフォルト値にリセットします。注: カスタム構成パラメータはすべて消去されます。
応答データ	1	完了コード: [00] OK [C1] コマンドがサポートされていない [CC] 要求内に無効なデータがある [C8] 要求データがフィールドの長さ制限を超えている (その他の完了コードについては、IPMI の仕様を参照)
	2	00
	3	00
	4	6F または 2A (Sun レガシー)



## Get Shelf FRU Record Data

この Pigeon Point 拡張コマンドを使用して、シェルフ FRU 情報内のマルチレコードから指定範囲のバイトを取得することができます。このコマンドは、アドレステーブルレコード (PICMG レコード ID = 10h)、シェルフ配電レコード (PICMG レコード ID = 11h) など、シェルフ FRU 情報の PICMG 定義または OEM 定義のすべてのレコードに適用することができます。

取得されるレコードのタイプは、製造元の IANA と製造元固有のレコードタイプによって指定されます。0 ベースのレコード番号は、同じタイプのレコードの異なるインスタンスを区別するために使用することができます (シェルフ FRU 情報内に、同じタイプのレコードが複数存在する場合があります)。

コマンド	NetFn	オペレーションコード
Get Shelf FRU Record Data	2Eh	1h

### 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	1	PPS IANA の下位バイト。値 0Ah が使用されます。
	2	PPS IANA の中位バイト。値 40h が使用されます。
	3	PPS IANA の上位バイト。値 00h が使用されます。
	4	レコードの製造元 IANA の下位バイト。
	5	レコードの製造元 IANA の中位バイト。
	6	レコードの製造元 IANA の上位バイト。
	7	レコードタイプ。
	8	レコード番号。このフィールドは、アクセスされるレコードの数を指定します。レコード番号は 0 から始まります。
	9	オフセット。このフィールドは、レコードの先頭からのオフセットをバイト数で指定します。
	10	バイト数。このフィールドは、読み取られるバイト数を指定します。

タイプ	バイト	データフィールド
応答データ	1	完了コード: [00] OK [C1] コマンドがサポートされていない [CC] 要求内に無効なデータがある (その他の完了コードについては、IPMI の仕様を参照)
	2	PPS IANA の下位バイト。値 0Ah が使用されます。
	3	PPS IANA の中位バイト。値 40h が使用されます。
	4	PPS IANA の上位バイト。値 00h が使用されます。
	5	読み取られる数。データフィールドのバイト数を示します。
	6:N+5	データ。この可変長フィールドには、レコードから取得したデータが格納されます。N は読み取りバイト数で指定されます。

たとえば、シェルフ FRU 情報内の最初のアドレステーブルレコードの最初の 10 バイトを取得するには、次のパラメータを使用します。

```
Record Manufacturer IANA Low Byte = 5Ah (PICMG)
Record Manufacturer IANA Middle Byte = 31h (PICMG)
Record Manufacturer IANA High Byte = 00h (PICMG)
Record type = 10h (Address Table)
Record Number = 00h (first record)
Offset = 00h
Byte Count = Ah (10 bytes)
```

## Set SOL Fail Over Link Change Timeouts

次のコマンドを使用して、一次リンクに障害が発生したときに IPMC が二次 Serial-over-LAN (SOL) リンクに切り替わるまで待機する時間、および、一次チャンネルリンクが復旧した場合に一次チャンネルに戻るまで待機する時間を設定します。待機時間を設定すると、リンクのアップとダウンの切り替わりを削減できます。

---

注 – ユーザーがカスタマイズした設定よりも、既定の設定を使用することをお勧めします。

---

待機時間は秒単位で指定します。たとえば、バイト 4 の数値を 10 (0xA) に設定すると、リンクを二次チャンネルに切り替えるまでに IPMC は 10 秒待機します。さらに、数値 15 (0xF) を設定すると、回復した一次チャンネルに戻るまでに IPMC は 15 秒待機します。

---

コマンド	NetFn	オペレーションコード
Set SOL Fail Over Link Change Timeouts	0x2E	0xE7

---

### 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	1	00
	2	00
	3	6F または 2A (Sun レガシー)
	4	一次リンクがダウンしてから、フェイルオーバーするまでの待機時間。
	5	一次リンクがアップしてから、一次リンクに切り替わるまでの待機時間。
応答データ	1	完了コード: [00] OK [C1] コマンドがサポートされていない [CC] 要求内に無効なデータがある (その他の完了コードについては、IPMI の仕様を参照)
	2	00
	3	00
	4	6F または 2A (Sun レガシー)

