



CHAPTER 60

システム プロセスおよびログのモニタ

この章では、スイッチ状態のモニタリングについて詳細に説明します。内容は次のとおりです。

- 「システム プロセスの表示」 (P.60-1)
- 「システム ステータスの表示」 (P.60-4)
- 「コア ファイルおよびログ ファイル」 (P.60-6)
- 「カーネル コアのダンプ」 (P.60-10)
- 「オンラインでのシステム ヘルス管理」 (P.60-12)
- 「オンボード障害ロギング」 (P.60-23)
- 「デフォルト設定」 (P.60-26)

システム プロセスの表示

すべてのプロセスに関する一般的な情報を表示するには、**show processes** コマンドを使用します (例 60-1 ~ 例 60-6 を参照)。

例 60-1 システム プロセスの表示

```
switch# show processes
PID      State  PC          Start_cnt  TTY  Process
-----  -----  -----  -----  ---  -----
  868     S    2ae4f33e         1    -   snmpd
  869     S    2acee33e         1    -   rscn
  870     S    2ac36c24         1    -   qos
  871     S    2ac44c24         1    -   port-channel
  872     S    2ac7a33e         1    -   ntp
    -     ER      -             1    -   mdog
    -     NR      -             0    -   vbuilder
```

それぞれの説明は次のとおりです。

- PID = プロセス ID
- State = プロセスの状態
 - D = 中断なしで休止 (通常 I/O)
 - R = 実行可能 (実行キュー上)
 - S = 休止中
 - T = トレースまたは停止

- Z = 存在しない (ゾンビ) プロセス。
- NR = 実行されていない。
- ER = 実行されているべきだが、現在は実行されていない
- PC = 現在のプログラム カウンタ (16 進形式)
- Start_cnt = プロセスがこれまでに開始 (または再開) された回数。
- TTY = プロセスを制御している端末 通常、「-」(ハイフン) は、特定の TTY 上で実行されていないデーモンを表します。
- Process = プロセスの名前

例 60-2 CPU 使用率情報の表示

```
switch# show processes cpu
PID      Runtime(ms)   Invoked    uSecs   1Sec   Process
-----
   842          3807    137001     27     0.0   sysmgr
  1112          1220    67974     17     0.0   syslogd
  1269           220    13568     16     0.0   fcfwd
  1276          2901    15419     188    0.0   zone
  1277           738    21010     35     0.0   xbar_client
  1278          1159     6789     170    0.0   wwn
  1279           515    67617      7     0.0   vsan
```

それぞれの説明は次のとおりです。

- Runtime (ms) = プロセスが使用した CPU 時間 (ミリ秒単位)
- Invoked = プロセスがこれまでに開始された回数
- uSecs = プロセスの呼び出しごとの平均 CPU 時間 (ミリ秒単位)
- 1Sec = 最近の 1 秒間における CPU 使用率 (パーセント単位)

例 60-3 プロセス ログ情報の表示

```
switch# show processes log
Process      PID      Normal-exit  Stack-trace  Core      Log-create-time
-----
 fspf         1339          N             Y            N      Jan  5 04:25
 lcm          1559          N             Y            N      Jan  2 04:49
 rib          1741          N             Y            N      Jan  1 06:05
```

それぞれの説明は次のとおりです。

- Normal-exit = プロセスが正常に終了したかどうか。
- Stack-trace = ログにスタック トレースがあるかどうか。
- Core = コア ファイルが存在するかどうか。
- Log-create-time = ログ ファイルが生成された時間。

例 60-4 プロセスに関する詳細ログ情報の表示

```
switch# show processes log pid 1339
Service: fspf
Description: FSPF Routing Protocol Application

Started at Sat Jan  5 03:23:44 1980 (545631 us)
Stopped at Sat Jan  5 04:25:57 1980 (819598 us)
```

```

Uptime: 1 hours 2 minutes 2 seconds

Start type: SRV_OPTION_RESTART_STATELESS (23)
Death reason: SYSMGR_DEATH_REASON_FAILURE_SIGNAL (2)
Exit code: signal 9 (no core)
CWD: /var/sysmgr/work

Virtual Memory:

      CODE      08048000 - 0809A100
      DATA     0809B100 - 0809B65C
      BRK       0809D988 - 080CD000
      STACK     7FFFD20
      TOTAL     23764 KB

Register Set:

      EBX 00000005      ECX 7FFFF8CC      EDX 00000000
      ESI 00000000      EDI 7FFF6CC      EBP 7FFF95C
      EAX FFFFD20      XDS 8010002B     XES 0000002B
      EAX 0000008E (orig) EIP 2ACE133E     XCS 00000023
      EFL 00000207      ESP 7FFF654      XSS 0000002B

Stack: 1740 bytes.ESP 7FFF654, TOP 7FFFD20

0x7FFF654: 00000000 00000008 00000003 08051E95 .....
0x7FFF664: 00000005 7FFF8CC 00000000 00000000 .....
0x7FFF674: 7FFF6CC 00000001 7FFF95C 080522CD .....\"..
0x7FFF684: 7FFF9A4 00000008 7FFFC34 2AC1F18C .....4.....*

```

例 60-5 すべてのプロセス ログの詳細表示

```

switch# show processes log details
=====
Service: snmpd
Description: SNMP Agent

Started at Wed Jan  9 00:14:55 1980 (597263 us)
Stopped at Fri Jan 11 10:08:36 1980 (649860 us)
Uptime: 2 days 9 hours 53 minutes 53 seconds

Start type: SRV_OPTION_RESTART_STATEFUL (24)
Death reason: SYSMGR_DEATH_REASON_FAILURE_SIGNAL (2)
Exit code: signal 6 (core dumped)
CWD: /var/sysmgr/work

Virtual Memory:

      CODE      08048000 - 0804C4A0
      DATA     0804D4A0 - 0804D770
      BRK       0804DFC4 - 0818F000
      STACK     7FFFCE0
      TOTAL     26656 KB

...

```

例 60-6 プロセスに関するメモリ情報の表示

```

switch# show processes memory
PID      MemAlloc  StackBase/Ptr      Process
-----
1277     120632   7ffffcd0/7ffffefe4  xbar_client
1278     56800   7ffffce0/7ffffb5c   wwn

```

```

1279 1210220 7ffffce0/7ffffbac vsan
1293 386144 7ffffcf0/7ffffebd4 span
1294 1396892 7ffffce0/7ffffdff4 snmpd
1295 214528 7ffffcf0/7ffff904 rscn
1296 42064 7ffffce0/7ffffb5c qos

```

それぞれの説明は次のとおりです。

- MemAlloc = プロセスで割り当てられたメモリの総容量。
- StackBase/Ptr = プロセス スタック ベースと現在のスタック ポインタ (16 進形式)。

システム ステータスの表示

システム関連のステータス情報を表示するには、**show system** コマンドを使用します (例 60-7 ~ 例 60-10 を参照)。

例 60-7 デフォルトのスイッチ ポート状態の表示

```

switch# show system default switchport
System default port state is down
System default trunk mode is on

```

例 60-8 指定された ID のエラー情報の表示

```

switch# show system error-id 0x401D0019
Error Facility: module
Error Description: Failed to stop Linecard Async Notification.

