



システム ステータス モニタリング

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。

- [システム ステータス モニタリングの機能履歴, on page 1](#)
- [システム ステータス モニタリングについての情報 \(2 ページ\)](#)
- [デフォルト設定, on page 7](#)
- [システム ヘルスの設定, on page 8](#)
- [オンボード障害ロギングの構成, on page 15](#)
- [モジュール カウンタのクリア, on page 17](#)
- [アラート、通知、およびカウンタのモニタリングの構成, on page 18](#)
- [コアの構成 \(21 ページ\)](#)
- [システム ステータスのモニタリング構成の確認, on page 24](#)
- [その他の参考資料, on page 35](#)

システム ステータス モニタリングの機能履歴

Table 1: システム ステータス モニタリングの機能履歴, on page 1 に、この機能のリリース履歴を示します。リリース 3.x 以降のリリースで導入または変更された機能のみが表に記載されています。

Table 1: システム ステータス モニタリングの機能履歴

| 機能名 | リリース | 機能情報 |
|---------------|---------|---|
| カーネル コア ロギング | 8.4(2c) | コア ファイルは、NX-OS で回復不能な障害が発生したときに作成されます。Cisco はコア ファイルを使用して障害を診断できます。 |
| 共通情報モデル (CIM) | 3.3(1a) | 共通情報モデルを表示するためのコマンドが追加されました。 |

| 機能名 | リリース | 機能情報 |
|---------------------------------|--------|--|
| オンラインシステム正常性メンテナンス (OHMS) の機能拡張 | 3.0(1) | 次の OHMS 機能拡張が含まれています。 <ul style="list-style-type: none"> • スイッチ上のすべてのモジュールのループバックテストのグローバルフレーム長を構成します。 • 特定のモジュールでのループバックテストのフレームカウントとフレーム長を指定します。 • 外部ループバックテスト用の送信元ポートと宛て先ポートの構成。 • ハードウェアをチェックするための serdes ループバックテストを提供します。 |
| オンボード障害ロギング (OBFL) | 3.0(1) | OBFL、第 2 世代モジュール用に OBFL を構成する方法、およびログ情報を表示する方法について説明します。 |

システムステータス モニタリングについての情報

オンラインヘルス管理システム

Online Health Management System (OHMS、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 シリーズ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。

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上で稼働するシステムヘルスアプリケーションは、そのモジュールが HA スタンバイモードで使用できる場合でも、スタンバイスーパーバイザモジュールだけを監視します。

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witch 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。

- 障害のあるコンポーネントを隔離するため、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、**Call Home** 通知、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、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テス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。

ループバックテストの設定頻度

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。

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。

ハードウェア障害時の処理

`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 NX-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。

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



Tip 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。



Tip 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)。

特定モジュールのテスト

NX-OS ソフトウェアのシステムヘルス機能は、次の領域のテストを実行します。

- アクティブな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。
- 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）。



Note Cisco MDS 9700 シリーズ スイッチでは、iSCSI ポートは適用されません。

前回のエラー レポートのクリア

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。



Tip 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。

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（[Table 2: テストおよびモジュールに関する OHMS の設定ス
テータス](#), on page 5 を参照）。

Table 2: 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 |
|-----------------|---|
| [有効 (Enabled)] | このモジュールのテストは有効化されていますが、現在は実行されていま せん。 |
| 無効 | 現在このモジュールのテストは無効化されています。 |
| Running | このモジュールのテストは有効化されていて、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。 |

| ステータス | 説明 |
|------------------|--|
| 失敗しました | 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。 |
| 停止 (Stopped) | テストは、Cisco NX-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。 |
| オン デマンド | 現在、このモジュールで、システム正常性の外部ループバックまたはシステム正常性の内部ループバック 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。 |
| 一時停止 | 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, on page 24を参照してください。

オンボード障害ロギング

第2世代ファイバチャ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On-Board Failure Logging (OBFL: オンボード障害ロギング) 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。

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

コアファイル

コア ファイルは、NX-OS で回復不能な障害が発生したときに作成されます。これらは `tar.gz` フォーマットのファイルのバンドルであり、シスコが障害を診断するために使用できます。

NX-OS は、スーパーバイザとモジュールの両方からプロセスとカーネル コア ファイルを生成できます。プロセス コア ファイルは、障害時にそれらが発生したモジュールからアクティブスーパーバイザにアップロードされます。コアファイルは揮発性であり、スーパーバイザがリセットされると失われます。カーネル コア ファイルは、作成されたスーパーバイザに保存され、スーパーバイザのリセット後も保持されます。

