



Embedded Event Manager の設定

この章は、次の項で構成されています。

- [組み込みイベント マネージャについて \(1 ページ\)](#)
- [Embedded Event Manager ポリシー \(2 ページ\)](#)
- [Embedded Event Manager の前提条件 \(5 ページ\)](#)
- [Embedded Event Manager の注意事項および制約事項 \(5 ページ\)](#)
- [Embedded Event Manager のデフォルト設定 \(6 ページ\)](#)
- [環境変数の定義 \(6 ページ\)](#)
- [CLI によるユーザ ポリシーの定義 \(7 ページ\)](#)
- [イベント文の設定 \(9 ページ\)](#)
- [アクション文の設定 \(12 ページ\)](#)
- [VSH スクリプトによるポリシーの定義 \(15 ページ\)](#)
- [VSH スクリプト ポリシーの登録およびアクティブ化 \(15 ページ\)](#)
- [システム ポリシーの上書き \(16 ページ\)](#)
- [EEM パブリッシャとしての syslog の設定 \(18 ページ\)](#)
- [イベント ログの自動収集とバックアップ \(19 ページ\)](#)

組み込みイベント マネージャについて

Cisco NX-OS システム内のクリティカル イベントを検出して処理する機能は、ハイ アベイラビリティにとって重要です。Embedded Event Manager (EEM) は、デバイス上で発生するイベントをモニターし、設定に基づいてこれらのイベントを回復またはトラブルシューティングするためのアクションを実行することによってシステム内のイベントを検出して処理する、中央のポリシー駆動型のフレームワークを提供します。

EEM は次の 3 種類の主要コンポーネントからなります。

イベント文

何らかのアクション、回避策、または通知が必要になる可能性のある、別の Cisco NX-OS コンポーネントからモニターするイベント。

アクション文

電子メールの送信やインターフェイスのディセーブル化などの、イベントから回復するために EEM が実行できるアクション。

ポリシー

イベントのトラブルシューティングまたはイベントからの回復を目的とした1つまたは複数のアクションとペアになったイベント。

EEM を使用しない場合は、個々のコンポーネントが独自のイベントの検出および処理を行います。たとえば、ポートでフラップが頻繁に発生する場合は、「errDisable ステートにする」のポリシーが ETHPM に組み込まれます。

Embedded Event Manager ポリシー

EEM ポリシーは、イベント文および1つまたは複数のアクション文からなります。イベント文では、探すイベントとともに、イベントのフィルタリング特性を定義します。アクション文では、イベントの発生時に EEM が実行するアクションを定義します。

たとえば、いつカードがデバイスから取り外されたかを識別し、カードの取り外しに関する詳細を記録する EEM ポリシーを設定できます。カードの取り外しのインスタンスすべてを探すようにシステムに指示するイベント文および詳細を記録するようにシステムに指示するアクション文を設定します。

コマンドラインインターフェイス (CLI) または VSH スクリプトを使用して EEM ポリシーを設定できます。

EEM からデバイス全体のポリシー管理ビューが得られます。EEM ポリシーが設定されると、対応するアクションがトリガーされます。トリガーされたイベントのすべてのアクション（システムまたはユーザー設定）がシステムによって追跡され、管理されます。

設定済みのシステム ポリシー

Cisco NX-OS には、設定済みのさまざまなシステム ポリシーがあります。これらのシステムポリシーでは、デバイスに関連する多数の一般的なイベントおよびアクションが定義されています。システム ポリシー名は、2 個の下線記号 (__) から始まります。

一部のシステムポリシーは上書きできます。このような場合、イベントまたはアクションに対する上書きを設定できます。設定した上書き変更がシステム ポリシーの代わりになります。



(注) 上書きポリシーにはイベント文を含める必要があります。イベント文が含まれていない上書きポリシーは、システム ポリシーで想定されるすべてのイベントを上書きします。

設定済みのシステムポリシーを表示し、上書きできるポリシーを決定するには、**show event manager system-policy** コマンドを使用します。

ユーザー作成ポリシー

ユーザー作成ポリシーを使用すると、ネットワークのEEMポリシーをカスタマイズできます。ユーザー ポリシーがイベントに対して作成されると、ポリシーのアクションは、EEM が同じイベントに関連するシステムポリシーアクションをトリガーした後にのみトリガーされます。

ログ ファイル

EEM ポリシーの一致に関連するデータが格納されたログファイルは、`/log/event_archive_1` ディレクトリにある `event_archive_1` ログ ファイルで維持されます。

イベント文

対応策、通知など、一部のアクションが実行されるデバイス アクティビティは、EEM によってイベントと見なされます。イベントは通常、インターフェイスやファンの誤動作といったデバイスの障害に関連します。

イベント文は、どのイベントがポリシー実行のトリガーになるかを指定します。



ヒント ポリシー内に複数の EEM イベントを作成し、区別してから、カスタムアクションをトリガーするためのイベントの組み合わせを定義することで、イベントの組み合わせに基づいた EEM ポリシーをトリガーするように EEM を設定できます。

EEM ではイベント フィルタを定義して、クリティカル イベントまたは指定された時間内で繰り返し発生したイベントだけが関連付けられたアクションのトリガーになるようにします。

一部のコマンドまたは内部イベントが他のコマンドを内部的にトリガーします。これらのコマンドは表示されませんが、引き続きアクションをトリガーするイベント指定と一致します。これらのコマンドがアクションをトリガーするのを防ぐことはできませんが、どのイベントがアクションを引き起こしたかを確認できます。

サポートされるイベント

EEM はイベント文で次のイベントをサポートします。

- カウンタ イベント
- ファン欠損イベント
- ファン不良イベント
- メモリしきい値イベント
- 上書きされたシステム ポリシーで使用するイベント
- SNMP 通知イベント
- syslog イベント
- システム マネージャ イベント

- 温度イベント
- 追跡イベント

アクション文

アクション文は、イベントが発生したときに、ポリシーによってトリガーされるアクションを説明します。各ポリシーに複数のアクション文を設定できます。ポリシーにアクションを関連付けなかった場合、EEM はイベント観察を続けますが、アクションは実行されません。

トリガーされたイベントがデフォルト アクションを処理するために、デフォルト アクションを許可する EEM ポリシーを設定する必要があります。たとえば、一致文で CLI コマンドを照合する場合、EEM ポリシーに `event-default` アクション文を追加する必要があります。この文がないと、EEM ではコマンドを実行できません。



(注) ユーザー ポリシーまたは上書きポリシー内のアクション文を設定する場合、アクション文が、相互に否定したり、関連付けられたシステムポリシーに悪影響を与えるようなことがないように確認することが重要です。

サポートされるアクション

EEM がアクション文でサポートするアクションは、次のとおりです。

- CLI コマンドの実行
- カウンタのアップデート
- デバイスのリロード
- syslog メッセージの生成
- SNMP 通知の生成
- システム ポリシー用デフォルト アクションの使用

VSH スクリプト ポリシー

テキスト エディタを使用して、VSH スクリプトでポリシーを作成できます。VSH スクリプトを使用して作成されたポリシーには、他のポリシーと同様にイベント文とアクション文が含まれます。また、これらのポリシーはシステムポリシーを拡張するか、または無効にすることができます。

VSH スクリプト ポリシーを定義したら、それをデバイスにコピーしてアクティブにします。

Embedded Event Manager の前提条件

EEM を設定するには、network-admin の権限が必要です。

Embedded Event Manager の注意事項および制約事項

EEM の設定を計画するときは、次の点を考慮します。

- 設定可能な EEM ポリシーの最大数は 500 です。
- ユーザポリシーまたは上書きポリシー内のアクション文が、相互に否定したり、関連付けられたシステムポリシーに悪影響を与えたりすることがないようにする必要があります。
- 発生したイベントでデフォルトのアクションを処理できるようにするには、デフォルトのアクションを許可する EEM ポリシーを設定する必要があります。たとえば、一致文でコマンドを照合する場合、EEM ポリシーに event-default アクション文を追加する必要があります。この文がないと、EEM ではコマンドを実行できません。
- イベント ログの自動収集とバックアップには、次の注意事項があります。
 - デフォルトでは、スイッチのログ収集を有効にすると、サイズ、規模、コンポーネントのアクティビティに応じて、15分から数時間のイベントログが利用できるようになります。
 - 長期間にわたる関連ログを収集できるようにするには、必要な特定のサービス/機能に対してのみイベントログの保持を有効にします。「単一サービスの拡張ログファイル保持の有効化」を参照してください。内部イベントログをエクスポートすることもできます。「外部ログ ファイルストレージ」を参照してください。
 - トラブルシューティングを行うときは、内部イベントログのスナップショットを手動によりリアルタイムで収集することをお勧めします。「最近のログファイルのローカル コピーの生成」を参照してください。
- イベント文が指定されていて、アクション文が指定されていない上書きポリシーを設定した場合、アクションは開始されません。また、障害も通知されません。
- 上書きポリシーにイベント文が含まれていないと、システムポリシーで可能性のあるイベントがすべて上書きされます。
- 通常コマンドの表現の場合：すべてのキーワードを拡張する必要があり、アスタリスク (*) 記号のみが引数の置換に使用できます。
- EEM イベント相関は 1 つのポリシーに最大 4 つのイベント文をサポートします。イベントタイプは同じでも別でもかまいませんが、サポートされるイベントタイプは、cli、カウンタ、snmp、syslog、追跡だけです。