```

例 60-9 システム リセット情報の表示

```

switch# Show system reset-reason module 5
----- reset reason for module 5 -----
1) At 224801 usecs after Fri Nov 21 16:36:40 2003
   Reason: Reset Requested by CLI command reload
   Service:
   Version: 1.3(1)
2) At 922828 usecs after Fri Nov 21 16:02:48 2003
   Reason: Reset Requested by CLI command reload
   Service:
   Version: 1.3(1)
3) At 318034 usecs after Fri Nov 21 14:03:36 2003
   Reason: Reset Requested by CLI command reload
   Service:
   Version: 1.3(1)
4) At 255842 usecs after Wed Nov 19 00:07:49 2003
   Reason: Reset Requested by CLI command reload
   Service:
   Version: 1.3(1)

```

show system reset-reason コマンドは、次の情報を表示します。

- Cisco MDS 9513 ディレクタでは、スロット 7 およびスロット 8 にあるスーパーバイザ モジュールの最後の 4 つのリセット理由コードが表示されます。いずれかのスーパーバイザ モジュールが不在の場合、そのスーパーバイザ モジュールのリセット理由コードは表示されません。

- Cisco MDS 9506 または Cisco MDS 9509 スイッチでは、スロット 5 およびスロット 6 にあるスーパーバイザ モジュールの最後の 4 つのリセット理由コードが表示されます。いずれかのスーパーバイザ モジュールが不在の場合、そのスーパーバイザ モジュールのリセット理由コードは表示されません。
- Cisco MDS 9200 シリーズ スイッチでは、スロット 1 にあるスーパーバイザ モジュールの最後の 4 つのリセット理由コードが表示されます。
- **show system reset-reason module number** コマンドは、指定されたスロットにある特定のモジュールの最後の 4 つのリセット理由コードを表示します。モジュールが不在の場合、そのモジュールのリセット理由コードは表示されません。

NVRAM および揮発性永続ストレージに保存されているリセット理由情報をクリアするには、**clear system reset-reason** コマンドを使用します。

- Cisco MDS 9500 シリーズ スイッチでは、このコマンドで、アクティブおよびスタンバイ スーパーバイザ モジュールの NVRAM および揮発性永続ストレージに保存されているリセット理由情報をクリアします。
- Cisco MDS 9200 シリーズ スイッチでは、このコマンドで、アクティブ スーパーバイザ モジュールの NVRAM および揮発性永続ストレージに保存されているリセット理由情報をクリアします。

例 60-10 システム稼働時間の表示

```
switch# show system uptime
Start Time: Sun Oct 13 18:09:23 2030
Up Time:    0 days, 9 hours, 46 minutes, 26 seconds
```

システム関連の CPU およびメモリの統計情報を表示するには、**show system resources** コマンドを使用します (例 60-11 を参照)。

例 60-11 システム関連の CPU およびメモリ情報の表示

```
switch# show system resources
Load average:  1 minute: 0.43   5 minutes: 0.17   15 minutes: 0.11
Processes   :  100 total, 2 running
CPU states  :  0.0% user,  0.0% kernel,  100.0% idle
Memory usage: 1027628K total,  313424K used,  714204K free
              3620K buffers,   22278K cache
```

それぞれの説明は次のとおりです。

- **Load average** : 実行中のプロセス数が表示されます。Load average には、過去 1 分間、5 分間、および 15 分間のシステム負荷が表示されます。
- **Processes** : システム内のプロセス数、およびコマンド発行時に実際に実行されていたプロセス数が表示されます。
- **CPU states** : 直前の 1 秒間における CPU のユーザ モードとカーネル モードでの使用率およびアイドル時間がパーセントで表示されます。
- **Memory usage** : 合計メモリ、使用中メモリ、空きメモリ、バッファに使用されているメモリ、およびキャッシュに使用されているメモリが KB 単位で表示されます。また、**buffers** および **cache** の値には、*使用中メモリ*の統計情報も含まれます。

コア ファイルおよびログ ファイル

この項は、次の内容で構成されています。

- 「コア ステータスの表示」 (P.60-6)
- 「コアの保存」 (P.60-7)
- 「CompactFlash への最後のコアの保存」 (P.60-8)
- 「コア ディレクトリのクリア」 (P.60-8)

コア ステータスの表示

現在設定されているコア コピー方式を表示するには、**show system cores** コマンドを使用します。例 60-12 ~ 60-14 を参照してください。

例 60-12 システム コア ステータスの表示

```
switch# show system cores
Transfer of cores is enabled
```

例 60-13 アクティブ スーパーバイザ モジュールからアップロードに使用可能なすべてのコアの表示

```
switch# show cores
Module-num  Process-name  PID      Core-create-time
-----
5           fspf             1524     Nov 9 03:11
6           fcc              919      Nov 9 03:09
8           acltcam          285      Nov 9 03:09
8           fib              283      Nov 9 03:08
```

Module-num は、コアが生成されたスロット番号を示します。この例で、fspf core がアクティブ スーパーバイザ モジュール (スロット 5) で生成され、fcc がスタンバイ スーパーバイザ モジュール (スロット 6) で生成され、acltcam および fib が、スイッチング モジュール (スロット 8) で生成されています。

例 60-14 ローカル システムのログの表示

```
switch# show processes log
Process      PID      Normal-exit  Stack  Core  Log-create-time
-----
ExceptionLog 2862     N            Y      N     Wed Aug 6 15:08:34 2003
acl          2299     N            Y      N     Tue Oct 28 02:50:01 2003
bios_daemon  2227     N            Y      N     Mon Sep 29 15:30:51 2003
capability   2373     N            Y      N     Tue Aug 19 13:30:02 2003
core-client  2262     N            Y      N     Mon Sep 29 15:30:51 2003
fcanalyzer   5623     N            Y      N     Fri Sep 26 20:45:09 2003
fcd          12996    N            Y      N     Fri Oct 17 20:35:01 2003
fcdomain     2410     N            Y      N     Thu Jun 12 09:30:58 2003
ficon        2708     N            Y      N     Wed Nov 12 18:34:02 2003
ficonstat    9640     N            Y      N     Tue Sep 30 22:55:03 2003
flogi        1300     N            Y      N     Fri Jun 20 08:52:33 2003
idehsd       2176     N            Y      N     Tue Jun 24 05:10:56 2003
lmgrd        2220     N            N      N     Mon Sep 29 15:30:51 2003
platform     2840     N            Y      N     Sat Oct 11 18:29:42 2003
port-security 3098     N            Y      N     Sun Sep 14 22:10:28 2003
```

port	11818	N	Y	N	Mon Nov 17 23:13:37 2003
rlir	3195	N	Y	N	Fri Jun 27 18:01:05 2003
rscn	2319	N	Y	N	Mon Sep 29 21:19:14 2003
securityd	2239	N	N	N	Thu Oct 16 18:51:39 2003
snmpd	2364	N	Y	N	Mon Nov 17 23:19:39 2003
span	2220	N	Y	N	Mon Sep 29 21:19:13 2003
syslogd	2076	N	Y	N	Sat Oct 11 18:29:40 2003
tcap	2864	N	Y	N	Wed Aug 6 15:09:04 2003
tftpd	2021	N	Y	N	Mon Sep 29 15:30:51 2003
vpm	2930	N	N	N	Mon Nov 17 19:14:33 2003

コアの保存

次の方法のいずれかで、(アクティブ スーパーバイザ モジュール、スタンバイ スーパーバイザ モジュール、または任意のスイッチング モジュールの) コアを外部 CompactFlash (スロット 0) または TFTP サーバに保存できます。

- オンデマンド：与えられたプロセス ID に基づいて 1 つのファイルをコピーします。
- 定期的：ユーザの設定に従ってコア ファイルを定期的にコピーします。

新しい方式が実行されると、その前に実行された方式は新しい方式で上書きされます。たとえば、別のコア ログ コピー タスクを実行すると、コアは、その新しい場所またはファイルに定期的に保存されません。