最初と最後のコア

一般に、プロセスによって生成された最初のコアと最新のコアには、デバッグに最も役立つ情報が含まれています。コア ファイルがアクティブなスーパーバイザ モジュールで生成された場合、コアリポジトリのスペースを節約するために、同じプロセス用に新しいコアが生成されると、最初と最後のコア機能によって中間コアが自動的に削除されます。

デフォルト設定

[Table 3: デフォルトのシステムステータス モニタリング, on page 7](#) に、デフォルト設定を示します。

Table 3: デフォルトのシステムステータス モニタリング

| パラメータ | デフォルト |
|----------|--------------|
| カーネルコア収集 | 無効 |
| システムヘルス | 有効 |
| ループバック頻度 | 5 秒 |
| 障害処理 | 有効 (Enabled) |

システムヘルスの設定

Online Health Management System (OHMS、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。

システムの正常性を構成するためのタスクフロー

システムの正常性を構成するには、次の手順を実行します。

Procedure

- ステップ1 システム正常性の開始を有効化します。
 - ステップ2 ループバックテストの構成頻度を構成します。
 - ステップ3 ループバックテスト構成のフレーム長を構成します。
 - ステップ4 ハードウェア障害アクションを構成します。
 - ステップ5 テストの実行要件を実施します。
 - ステップ6 前回のエラーレポートをクリアします。
 - ステップ7 内部ループバックテストを実施します。
 - ステップ8 外部ループバックテストを実施します。
 - ステップ9 Serdes ループバックを実施します。
-

システムの正常性開始の構成

デフォルトでは、システムの正常性機能はCisco MDS 9000ファミリの各スイッチで有効です。

Cisco MDS 9000ファミリの任意のスイッチでこの機能を無効化または有効化するには、次の手順を実行します。

Procedure

- ステップ1 `switch# configure terminal`
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`
システム正常性が無効になっています。
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。

ステップ 3 switch(config)# **system health**

システム正常性が有効になっています。

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 4 switch(config)# **no system health interface fc8/1**

インターフェイス fc8/13 のシステム正常性が無効になっています。

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正常性を設定します。

ステップ 5 switch(config)# **system health interface fc8/1**

インターフェイス fc8/13 のシステム正常性が有効になっています。

システム正常性を有効（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 テストの構成頻度の構成

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構成するには、次の手順を実行します。

Procedure**ステップ 1** switch# **configure terminal**

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。指定できる範囲は 5 ~ 255 秒です。

ループバック テスト構成のフレーム長の構成

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構成するには、次の手順を実行します。

Procedure**ステップ 1** switch# **configure terminal**

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 バイトに構成します。有効な範囲は 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）を生成するように構成します。

ハードウェア障害アクションの構成

スイッチの障害アクションを構成するには、次の手順を実行します。

Procedure

ステップ 1 switch# **configure terminal**

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

特定のモジュールで必要なテストを実行するには、次の手順を実行します。

Procedure

ステップ 1 `switch# configure terminal`

コンフィギュレーション モードに入ります。

Note 次のステップは、任意の順序で実行できます。

Note 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 2 `switch(config)# system health module 8 bootflash`

スロット 8 のモジュールでブートフラッシュ テストを有効にします。

ステップ 3 `switch(config)# system health module 8 bootflash frequency 200`

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 4 `switch(config)# system health module 8 eobc`

スロット 8 のモジュールで EOBC テストを有効にします。

ステップ 5 `switch(config)# system health module 8 loopback`

スロット 8 のモジュールでループバック テストを有効にします。

ステップ 6 `switch(config)# system health module 5 management`

スロット 5 のモジュールで管理テストを有効にします。

前回のエラー レポートのクリア

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、システム正常性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。`bootflash`、`eobc`、`inband`、`loopback`、および `mgmt` テスト オプションは所定のモジュールに対して個別に指定することができます。

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 テストは同一のポートに対して FC2 フレームを送受信し、ラウンドトリップ時間をマイクロ秒単位で示します。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、スイッチに構成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、スイッチに構成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
```



Note テストが正常に完了しなかった場合、ソフトウェアは失敗を分析し、次のエラーを出力します。「インターフェイス fc 7/2 の外部ループバック テストが失敗しました。」失敗の理由：ループバックが失敗しました。モジュール 1 での失敗したデバイス ID 3 の分析を完了します

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、同一のポートの間または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。同じポートの間でテストする場合は、特殊なループケーブルが必要です。異なるポートとの間でテストする場合は、通常のケーブルを使用できます。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 interface** 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
```