- 複数のイベント文が EEM ポリシーに存在する場合は、各イベント文に **tag** キーワードと一意な **tag** 引数が必要です。
- EEM イベント関連はシステムのデフォルト ポリシーを上書きしません。
- デフォルトアクション実行は、タグ付きのイベントで設定されているポリシーではサポートされません。
- イベント指定が CLI のパターンと一致する場合、SSH 形式のワイルドカード文字を使用できます。
たとえば、すべての **show** コマンドを照合する場合は、**show *** コマンドを入力します。
show . * コマンドを入力すると、機能しません。
- イベント指定が一致する syslog メッセージの正規表現の場合、適切な正規表現を使用できます。
たとえば、syslog が生成されているポート上で **ADMIN_DOWN** イベントを検出するには、**.ADMIN_DOWN.** を使用します。**ADMIN_DOWN** コマンドを入力すると、機能しません。
- syslog のイベント指定では、regex は、EEM ポリシーのアクションとして生成される syslog メッセージと一致しません。
- EEM イベントが CLI の **show** コマンドと一致し、画面に表示するために（および EEM ポリシーによってブロックされないために）**show** コマンドの出力が必要な場合は、EEM ポリシーの最初のアクションに対して、**event-default** コマンドを指定する必要があります。
- Cisco Nexus 3500 シリーズ スイッチは、Cisco NX-OS リリース 7.0(3)I7(2) およびそれ以前のリリースの Embedded Event Manager をサポートしていません。

Embedded Event Manager のデフォルト設定

表 1: デフォルトの EEM パラメータ

パラメータ	デフォルト
システム ポリシー	アクティブ

環境変数の定義

環境変数の定義はオプションの手順ですが、複数のポリシーで繰り返し使用する共通の値を設定する場合に役立ちます。

手順の概要

1. configure terminal

2. **event manager environment** *variable-name variable-value*
3. (任意) **show event manager environment** {*variable-name* | **all**}
4. (任意) **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例 : <pre>switch# configure terminal switch(config)#</pre>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	event manager environment <i>variable-name variable-value</i> 例 : <pre>switch(config) # event manager environment emailto "admin@anyplace.com"</pre>	EEM 用の環境変数を作成します。 <i>variable-name</i> は大文字と小文字を区別し、最大 29 文字の英数字を使用できます。 <i>variable-value</i> は大文字と小文字が区別され、引用符で囲んだ最大 39 文字の英数字を使用できます。
ステップ 3	(任意) show event manager environment { <i>variable-name</i> all }	設定した環境変数に関する情報を表示します。
ステップ 4	(任意) copy running-config startup-config 例 : <pre>switch(config)# copy running-config startup-config</pre>	リブートおよびリスタート時に実行コンフィギュレーションをスタートアップコンフィギュレーションにコピーして、変更を継続的に保存します。

次のタスク

ユーザ ポリシーを設定します。

CLI によるユーザ ポリシーの定義

手順の概要

1. **configure terminal**
2. **event manager applet** *applet-name*
3. (任意) **description** *policy-description*
4. **event** *event-statement*

5. (任意) **tag** *tag* {**and** | **andnot** | **or**} *tag* [**and** | **andnot** | **or** {*tag*}] { **happens occurs in seconds**}
6. **action** *number*[*.number2*] *action-statement*
7. (任意) **show event manager policy-state** *name* [**module** *module-id*]
8. (任意) **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例： switch# configure terminal switch(config)#	グローバル コンフィギュレーション モードを開始します。
ステップ 2	event manager applet <i>applet-name</i> 例： switch(config)# event manager applet monitorShutdown switch(config-applet)#	EEM にアプレットを登録し、アプレット コンフィギュレーション モードを開始します。 applet-name は大文字と小文字を区別し、最大 29 文字の英数字を使用できます。
ステップ 3	(任意) description <i>policy-description</i> 例： switch(config-applet)# description "Monitors interface shutdown."	ポリシーの説明になるストリングを設定します。 string には最大 80 文字の英数字を使用できます。ストリングは引用符で囲みます。
ステップ 4	event <i>event-statement</i> 例： switch(config-applet)# event cli match "shutdown"	ポリシーのイベント文を設定します。
ステップ 5	(任意) tag <i>tag</i> { and andnot or } <i>tag</i> [and andnot or { <i>tag</i> }] { happens occurs in seconds }	ポリシー内の複数のイベントを相互に関連付けます。 occurs 引数の範囲は 1 ～ 4294967295 です。 seconds 引数の範囲は 0 ～ 4294967295 秒です。
ステップ 6	action <i>number</i> [<i>.number2</i>] <i>action-statement</i> 例： switch(config-applet)# action 1.0 cli show interface e 3/1	ポリシーのアクション文を設定します。アクション文が複数ある場合、このステップを繰り返します。
ステップ 7	(任意) show event manager policy-state <i>name</i> [module <i>module-id</i>] 例： switch(config-applet)# show event manager policy-state monitorShutdown	設定したポリシーの状態に関する情報を表示します。

	コマンドまたはアクション	目的
ステップ 8	(任意) copy running-config startup-config 例 : <pre>switch(config)# copy running-config startup-config</pre>	リブートおよびリスタート時に実行コンフィギュレーションをスタートアップコンフィギュレーションにコピーして、変更を継続的に保存します。

次のタスク

イベント文およびアクション文を設定します。

イベント文の設定

イベント文を設定するには、EEM コンフィギュレーションモード (config-applet) で次のいずれかのコマンドを使用します。

始める前に

ユーザー ポリシーを定義します。

手順の概要

1. **event cli** [tag tag] **match** expression [count repeats | time seconds
2. **event counter** [tag tag] **name** counter **entry-val** entry **entry-op** {eq | ge | gt | le | lt | ne} { **exit-val** exit **exit-op** {eq | ge | gt | le | lt | ne}
3. **event fanabsent** [fan number] **time** seconds
4. **event fanbad** [fan number] **time** seconds
5. **event memory** {critical | minor | severe}
6. **event policy-default** count repeats [time seconds]
7. **event snmp** [tag tag] **oid** oid **get-type** {exact | next} **entry-op** {eq | ge | gt | le | lt | ne} **entry-val** entry [exit-comb {and | or}] **exit-op** {eq | ge | gt | le | lt | ne} **exit-val** exit **exit-time** time **polling-interval** interval
8. **event sysmgr memory** [module module-num] **major** major-percent **minor** minor-percent **clear** clear-percent
9. **event temperature** [module slot] [sensor number] **threshold** {any | down | up}
10. **event track** [tag tag] **object-number** state {any | down | up

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	event cli [tag tag] match expression [count repeats time seconds] 例 : <pre>switch(config-applet) # event cli match "shutdown"</pre>	正規表現と一致するコマンドが入力された場合に、イベントを発生させます。 tag tag キーワードと引数のペアは、複数のイベントがポリシーに含まれている場合、この特定のイベントを識別します。 repeats の範囲は 1 ～ 65000 です。 time の範囲は 0 ～ 4294967295 です。0 は無制限を示します。
ステップ 2	event counter [tag tag] name counter entry-val entry entry-op {eq ge gt le lt ne} { exit-val exit exit-op {eq ge gt le lt ne}} 例 : <pre>switch(config-applet) # event counter name mycounter entry-val 20 gt</pre>	カウンタが、開始演算子に基づいて開始のしきい値を超えた場合にイベントを発生させます。イベントはただちにリセットされます。任意で、カウンタが終了のしきい値を超えたあとでリセットされるように、イベントを設定できます。 tag tag キーワードと引数のペアは、複数のイベントがポリシーに含まれている場合、この特定のイベントを識別します。 counter name は大文字と小文字を区別し、最大 28 の英数字を使用できます。 entry および exit の値の範囲は 0 ～ 2147483647 です。
ステップ 3	event fanabsent [fan number] time seconds 例 : <pre>switch(config-applet) # event fanabsent time 300</pre>	秒数で設定された時間を超えて、ファンがデバイスから取り外されている場合に、イベントを発生させます。 number の範囲はモジュールに依存します。 seconds の範囲は 10 ～ 64000 です。
ステップ 4	event fanbad [fan number] time seconds 例 : <pre>switch(config-applet) # event fanbad time 3000</pre>	秒数で設定された時間を超えて、ファンが故障状態の場合に、イベントを発生させます。 number の範囲はモジュールに依存します。 seconds の範囲は 10 ～ 64000 です。
ステップ 5	event memory {critical minor severe} 例 : <pre>switch(config-applet) # event memory critical</pre>	メモリのしきい値を超えた場合にイベントを発生させます。