---

## Get SOL Fail Over Link Change Timeouts

次のコマンドを使用して、一次リンクに障害が発生したときに IPMC が二次 Serial-over-LAN (SOL) リンクに切り替わるまで待機する時間、および、一次チャンネルリンクが復旧した場合に一次チャンネルに戻るまで待機する時間を読み取ります。

コマンド	NetFn	オペレーションコード
Get SOL Fail Over Link Change Timeouts	0x2E	0xE6

### 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	1	00
	2	00
	3	6F または 2A (Sun レガシー)
応答データ	1	完了コード: [00] OK [C1] コマンドがサポートされていない [CC] 要求内に無効なデータがある (その他の完了コードについては、IPMI の仕様を参照)
	2	00
	3	00
	4	6F または 2A (Sun レガシー)
	5	一次リンクがダウンしてから、フェイルオーバーするまでの待機時間。
	6	一次リンクがアップしてから、一次リンクに切り替わるまでの待機時間。

## Get Version

次のコマンドを使用して、IPMC ファームウェアとスタンバイ CPLD のバージョンを読み取ります。このコマンドは IPMC ファームウェアのバージョンと CPLD のバージョンを返しますが、このコマンドを使用する主な理由は、IPMC のバージョンに対応する CPLD のバージョンを表示することです。IPMC バージョンを取得するには、IPMI の「デバイス ID の取得」コマンドを使用します。

コマンド	NetFn	オペレーションコード
Get Version	0x2E	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	1	00
	2	00
	3	6F または 2A (Sun レガシー)
応答データ	1	完了コード: [00] OK [C1] コマンドがサポートされていない [CC] 要求内に無効なデータがある (その他の完了コードについては、IPMI の仕様を参照)
	2	00
	3	00
	4	6F または 2A (Sun レガシー)
	5	CPLD バージョン
	6	IPMC ファームウェアの REV1 バイト
	7	IPMC ファームウェアの REV2 バイト
	8	ビット 7 - ビット 1: 予約済み ビット 8 - ビット 1: 予約済み 1 => テストリリース。 0 => 通常リリース。
	9	将来用に予約済み (無視される)。
	A	将来用に予約済み (無視される)。

注 - IPMC のバージョンは、REV1 の下位ニブル、REV2 の上位ニブル、および REV2 の下位ニブルとして読み取られます。

## Set Thermal Trip

次のコマンドを使用して、最大温度に達した場合にボードを停止するタイミングを決定するサーマルトリップを有効または無効にします。



**注意** – 温度のしきい値に達しても停止しない場合、ボードとシステムが損傷する可能性があります。デフォルトの上書きが正当化される運用状況でないかぎり、デフォルト値を使用してください。

WAR ゾーンでの動作のような極限の状況においては、ユーザーが最大温度のしきい値を上書きしなければならない場合があります。「WAR ゾーンモード」と呼ばれるこの状況では、ユーザーがサーマルトリップを上書きすることで、最大温度のしきい値に達してもボードとシステムの動作を維持することができます。

コマンド	NetFn	オペレーションコード
Set Thermal Trip	0x2E	E5

### 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	1	00
	2	00
	3	6F または 2A (Sun レガシー)
	4	ビット 7- ビット 1: 予約済み。0 に設定されます。 ビット 0: サーマルトリップビットの値: 1 => サーマルトリップを有効にします (デフォルト)。 0 => サーマルトリップを無効にします。
応答データ	1	完了コード: [00] OK [C1] コマンドがサポートされていない [CC] 要求内に無効なデータがある (その他の完了コードについては、IPMI の仕様を参照)
	2	00
	3	00
	4	6F または 2A (Sun レガシー)

# Get Thermal Trip

次のコマンドを使用して、現在のサーマルトリップの値を読み取ります。

コマンド	NetFn	オペレーションコード
Get Thermal Trip	0x2E	0xE4

## 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	1	00
	2	00
	3	6F または 2A (Sun レガシー)
応答データ	1	完了コード: [00] OK [C1] コマンドがサポートされていない [CC] 要求内に無効なデータがある (その他の完了コードについては、IPMI の仕様を参照)
	2	00
	3	00
	4	6F または 2A (Sun レガシー)
	5	サーマルトリップの値: 1 => サーマルトリップが有効。最大温度に達すると ボードが停止します。 0 => サーマルトリップが無効。最大温度に達しても ボードは停止しません。