ヒント

この操作を実行する前に、必要なディレクトリを作成してください。この操作で指定されたディレクトリが存在しない場合、スイッチ ソフトウェアはコピー コアが試行されるたびにシステム メッセージを記録します。

コア ファイルおよびログ ファイルをオンデマンドでコピーするには、次の手順を実行します。

	コマンド	目的
ステップ 1	switch# show cores	
ステップ 2	switch# copy core:7407 slot0:coreSample	プロセス ID が 7407 のコア ファイルをスロット 0 に coreSample としてコピーします。
	switch# copy core://5/1524 tftp://1.1.1.1/abcd	スロット 5 ¹ またはスロット 7 ² で生成された、PID が 1524 のプロセスのコア (存在する場合) を IPv4 アドレスが 1.1.1.1 の TFTP サーバにコピーします。 (注) IPv6 アドレスを使用して TFTP サーバを識別することもできます。

1. Cisco MDS 9506 または Cisco MDS 9509 スイッチ
2. Cisco MDS 9513 ディレクタ

- 指定されたプロセス ID のコア ファイルが使用できない場合は、次の応答が表示されます。

```
switch# copy core:133 slot0:foo
No core file found with pid 133
```

- 同じプロセス ID のコア ファイルが 2 つ存在する場合は、1 つのファイルだけがコピーされます。

```
switch# copy core:7407 slot0:fool
2 core files found with pid 7407
Only "/isan/tmp/logs/calc_server_log.7407.tar.gz" will be copied to the destination.
```

コア ファイルおよびログ ファイルを定期的にコピーするには、次の手順を実行します。

	コマンド	目的
ステップ1	switch# show system cores	
ステップ2	switch# config t	コンフィギュレーション モードに入ります。
ステップ3	switch(config)# system cores slot0:coreSample	コア ファイル (coreSample) をスロット 0 にコピーします。
	switch(config)# system cores tftp://1.1.1.1/abcd	指定されたディレクトリ内のコア ファイル (abcd) を IPv4 アドレス 1.1.1.1 の TFTP サーバにコピーします。 (注) IPv6 アドレスを使用して TFTP サーバを識別することもできます。
	switch(config)# no system cores	コア ファイルのコピー機能をディセーブルにします。

CompactFlash への最後のコアの保存

この最後のコア ダンプは、スイッチオーバーまたはリブートが起こる前に、/mnt/pss/ パーティションにある CompactFlash に自動的に保存されます。スーパーバイザ モジュールがリブートしてから 3 分後に、保存された最後のコアがフラッシュ パーティション (/mnt/pss) から元のメモリ上に復元されます。この復元はバックグラウンドプロセスであり、ユーザからは見えません。



ヒント

復元された最後のコア ファイルのタイムスタンプは、最後のコアが実際にダンプされた時刻ではなく、スーパーバイザのブート時刻を表します。最後のコア ダンプの正確な時刻を知るには、PID が同じ、対応するログ ファイルを確認してください。

最後のコア情報を表示するには、EXEC モードで **show cores** コマンドを発行します。

実際の最後のコア ダンプの時間を表示するには、EXEC モードで **show process log** コマンドを発行します。

コア ディレクトリのクリア

コア ディレクトリをクリアするには、**clear cores** コマンドを使用します。ソフトウェアはサービスごと、およびスロットごとに直前のコアをいくつか保管し、アクティブ スーパーバイザ モジュール上のその他のすべてのコアをクリアします。

```
switch# clear cores
```

最初と最後のコア

最初と最後のコアの機能は、限られたシステム リソースで最も重要なコア ファイルを保持します。一般に、最初のコアと最後に生成されたコアにデバッグの情報が格納されています。最初と最後のコアの機能は、最初と最後のコア情報を保持しようとします。

コア ファイルがアクティブ スーパーバイザ モジュールから生成された場合、サービスのコア ファイル数は `service.conf` ファイルで定義されます。アクティブ スーパーバイザ モジュールのコア ファイルの総数に上限はありません。コア ファイルの定義された数はすべての VDC に適用されます。

システムに保存されたコア ファイルを表示するには、次のコマンドを使用します。

コマンド	目的
<code>switch# show cores</code>	デフォルト VDC に保存されたすべてのコア ファイルを表示します。
<code>switch# show cores vdc-all</code>	システムに保存されたすべてのコア ファイルを表示します。コア ファイルの数は、 <code>service.conf</code> ファイルで定義されます。

最初と最新のコアの確認

保存したコア ファイルに関する詳細な情報を表示できます。例 60-15 ~ 例 60-16 に、保存したコア ファイルの詳細を示します。

例 60-15 ローカル ノードのデフォルト VDC 上の通常サービス

たとえば、`pixm` が 5 回クラッシュします。`show cores vdc-all` の出力に 5 つのコア ファイルが表示されます。3 分後に、`service.conf` ファイルで定義されているコア数に合うように、2 番目に古いコア ファイルが削除されます。

```
switch# show cores vdc-all
```

VDC No	Module-num	Process-name	PID	Core-create-time
1	5	pixm	4103	Jan 29 01:30
1	5	pixm	5105	Jan 29 01:32
1	5	pixm	5106	Jan 29 01:32
1	5	pixm	5107	Jan 29 01:33
1	5	pixm	5108	Jan 29 01:40

```
switch# show cores vdc-all
```

VDC No	Module-num	Process-name	PID	Core-create-time
1	5	pixm	4103	Jan 29 01:30
1	5	pixm	5106	Jan 29 01:32
1	5	pixm	5107	Jan 29 01:33
1	5	pixm	5108	Jan 29 01:40

例 60-16 アクティブ スーパーバイザ モジュール上の vdc 2 上の通常のサービス

たとえば、アクティブ スーパーバイザ モジュール上の `vdc 2` には 5 つの `radius` コア ファイルがあります。`service.conf` ファイルに定義されたコア ファイル数に合うように、2 番めと 3 番めに古いファイルは削除されます。

```
switch# show cores vdc vdc2
```

VDC No	Module-num	Process-name	PID	Core-create-time
2	5	radius	6100	Jan 29 01:47
2	5	radius	6101	Jan 29 01:55

```

2      5      radius      6102      Jan 29 01:55
2      5      radius      6103      Jan 29 01:55
2      5      radius      6104      Jan 29 01:57

```

```
switch# show cores vdc vdc2
```

VDC No	Module-num	Process-name	PID	Core-create-time
2	5	radius	6100	Jan 29 01:47
2	5	radius	6103	Jan 29 01:55
2	5	radius	6104	Jan 29 01:57

カーネル コアのダンプ



注意

カーネル コアの変更は、管理者または、スイッチの操作に精通した担当者が実行してください。

特定のモジュールのオペレーティング システム (OS) がクラッシュした場合、メモリ イメージの完全なコピー (カーネル コア ダンプ) を取得すると、クラッシュの原因の確認に役立つことがあります。モジュールにカーネル コア ダンプが発生すると、スーパーバイザに設定されたプロキシ サーバがトリガーされます。スーパーバイザは、Cisco MDS 9000 システム デバッグ サーバにモジュールの OS カーネル コア ダンプを送信します。同様に、スーパーバイザ OS に障害が発生した場合、スーパーバイザは Cisco MDS 9000 システム デバッグ サーバに OS カーネル コア ダンプを送信します。



(注)