	コマンドまたはアクション	目的
ステップ 6	event policy-default count repeats [time seconds] 例 : <pre>switch(config-applet) # event policy-default count 3</pre>	システム ポリシーで設定されているイベントを使用します。このオプションは、ポリシーを上書きする場合に使用します。 <i>repeats</i> の範囲は 1 ～ 65000 です。 <i>seconds</i> の範囲は 0 ～ 4294967295 秒です。0 は無制限を示します。
ステップ 7	event snmp [tag tag] oid oid get-type {exact next} entry-op {eq ge gt le lt ne} entry-val entry [exit-comb {and or}] exit-op {eq ge gt le lt ne} exit-val exit exit-time time polling-interval interval 例 : <pre>switch(config-applet) # event snmp oid 1.3.6.1.2.1.31.1.1.1.6 get-type next entry-op lt 300 entry-val 0 exit-op eq 400 exit-time 30 polling-interval 300</pre>	SNMP OID が、開始演算子に基づいて開始のしきい値を超えた場合にイベントを発生させます。イベントはただちにリセットされます。または任意で、カウンタが終了のしきい値を超えたあとでリセットされるように、イベントを設定できます。OID はドット付き 10 進表記です。 tag tag キーワードと引数のペアは、複数のイベントがポリシーに含まれている場合、この特定のイベントを識別します。 <i>entry</i> および <i>exit</i> の値の範囲は 0 ～ 18446744073709551615 です。 <i>time</i> の範囲は 0 ～ 2147483647 秒です。 <i>interval</i> の範囲は 0 ～ 2147483647 秒です。
ステップ 8	event sysmgr memory [module module-num] major major-percent minor minor-percent clear clear-percent 例 : <pre>switch(config-applet) # event sysmgr memory minor 80</pre>	指定したシステム マネージャのメモリのしきい値を超えた場合にイベントを発生させます。 <i>percent</i> の範囲は 1 ～ 99 です。
ステップ 9	event temperature [module slot] [sensor number] threshold {any down up} 例 : <pre>switch(config-applet) # event temperature module 2 threshold any</pre>	温度センサーが設定されたしきい値を超えた場合に、イベントを発生させます。 <i>sensor</i> の範囲は 1 ～ 18 です。
ステップ 10	event track [tag tag] object-number state {any down up} 例 : <pre>switch(config-applet) # event track 1 state down</pre>	トラッキング対象オブジェクトが設定された状態になった場合に、イベントを発生させます。 tag tag キーワードと引数のペアは、複数のイベントがポリシーに含まれている場合、この特定のイベントを識別します。 指定できる <i>object-number</i> の範囲は 1 ～ 500 です。

次のタスク

アクション文を設定します。

すでにアクション文を設定した場合、または設定しないことを選択した場合は、次のオプション作業のいずれかを実行します。

- VSH スクリプトを使用してポリシーを定義します。その後、VSH スクリプト ポリシーを登録し、アクティブにします。
- メモリのしきい値を設定します。
- EEM パブリッシャとして syslog を設定します。
- EEM 設定を確認します。

アクション文の設定

EEM のコンフィギュレーション モード (config-applet) で次のいずれかのコマンドを使用して、アクションを設定できます。



(注) 発生したイベントでデフォルトのアクションを処理できるようにする場合は、デフォルトのアクションを許可する EEM ポリシーを設定する必要があります。たとえば、一致文でコマンドを照合する場合、EEM ポリシーに **event-default** アクション文を追加する必要があります。この文がないと、EEM ではコマンドを実行できません。 **terminal event-manager bypass** コマンドを使用すると、一致するすべての EEM ポリシーでコマンドを実行できます。

始める前に

ユーザー ポリシーを定義します。

手順の概要

1. **action** *number*[.*number2*] **cli** *command1*[*command2*.] [**local**]
2. **action** *number*[.*number2*] **counter** *name* *counter* *value* *val* **op** {**dec** | **inc** | **nop** | **set**}
3. **action** *number*[.*number2*] **event-default**
4. **action** *number*[.*number2*] **policy-default**
5. **action** *number*[.*number2*] **reload** [**module** *slot* [- *slot*]]
6. **action** *number*[.*number2*] **snmp-trap** [**intdata1** *integer-data1*] [**intdata2** *integer-data2*] [**strdata** *string-data*]
7. **action** *number*[.*number2*] **syslog** [**priority** *prio-val*] **msg** *error-message*

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	action <i>number</i> [. <i>number2</i>] cli <i>command1</i> [<i>command2</i> .] [local] 例 : <pre>switch(config-applet) # action 1.0 cli "show interface e 3/1"</pre>	設定済みコマンドを実行します。任意で、イベントが発生したモジュール上でコマンドを実行できます。 アクションラベルのフォーマットは <i>number1.number2</i> です。 <i>number</i> には 1 ～ 16 桁の任意の番号を指定できます。 <i>number2</i> の範囲は 0 ～ 9 です。
ステップ 2	action <i>number</i> [. <i>number2</i>] counter <i>name</i> <i>counter</i> value <i>val</i> op { dec inc nop set } 例 : <pre>switch(config-applet) # action 2.0 counter name mycounter value 20 op inc</pre>	設定された値および操作でカウンタを変更します。 アクションラベルのフォーマットは <i>number1.number2</i> です。 <i>number</i> には 1 ～ 16 桁の任意の番号を指定できます。 <i>number2</i> の範囲は 0 ～ 9 です。 <i>counter</i> は大文字と小文字を区別し、最大 28 文字の英数字を使用できます。 <i>val</i> には 0 ～ 2147483647 の整数または置換パラメータを指定できます。
ステップ 3	action <i>number</i> [. <i>number2</i>] event-default 例 : <pre>switch(config-applet) # action 1.0 event-default</pre>	関連付けられたイベントのデフォルトアクションを実行します。 アクションラベルのフォーマットは <i>number1.number2</i> です。 <i>number</i> には 1 ～ 16 桁の任意の番号を指定できます。 <i>number2</i> の範囲は 0 ～ 9 です。
ステップ 4	action <i>number</i> [. <i>number2</i>] policy-default 例 : <pre>switch(config-applet) # action 1.0 policy-default</pre>	上書きしているポリシーのデフォルトアクションを実行します。 アクションラベルのフォーマットは <i>number1.number2</i> です。 <i>number</i> には 1 ～ 16 桁の任意の番号を指定できます。 <i>number2</i> の範囲は 0 ～ 9 です。
ステップ 5	action <i>number</i> [. <i>number2</i>] reload [module <i>slot</i> [- <i>slot</i>]] 例 :	システム全体に 1 つ以上のモジュールをリロードします。

	コマンドまたはアクション	目的
	switch(config-applet) # action 1.0 reload module 3-5	アクションラベルのフォーマットは <code>number1.number2</code> です。 <i>number</i> には 1 ～ 16 桁の任意の番号を指定できます。 <i>number2</i> の範囲は 0 ～ 9 です。
ステップ 6	action <i>number</i>[<i>number2</i>] snmp-trap [<i>intdata1</i> <i>integer-data1</i>] [<i>intdata2</i> <i>integer-data2</i>] [<i>strdata</i> <i>string-data</i>] 例 : switch(config-applet) # action 1.0 snmp-trap strdata "temperature problem"	設定されたデータを使用して SNMP トラップを送信します。アクションラベルのフォーマットは <code>number1.number2</code> です。 <i>number</i> には 1 ～ 16 桁の任意の番号を指定できます。 <i>number2</i> の範囲は 0 ～ 9 です。 <i>data</i> 要素には 80 桁までの任意の数を指定できます。 <i>string</i> には最大 80 文字の英数字を使用できます。
ステップ 7	action <i>number</i>[<i>number2</i>] syslog [<i>priority prio-val</i>] msg <i>error-message</i> 例 : switch(config-applet) # action 1.0 syslog priority notifications msg "cpu high"	設定されたプライオリティで、カスタマイズした syslog メッセージを送信します。 アクションラベルのフォーマットは <code>number1.number2</code> です。 <i>number</i> には 1 ～ 16 桁の任意の番号を指定できます。 <i>number2</i> の範囲は 0 ～ 9 です。 <i>error-message</i> には最大 80 文字の英数字を引用符で囲んで使用できます。

次のタスク

イベント文を設定します。

すでにイベント文を設定した場合、または設定しないことを選択した場合は、次のオプション作業のいずれかを実行します。

- VSH スクリプトを使用してポリシーを定義します。その後、VSH スクリプト ポリシーを登録し、アクティブにします。
- メモリのしきい値を設定します。
- EEM パブリッシャとして syslog を設定します。
- EEM 設定を確認します。