スイッチの2つの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レベルの **system health external-loopback source interface destination interface interface** コマンドを使用します。

```
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、スイッチ上で構成されたフレームカウントを上書きするには、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、スイッチ上で構成された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
```

system health external-loopback interface force コマンドを使用して、バックアウトの確認なしに必要なインターフェイスを直接シャットダウンします。

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



Note テストが正常に完了しなかった場合、ソフトウェアは失敗を分析し、次のエラーを出力します。「インターフェイス fc 7/2 の外部ループバック テストが失敗しました。」失敗の理由：ループバックが失敗しました。モジュール 1 での失敗したデバイス ID 3 の分析を完了します

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、スイッチに構成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、スイッチに構成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
```



Note テストが正常に完了しなかった場合、ソフトウェアは失敗を分析し、次のエラーを出力します。「インターフェイス fc 3/1 の外部ループバック テストが失敗しました。」失敗の理由：ループバックが失敗しました。モジュール 3 での失敗したデバイス ID 3 の分析を完了します。

オンボード障害ロギングの構成

各ハードウェアモジュールは障害データをオンモジュールの永続的ストレージに記録し、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 On-Board Failure Logging (OBFL: オンボード障害ロギング) 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。

スイッチの OBFL の構成

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

Procedure

ステップ 1 switch# **configure terminal**

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。

Note この CLI は、no hw-module logging onboard コマンドによって無効にされた OBFL 機能のみを有効にします。個別に無効にされていた OBFL 機能については、hw-module logging onboard obfl-feature コマンドを使用して有効にしてください。

ステップ 3 switch(config)# **hw-module logging onboard cpu-hog**

OBFL CPU hog イベントを有効にします。

ステップ 4 switch(config)# **hw-module logging onboard environmental-history**

OBFL 環境履歴をイネーブルにします。

ステップ 5 switch(config)# **hw-module logging onboard error-stats**

OBFL エラー統計をイネーブルにします。

ステップ 6 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。

ステップ 7 switch(config)# **hw-module logging onboard mem-leak**

OBFL メモリ リーク イベントを有効にします。

ステップ 8 switch(config)# **hw-module logging onboard miscellaneous-error**

OBFL のその他の情報を有効にします。

ステップ 9 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。

ステップ 10 `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。

モジュールの OBFL の構成

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

Procedure

ステップ 1 `switch# configure terminal`

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 機能を有効にします。

ステップ 3 `switch(config)# hw-module logging onboard module 1 cpu-hog`

モジュールの OBFL CPU hog イベントを無効にします。

ステップ 4 `switch(config)# hw-module logging onboard module 1 environmental-history`

モジュールの OBFL 環境履歴を有効にします。

ステップ 5 `switch(config)# hw-module logging onboard module 1 error-stats`

モジュールの OBFL エラー統計を有効にします。

ステップ 6 `switch(config)# hw-module logging onboard module 1 interrupt-stats`

モジュールの OBFL 割り込み統計を有効にします。

ステップ 7 `switch(config)# hw-module logging onboard module 1 mem-leak`

モジュールの OBFL メモリ リーク イベントを有効にします。

ステップ 8 `switch(config)# hw-module logging onboard module 1 miscellaneous-error`

モジュールの OBFL のその他の情報を有効にします。

ステップ 9 `switch(config)# hw-module logging onboard module 1 obfl-log`

モジュールのブート稼働時間、デバイスバージョン、および OBFL 履歴を有効にします。

ステップ 10 `switch(config)# no hw-module logging onboard module 1`

モジュールのすべての OBFL 機能を無効にします。

モジュール カウンタのクリア



Note モジュール カウンタは、Device Manager または DCNM-SAN を使用してクリアできません。

モジュール カウンタをリセットする手順は、次のとおりです。

Procedure

ステップ 1 `switch# attach module 1`

`ModuleX#`

モジュール 1 をシャーシに取り付けます。

ステップ 2 `ModuleX# clear ASIC-cnt all`

モジュール内のすべてのデバイスのカウンタをクリアします。

ステップ 3 `ModuleX# clear ASIC-cnt list-all-devices`

`ModuleX# clear ASIC-cnt device-id device-id`

指定されたデバイス ID のみのカウンタをクリアします。デバイス ID は、1 ~ 255 の範囲で指定できます。

すべてのモジュールのカウンタのリセット

すべてのモジュールのカウンタをリセットするには、次の手順に従います。

Procedure

`switch# debug system internal clear-counters all`

スイッチ内のすべてのモジュールのカウンタをクリアします。

アラート、通知、およびカウンタのモニタリングの構成

このセクションでは、アラート、通知、およびモニタのカウンタを構成する方法について説明します。

CPU 使用率のモニタリング

システム CPU の使用状況を表示するには、**show processes cpu** コマンドを使用します。

次の例は、現在の VDC のプロセスと CPU 使用率を表示する方法を示しています。