## Subscribe for Event Notifications

RMCP クライアントでこの Pigeon Point 拡張コマンドを使用して、現在の RMCP セッション上でイベント通知の購読と購読解除を行うことができます。このコマンドは、シェルフマネージャの論理アドレス (20h) を対象にする必要があります。このコマンドは、パラメータフラグの値に従って、現在のセッション上でイベント通知を購読または購読解除します。セッションで通知が購読されると、SEL に新しいエントリが配置されるたびに、「SEL エントリの追加」要求の形式の通知がそのセッションを介して RMCP クライアントに送信されます。クライアントは、標準の IPMI 規則に従って、「SEL エントリの追加」応答を送信することで通知を確認する必要があります。対応するセッションを閉じると、購読が自動的に解除されます。

コマンド	NetFn	オペレーションコード
Subscribe for Event Notifications	2Eh	4h

### 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	1	PPS IANA の下位バイト。値 0Ah が使用されます。
	2	PPS IANA の中位バイト。値 40h が使用されます。
	3	PPS IANA の上位バイト。値 00h が使用されます。
	4	フラグ。アクションを指定するビットフィールド。 [0]: 1b = 現在のセッションでイベント通知を購読します。 0b = 購読を解除します。 [7:1] 予約済み。0 に設定されます。
応答データ	1	完了コード: [00] OK [C1] コマンドがサポートされていない [CC] 要求内に無効なデータがある (その他の完了コードについては、IPMI の仕様を参照)
	2	SPPS IANA の下位バイト。値 0Ah が使用されます。
	3	PPS IANA の中位バイト。値 40h が使用されます。
	4	PPS IANA の上位バイト。値 00h が使用されます。



# 用語集

---

以下に、Oracle の Sun Netra CT900 サーバーの管理に役立つ用語および頭字語について説明します。

---

## A

**ATCA** Advanced Telecom Computing Architecture の略語で、AdvancedTCA とも呼ばれます。次世代のキャリアグレード通信機器のための一連の業界標準仕様です。AdvancedTCA には、高速インターコネクト技術の最新動向、次世代プロセッサ、および強化された信頼性、管理性と保守性が統合されており、標準化により、もっとも低コストで通信に最適な新しいブレード (ボード) およびシャーシ (シェルフ) フォームファクタを実現します。

---

## B

**backup (バックアップ)** 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。

**Base channel (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。

**Base switch (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。Sun Netra CT900 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、単に「スイッチ」とも呼ばれます。

Base interface  
(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。Sun Netra CT900 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。

---

## D

data transport interface  
(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ッ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。

Dual Star topology  
(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、インターコネクットファブリックポロジ。1 組のスイッチが、ノードボード間の冗長インターコネクットを提供します。

---

## E

Electronic Keying  
(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。

ETSI

ヨーロッパ電気通信標準化協会 (European Telecommunications Standards Institute)。

---

## F

Fabric channel  
(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。そのため、各コネクタが、ボード間接続に使用可能な最大 5 つのチャンネルをサポートします。1 つのチャンネルは、4 組の、2 つのポートの組で構成されるとみなすこともできます。

Fabric interface (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 の他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

FRU (現場交換可能 ユニット)	ボード実装は、PICMG 3.x 補足仕様に定義されています。 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。一方、空のサーバーは FRU です。
frame (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。
front board (フロントボード)	PCB およびパネルを含む、PICMG 3.0 機械仕様 (8U x 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。ゾーン 3 のミッドプレーンコネクタまたは直接背面切り替えモジュールコネクタに接続することもできます。フロントボードはシェルフの前面部分に取り付けられます。
Full channel (フルチャネル)	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ネル接続。
Full Mesh topology (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可能な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専用チャネルを 1 つ提供します。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。

---

## H

hot-swap (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。
--------------------	--

---

## 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。
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 IPMB-0 リンクを持つ ShMM 内に存在します。

**IPMB-0 リンク** 放射状トポロジで、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。

**IPMC (IPM コントローラ)** 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。

**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。

---

## L

**logic ground (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。

---

## M

**Mesh Enabled board (メッシュ対応ボード)** ミッドプレーン内の他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のサポートも可能です。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ンネル (通常は 15 チャンネルすべて) を使用して、シェルフ内の他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ンネルの数によって、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、もっとも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、ベーススイッチにすることができます。この場合、このボードはベースチャンネル 1 および 2 をサポートでき、論理スロット 3 - 16 に取り付けすることができます。ベースインタフェースをサポートするボードは、ベースチャンネル 1 および 2 を使用して、10/100/1000BASE-T Ethernet のみをサポートします。