VSH スクリプトによるポリシーの定義

これはオプションのタスクです。VSH スクリプトを使用して EEM ポリシーを記述する場合は、次の手順を実行します。

手順の概要

1. テキスト エディタで、ポリシーを定義するコマンドリストを指定します。
2. テキスト ファイルに名前をつけて保存します。
3. 次のシステム ディレクトリにファイルをコピーします。bootflash://eem/user_script_policies

手順の詳細

手順

ステップ 1 テキスト エディタで、ポリシーを定義するコマンドリストを指定します。

ステップ 2 テキスト ファイルに名前をつけて保存します。

ステップ 3 次のシステム ディレクトリにファイルをコピーします。bootflash://eem/user_script_policies

次のタスク

VSH スクリプト ポリシーを登録してアクティブにします。

VSH スクリプト ポリシーの登録およびアクティブ化

これはオプションのタスクです。VSH スクリプトを使用して EEM ポリシーを記述する場合は、次の手順を実行します。

始める前に

ポリシーを VSH スクリプトを使用して定義し、システム ディレクトリにファイルをコピーします。

手順の概要

1. **configure terminal**
2. **event manager policy policy-script**
3. (任意) **event manager policy internal name**
4. (任意) **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例： switch# configure terminal switch(config)#	グローバル コンフィギュレーション モードを開始します。
ステップ 2	event manager policy <i>policy-script</i> 例： switch(config)# event manager policy moduleScript	EEM スクリプト ポリシーを登録してアクティブにします。 <i>policy-script</i> は大文字と小文字を区別し、最大 29 文字の英数字を使用できます。
ステップ 3	(任意) event manager policy <i>internal name</i> 例： switch(config)# event manager policy internal moduleScript	EEM スクリプト ポリシーを登録してアクティブにします。 <i>policy-script</i> は大文字と小文字を区別し、最大 29 の英数字を使用できます。
ステップ 4	(任意) copy running-config startup-config 例： switch(config)# copy running-config startup-config	リブートおよびリスタート時に実行コンフィギュレーションをスタートアップコンフィギュレーションにコピーして、変更を継続的に保存します。

次のタスク

システム要件に応じて、次のいずれかを実行します。

- メモリのしきい値を設定します。
- EEM パブリッシャとして syslog を設定します。
- EEM 設定を確認します。

システム ポリシーの上書き

手順の概要

1. **configure terminal**
2. (任意) **show event manager policy-state *system-policy***
3. **event manager applet *applet-name* override *system-policy***
4. **description *policy-description***
5. **event *event-statement***

6. **section number action-statement**
7. (任意) **show event manager policy-state name**
8. (任意) **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例 : <pre>switch# configure terminal switch(config)#</pre>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	(任意) show event manager policy-state system-policy 例 : <pre>switch(config-applet)# show event manager policy-state __ethpm_link_flap Policy __ethpm_link_flap Cfg count : 5 Cfg time interval : 10.000000 (seconds) Hash default, Count 0</pre>	上書きするシステムポリシーの情報をしきい値を含めて表示します。 show event manager system-policy コマンドを使用して、システムポリシーの名前を探します。
ステップ 3	event manager applet applet-name override system-policy 例 : <pre>switch(config-applet)# event manager applet ethport override __ethpm_link_flap switch(config-applet)#</pre>	システムポリシーを上書きし、アプレットコンフィギュレーション モードを開始します。 applet-name は大文字と小文字を区別し、最大 80 文字の英数字を使用できます。 system-policy は、システム ポリシーの 1 つにする必要があります。
ステップ 4	description policy-description 例 : <pre>switch(config-applet)# description "Overrides link flap policy"</pre>	ポリシーの説明になるストリングを設定します。 policy-description は大文字と小文字を区別し、最大 80 文字の英数字を使用できますが、引用符で囲む必要があります。
ステップ 5	event event-statement 例 : <pre>switch(config-applet)# event policy-default count 2 time 1000</pre>	ポリシーのイベント文を設定します。
ステップ 6	section number action-statement 例 : <pre>switch(config-applet)# action 1.0 syslog priority warnings msg "Link is flapping."</pre>	ポリシーのアクション文を設定します。複数のアクション文では、この手順を繰り返します。

	コマンドまたはアクション	目的
ステップ 7	(任意) show event manager policy-state name 例 : <pre>switch(config-applet)# show event manager policy-state ethport</pre>	設定したポリシーに関する情報を表示します。
ステップ 8	(任意) copy running-config startup-config 例 : <pre>switch(config)# copy running-config startup-config</pre>	リブートおよびリスタート時に実行コンフィギュレーションをスタートアップコンフィギュレーションにコピーして、変更を継続的に保存します。

EEM パブリッシャとしての syslog の設定

EEM パブリッシャとして syslog を設定すると、スイッチから syslog メッセージをモニターできます。



(注) syslog メッセージをモニターする検索文字列の最大数は 10 です。

始める前に

- EEM が syslog による登録で利用できることを確認します。
- syslog デーモンが設定され、実行されていることを確認します。

手順の概要

1. **configure terminal**
2. **event manager applet applet-name**
3. **event syslog [tag tag] { occurs number | period seconds | pattern msg-text | priority priority }**
4. (任意) **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例 : <pre>switch# configure terminal switch(config)#</pre>	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 2	event manager applet <i>applet-name</i> 例 : <pre>switch(config)# event manager applet abc switch (config-applet)#</pre>	EEM にアプレットを登録し、アプレット コンフィギュレーション モードを開始します。
ステップ 3	event syslog [tag <i>tag</i>] { occurs <i>number</i> period <i>seconds</i> pattern <i>msg-text</i> priority <i>priority</i> } 例 : <pre>switch(config-applet)# event syslog occurs 10</pre>	EEM にアプレットを登録し、アプレット コンフィギュレーション モードを開始します。
ステップ 4	(任意) copy running-config startup-config 例 : <pre>switch(config)# copy running-config startup-config</pre>	リブートおよびリスタート時に実行コンフィギュレーションをスタートアップコンフィギュレーションにコピーして、変更を継続的に保存します。

次のタスク

EEM 設定を確認します。

イベント ログの自動収集とバックアップ

自動的に収集されたイベントログは、スイッチのメモリにローカルに保存されます。イベントログ ファイル ストレージは、一定期間ファイルを保存する一時バッファです。時間が経過すると、バッファのロールオーバーによって次のファイルのためのスペースが確保されます。ロールオーバーでは、先入れ先出し方式が使用されます。

Cisco NX-OS リリース 9.3(3) 以降、EEM は以下の収集およびバックアップ方法を使用します。

- 拡張ログ ファイルの保持
- トリガーベースのイベント ログの自動収集

拡張ログ ファイルの保持

Cisco NX-OS リリース 9.3 (3) 以降、すべての Cisco Nexus プラットフォーム スイッチは、少なくとも 8 GB のシステムメモリを備え、イベントロギングファイルの拡張保持をサポートします。ログファイルをスイッチにローカルに保存するか、外部コンテナを介してリモートに保存すると、ロールオーバーによるイベント ログの損失を削減できます。

すべてのサービスの拡張ログ ファイル保持のイネーブル化

拡張ログファイル保持は、スイッチで実行されているすべてのサービスに対してデフォルトで有効になっています。スイッチでログファイル保持機能がイネーブルになっていない場合（**no bloggerd log-dump** が設定されている場合）、次の手順を使用してイネーブルにします。

すべてのサービスの拡張ログ ファイル保持の無効化

手順の概要

1. **configure terminal**
2. **bloggerd log-dump all**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例 : <pre>switch# configure terminal switch(config)#</pre>	グローバル コンフィギュレーション モードを開始します
ステップ 2	bloggerd log-dump all 例 : <pre>switch(config)# bloggerd log-dump all switch(config)#</pre>	すべてのサービスのログ ファイル保持機能をイネーブルにします。

例

```
switch# configure terminal
switch(config)# bloggerd log-dump all
Sending Enable Request to Bloggerd
Bloggerd Log Dump Successfully enabled
switch(config)#
```

すべてのサービスの拡張ログ ファイル保持の無効化

拡張ログ ファイル保持は、スイッチ上のすべてのサービスに対してデフォルトで無効になっています。スイッチのログ ファイル保持機能がすべてのサービスに対して有効になっている場合は、次の手順を実行します。

手順の概要

1. **configure terminal**
2. **no bloggerd log-dump all**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例 : <pre>switch# configure terminal switch(config)#</pre>	グローバル コンフィギュレーション モードを開始します
ステップ 2	no bloggerd log-dump all 例 : <pre>switch(config)# no bloggerd log-dump all switch(config)#</pre>	スイッチ上のすべてのサービスのログファイル保持機能を無効にします。

例

```
switch# configure terminal
switch(config)# no bloggerd log-dump all
Sending Disable Request to Bloggerd
Bloggerd Log Dump Successfully disabled
switch(config)#
```