```
switch# show processes cpu
PID      Runtime(ms)   Invoked    uSecs   1Sec   Process
-----
      4         386829    67421866      5     0.9%   ksoftirqd/0
    3667         270567    396229      682     9.8%   syslogd
    3942           262         161     1632     7.8%   netstack
    4006    106999945   354495641     301    28.2%   snmpd
    4026     4454796     461564     9651     0.9%   sac_usd
    4424         84187     726180     115     0.9%   vpc
    4426         146378     919073     159     0.9%   tunnel
CPU util  :   25.0% user,   30.5% kernel,   44.5% idle
```

RAM 使用量情報の取得

プロセッサの RAM 使用量は、次の SNMP 変数を使用して取得できます。ceExtProcessorRam。

```
ceExtProcessorRam OBJECT-TYPE
    SYNTAX  Unsigned32
    UNITS   "bytes"
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Total number of bytes of RAM available on the
        Processor."
    ::= { ceExtPhysicalProcessorEntry 1 }
```

Rx および Tx トラフィック カウンタのモニタリング

Rx および Tx トラフィック カウンタをモニタするときは、Rx カウンタ OID を含める必要があります。

```
ifHCInOctets
```

インターフェイスのステータスのモニタリング

インターフェイスのステータスをモニタするには、ifAlias（このトラップはインターフェイスの説明を設定できます）と ifDescr を持つ IETF 拡張リンクダウン トラップを使用し、次に示すように ASCII 形式でポート名を表示します。

```
switch (config)# snmp-server enable traps link
  cieLinkDown          Cisco extended link state down notification
  cieLinkUp            Cisco extended link state up notification
  cisco-xcvr-mon-status-chg Cisco interface transceiver monitor status change
                        notification
  delayed-link-state-change Delayed link state change
  extended-linkDown    IETF extended link state down notification
  extended-linkUp      IETF extended link state up notification
  linkDown             IETF Link state down notification
  linkUp               IETF Link state up notification
switch (config)#
```

次に、トラップの例を示します。

```
[+]          10          16:41:39.79          IF-MIB:linkDown trap:SNMPv2c from
[172.25.234.200 Port: 162 Community: public]
SNMPv2-MIB:sysUpTime.0 : (35519336)          Syntax: TimeTicks
SNMPv2-MIB:snmpTrapOID.0 : (IF-MIB:linkDown)          Syntax: ObjectID
IF-MIB:ifIndex.440414208 : (440414208)          Syntax: INTEGER, Instance IDs: (440414208)
IF-MIB:ifAdminStatus.440414208 : (down)          Syntax: INTEGER, Instance IDs: (440414208)
IF-MIB:ifOperStatus.440414208 : (down)          Syntax: INTEGER, Instance IDs: (440414208)
IF-MIB:ifDescr.440414208 : (Ethernet9/4)          Syntax: RFC1213-MIB:DisplayString, Instance
IDs: (440414208)
IF-MIB:ifAlias.440414208 : (eth9/4)          Syntax: SNMPv2-TC:DisplayString, Instance IDs:
(440414208)
SNMPv2-MIB:snmpTrapEnterprise.0 : (IF-MIB:linkDown)          Syntax: ObjectID
```

トランシーバしきい値のモニタリング

cisco-xcvr-mon-status-chg トラップ方法を使用して、次に示すように、しきい値のデジタル診断統計をモニタします。

```
switch (config)# snmp-server enable traps link cisco-xcvr-mon-status-chg
switch (config)#
```

トラップ MIB は次のとおりです。

```
cIfXcvrMonStatusChangeNotif NOTIFICATION-TYPE
  OBJECTS          {
                    ifName,
                    cIfXcvrMonDigitalDiagTempAlarm,
                    cIfXcvrMonDigitalDiagTempWarning,
                    cIfXcvrMonDigitalDiagVoltAlarm,
                    cIfXcvrMonDigitalDiagVoltWarning,
                    cIfXcvrMonDigitalDiagCurrAlarm,
                    cIfXcvrMonDigitalDiagCurrWarning,
                    cIfXcvrMonDigitalDiagRxPwrAlarm,
                    cIfXcvrMonDigitalDiagRxPwrWarning,
                    cIfXcvrMonDigitalDiagTxPwrAlarm,
                    cIfXcvrMonDigitalDiagTxPwrWarning,
                    cIfXcvrMonDigitalDiagTxFaultAlarm
                    }
  STATUS          current
```

次の例は、トランシーバの詳細情報を表示する方法を示します。