Cisco MDS 9000 システム デバッグ サーバは、Linux 上で動作するシスコ アプリケーションです。これにより、カーネル コア ダンプのリポジトリが作成されます。Cisco MDS 9000 システム デバッグ サーバは、Cisco.com Web サイト (<http://www.cisco.com/kobayashi/sw-center/sw-stornet.shtml>) からダウンロードできます。

カーネル コア ダンプは、テクニカルサポート担当者だけに役立ちます。大きなバイナリ ファイルであるカーネル コア ダンプ ファイルは、スイッチと同じ物理 LAN に存在する外部サーバに転送する必要があります。その後、コア ダンプは、ソース コードおよび詳細なメモリ マップにアクセスできる技術担当者によって解釈されます。



ヒント

コア ダンプは Cisco MDS 9000 システム デバッグ サーバ アプリケーションのディスク領域を消費します。すべてのレベルのコア ダンプ (**level all** オプション) が設定されている場合は、Cisco MDS 9000 システム デバッグ サーバ アプリケーションが稼働している Linux サーバに、ダンプを保存するためのディスク領域を 1 GB 以上確保する必要があります。生成を完了するのに十分な領域がプロセスない場合、モジュールは自動的にリセットされます。カーネル コアへのすべての変更は、実行コンフィギュレーションに保存されます。

この項では、次のトピックについて取り上げます。

- 「外部サーバの設定」 (P.60-11)
- 「モジュール パラメータの設定」 (P.60-11)
- 「カーネル コア情報の表示」 (P.60-11)

外部サーバの設定

IPv4 を使用して外部サーバを設定するには、次の手順を実行します。

	コマンド	目的
ステップ1	switch# config terminal switch(config)#	コンフィギュレーション モードに入ります。
ステップ2	switch(config)# kernel core target 10.50.5.5 succeeded	外部サーバの IPv4 アドレスを設定します。 (注) IPv6 アドレスは、カーネル コア ターゲットではサポートされていません。

モジュール パラメータの設定

モジュール パラメータを設定するには、次の手順を実行します。

	コマンド	目的
ステップ1	switch# config terminal switch(config)#	コンフィギュレーション モードに入ります。
ステップ2	switch(config)# kernel core module 5 succeeded	モジュール 5 のカーネル コアの生成を設定します。
	switch(config)# kernel core module 5 level header succeeded	モジュール 5 のカーネル コアの生成を設定し、生成をヘッダー レベル コアに制限します。
ステップ3	switch(config)# kernel core limit 2 succeeded	2 つのモジュールのカーネル コアの生成を設定します。デフォルトは 1 つのモジュールです。

カーネル コア情報の表示

show running-config コマンドを使用すると、カーネル コアへの変更をすべて表示できます。また、**show kernel cores** コマンドを使用して、特定の設定変更を確認することもできます (例 60-17 ~ 例 60-19 を参照)。

例 60-17 コア制限の表示

```
switch# show kernel core limit
2
```

例 60-18 外部サーバの表示

```
switch# show kernel core target
10.50.5.5
```

例 60-19 指定されたモジュールのコア設定の表示

```
switch# show kernel core module 5
module 5 core is enabled
    level is header
    dst_ip is 10.50.5.5
    src_port is 6671
```

```
dst_port is 6666
dump_dev_name is eth1
dst_mac_addr is 00:00:0C:07:AC:01
```

オンラインでのシステムヘルス管理

Online Health Management System (システムヘルス) は、ハードウェア障害検出および復旧機能です。OHMS は、Cisco MDS 9000 ファミリのすべてのスイッチのスイッチング モジュール、サービス モジュール、スーパーバイザ モジュールの全般的な状態を確認します。

この項では、次のトピックについて取り上げます。

- 「オンライン システムヘルス管理の概要」 (P.60-12)
- 「システムヘルスの初期化」 (P.60-13)
- 「ループバック テストの頻度の設定」 (P.60-14)
- 「ループバック テストのフレーム長の設定」 (P.60-14)
- 「ハードウェアの障害処理」 (P.60-15)
- 「テストの実行要件」 (P.60-15)
- 「指定モジュールのテスト」 (P.60-16)
- 「古いエラー通知のクリア」 (P.60-17)
- 「内部ループバック テストの実行」 (P.60-17)
- 「外部ループバック テストの実行」 (P.60-18)
- 「Serdes ループバックの実行」 (P.60-19)
- 「現在のステータスの説明」 (P.60-19)
- 「システムヘルスの表示」 (P.60-20)

オンライン システムヘルス管理の概要

Online Health Management System (OHMS) は、ハードウェア障害検出および回復機能です。すべての Cisco MDS スwitching モジュール、サービス モジュール、およびスーパーバイザ モジュールで動作し、任意の Cisco MDS 9000 ファミリのスイッチの一般的なヘルスを保証します。OHMS は、システムハードウェアを次のようにモニタリングします。

- アクティブ スーパーバイザ稼働する OHMS コンポーネントは、スイッチ内の他のモジュール上で稼働する他のすべての OHMS コンポーネントを制御します。
- スタンバイ スーパーバイザ モジュールで稼働するシステムヘルス アプリケーションは、スタンバイ スーパーバイザ モジュールが HA スタンバイ モードで使用できる場合、このモジュールだけをモニタします。「HA スイッチオーバーの特長」 (P.10-2) を参照してください。

OHMS アプリケーションはすべてのモジュールでデーモン プロセスを起動して、各モジュール上で複数のテストを実行し、モジュールの個々のコンポーネントをテストします。これらのテストは、事前に設定されたインターバルで実行され、すべての主要な障害ポイントを対象として、障害が発生している MDS スイッチのコンポーネントを隔離します。アクティブ スーパーバイザ上で稼働する OHMS は、スイッチ内の他のすべてのモジュール上で稼働する他のすべての OHMS コンポーネントを制御します。

障害を検出すると、システムヘルス アプリケーションは次のリカバリ アクションを試行します。

- 障害が発生しているコンポーネントを隔離するための追加のテストを実行します。
- 永続的ストレージから設定情報を取得し、コンポーネントの再設定を試みます。
- 復旧できない場合は **Call Home** 通知、システム メッセージ、および例外ログを送信し、障害モジュールまたは障害コンポーネント（インターフェイスなど）をシャットダウンしてテストを中止します。
- 障害を検出すると、ただちに **Call Home** メッセージ、システム メッセージ、および例外ログを送信します。
- 障害の発生しているモジュールまたはコンポーネント（インターフェイスなど）をシャットダウンします。
- 詳細なテストが実行されないように、障害が発生したポートを隔離します。
- その障害を適切なソフトウェア コンポーネントに報告します。
- スタンバイ スーパーバイザ モジュールに切り替えます（障害がアクティブ スーパーバイザ モジュールで検出され、Cisco MDS スイッチにスタンバイ スーパーバイザ モジュールが搭載されている場合）。スイッチオーバーが完了すると、新しいアクティブ スーパーバイザ モジュールはアクティブ スーパーバイザ テストを再開します。
- スイッチをリロードします（スイッチにスタンバイ スーパーバイザ モジュールが搭載されていない場合）。
- テストの実行統計情報を表示、テスト、および取得したり、スイッチのシステムヘルスマネジメント設定を変更したりするための CLI サポートを提供します。
- 問題領域に焦点を当てるためのテストを実行します。