単一サービスの拡張ログファイル保持の有効化

拡張ログファイル保持は、スイッチで実行されているすべてのサービスに対してデフォルトで有効になっています。スイッチで（**no bloggerd log-dump**が設定されていて）ログファイル保持機能が有効になっていない場合、次の手順を使用して単一のサービスに対して有効にします。

手順の概要

1. **show system internal sysmgr service name service-type**
2. **configure terminal**
3. **bloggerd log-dump sap number**
4. **show system internal bloggerd info log-dump-info**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	show system internal sysmgr service name service-type 例 :	サービス SA P番号を含む ACL Manager に関する情報を表示します。

	コマンドまたはアクション	目的
	switch# show system internal sysmgr service name aclmgr	
ステップ 2	configure terminal 例 : switch# configure terminal switch(config)#	グローバル コンフィギュレーション モードを開始 します。
ステップ 3	bloggerd log-dump sap number 例 : switch(config)# bloggerd log-dump sap 351	ACL Manager サービスのログファイル保持機能をイ ネーブルにします。
ステップ 4	show system internal bloggerd info log-dump-info 例 : switch(config)# show system internal bloggerd info log-dump-info	スイッチ上のログファイル保持機能に関する情報を 表示します。

例

```

switch# show system internal sysmgr service name aclmgr
Service "aclmgr" ("aclmgr", 80):
  UUID = 0x182, PID = 653, SAP = 351
  State: SRV_STATE_HANDSHAKED (entered at time Mon Nov  4 11:10:41 2019).
  Restart count: 1
  Time of last restart: Mon Nov  4 11:10:39 2019.
  The service never crashed since the last reboot.
  Tag = N/A
  Plugin ID: 0

switch(config)# configure terminal
switch(config)# bloggerd log-dump sap 351
Sending Enable Request to Bloggerd
Bloggerd Log Dump Successfully enabled
switch(config)# show system internal bloggerd info log-dump-info
-----
Log Dump config is READY
Log Dump is DISABLED for ALL application services in the switch
Exceptions to the above rule (if any) are as follows:
-----
Module      | VDC      | SAP                      | Enabled?
-----
1           | 1        | 351 (MTS_SAP_ACLMGR     ) | Enabled
-----

Log Dump Throttle Switch-Wide Config:
-----
Log Dump Throttle                               : ENABLED
Minimum buffer rollover count (before throttling) : 5
Maximum allowed rollover count per minute         : 1
-----

switch(config)#

```

拡張ログ ファイルの表示

スイッチに現在保存されているイベント ログ ファイルを表示するには、次の作業を実行します。

手順の概要

1. `dir debug:log-dump/`

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	<code>dir debug:log-dump/</code> 例 : <code>switch# dir debug:log-dump/</code>	スイッチに現在保存されているイベント ログ ファイルを表示します。

例

```
switch# dir debug:log-dump/

3676160 Dec 05 02:43:01 2019 20191205023755_evtlog_archive.tar
3553280 Dec 05 06:05:06 2019 20191205060005_evtlog_archive.tar

Usage for debug://sup-local
913408 bytes used
4329472 bytes free
5242880 bytes total
```

単一サービスに対する拡張ログファイル保持の無効化

拡張ログファイル保持は、スイッチ上のすべてのサービスに対してデフォルトで有効になっています。スイッチで単一またはすべてのサービス（Cisco NX-OSリリース9.3(5)ではデフォルト）に対してログファイル保持機能が有効になっている場合に、特定のサービスを無効にするには、次の手順を実行します。

手順の概要

1. `show system internal sysmgr service name service-type`
2. `configure terminal`
3. `no bloggerd log-dump sap number`
4. `show system internal bloggerd info log-dump-info`

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	show system internal sysmgr service name service-type 例 : <pre>switch# show system internal sysmgr service name aclmgr</pre>	サービス SA P 番号を含む ACL Manager に関する情報を表示します。
ステップ 2	configure terminal 例 : <pre>switch# configure terminal switch(config)#</pre>	グローバル コンフィギュレーション モードを開始します。
ステップ 3	no bloggerd log-dump sap number 例 : <pre>switch(config)# no bloggerd log-dump sap 351</pre>	ACL Manager サービスのログファイル保持機能を無効にします。
ステップ 4	show system internal bloggerd info log-dump-info 例 : <pre>switch(config)# show system internal bloggerd info log-dump-info</pre>	スイッチ上のログファイル保持機能に関する情報を表示します。

例

次に、「aclmgr」という名前のサービスの拡張ログファイル保持を無効にする例を示します。

```
switch# show system internal sysmgr service name aclmgr
Service "aclmgr" ("aclmgr", 80):
  UUID = 0x182, PID = 653, SAP = 351
  State: SRV_STATE_HANDSHAKED (entered at time Mon Nov  4 11:10:41 2019).
  Restart count: 1
  Time of last restart: Mon Nov  4 11:10:39 2019.
  The service never crashed since the last reboot.
  Tag = N/A
  Plugin ID: 0
switch(config)# configure terminal
switch(config)# no bloggerd log-dump sap 351
Sending Disable Request to Bloggerd
Bloggerd Log Dump Successfully disabled
switch(config)# show system internal bloggerd info log-dump-info
-----
Log Dump config is READY
Log Dump is DISABLED for ALL application services in the switch
Exceptions to the above rule (if any) are as follows:
-----
Module      | VDC      | SAP              | Enabled?
-----
1           | 1        | 351 (MTS_SAP_ACLMGR) | Disabled
```



```

-----
Log Dump Throttle Switch-Wide Config:
-----
Log Dump Throttle                               : ENABLED
Minimum buffer rollover count (before throttling) : 5
Maximum allowed rollover count per minute         : 1
-----

switch(config)#

```

トリガーベースのイベント ログの自動収集

トリガーベースのログ収集機能：

- 問題発生時に関連データを自動的に収集します。
- コントロール プレーンへの影響なし
- カスタマイズ可能な設定ですか：
 - シスコが入力するデフォルト
 - 収集対象は、ネットワーク管理者または Cisco TACによって、選択的に上書きされます。
 - イメージのアップグレード時は新しいトリガーを自動的に更新します。
- ログをスイッチにローカルに保存するか、外部サーバにリモートで保存します。
- 重大度 0、1、および 2 の syslog をサポートします：
- アドホック イベントのカスタム syslog (syslog と接続する自動収集コマンド)

トリガーベースのログ ファイルの自動収集の有効化

ログ ファイルのトリガーベースの自動作成を有効にするには、`__syslog_trigger_default` システム ポリシーのオーバーライドポリシーをカスタム YAML ファイルで作成し、情報を収集する特定のログを定義する必要があります。

ログ ファイルの自動収集を有効にするカスタム YAML ファイルの作成の詳細については、[自動収集 YAML ファイルの設定 \(26 ページ\)](#) を参照してください。

自動収集 YAML ファイル

EEM 機能の **action** コマンドで指定される自動収集 YAML ファイルは、さまざまなシステムまたは機能コンポーネントのアクションを定義します。このファイルは、スイッチ ディレクトリ `/bootflash/scripts` にあります。デフォルトの YAML ファイルに加えて、コンポーネント固有の YAML ファイルを作成し、同じディレクトリに配置できます。コンポーネント固有の YAML ファイルの命名規則は **component-name.yaml** です。コンポーネント固有のファイルが同じディレクトリに存在する場合は、**action** コマンドで指定されたファイルよりも優先されます。たとえば、アクションファイル **bootflash/scripts/platform.yaml** がデフォルトのアクション

ンファイル `/bootflash/scripts` とともに `bootflash/scripts/test.yaml` ディレクトリにある場合、`platform.yaml` ファイルで定義された命令がデフォルトの `test.yaml` ファイルに存在するプラットフォーム コンポーネントの手順よりも優先します。

コンポーネントの例としては、ARP、BGP、IS-IS などがあります。すべてのコンポーネント名に精通していない場合は、シスコ カスタマー サポートに連絡して、コンポーネント固有のアクション（およびデフォルトの `test.yaml` ファイル）の YAML ファイルを定義してください。

例：

```
event manager applet test_1 override __syslog_trigger_default
  action 1.0 collect test.yaml $_syslog_msg
```

自動収集 YAML ファイルの設定

YAML ファイルの内容によって、トリガーベースの自動収集時に収集されるデータが決まります。スイッチには YAML ファイルが 1 つだけ存在しますが、任意の数のスイッチ コンポーネントとメッセージの自動収集メタデータを含めることができます。