```

switch(config)# show interface ethernet 1/17 transceiver details
Ethernet1/17
  transceiver is present
  type is 10Gbase-SR
  name is CISCO-AVAGO
  part number is SFBR-7702SDZ
  revision is G2.3
  serial number is AGA1427618P
  nominal bitrate is 10300 MBit/sec
  Link length supported for 50/125um OM2 fiber is 82 m
  Link length supported for 62.5/125um fiber is 26 m
  Link length supported for 50/125um OM3 fiber is 300 m
  cisco id is --
  cisco extended id number is 4
    SFP Detail Diagnostics Information (internal calibration)
-----
                Current           Alarms           Warnings
                Measurement       High           Low           High           Low
-----
Temperature    27.65 C           75.00 C       -5.00 C       70.00 C       0.00 C
Voltage         3.29 V           3.63 V        2.97 V        3.46 V        3.13 V
Current        5.42 mA          10.50 mA      2.50 mA       10.50 mA      2.50 mA
Tx Power       -2.51 dBm        1.69 dBm     -11.30 dBm    -1.30 dBm     -7.30 dBm
Rx Power       -2.64 dBm        1.99 dBm     -13.97 dBm    -1.00 dBm     -9.91 dBm
Transmit Fault Count = 0
-----
Note: ++ high-alarm; + high-warning; -- low-alarm; - low-warning
switch(config)#

```

スーパーバイザスイッチオーバー通知の構成

スーパーバイザスイッチオーバー通知は、`ciscoRFSwactNotif` トラップをリスンすることでモニタできます。

```

ciscoRFSwactNotif NOTIFICATION-TYPE
OBJECTS {
  cRFStatusUnitId,
  sysUpTime,
  cRFStatusLastSwactReasonCode
}

```

CRC および FCS エラーを含むカウンタの構成

次の例に示すように、`dot3StatsFCSErrors` カウンタをポーリングすることにより、インターフェースの CRC および FCS エラーを含めることができます。

`dot3StatsFCSErrors Counter32`

```

Dot3StatsEntry ::= SEQUENCE {
  dot3StatsIndex           InterfaceIndex,
  dot3StatsAlignmentErrors Counter32,
  dot3StatsFCSErrors      Counter32,
  dot3StatsSingleCollisionFrames Counter32,
  dot3StatsMultipleCollisionFrames Counter32,
  dot3StatsSQETestErrors  Counter32,
  dot3StatsDeferredTransmissions Counter32,
  dot3StatsLateCollisions Counter32,
}

```

```

dot3StatsExcessiveCollisions      Counter32,
dot3StatsInternalMacTransmitErrors Counter32,
dot3StatsCarrierSenseErrors       Counter32,
dot3StatsFrameTooLongs            Counter32,
dot3StatsInternalMacReceiveErrors Counter32,
dot3StatsEtherChipSet             OBJECT IDENTIFIER,
dot3StatsSymbolErrors             Counter32,
dot3StatsDuplexStatus              INTEGER,
dot3StatsRateControlAbility        TruthValue,
dot3StatsRateControlStatus         INTEGER
}

```

アラートの Call Home の構成

Call Home 機能を使用すると、システムで例外が発生したときに Call Home 電子メールを受信できます。次の CLI または SNMP を使用して、Call Home 構成をセットアップし、すべてのアラート グループを有効にします。

```

switch (config)# callhome
switch-FC-VDC(config-callhome)# destination-profile full-txt-destination alert-group
All This alert group consists of all of the callhome
    messages
    Cisco-TAC Events which are meant for Cisco TAC only
    Configuration Events related to Configuration
    Diagnostic Events related to Diagnostic
    EEM EEM events
    Environmental Power,fan,temperature related events
    Inventory Inventory status events
    License Events related to licensing
    Linecard-Hardware Linecard related events
    Supervisor-Hardware Supervisor related events
    Syslog-group-port Events related to syslog messages filed by port manager
    System Software related events
    Test User generated test events
switch-FC-VDC(config-callhome)#

```

ユーザ認証失敗のモニタリング

authenticationFailure トラップをリッスンすることで、ユーザ認証の失敗をモニタできます。