各モジュールはそれぞれに対応するテストを実行するように設定されています。必要に応じて、各モジュールのデフォルト パラメータを変更できます。

システムヘルスの初期化

デフォルトでは、システムヘルスマネジメント機能は Cisco MDS 9000 ファミリの各スイッチでイネーブルです。

Cisco MDS 9000 ファミリの任意のスイッチでこの機能をディセーブルまたはイネーブルにするには、次の手順を実行します。

	コマンド	目的
ステップ 1	switch# config terminal switch(config)#	コンフィギュレーションモードに入ります。
ステップ 2	switch(config)# no system health System Health is disabled.	このスイッチでテストを実行できないようにシステムヘルスを設定します。
	switch(config)# system health System Health is enabled.	このスイッチでテストを実行できるようにシステムヘルスを設定します（デフォルト）。
ステップ 3	switch(config)# no system health interface fc8/1 System health for interface fc8/13 is disabled.	指定されたインターフェイスのテストを実行できないようにシステムヘルスを設定します。
	switch(config)# system health interface fc8/1 System health for interface fc8/13 is enabled.	システムヘルスをイネーブル（デフォルト）にして、指定されたインターフェイスをテストします。

ループバック テストの頻度の設定

ループバック テストは、モジュール内のデータ パスおよびスーパーバイザ内の制御パスにおいてハードウェア エラーを特定するように設計されています。事前に設定された頻度でループバック フレームが各モジュールに 1 つずつ送信されます。このフレームは、それぞれに設定されたインターフェイスを通過した後、スーパーバイザ モジュールに戻ります。

ループバック テストは 5 (デフォルト) ~ 255 秒の範囲の頻度で実行できます。ループバック 頻度の値を設定しなければ、デフォルトの頻度である 5 秒がスイッチ内のすべてのモジュールに対して使用されます。ループバック テストの頻度は、モジュールごとに変更できます。

スイッチのすべてのモジュールにループバック テストの頻度を設定するには、次の手順を実行します。

	コマンド	目的
ステップ1	switch# config terminal switch(config)#	コンフィギュレーション モードに入ります。
ステップ2	switch(config)# system health loopback frequency 50 The new frequency is set at 50 Seconds.	ループバック 頻度を 50 秒に設定します。デフォルトのループバック 頻度は 5 秒です。有効な範囲は 5 ~ 255 秒です。

ループバック テストのフレーム長の設定

ループバック テストは、モジュール内のデータ パスおよびスーパーバイザ内の制御パスにおいてハードウェア エラーを特定するように設計されています。事前に設定されたサイズでループバック フレームが各モジュールに 1 つずつ送信されます。このフレームは、それぞれに設定されたインターフェイスを通過した後、スーパーバイザ モジュールに戻ります。

ループバック テストは、0 ~ 128 バイトの範囲のフレーム サイズで実行できます。ループバック フレーム長の値を設定しなければ、スイッチ内のすべてのモジュールに対してランダムなフレーム長がスイッチによって生成されます (自動モード)。ループバック テストのフレーム長は、モジュールごとに変更できます。

スイッチのすべてのモジュールにループバック テストのフレーム長を設定するには、次の手順を実行します。

	コマンド	目的
ステップ1	switch# config terminal switch(config)#	コンフィギュレーション モードに入ります。
ステップ2	switch(config)# system health loopback frame-length 128	ループバック フレーム長を 128 バイトに設定します。有効範囲は 0 ~ 128 バイトです。
ステップ3	switch(config)# system health loopback frame-length auto	ループバック フレーム長を自動的にランダム長 (デフォルト) を生成するように設定します。

ループバック 頻度の設定を確認するには、**show system health loopback frame-length** コマンドを使用します。

```
switch# show system health loopback frame-length
Loopback frame length is set to auto-size between 0-128 bytes
```

ハードウェアの障害処理

failure-action コマンドは、テストの実行中にハードウェア障害が発見された場合に、Cisco SAN-OS ソフトウェアによる処理の実行を抑制します。

デフォルトでは、Cisco MDS 9000 ファミリのすべてのスイッチでこの機能はイネーブルになります。障害が発見されると処理が実行され、障害が発生したコンポーネントはそれ以降のテストから隔離されます。

障害処理は、個々のテスト レベル (モジュール単位)、モジュール レベル (すべてのテスト)、またはスイッチ全体で制御されます。

スイッチの障害処理を設定するには、次の手順を実行します。

	コマンド	目的
ステップ 1	switch# config terminal switch(config)#	コンフィギュレーション モードに入ります。
ステップ 2	switch(config)# system health failure-action System health global failure action is now enabled.	障害処理を実行できるようにスイッチを設定します (デフォルト)。
ステップ 3	switch(config)# no system health failure-action System health global failure action now disabled.	障害処理が実行されないようにスイッチの設定を取り消します。
ステップ 4	switch(config)# system health module 1 failure-action System health failure action for module 1 is now enabled.	モジュール 1 の障害処理を実行できるようにスイッチを設定します。
ステップ 5	switch(config)# no system health module 1 loopback failure-action System health failure action for module 1 loopback test is now disabled.	モジュール 1 のループバック テストによって発見された障害に対する障害処理を実行しないようにスイッチを設定します。

テストの実行要件

テストをイネーブルにしても、テストの実行が保障されるわけではありません。

特定のインターフェイスまたはモジュールのテストが実行されるのは、次のすべての項目に対してシステムヘルスをイネーブルにしている場合だけです。

- スイッチ全体
- 必要なモジュール
- 必要なインターフェイス



ヒント

上記のいずれかによってシステムヘルスがディセーブルになっている場合、テストは実行されません。システムヘルスでテストの実行がディセーブルになっている場合、テストステータスはディセーブル (Disabled) と表示されます。



ヒント

特定のモジュールまたはインターフェイスでテストの実行がイネーブルになっているが、システムヘルスがディセーブルであるためにテストが実行されない場合、テストはイネーブル (Enabled) と表示されます (実行中 (Running) にはなりません)。

指定モジュールのテスト

SAN-OS ソフトウェアのシステムヘルスマネジメント機能は、次の領域のテストを実行します。

- アクティブ スーパーバイザのファブリックへのインバンド接続。
- スタンバイ スーパーバイザのアービターの可用性。
- すべてのモジュール上でのブートフラッシュの接続性とアクセシビリティ。
- すべてのモジュール上での EOBC の接続性とアクセシビリティ。
- すべてのモジュール上の各インターフェイスのデータパスの完全性。
- 管理ポートの接続性。
- キャッシング サービス モジュール (CSM) のバッテリー (温度、使用年数、フル充電容量、(放)充電機能、およびバックアップ機能) およびキャッシュ ディスク (接続性、アクセシビリティ、およびロー ディスク I/O)。
- 外部接続性検証のためのユーザによるテスト。テスト中はポートがシャットダウンされます (ファイバ チャネル ポートのみ)。
- 内部接続性検証のためのユーザによるテスト (ファイバ チャネル ポートと iSCSI ポート)。