スイッチの次のディレクトリで YAML ファイルを見つけます。

```
/bootflash/scripts
```

次の例を使用して、トリガーベース収集の YAML ファイルを呼び出します。この例は、ユーザ定義の YAML ファイルを使用してトリガーベース収集を実行するために最低限必要な設定を示しています。

```
switch# show running-config eem
!Command: show running-config eem
!Running configuration last done at: Mon Sep 30 19:34:54 2019
!Time: Mon Sep 30 22:24:55 2019
version 9.3(3) Bios:version 07.59
event manager applet test_1 override __syslog_trigger_default
  action 1.0 collect test.yaml $_syslog_msg
```

上記の例では、「test_1」がアプレットの名前で、「test.yaml」が `/bootflash/scripts` ディレクトリにあるユーザ設定の YAML ファイルの名前です。

YAML ファイルの例

次に、トリガーベースのイベント ログ自動収集機能をサポートする基本的な YAML ファイルの例を示します。ファイル内のキー/値の定義を次の表に示します。



- (注) YMAL ファイルに適切なインデントがあることを確認します。ベストプラクティスとして、スイッチで使用する前に任意の「オンライン YAML 検証」を実行します。

```
bash-4.3$ cat /bootflash/scripts/test.yaml
version: 1
components:
  securityd:
    default:
      tech-sup: port
      commands: show module
  platform:
    default:
```

```
tech-sup: port
commands: show module
```

キー : 値	説明
バージョン : 1	1 に設定します。他の番号を使用すると、自動収集スクリプトに互換性がなくなります。
コンポーネント :	以下がスイッチ コンポーネントであることを指定するキーワード。
securityd :	syslog コンポーネントの名前 (securityd は syslog のファシリティ名)。
デフォルト :	コンポーネントに属するすべてのメッセージを識別します。
tech-sup : port	securityd syslog コンポーネントのポート モジュールのテクニカル サポートを収集します。
コマンド : show module	securityd syslog コンポーネントの show module コマンド出力を収集します。
プラットフォーム :	syslog コンポーネントの名前 (platform は syslog のファシリティ名)。
tech-sup : port	platform syslog コンポーネントのポート モジュールのテクニカル サポートを収集します。
コマンド : show module	platform syslog コンポーネントの show module コマンド出力を収集します。

特定のログにのみ自動収集メタデータを関連付けるには、次の例を使用します。たとえば、SECURITYD-2-FEATURE_ENABLE_DISABLE

```
securityd:
    feature_enable_disable:
        tech-sup: security
        commands: show module
```

キー : 値	説明
securityd :	syslog コンポーネントの名前 (securityd は syslog のファシリティ名)。
feature_enable_disable :	syslog メッセージのメッセージ ID。
tech-sup : security	securityd syslog コンポーネントのセキュリティモジュールのテクニカル サポートを収集します。
コマンド : show module	セキュリティ syslog コンポーネントの show module コマンド出力を収集します。

上記の YAML エントリの syslog 出力の例 :

```
2019 Dec 4 12:41:01 n9k-c93108tc-fx %SECURITYD-2-FEATURE_ENABLE_DISABLE: User
has enabled the feature bash-shell
```

複数の値を指定するには、次の例を使用します。

```
version: 1
components:
  securityd:
    default:
      commands: show module;show version;show module
      tech-sup: port;lldp
```



(注) 複数の show コマンドとテクニカルサポート キーの値を区切るには、セミコロンを使用します (前の例を参照)。

リリース 10.1(1) 以降では、test.yaml は複数の YAML ファイルが存在するフォルダに置き換えることができます。フォルダ内のすべての YAML ファイルは、ComponentName.yaml 命名規則に従う必要があります。

次の例では、test.yaml が test_folder に置き換えられます。

```
test.yaml:
event manager applet logging2 override __syslog_trigger_default
  action 1.0 collect test.yaml rate-limit 30 $_syslog_msg

test_folder:
event manager applet logging2 override __syslog_trigger_default
  action 1.0 collect test_folder rate-limit 30 $_syslog_msg
```

次の例は、test_folder のパスとコンポーネントを示しています。

```
ls /bootflash/scripts/test_folder
bgp.yaml ppm.yaml
```

コンポーネントあたりの自動収集の量の制限

自動収集の場合、コンポーネントイベントあたりのバンドル数の制限はデフォルトで3に設定されています。1つのコンポーネントで3つ以上のイベントが発生すると、イベントはドロップされ、ステータス メッセージ **EVENTLOGLIMITREACHED** が表示されます。イベント ログがロールオーバーすると、コンポーネント イベントの自動収集が再開されます。

例：

```
switch# show system internal event-logs auto-collect history
DateTime      Snapshot ID  Syslog                               Status/Secs/Logsize(Bytes)
2020-Jun-27 07:20:03 1140276903 ACLMGR-0-TEST_SYSLOG                EVENTLOGLIMITREACHED
2020-Jun-27 07:15:14 1026359228 ACLMGR-0-TEST_SYSLOG                RATELIMITED
2020-Jun-27 07:15:09 384952880   ACLMGR-0-TEST_SYSLOG                RATELIMITED
2020-Jun-27 07:13:55 1679333688 ACLMGR-0-TEST_SYSLOG                PROCESSED:2:9332278
2020-Jun-27 07:13:52 1679333688 ACLMGR-0-TEST_SYSLOG                PROCESSING
2020-Jun-27 07:12:55 502545693  ACLMGR-0-TEST_SYSLOG                RATELIMITED
2020-Jun-27 07:12:25 1718497217 ACLMGR-0-TEST_SYSLOG                RATELIMITED
2020-Jun-27 07:08:25 1432687513 ACLMGR-0-TEST_SYSLOG                PROCESSED:2:10453823
2020-Jun-27 07:08:22 1432687513 ACLMGR-0-TEST_SYSLOG                PROCESSING
2020-Jun-27 07:06:16 90042807   ACLMGR-0-TEST_SYSLOG                RATELIMITED
2020-Jun-27 07:03:26 1737578642 ACLMGR-0-TEST_SYSLOG                RATELIMITED
```

```

2020-Jun-27 07:02:56 40101277 ACLMGR-0-TEST_SYSLOG PROCESSED:3:10542045
2020-Jun-27 07:02:52 40101277 ACLMGR-0-TEST_SYSLOG PROCESSING

```

自動収集ログ ファイル

自動収集ログ ファイルについて

YAML ファイルの設定によって、自動収集ログ ファイルの内容が決まります。収集ログ ファイルで使用するメモリの量は設定できません。保存後のファイルが消去される頻度は設定できます。

自動収集ログ ファイルは、次のディレクトリに保存されます。

```

switch# dir bootflash:eem_snapshots
 44205843 Sep 25 11:08:04 2019
1480625546_SECURITYD_2_FEATURE_ENABLE_DISABLE_eem_snapshot.tar.gz
  Usage for bootflash://sup-local
 6940545024 bytes used
44829761536 bytes free
51770306560 bytes total

```

ログ ファイルへのアクセス

コマンドキーワード「debug」を使用してログを検索します。

```

switch# dir debug:///
...
 26 Oct 22 10:46:31 2019 log-dump
 24 Oct 22 10:46:31 2019 log-snapshot-auto
 26 Oct 22 10:46:31 2019 log-snapshot-user

```

次の表に、ログの場所と保存されるログの種類を示します。

場所	説明
log-dump	このフォルダには、ログロールオーバー時にイベントログが保存されます。
log-snapshot-auto	このフォルダには、syslogイベント0、1、2の自動収集ログが含まれます。
log-snapshot-user	このフォルダには、bloggerd log-snapshot の実行時に収集されたログが保存されます。

ログロールオーバーで生成されたログファイルを表示するには、次の例を参考にしてください。

```

switch# dir debug:log-dump/
debug:log-dump/20191022104656_evtlog_archive.tar
debug:log-dump/20191022111241_evtlog_archive.tar
debug:log-dump/20191022111841_evtlog_archive.tar
debug:log-dump/20191022112431_evtlog_archive.tar
debug:log-dump/20191022113042_evtlog_archive.tar
debug:log-dump/20191022113603_evtlog_archive.tar

```

ログ tar ファイルの解析

tar ファイル内のログを解析するには、次の例を参考にしてください。

```

switch# show system internal event-logs parse
debug:log-dump/20191022104656_evtlog_archive.tar
-----LOGS:/tmp/BLOGGERD0.991453012199/tmp/1-191022104658-191022110741-device_test-M27-V1-I1:0-P884.gz-----
2019 Oct 22 11:07:41.597864 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):Data Space
Limits(bytes): Soft: -1  Ha rd: -1
2019 Oct 22 11:07:41.597857 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):Stack Space
Limits(bytes): Soft: 500000  Hard: 500000
2019 Oct 22 11:07:41.597850 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):AS: 1005952076
-1
2019 Oct 22 11:07:41.597406 E_DEBUG Oct 22 11:07:41 2019(device_test_process_events):Sdwrap
msg unknown
2019 Oct 22 11:07:41.597398 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):Going back to
select
2019 Oct 22 11:07:41.597395 E_DEBUG Oct 22 11:07:41 2019(nvram_test):TestNvram examine
27 blocks
2019 Oct 22 11:07:41.597371 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):Parent: Thread
created test index:4 thread_id:-707265728
2019 Oct 22 11:07:41.597333 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):Node inserted
2019 Oct 22 11:07:41.597328 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):The test index
in diag is 4
2019 Oct 22 11:07:41.597322 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):result severity
level
2019 Oct 22 11:07:41.597316 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):callhome alert
level