```
SNMPv2-MIB: authenticationFailure trap
```

コアの構成

コアファイルは、ユーザが手動で保存することも、障害発生時に自動的に保存することもできます。コア ファイルが作成された場合は、それを不揮発性ファイル スペース（ホストなど）にコピーして保存し、診断のためにシスコに報告します。

コアは複数回コピーできます。コアをリモートホスト上のファイルスペースにコピーするために、IPv4、IPv6、および多くのプロトコルの両方がサポートされています。これには、安全な環境での自動コピーに便利なパスワードなしの SSH が含まれます。リモート ホストへのパスワードレス アクセスの構成の詳細については、『Cisco MDS 9000 シリーズセキュリティの設

定ガイド、リリース 8.x』の「SSH サービスおよび Telnet の構成」の章の「パスワードレス ファイル コピーおよび SSH」セクションを参照してください。

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。



ヒント コアをコピーする前に、ユーザの書き込み権限を持つ接続先ディレクトリを作成していることを確認してください。

カーネル コア収集の構成

カーネル コア収集を構成する手順は、次のとおりです。

手順

ステップ 1 switch# **configure terminal**

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 kernel core**

カーネル クラッシュが発生した場合に、カーネル コアの収集を有効にします。

ステップ 3 switch(config)# **no system kernel core**

(オプション) カーネル コアの収集を無効にします。

コアの手動コピー

サポート対象のスイッチ上の接続先は、slot0 です。コアをリモートの接続先に転送するサポート対象プロトコルは、TFTP、SFTP、および SCP です。

コアの手動保存を構成するには、次の手順を実行します。

Procedure

```
switch# copy core://module/process-id[/instance] destination://[[user@]host/][directory]
```

プロセスのコアを指定された場所にコピーします。

コアの自動コピー

サポートされているスイッチ上の接続先は、bootflash、slot0、およびusb1です。コアをリモートの接続先に転送するサポート対象プロトコルは、HTTP、HTTPS、TFTP、FTP、SFTP、およびSCPです。

コアの自動保存を構成するには、次の手順を実行します。

Procedure

ステップ 1 switch# **configure**

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 cores destination://[[user@]host][directory]**

コアファイルが作成されるとすぐに、指定された接続先にコアファイルを保存します。

ステップ 3 switch(config)# **no system cores**

(オプション) コアファイルの自動保存を無効にします。

コアの削除

コアファイルはコピー後に自動的に削除されません。コアがコピーされたら、スイッチコアリポジトリから削除してスペースを再利用し、分析のためにシスコサポートに報告します。

clear core_file コマンドを使用して、スイッチコアリポジトリから1つのコアを削除します。

```
switch# clear core_file module module pid pid
```

clear cores コマンドを使用して、スイッチコアリポジトリ内のすべてのコアをクリアします。

```
switch# clear cores
```

例：コアの構成

次の例では、スロット5で生成されたPID 1524のプロセスのコアを、ユーザ *mdsadmin* としてHTTPSを持つホストの *cores* ディレクトリにコピーします。

```
switch# copy core://5/1524 https://mdsadmin@192.168.1.2/cores
```

次の例では、コアファイルが作成された直後に、SCPがユーザ *mdsadmin* としてホスト上の */tftpboot/cores* ディレクトリに自動的にコピーされます。これを機能させるには、最初にパスワードなしのSSHを構成します。

```
switch# configure
```

```
switch(config)# system cores scp://mdsadmin@192.168.1.2/tftpboot/cores
```

次の例では、PID 1234 のプロセスのモジュール 1 から生成されたコアを削除します。

```
switch# clear core_file module 1 pid 1234
```

システムステータスのモニタリング構成の確認

システムステータスのモニタリング構成情報を表示するには、次の作業のいずれかを行います。

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（スイッチ内のすべてのモジュールの現在の正常性、[on page 24](#)～指定されたモジュールのループバック テスト時間ログ、[on page 26](#) を参照）。

スイッチ内のすべてのモジュールの現在の正常性

次の例は、スイッチ内のすべてのモジュールの現在の正常性を表示しています。

```
switch# show system health

Current health information for module 2.
Test                Frequency      Status        Action
-----
Bootflash           5 Sec         Running       Enabled
EOBC                 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
EOBC                 5 Sec         Running       Enabled
Management Port    5 Sec         Running       Enabled
-----
```

指定されたモジュールの現在の正常性

次の例は、指定されたモジュールの現在の正常性を表示しています。

```
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
-----
```

すべてのモジュールの正常性統計

次の例は、すべてのモジュールの正常性統計を表示しています。

```
switch# show system health statistics
Test statistics for module # 1
-----
Test Name          State          Frequency 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          Frequency 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          Frequency 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          Frequency 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          Frequency 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
-----
```

指定されたモジュールの統計情報の表示

次の例は、指定されたモジュールの統計を表示しています。

```
switch# show system health statistics module 3
Test statistics for module # 3
-----
Test Name          State          Frequency 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
-----
```

スイッチ全体のループバック テストの統計

次の例は、スイッチ全体のループバック テストの統計を表示しています。