特定のモジュールに必要なテストを実行するには、次の手順を実行します。

	コマンド	目的
ステップ1	switch# config terminal switch(config)#	コンフィギュレーション モードに入ります。
(注) 次のステップは、任意の順序で実行できます。		
ステップ2	switch(config)# system health module 8 battery-charger battery-charger test is not configured to run on module 8.	スロット 8 に存在する CSM の両方のバッテリーで、バッテリー充電器テストをイネーブルにします。スイッチのスロット 8 に CSM がない場合、このメッセージが発行されます。
ステップ3	switch(config)# system health module 8 cache-disk cache-disk test is not configured to run on module 8.	スロット 8 に存在する CSM の両方のディスクで、キャッシュ ディスク テストをイネーブルにします。スイッチのスロット 8 に CSM がない場合、このメッセージが発行されます。
(注) それぞれのテストの各種オプションについては、次のステップで説明します。各コマンドは任意の順序で設定できます。説明のため、各種オプションを同じステップに記述しています。		
ステップ4	switch(config)# system health module 8 bootflash System health for module 8 Bootflash is already enabled.	スロット 8 のモジュールでブートフラッシュ テストをイネーブルにします。
	switch(config)# system health module 8 bootflash frequency 200 The new frequency is set at 200 Seconds.	モジュール 8 のブートフラッシュ テストの新しい頻度を 200 秒に設定します。
ステップ5	switch(config)# system health module 8 eobc System health for module 8 EOBC is now enabled.	スロット 8 のモジュールで EOBC テストをイネーブルにします。
ステップ6	switch(config)# system health module 8 loopback System health for module 8 EOBC is now enabled.	スロット 8 のモジュールでループバック テストをイネーブルにします。
ステップ7	switch(config)# system health module 5 management System health for module 8 EOBC is now enabled.	スロット 5 のモジュールで管理テストをイネーブルにします。

古いエラー通知のクリア

ファイバチャネル インターフェイス、iSCSI インターフェイス、モジュール全体、またはモジュール全体の特定の 1 つのテストについて、エラー履歴をクリアできます。履歴をクリアすると、障害が発生してテストから除外されていたコンポーネントはすべて再度テストされます。

障害発生時に OHMS が一定期間（たとえば、1 週間）の間処理を実行しないようにオプション `failure-action` オプションをイネーブルにしている、指定期間が経過した後でエラー受信を再開する準備が整った場合には、それぞれのテストのシステムヘルスエラーステータスをクリアする必要があります。



ヒント

管理ポートテストは、スタンバイスーパーバイザモジュール上で実行することはできません。

インターフェイスまたはモジュールレベルで EXEC レベルの `system health clear-errors` コマンドを使用すると、システムヘルスアプリケーションで記録された古いエラー状態はすべて消去されます。特定のモジュールに対して、`battery-charger`、`bootflash`、`cache-disk`、`eobc`、`inband`、`loopback`、および `mgmt` のテストオプションを個別に指定できます。

次の例では、指定されたファイバチャネルインターフェイスのエラー履歴がクリアされます。

```
switch# system health clear-errors interface fc 3/1
```

次の例では、指定されたモジュールのエラー履歴がクリアされます。

```
switch# system health clear-errors module 3
```

次の例では、指定されたモジュールの管理テストのエラー履歴がクリアされます。

```
switch# system health clear-errors module 1 mgmt
```

内部ループバックテストの実行

手動ループバックテストを実行すると、スイッチングモジュールまたはサービスモジュールのデータパスや、スーパーバイザモジュールの制御パスにおけるハードウェアエラーを特定できます。内部ループバックテストは、同じポートとの間で FC2 フレームを送受信します。ラウンドトリップ時間はミリ秒単位で指定します。このテストは、ファイバチャネルインターフェイス、IPS インターフェイス、iSCSI インターフェイスで使用できます。

モジュール全体のポート内でこのテストを（ユーザが要求したときに）オンデマンドで明示的に実行するには、EXEC レベルで `system health internal-loopback` コマンドを使用します。

```
switch# system health internal-loopback interface iscsi 8/1
Internal loopback test on interface iscsi8/1 was successful.
Sent 1 received 1 frames
Round trip time taken is 79 useconds
```

モジュール全体のポート内でこのテストを（ユーザが要求したときに）オンデマンドで明示的に実行し、スイッチに設定されているフレーム数を上書きするには、EXEC レベルで `system health internal-loopback` コマンドを使用します。

```
switch# system health internal-loopback interface iscsi 8/1 frame-count 20
Internal loopback test on interface iscsi8/1 was successful.
Sent 1 received 1 frames
Round trip time taken is 79 useconds
```

モジュール全体のポート内でこのテストを（ユーザが要求したときに）オンデマンドで明示的に実行し、スイッチに設定されているフレーム長を上書きするには、EXEC レベルで **system health internal-loopback** コマンドを使用します。

```
switch# system health internal-loopback interface iscsi 8/1 frame-count 32
Internal loopback test on interface iscsi8/1 was successful.
Sent 1 received 1 frames
Round trip time taken is 79 useconds
```



(注)

テストが正常に完了しなかった場合は、ソフトウェアが障害を分析し、次のエラーを出力します。

```
External loopback test on interface fc 7/2 failed.Failure reason: Failed to loopback,
analysis complete Failed device ID 3 on module 1
```

外部ループバック テストの実行

手動ループバック テストを実行すると、スイッチング モジュールまたはサービス モジュールのデータパスや、スーパーバイザ モジュールの制御パスにおけるハードウェア エラーを特定できます。外部ループバック テストは、同じポート間または 2 つのポート間で FC2 フレームを送受信します。

テストを実行する前に、Rx ポートから Tx ポートへループさせるためにケーブル（またはプラグ）を接続する必要があります。同じポートでテストを送受信する場合、特別なループ ケーブルが必要です。別のポートとの間でテストを送受信する場合は、通常のケーブルが使用できます。このテストを使用できるのは、ファイバ チャネル インターフェイスだけです。

長距離ネットワークに属するスイッチに接続されている外部デバイスに対してこのテストをオンデマンドで実行するには、EXEC レベルで **system health external-loopback interface** インターフェイス コマンドを使用します。

```
switch# system health external-loopback interface fc 3/1
This will shut the requested interfaces Do you want to continue (y/n)?[n] y
External loopback test on interface fc3/1 was successful.
Sent 1 received 1 frames
```

スイッチ上の 2 ポート間でこのテストをオンデマンドで実行するには、EXEC レベルで **system health external-loopback source interface destination interface** インターフェイス コマンドを実行します。

```
switch# system health external-loopback source interface fc 3/1 destination interface fc
3/2
This will shut the requested interfaces Do you want to continue (y/n)?[n] y
External loopback test on interface fc3/1 and interface fc3/2 was successful.
Sent 1 received 1 frames
```

長距離ネットワークに属するスイッチに接続されている外部デバイスに対してこのテストをオンデマンドで実行し、スイッチに設定されているフレーム数を上書きするには、EXEC レベルで **system health external-loopback interface frame-count** コマンドを使用します。

```
switch# system health external-loopback interface fc 3/1 frame-count 10
This will shut the requested interfaces Do you want to continue (y/n)?[n] y
External loopback test on interface fc3/1 was successful.
Sent 1 received 1 frames
```

長距離ネットワークに属するスイッチに接続されている外部デバイスに対してこのテストをオンデマンドで実行し、スイッチに設定されているフレーム長を上書きするには、EXEC レベルで **system health external-loopback interface frame-length** コマンドを使用します。

```
switch# system health external-loopback interface fc 3/1 frame-length 64
This will shut the requested interfaces Do you want to continue (y/n)?[n] y
External loopback test on interface fc3/1 was successful.
```

```
Sent 1 received 1 frames
```

特定のインターフェイスを確認なしで直接シャットダウンするには、**system health external-loopback interface force** コマンドを使用します。

```
switch# system health external-loopback interface fc 3/1 force
External loopback test on interface fc3/1 was successful.
Sent 1 received 1 frames
```



(注)

テストが正常に完了しなかった場合は、ソフトウェアが障害を分析し、次のエラーを出力します。

```
External loopback test on interface fc 7/2 failed.Failure reason: Failed to loopback,
analysis complete Failed device ID 3 on module 1
```