```

次の表に、特定の tar ファイルの解析に使用できる追加のキーワードを示します。

キーワード	説明
component	プロセス名で識別されるコンポーネントに属するログをデコードします。
from-datetime	yy [mm [dd [HH [MM [SS]]]]] 形式で指定した、特定の日時のログをデコードします。
instance	デコードする SDWRAP バッファ インスタンスのリスト（カンマ区切り）。
module	SUPやLCなどのモジュールからのログをデコードします（モジュールIDを使用）。
to-datetime	yy [mm [dd [HH [MM [SS]]]]] 形式で指定した、特定の日時までのログをデコードします。

別の場所へログをコピーする

リモート サーバなどの別の場所にログをコピーするには、次の例を参考にしてください。

```

switch# copy debug:log-dump/20191022104656_evtlog_archive.tar
scp://<ip-address>/nobackup/<user> vrf management use-kstack
Enter username: user@<ip-address>'s password:
20191022104656_evtlog_archive.tar                                100% 130KB
130.0KB/s 00:00
Copy complete, now saving to disk (please wait)...
Copy complete.

```

自動収集ログファイルの消去

生成されるトリガー ベースの自動収集ログには、EventHistory と EventBundle の 2 種類があります。

EventHistory ログの消去ロジック

イベント履歴の場合は、/var/sysmgr/srv_logs/xport フォルダで消去が行われます。250 MB のパーティション RAM が、/var/sysmgr/srv_logs ディレクトリにマウントされます。

/var/sysmgr/srv_logs のメモリ使用率が、割り当てられた 250 MB の 65% 未満の場合、ファイルは消去されません。メモリ使用率が 65% の制限レベルに達すると、新しいログの保存を続行するのに十分なメモリが使用可能になるまで、最も古いファイルから消去されます。

EventBundle ログの消去ロジック

イベントバンドルの場合、消去ロジックは /bootflash/eem_snapshots フォルダでの状態に基づいて実行されます。自動収集されたスナップショットを保存するために、EEM 自動収集スクリプトは、ブートフラッシュストレージの 5% を割り当てます。ブートフラッシュ容量の 5% が使用されると、ログは消去されます。

新しい自動収集ログが利用可能になっているものの、ブートフラッシュに保存するスペースがない場合（すでに 5% の容量に達している）、システムは次のことを確認します。

1. 12 時間以上経過した既存の自動収集ファイルがある場合、システムはファイルを削除し、新しいログをコピーします。
2. 既存の自動収集ファイルが 12 時間未満の場合、新しく収集されたログは保存されずに廃棄されます。

デフォルトページ時間である 12 時間は、次のコマンドを使用して変更できます。コマンドで指定する時間は分単位です。

```
switch(config)# event manager applet test override __syslog_trigger_default
switch(config-applet)# action 1.0 collect test.yaml purge-time 300 $_syslog_msg
```

event manager command: test は、ポリシー例の名前です。__syslog_trigger_default は、オーバーライドする必要があるシステムポリシーの名前です。この名前は、二重アンダースコア (__) で始まる必要があります。

action command: 1.0 は、アクションの実行順番を示している例となっています。**collect** は、データが YAMU ファイルを使用して収集されることを示しています。**test.yaml** は、YAML ファイルの名前の例です。**\$_syslog_msg** は、コンポーネントの名前です。



- (注) どの時点でも、進行中のトリガーベースの自動収集イベントは 1 つだけです。自動収集がすでに発生しているときに別の新しいログ イベントを保存しようとする、新しいログ イベントは破棄されます。

デフォルトでは、トリガーベースのバンドルは 5 分 (300 秒) ごとに 1 つだけ収集されます。このレート制限は、次のコマンドでも設定できます。コマンドで指定する時間は秒単位です。

```
switch(config)# event manager applet test override __syslog_trigger_default
switch(config-applet)# action 1.0 collect test.yaml rate-limit 600 $_syslog_msg
```

event manager command: test はポリシーの名前の例です。__syslog_trigger_default は、オーバーライドするシステムポリシーの名前の例です。この名前は、二重アンダースコア (__) で始まる必要があります。

action command: 1.0 は、アクションの実行順番を示している例となっています。**collect** は、データが YAMU ファイルを使用して収集されることを示しています。**test.yaml** は、YAML ファイルの名前の例です。**\$_syslog_msg** は、コンポーネントの名前です。

リリース 10.1(1) 以降では、トリガーの最大数オプションを使用して収集レートを調整することもできます。これは、この数のトリガーだけを保つものです。**max-triggers** の値に達すると、syslog が発生しても、これ以上バンドルは収集されなくなります。

```
event manager applet test_1 override __syslog_trigger_default
  action 1.0 collect test.yaml rate-limit 30 max-triggers 5 $_syslog_msg
```



(注) 自動収集されたバンドルを debug:log-snapshot-auto/ により手動で削除すれば、次のイベントが発生したとき、**max-triggers** の設定数に基づいて収集が再開されます。

自動収集の統計情報と履歴

トリガーベースの収集統計情報の例を次に示します。

```
switch# show system internal event-logs auto-collect statistics
-----EEM Auto Collection Statistics-----
Syslog Parse Successful :88 Syslog Parse Failure :0
Syslog Ratelimited :0 Rate Limit Check Failed :0
Syslog Dropped(Last Action In Prog) :53 Storage Limit Reached :0
User Yaml Action File Unavailable :0 User Yaml Parse Successful :35
User Yaml Parse Error :0 Sys Yaml Action File Unavailable :11
Sys Yaml Parse Successful :3 Sys Yaml Parse Error :0
Yaml Action Not Defined :0 Syslog Processing Initiated :24
Log Collection Failed :0 Tar Creation Error :0
Signal Interrupt :0 Script Exception :0
Syslog Processed Successfully :24 Logfiles Purged :0
```

次の例は、CLI コマンドを使用して取得されたトリガーベースの収集履歴（処理された syslog 数、処理時間、収集されたデータのサイズ）を示しています。

```
switch# show system internal event-logs auto-collect history
DateTime Snapshot ID Syslog Status/Secs/Logsize(Bytes)
2019-Dec-04 05:30:32 1310232084 VPC-0-TEST_SYSLOG PROCESSED:9:22312929
2019-Dec-04 05:30:22 1310232084 VPC-0-TEST_SYSLOG PROCESSING
2019-Dec-04 04:30:13 1618762270 ACLMGR-0-TEST_SYSLOG PROCESSED:173:33194665
2019-Dec-04 04:28:47 897805674 SYSLOG-1-SYSTEM_MSG DROPPED-LASTACTIONINPROG
2019-Dec-04 04:28:47 947981421 SYSLOG-1-SYSTEM_MSG DROPPED-LASTACTIONINPROG
2019-Dec-04 04:27:19 1618762270 ACLMGR-0-TEST_SYSLOG PROCESSING
2019-Dec-04 02:17:16 1957148102 CARDCLIENT-2-FPGA_BOOT_GOLDEN NOYAMLFILFOUND
```

トリガーベースのログ収集の確認

次の例のように **show event manager system-policy | i trigger** コマンドを入力して、トリガーベースのログ収集機能が有効になっていることを確認します。

```
switch# show event manager system-policy | i trigger n 2
      Name : __syslog_trigger_default
  Description : Default policy for trigger based logging
  Overridable : Yes
    Event type : 0x2101
```


トリガーベースのログ ファイル生成の確認

トリガーベースの自動収集機能によってイベント ログ ファイルが生成されたかどうかを確認できます。次の例のいずれかのコマンドを入力します。

```
switch# dir bootflash:eem_snapshots
9162547 Nov 12 22:33:15 2019
1006309316_SECURITYD_2_FEATURE_ENABLE_DISABLE_eem_snapshot.tar.gz

Usage for bootflash://sup-local
8911929344 bytes used
3555950592 bytes free
12467879936 bytes total

switch# dir debug:log-snapshot-auto/
63435992 Dec 03 06:28:52 2019
20191203062841_1394408030_PLATFORM_2_MOD_PWRDN_eem_snapshot.tar.gz

Usage for debug://sup-local
544768 bytes used
4698112 bytes free
5242880 bytes total
```