```
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
-----
```

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

次の例は、指定されたインターフェイスのループバック テスト統計を表示しています。

```
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
-----
```



Note 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。

すべてのモジュールのループバック テスト時間ログ

次の例では、すべてのモジュールのループバック テスト時間ログを表示しています。

```
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
-----
```

指定されたモジュールのループバック テスト時間ログ

次の例では、指定されたモジュールのループバック テスト時間ログを表示しています。

```
switch# show system health statistics loopback module 8 timelog
-----
Mod          Samples      Min(usecs)    Max(usecs)    Ave(usecs)
8            1867         134           455           349
-----
```

ループバック テスト構成のフレーム長の確認

ループバック周波数の構成を確認するには、**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
```

スイッチの OBFL の確認

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

モジュールの OBFL の確認

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

カーネル コア収集の確認

カーネル コア 収集の構成は、実行構成をチェックすることで確認できます。

```
switch# show running-config | include 'kernel core'
system kernel core
```

自動コアコピーの確認

`show system cores` コマンドを使用して、自動コアコピー機能の構成を表示します。

```
switch# show system cores
Cores are transferred to scp://mdsadmin@192.168.1.2/tftpboot/cores
```

OBFL ログの表示

モジュールに保存されている 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 counter-stats</code> | カウンタ統計を表示します。 Note Cisco MDS 9132T および Cisco MDS 9396T スイッチでは、このコマンドの出力に、削除された LEM ポートに関する情報が表示されます。 |
| <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。 |

| コマンド | 目的 |
|---|------------------|
| show logging onboard system-health | システムヘルス情報を表示します。 |

モジュール カウンタ情報の表示

この例では、モジュール内のすべてのデバイスのデバイス ID を表示しています。

```
switch# attach module 4
Attaching to module 4 ...
To exit type 'exit', to abort type '$.'
Linux lc04 2.6.10_mvl401-pc_target #1 Tue Dec 16 22:58:32 PST 2008 ppc GNU/Linux

module-4# clear asic-cnt list-all-devices
      Asic Name |          Device ID
-----|-----
Stratosphere  |                63
transceiver   |                46
Skyline-asic  |                57
Skyline-ni    |                60
Skyline-xbar  |                59
Skyline-fwd   |                58
Tuscany-asic  |                52
Tuscany-xbar  |                54
Tuscany-que   |                55
Tuscany-fwd   |                53
Fwd-spi-group|                73
Fwd-parser    |                74
      eobc     |                10
      X-Bus IO |                 1
Power Mngmnt  |                25
      Epld    |
```

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（CPU 使用率情報, [on page 30](#) ~ プロセスに関するメモリ情報, [on page 32](#) を参照）。

システム プロセスの表示

次の例では、システム プロセスを表示します。

```
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     -          -          -    mdog
-        NR     -          -          0    vbuilder
```

それぞれの説明は次のとおりです。

- ProcessId = 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 = defunct (「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 Name = プロセスの名前

CPU 使用率情報

次の例は、CPU 使用率情報を表示しています。

```
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。

- MemAllocated = このプロセスがシステムから動的に割り当てられているすべてのメモリの合計。すでにシステムに返されたメモリが含まれている場合があります。
- Runtime CPU Time (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)

プロセス ログ情報

次の例では、プロセス ログ情報を表示しています。

```
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刻。

プロセスに関する詳細ログ情報

次の例では、プロセスに関する詳細なログ情報を表示しています。

```
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     7FFFFFFD20
  TOTAL    23764 KB
Register Set:
  EBX 00000005      ECX 7FFFFFF8CC      EDX 00000000
  ESI 00000000      EDI 7FFFFFF6CC      EBP 7FFFFFF95C
  EAX FFFFFFFDFE    XDS 8010002B        XES 0000002B
  EAX 0000008E (orig) EIP 2ACE133E        XCS 00000023
  EFL 00000207      ESP 7FFFFFF654      XSS 0000002B
Stack: 1740 bytes. ESP 7FFFFFF654, TOP 7FFFFFFD20
0x7FFFFFF654: 00000000 00000008 00000003 08051E95 .....
0x7FFFFFF664: 00000005 7FFFFFF8CC 00000000 00000000 .....
0x7FFFFFF674: 7FFFFFF6CC 00000001 7FFFFFF95C 080522CD .....\"..
0x7FFFFFF684: 7FFFFFF9A4 00000008 7FFFFFFC34 2AC1F18C .....4.....*
```

すべてのプロセス ログの詳細

次の例では、すべてのプロセス ログの詳細を表示しています。

```
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
...
```