Serdes ループバックの実行

シリアライザ/デシリアライザ (serdes) ループバックでは、ポートのハードウェアがテストされます。このテストは、ファイバチャネルインターフェイスで使用できます。

モジュール全体のポート内でこのテストを (ユーザが要求したときに) オンデマンドで明示的に実行するには、EXEC レベルで **system health serdes-loopback** コマンドを使用します。

```
switch# system health serdes-loopback interface fc 3/1
This will shut the requested interfaces Do you want to continue (y/n)?[n] y
Serdes loopback test passed for module 3 port 1
```

モジュール全体のポート内でこのテストを (ユーザが要求したときに) オンデマンドで明示的に実行し、スイッチに設定されているフレーム数を上書きするには、EXEC レベルで **system health serdes-loopback** コマンドを使用します。

```
switch# system health serdes-loopback interface fc 3/1 frame-count 10
This will shut the requested interfaces Do you want to continue (y/n)?[n] y
Serdes loopback test passed for module 3 port 1
```

モジュール全体のポート内でこのテストを (ユーザが要求したときに) オンデマンドで明示的に実行し、スイッチに設定されているフレーム長を上書きするには、EXEC レベルで **system health serdes-loopback** コマンドを使用します。

```
switch# system health serdes-loopback interface fc 3/1 frame-length 32
This will shut the requested interfaces Do you want to continue (y/n)?[n] y
Serdes loopback test passed for module 3 port 1
```



(注)

テストが正常に完了しなかった場合は、ソフトウェアが障害を分析し、次のエラーを出力します。

```
External loopback test on interface fc 3/1 failed.Failure reason: Failed to loopback,
analysis complete Failed device ID 3 on module 3
```

現在のステータスの説明

各モジュールまたはテストのステータスは、その特定のモジュールでの OHMS テストの現在の設定状態によって異なります (表 60-1 を参照)。

表 60-1 テストおよびモジュールに関する OHMS の設定ステータス

ステータス	説明
Enabled	このモジュールのテストはイネーブルに設定されていますが、現在は実行されていません。
Disabled	現在このモジュールのテストはディセーブルに設定されています。
Running	このモジュールのテストはイネーブルに設定され、現在実行中です。
Failing	このステータスは、このモジュールで実行中のテストで障害が発生しそうな場合に表示されます。このステータスは、テストで回復できる可能性があります。
Failed	このモジュールのテストで障害が発生しました。ステータスは回復できません。
Stopped	テストは、Cisco SAN-OS ソフトウェアによってこのモジュールのテストが内部的に停止されました。
Internal failure	このモジュールのテストで、内部障害が発生しました。たとえば、システムヘルスアプリケーションがテスト手順の一部でソケットをオープンできません。
Diags failed	このモジュールまたはインターフェイスの起動時の診断で障害が発生しました。
On demand	現在、このモジュールで、システムヘルス外部ループバックまたはシステムヘルス内部ループバックテストが実行中です。オンデマンドで発行できるのは、これらの 2 つのコマンドだけです。
Suspended	1 つのオーバーサブスクライブポートが E または TE ポートモードに移行することにより、MDS 9100 シリーズでのみ発生します。1 つのオーバーサブスクライブポートがこのモードに移行すると、グループ内の他の 3 つのオーバーサブスクライブポートは中断されます。

各モジュールの各テストのステータスは、**show system health** コマンドで表示できます。「システムヘルスの表示」(P.60-20) を参照してください。

システムヘルスの表示

システム関連のステータス情報を表示するには、**show system health** コマンドを使用します(例 60-20 ~ 例 60-25 を参照)。

例 60-20 スイッチ内のすべてのモジュールの現在のヘルス情報の表示

```
switch# show system health
```

```
Current health information for module 2.
```

Test	Frequency	Status	Action
Bootflash	5 Sec	Running	Enabled
EIBC	5 Sec	Running	Enabled
Loopback	5 Sec	Running	Enabled

```
Current health information for module 6.
```

Test	Frequency	Status	Action
InBand	5 Sec	Running	Enabled
Bootflash	5 Sec	Running	Enabled
EIBC	5 Sec	Running	Enabled

```
Management Port          5 Sec          Running          Enabled
-----
```

例 60-21 指定されたモジュールの現在のヘルス情報の表示

```
switch# show system health module 8

Current health information for module 8.
```

```
Test                      Frequency      Status          Action
-----
Bootflash                 5 Sec         Running         Enabled
EOBC                      5 Sec         Running         Enabled
Loopback                  5 Sec         Running         Enabled
-----
```

例 60-22 すべてのモジュールのヘルス統計情報の表示

```
switch# show system health statistics
```

```
Test statistics for module # 1
```

```
Test Name                 State          Freq(s)      Run    Pass    Fail CFail Errs
-----
Bootflash                 Running        5s           12900 12900    0     0     0
EOBC                      Running        5s           12900 12900    0     0     0
Loopback                  Running        5s           12900 12900    0     0     0
-----
```

```
Test statistics for module # 3
```

```
Test Name                 State          Freq(s)      Run    Pass    Fail CFail Errs
-----
Bootflash                 Running        5s           12890 12890    0     0     0
EOBC                      Running        5s           12890 12890    0     0     0
Loopback                  Running        5s           12892 12892    0     0     0
-----
```

```
Test statistics for module # 5
```

```
Test Name                 State          Freq(s)      Run    Pass    Fail CFail Errs
-----
InBand                    Running        5s           12911 12911    0     0     0
Bootflash                 Running        5s           12911 12911    0     0     0
EOBC                      Running        5s           12911 12911    0     0     0
Management Port          Running        5s           12911 12911    0     0     0
-----
```

```
Test statistics for module # 6
```

```
Test Name                 State          Freq(s)      Run    Pass    Fail CFail Errs
-----
InBand                    Running        5s           12907 12907    0     0     0
Bootflash                 Running        5s           12907 12907    0     0     0
EOBC                      Running        5s           12907 12907    0     0     0
-----
```

```
Test statistics for module # 8
```

```
Test Name                 State          Freq(s)      Run    Pass    Fail CFail Errs
-----
```

```
-----
Bootflash          Running          5s   12895   12895       0     0     0
EOBC               Running          5s   12895   12895       0     0     0
Loopback          Running          5s   12896   12896       0     0     0
-----
```

例 60-23 指定されたモジュールの統計情報の表示

```
switch# show system health statistics module 3
```

```
Test statistics for module # 3
```

```
-----
Test Name          State           Freq(s)   Run    Pass    Fail  CFail  Errs
-----
Bootflash          Running        5s        12932  12932    0     0     0
EOBC               Running        5s        12932  12932    0     0     0
Loopback          Running        5s        12934  12934    0     0     0
-----
```

例 60-24 スイッチ全体のループバック テストの統計情報の表示

```
switch# show system health statistics loopback
```

```
-----
Mod Port Status          Run    Pass    Fail    CFail  Errs
-----
 1  16 Running        12953  12953    0        0     0
 3  32 Running        12945  12945    0        0     0
 8   8 Running        12949  12949    0        0     0
-----
```

例 60-25 指定されたインターフェイスのループバック テスト統計情報の表示

```
switch# show system health statistics loopback interface fc 3/1
```

```
-----
Mod Port Status          Run    Pass    Fail    CFail  Errs
-----
 3   1 Running           0      0      0        0     0
-----
```