**midplane (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。

---

## 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、それぞれ下のレベルを含みます。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。

- node board (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、スタートボロジの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 を使用して、10/100/1000BASE-T Ethernet のみをサポートします。
- node slot (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、スタートボ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 への接続を確立します。

---

## 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。20 - 33 MHz で稼動し、一度に、124 ピンコネクタを介して 32 ビット、または 188 ピンコネクタを介して 64 ビットを搬送します。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。

**physical address (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番号で構成されます。

**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。

---

## R

**rear-access (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、Sun Netra CT900 サーバーの構成オプション。

**rear transition module (背面切り替えモジュール)** Sun Netra CT900 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。

**RAS: Reliability, Availability, Serviceability (信頼性、可用性、保守性)** サーバーの信頼性、可用性、および保守性を実現または向上させるハードウェアおよびソフトウェア機能。

---

## S

**shelf (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。

shelf address (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。
shelf ground (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。
shelf manager (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) の管理を担当するシステムの構成要素。またシェルフマネージャー (ShMM) は、システムマネージャーインタフェースと IPMB-0 との間でのメッセージのルーティング、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への応答も実行します。Oracle では管理システム全体を ShMM と呼んでいますが、ほかの ATCA ベンダーでは ShMC と呼ばれる場合がある点に注意してください。
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可能な IPMC。
SNMP	Simple Network Management Protocol の略語。
star topology (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。
switch (スイッチ)	スタートポロジのミッドプレーンでの使用を目的としたボードで、ミッドプレーン内の多数のノードボードへの接続を提供する。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ンネルに対してスイッチ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ンネルを使用して、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 のスイッチリソースを提供します。1 つのベースチャンネルが、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。
switch slot (スイッチスロット)	スタートポロジの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 に位置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 のベースチャンネルと最大 15 のファブリックチャンネルをそれぞれサポートします。
system (システム)	ノードおよびスイッチ、シェルフ、フレームなどのコンポーネントを 1 つ以上含むことができる管理対象。

---

## U

U 測定単位は 1.75 インチです (44.45 mm)。

update channel interface  
(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。2つのボード間の10組の差動信号ペアから構成される接続を提供するゾーン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化に使用できます。ボード上のアップデートチャンネル用に実装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、単一ベンダーによって製造された、同等機能を持つ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。

---

## Z

- Zone 1 (ゾーン 1) 電力、管理、およびその他の補助機能用に割り当てられた ATCA スロットの高さ方向に沿った線形空間。
- Zone 2 (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た ATCA スロットの高さ方向に沿った線形空間。
- Zone 3 (ゾーン 3) 背面アクセスシステムの背面切り替えモジュールへのユーザー定義の接続用および相互接続用に予約された ATCA スロットの高さ方向に沿った線形空間。



# 索引

---

## A

AMC (Advanced Mezzanine Card), 6  
AMC キャリア, 6  
ATCA (Advanced Telecommunications Computing Architecture), 1  
ATCA シェルフ, 6

## C

Command-Line Interface (コマンド行インタフェース), 2  
commands  
  setenv, 22  
CPLD  
  cpldtool ユーティリティ, 123

## E

Ethernet ポート, 4, 25  
E-キーイング, 164

## F

FRU (現場交換可能ユニット), 3  
FRU 情報, 104, 106, 108, 109, 110  
  IPMI, 104  
  Sun, 106

## I

IPMB  
  スロット番号, 159  
  論理スロット番号, 159  
IPMB アドレス, 13, 159, 162  
  範囲, 159

IPMC, 2

## IPMI

  FRU 情報の配置, 105  
  FRU 情報, 104  
  LAN インタフェース, 12, 80  
  OEM 拡張コマンド, 87  
  概要, 6

IPMI コマンド, 80

IP アドレス, 26  
  RMCP, 28

## M

MMC (モジュール管理コントローラ), 6

## N

netconsole.conf ファイル, 148

## O

OpenHPI, 70  
  /etc/openhpi.conf, 70  
  libipigeonpoint パラメータ, 71  
  設定, 70

## P

POST, 2

## R

RADIUS  
  変数, 77  
Remote Authentication Dial-In User Service (RADIUS), 76

RMCP, 11, 12, 26  
アドレス, 28  
RMCP (Remote Management Control Protocol), 80  
rupgrade\_tool, 128, 129, 130, 132