プロセスに関するメモリ情報

次の例では、プロセスに関するメモリ情報を表示しています。

```
switch# show processes memory
PID      MemAlloc  MemLimit  MemUsed   StackBase/Ptr  Process
-----
  1      147456   0         1667072   7ffffe50/7ffff950  init
  2         0 0         0         0/0             ksoftirqd/0
  3         0 0         0         0/0             desched/0
  4         0 0         0         0/0             events/0
  5         0 0         0         0/0             khelper
```

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 (デフォルトのスイッチポートの状態, [on page 32](#) ~ システム関連の CPU およびメモリ情報, [on page 34](#) を参照)。

デフォルトのスイッチポートの状態

次の例は、デフォルトのスイッチポートの状態を示しています。

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

指定 ID のエラー情報

次の例では、指定された ID のエラー情報を表示します。

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


システム リセット情報

次の例は、システム リセット情報を表示します。

```
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 コマンドにより、以下の情報が表示されます。

- 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。どのスーパーバイザ モジュールも存在しない場合には、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。どのスーパーバイザ モジュールも存在しない場合には、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、特定のスロットの特定のモジュールでの、最後の 4 つのリセット理由コードを表示します。モジュールが存在しない場合には、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 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 に保存されているリセット理由情報をクリアします。

システム稼働時間

次の例は、システムの稼働時間を表示します。

```
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 およびメモリ統計を表示するには、**show system resources** コマンドを使用します（システム関連の CPU およびメモリ情報, on page 34 を参照）。

システム関連の CPU およびメモリ情報

次の例は、システム関連の CPU およびメモリ情報を表示します。

```
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、バッファおよびキャッシュの値には、*used* メモリの統計も含まれます。

プロセス障害ログの表示

プロセス障害ログの概要の表示

致命的なプロセス障害の履歴や、イベントごとに収集されたログをモジュール単位で表示できます。**slot** コマンドを使用して、特定のモジュールで **show processes log** コマンドを実行します。

次の例は、モジュール 2 のプロセス障害ログの概要を表示します。

```
switch# slot 2 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
```

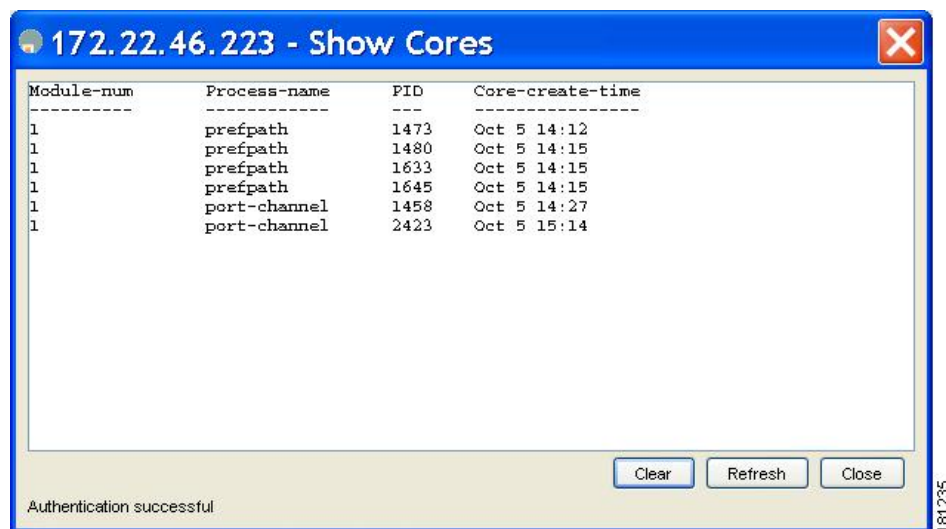
```

bios_daemon      2227          N      Y      N  Mon Sep 29 15:30:51 2003

```

次の例では、デバイスマネージャでシステムのプロセス コアを表示します。

Figure 1: [Show Cores] ダイアログボックス



プロセス コアの表示

次の例では、アクティブスーパーバイザモジュールに保存されているすべてのコアを表示します。

```

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

```

その他の参考資料

システムプロセスとログの実装に関する詳細情報については、次のセクションを参照してください。

MIB

| MIB | MIB のリンク |
|--|--|
| <ul style="list-style-type: none"> • CISCO-SYSTEM-EXT-MIB • CISCO-SYSTEM-MIB | <p>MIB を検索およびダウンロードするには、次の URL にアクセスしてください。</p> <p>http://www.cisco.com/en/US/products/ps5989/prod_technical_reference_list.html</p> |

翻訳について

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。