(注) モジュール固有のループバック テストでエラーまたは障害が報告されない限り、インターフェイス固有のカウンタはゼロのままです。

例 60-26 すべてのモジュールのループバック テスト時間ログの表示

```
switch# show system health statistics loopback timelog
```

```
-----
Mod      Samples    Min(usecs)    Max(usecs)    Ave(usecs)
-----
 1         1872         149           364           222
 3         1862         415           743           549
 8         1865         134           455           349
-----
```

例 60-27 指定されたモジュールのループバック テスト時間ログの表示

```
switch# show system health statistics loopback module 8 timelog
```

Mod	Samples	Min (usecs)	Max (usecs)	Ave (usecs)
8	1867	134	455	349

オンボード障害ロギング

第 2 世代ファイバ チャンネル スイッチング モジュールでは、障害データを永続的ストレージに記録する機能が提供されます。この記録は、分析用に取得したり、表示したりできます。このオンボード障害ロギング (OBFL) 機能は、障害および環境情報をモジュールの不揮発性メモリに保管します。この情報は、障害が発生したカードの事後分析に役立ちます。

この項では、次のトピックについて取り上げます。

- 「OBFL の概要」 (P.60-23)
- 「スイッチの OBFL の設定」 (P.60-24)
- 「モジュールに対する OBFL の設定」 (P.60-25)
- 「OBFL ログの表示」 (P.60-26)

OBFL の概要

OBFL データは、モジュール上の既存の CompactFlash に保存されます。OBFL では、モジュールのファームウェアで使用できる永続的ロギング (PLOG) 機能を使用して CompactFlash にデータを保存します。保存されたデータを取得するためのメカニズムも提供されます。

OBFL 機能によって保存されるデータは、次のとおりです。

- 最初の電源投入時刻
- カードのシャーシ スロット番号
- カードの初期温度
- ファームウェア、BIOS、FPGA、および ASIC のバージョン
- カードのシリアル番号
- クラッシュのスタック トレース
- CPU hog 情報
- メモリ リーク情報
- ソフトウェア エラー メッセージ
- ハードウェア例外ログ
- 環境履歴
- OBFL 固有の履歴情報
- ASIC 割り込みおよびエラー統計の履歴
- ASIC レジスタ ダンプ

スイッチの OBFL の設定

スイッチのすべてのモジュールに OBFL を設定するには、次の手順を実行します。

	コマンド	目的
ステップ1	switch# config terminal switch(config)#	コンフィギュレーション モードに入ります。
ステップ2	switch(config)# hw-module logging onboard	すべての OBFL 機能をイネーブルにします。
	switch(config)# hw-module logging onboard cpu-hog	OBFL CPU hog イベントをイネーブルにします。
	switch(config)# hw-module logging onboard environmental-history	OBFL 環境履歴をイネーブルにします。
	switch(config)# hw-module logging onboard error-stats	OBFL エラー統計をイネーブルにします。
	switch(config)# hw-module logging onboard interrupt-stats	OBFL 割り込み統計をイネーブルにします。
	switch(config)# hw-module logging onboard mem-leak	OBFL メモリ リーク イベントをイネーブルにします。
	switch(config)# hw-module logging onboard miscellaneous-error	OBFL のその他の情報をイネーブルにします。
	switch(config)# hw-module logging onboard obfl-log	ブート動作時間、デバイス バージョン、および OBFL 履歴をイネーブルにします。
	switch(config)# no hw-module logging onboard	すべての OBFL 機能をディセーブルにします。

OBFL の設定ステータスを表示するには、**show logging onboard status** コマンドを使用します。

```
switch# show logging onboard status
```

```
Switch OBFL Log:                               Enabled

Module: 6 OBFL Log:
error-stats                                     Enabled
exception-log                                   Enabled
miscellaneous-error                             Enabled
obfl-log (boot-uptime/device-version/obfl-history) Enabled
system-health                                   Enabled
stack-trace                                     Enabled
```


モジュールに対する OBFL の設定

スイッチの特定のモジュールに OBFL を設定するには、次の手順を実行します。

	コマンド	目的
ステップ 1	switch# config terminal switch(config)#	コンフィギュレーション モードに入ります。
ステップ 2	switch(config)# hw-module logging onboard module 1	モジュールのすべての OBFL 機能をイネーブルにします。
	switch(config)# hw-module logging onboard module 1 cpu-hog	モジュールの OBFL CPU hog イベントをイネーブルにします。
	switch(config)# hw-module logging onboard module 1 environmental-history	モジュールの OBFL 環境履歴をイネーブルにします。
	switch(config)# hw-module logging onboard module 1 error-stats	モジュールの OBFL エラー統計情報をイネーブルにします。
	switch(config)# hw-module logging onboard module 1 interrupt-stats	モジュールの OBFL 割り込み統計情報をイネーブルにします。
	switch(config)# hw-module logging onboard module 1 mem-leak	モジュールの OBFL メモリ リーク イベントをイネーブルにします。
	switch(config)# hw-module logging onboard module 1 miscellaneous-error	モジュールの OBFL のその他の情報をイネーブルにします。
	switch(config)# hw-module logging onboard module 1 obfl-log	モジュールのブート動作時間、デバイスバージョン、および OBFL 履歴をイネーブルにします。
	switch(config)# no hw-module logging onboard module 1	モジュールのすべての OBFL 機能をディセーブルにします。

OBFL の設定ステータスを表示するには、**show logging onboard status** コマンドを使用します。

```
switch# show logging onboard status

Switch OBFL Log:                               Enabled

Module: 6 OBFL Log:
error-stats                                   Enabled
exception-log                                 Enabled
miscellaneous-error                           Enabled
obfl-log (boot-uptime/device-version/obfl-history) Enabled
system-health                                 Enabled
stack-trace                                   Enabled
```

OBFL ログの表示

モジュールの CompactFlash に保存されている OBFL 情報を表示するには、次のコマンドを使用します。

コマンド	目的
<code>show logging onboard boot-uptime</code>	ブートおよび動作時間の情報を表示します。
<code>show logging onboard cpu-hog</code>	CPU hog イベントの情報を表示します。
<code>show logging onboard device-version</code>	デバイス バージョン情報を表示します。
<code>show logging onboard endtime</code>	終了時刻までの OBFL ログを表示します。
<code>show logging onboard environmental-history</code>	環境履歴を表示します。
<code>show logging onboard error-stats</code>	エラー統計情報を表示します。
<code>show logging onboard exception-log</code>	例外ログ情報を表示します。
<code>show logging onboard interrupt-stats</code>	割り込み統計情報を表示します。
<code>show logging onboard mem-leak</code>	メモリ リーク情報を表示します。
<code>show logging onboard miscellaneous-error</code>	各種エラー情報を表示します。
<code>show logging onboard module slot</code>	指定したモジュールの OBFL 情報を表示します。
<code>show logging onboard obfl-history</code>	履歴情報を表示します。
<code>show logging onboard register-log</code>	登録ログ情報を表示します。
<code>show logging onboard stack-trace</code>	カーネル スタック トレース情報を表示します。
<code>show logging onboard starttime</code>	指定した開始時刻からの OBFL ログを表示します。
<code>show logging onboard system-health</code>	システム ヘルス情報を表示します。

デフォルト設定

表 60-2 に、システム ヘルスおよびログのデフォルト設定値を示します。

表 60-2 システム ヘルスおよびログのデフォルト設定値

パラメータ	デフォルト
カーネル コアの生成	1 つのモジュール
システム ヘルス	イネーブル
ループバック 頻度	5 秒
障害処理	イネーブル