ローカル ログ ファイルのストレージ

ローカル ログ ファイルのストレージ機能：

- ローカルデータストレージ時間の量は、導入の規模とタイプによって異なります。モジュラスイッチと非モジュラスイッチの両方で、ストレージ時間は15分から数時間のデータです。長期間にわたる関連ログを収集するには、次の手順を実行します。
 - 必要な特定のサービス/機能に対してのみイベント ログの保持を有効にします。「[単一サービスの拡張ログファイル保持の有効化 \(21 ページ\)](#)」を参照してください。
 - スイッチから内部イベント ログをエクスポートします。「[外部ログ ファイルのストレージ \(36 ページ\)](#)」を参照してください。
- 圧縮されたログは RAM に保存されます。
- 250MB のメモリは、ログ ファイル ストレージ用に予約されています。
- ログ ファイルは tar 形式で最適化されます (5 分ごとに 1 ファイルまたは 10 MB のいずれか早い方)。
- スナップショット収集を許可します。

最近のログ ファイルのローカル コピーの生成

拡張ログファイル保持は、スイッチで実行されているすべてのサービスに対してデフォルトで有効になっています。ローカルストレージの場合、ログファイルは、フラッシュメモリに保存されます。次の手順を使用して、最新のイベントログファイルのうち最大10個のイベントログファイルを生成します。

手順の概要

1. **bloggerd log-snapshot** [*file-name*] [**bootflash:** *file-path* | **logflash:** *file-path* | **usb1:**] [**size** *file-size*] [**time** *minutes*]

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	bloggerd log-snapshot [<i>file-name</i>] [bootflash: <i>file-path</i> logflash: <i>file-path</i> usb1:] [size <i>file-size</i>] [time <i>minutes</i>] 例 : <pre>switch# bloggerd log-snapshot snapshot1</pre>	スイッチに保存されている最新の 10 個のイベント ログのスナップショット バンドル ファイルを作成します。この操作のデフォルトのストレージは logflash です。 file-name : 生成されたスナップショット ログ ファイル バンドルのファイル名。 <i>file-name</i> には最大 64 文字を使用します。 (注) この変数はオプションです。設定されていない場合、システムはタイムスタンプと「_snapshot_bundle.tar」をファイル名として適用します。例 : <pre>20200605161704_snapshot_bundle.tar</pre> bootflash: <i>file-path</i> : スナップショット ログ ファイル バンドルがブートフラッシュに保存されているファイルパス。次の初期パスのいずれかを選択します。 • bootflash:/// • bootflash://module-1/ • bootflash://sup-1/ • bootflash://sup-active/ • bootflash://sup-local/ logflash: <i>file-path</i> : スナップショット ログ ファイル バンドルがログ フラッシュに保存されるファイルパス。次の初期パスのいずれかを選択します。 • logflash:/// • logflash://module-1/ • logflash://sup-1/ • logflash://sup-active/

	コマンドまたはアクション	目的
		• logflash://sup-local/ usb1: : USB デバイス上のスナップショット ログ ファイルバンドルが保存されているファイルパス。 size file-size : メガバイト (MB) 単位のサイズに基づくスナップショット ログ ファイル バンドル。範囲は 5MB〜250MB です。 time minutes : 最後の x 時間 (分) に基づくスナップショット ログ ファイル バンドル。範囲は 1 ~ 30 分です。

例

```
switch# bloggerd log-snapshot snapshot1
Snapshot generated at logflash:evt_log_snapshot/snapshot1_snapshot_bundle.tar Please
cleanup once done.
switch#
switch# dir logflash:evt_log_snapshot
159098880 Dec 05 06:40:24 2019 snapshot1_snapshot_bundle.tar
159354880 Dec 05 06:40:40 2019 snapshot2_snapshot_bundle.tar

Usage for logflash://sup-local
759865344 bytes used
5697142784 bytes free
6457008128 bytes total
```

次の例のコマンドを使用して、同じファイルを表示します。

```
switch# dir debug:log-snapshot-user/
159098880 Dec 05 06:40:24 2019 snapshot1_snapshot_bundle.tar
159354880 Dec 05 06:40:40 2019 snapshot2_snapshot_bundle.tar

Usage for debug://sup-local
929792 bytes used
4313088 bytes free
5242880 bytes total
```



(注) ファイル名は、例の最後に示されています。個々のログファイルは、生成された日時によっても識別されます。

リリース 10.1(1) 以降、LC コアファイルには log-snapshot バンドルが含まれています。log-snapshot バンドル ファイル名は、tac_snapshot_bundle.tar.gz です。次に例を示します。

```
bash-4.2$ tar -tvf 1610003655_0x102_aclqos_log.17194.tar.gz
drwxrwxrwx root/root 0 2021-01-07 12:44 pss/
-rw-rw-rw- root/root 107 2021-01-07 12:44 pss/dev_shm_aclqos_runtime_info_lc.gz
-rw-rw-rw- root/root 107 2021-01-07 12:44 pss/dev_shm_aclqos_runtime_cfg_lc.gz
-rw-rw-rw- root/root 107 2021-01-07 12:44 pss/dev_shm_aclqos_debug.gz
```

```

-rw-rw-rw- root/root 129583 2021-01-07 12:44 pss/clqosdb_ver1_0_user.gz
-rw-rw-rw- root/root 20291 2021-01-07 12:44 pss/clqosdb_ver1_0_node.gz
-rw-rw-rw- root/root 444 2021-01-07 12:44 pss/clqosdb_ver1_0_ctrl.gz
drwxrwxrwx root/root 0 2021-01-07 12:44 proc/
-rw-rw-rw- root/root 15159 2021-01-07 12:44 0x102_aclqos_compress.17194.log.25162
-rw-rw-rw- root/root 9172392 2021-01-07 12:43 0x102_aclqos_core.17194.gz
-rw-rw-rw- root/root 43878 2021-01-07 12:44 0x102_aclqos_df_dmesg.17194.log.gz
-rw-rw-rw- root/root 93 2021-01-07 12:44 0x102_aclqos_log.17194
-rw-rw-rw- root/root 158 2021-01-07 12:44 0x102_aclqos_mcore.17194.log.gz
drwxrwxrwx root/root 0 2021-01-07 12:44 usr17194/
-rw-rw-rw- root/root 11374171 2021-01-07 12:44 tac_snapshot_bundle.tar.gz

```

外部ログ ファイルのストレージ

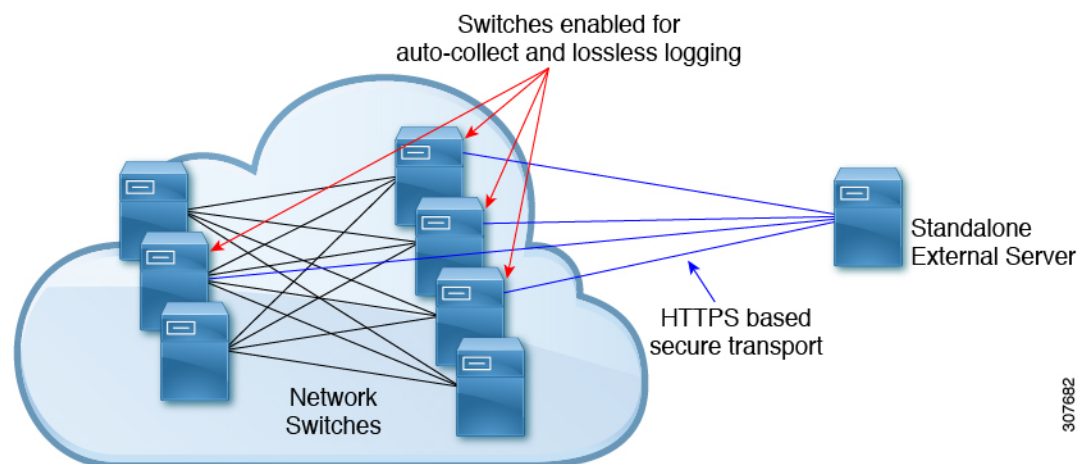
外部サーバ ソリューションは、ログを安全な方法でオフスイッチに保存する機能を提供します。



(注) 外部ストレージ機能を作成するため、Cisco Technical Assistance Center (TAC) に連絡して、外部サーバ ソリューションの展開をサポートを求めてください。

次に、外部ログ ファイルの保存機能を示します。

- オンデマンドで有効
- HTTPS ベースの転送
- ストレージ要件 :
 - 非モジュラ スイッチ : 300 MB
 - モジュラ スイッチ : 12 GB (1 日あたり、スイッチあたり)
- 通常、外部サーバには 10 台のスイッチのログが保存されます。ただし、外部サーバでサポートされるスイッチの数に厳密な制限はありません。



307682

外部サーバ ソリューションには、次の特性があります。

- コントローラレス環境
- セキュリティ証明書の手動管理
- サポートされている 3 つの使用例：
 - 選択したスイッチからのログの継続的な収集
 - TAC のサポートによる、シスコ サーバへのログの展開とアップロード。
 - 限定的なオンプレミス処理



(注) 外部サーバでのログ ファイルの設定と収集については、Cisco TAC にお問い合わせください。

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。