## 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, 70  
/etc/snmpd.conf ファイルの更新, 75  
アクセス制御, 71  
トラップの設定, 75  
SNMPv3 の構成, 74  
SNMP サブエージェント, 71  
SNMP サブエージェント構成ファイル, 70  
Sun の FRU 情報, 106  
switchover, 10

## T

Telco アラーム, 8  
telnet, 16  
tip プログラム, 16

## U

U-Boot, 2, 17, 26  
アクセス, 18  
デフォルトに戻す, 120  
変数, 18, 22, 26  
変数の変更, 24

## USB

インタフェース, 10  
接続, 28

## W

WDT (ウォッチドッグタイマー), 133, 131, 132, 133

## か

拡張インタフェース, 4  
環境 FRU, 105  
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, 64

## こ

コネクタ  
シェルフアラームパネル, 16

## コマンド

clia shmstatus, 17, 147  
console, 146, 147  
getenv, 22, 30  
printenv, 21, 22  
setenv, 22, 26  
showcpustate, 96  
switchover, 10  
useradd, 67

## コンソール, 16, 148

エスケープ文字, 148  
セッション, 146, 170  
ユーザーラベルの表示, 148

## さ

ファイルシステム  
再初期化, 122  
再初期化, 11

## し

シェルフ管理カード, 2, 6, 8, 146  
Ethernet ポート, 25  
アクセス, 16  
アクティブ, 16  
コンソール, 16, 148  
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, 66  
リセット, 121  
ログイン, 16  
シェルフ情報, 114  
シェルフマネージャ  
機能, 8  
構成パラメータ, 36

シェルフマネージャー, 2, 6, 8  
CLI コマンド, 153, 153 ~ 336  
CLI コマンドの概要, 89  
CLI の開始, 88  
Command-Line Interface (コマンド行インタ  
フェース), 88  
IP アドレス, 16  
shelfman.conf ファイル, 35  
インタフェースオプション, 12  
概要, 6  
構成ファイル, 29  
コマンド行インタフェース (CLI), 2  
再プログラミング, 123  
詳細レベル, 63  
スイッチオーバー, 9  
ソフトウェア, 2  
デバッグレベル, 171  
ネットワークパラメータ, 30  
バックアップ, 11  
シェルフマネージャーの再プログラミング, 123  
システムマネージャー, 7  
シリアルポート, 16  
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, 124  
シナリオ, 134  
状態ファイル, 127  
フラッシュのパーティション分割, 124  
ユーティリティ, 128  
例, 135  
信頼できるアップグレード処理, 123, 128  
信頼できるアップグレードユーティリティ, 134  
**す**  
スイッチオーバー, 10, 11  
強制的, 10  
協動的, 10  
スイッチカード, 146, 170  
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/step4hsh, 131

**せ**  
制御された非アクティブ化, 150, 151  
正常な停止, 149

**た**  
タイムサーバー, 65

**て**  
停止  
手動による正常な, 149  
ノードボード, 150

**と**  
ドキュメント, xxxvi

**ね**  
ネットコンソール, 146, 149  
「コマンド」の「コンソール」も参照, 146  
ネットワークインタフェース, 4

**の**  
ノードボード  
コンソール, 146  
Ethernet ポート, 4  
SAS ポート, 4  
アクセス, 4  
シリアルポート, 4  
説明, 3  
停止, 150  
「ノードボード、サードパーティー」も参照  
ノードボード、サードパーティー, 4  
説明, 3

**は**  
ハードウェアインタフェース, 4, 5  
背面切り替えカード, 3  
パスワード, 68

## ふ

- ファイルシステム
  - 再初期化, 122
- フラッシュパーティション, 125
- フラッシュメモリー, 123
- 物理スロットから論理スロットへのマッピング, 13

## へ

- ベースインタフェース, 4

## ま

- マッピング
  - IPMB アドレス, 13
  - ハードウェアアドレス, 13
  - 物理アドレス, 162

## 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, 128

## ろ

- ログイン
  - 出荷時のデフォルト設定, 122
  - パスワードのリセット, 122
- ログイン、シェルフ管理カード, 16