Embedded Event Manager の設定

この章では、Embedded Event Manager（EEM）を設定して Cisco NX-OS デバイス上のクリティカル イベントを検出し、対処する方法について説明します。

この章は、次の項で構成されています。

- EEM について，1 ページ
- EEM のライセンス要件，6 ページ
- EEM の前提条件，6 ページ
- EEM の注意事項と制約事項，6 ページ
- EEM のデフォルト設定，7 ページ
- EEM の設定，7 ページ
- EEM 設定の確認，23 ページ
- EEM のコンフィギュレーション例，24 ページ

EEM について

EEMはデバイス上で発生するイベントをモニタし、設定に基づいて各イベントの回復またはトラブルシューティングのためのアクションを実行します。

EEM は次の 3 種類の主要コンポーネントからなります。

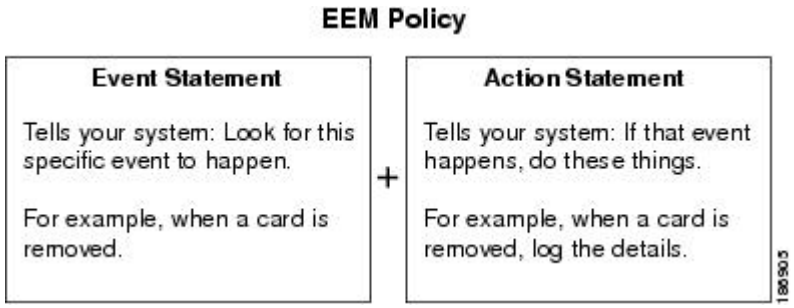
- イベント文：別の Cisco NX-OS コンポーネントからモニタし、アクション、回避策、または通知が必要になる可能性のあるイベント。
- アクション文：電子メールの送信、インターフェイスのディセーブル化など、イベントから回復するために EEM が実行できるアクション。
- ポリシー：イベントのトラブルシューティングまたはイベントからの回復を目的とした 1 つまたは複数のアクションとペアになったイベント。

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。

この図は、EEM ポリシーの基本的な 2 種類の文を示します。

図 *1*：*EEM* ポリシー文



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。スーパーバイザ上で EEM ポリシーを設定すると、EEM がイベント タイプに基づいて、正しいモジュールにポリシーをプッシュします。EEM はモジュール上でローカルに、またはスーパーバイザ上で（デフォルトのオプション）、発生したイベントに対応するアクションを実行します。

EEM はスーパーバイザ上でイベント ログを維持します。

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。

使用するネットワークに合わせてユーザ ポリシーを作成できます。ユーザ ポリシーを作成すると、そのポリシーと同じイベントに関連するシステム ポリシー アクションが EEM によって発生したあと、ユーザ ポリシーで指定したアクションが行われます。

一部のシステム ポリシーは上書きすることもできます。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。イベントまたはアクションの上書きが可能です。

設定済みのシステム ポリシーを表示して、上書き可能なポリシーを判断するには、**show event manager system-policy** コマンドを使用します。

（注） **show running-config eem** コマンドを使用して、各ポリシーのコンフィギュレーションを確認してください。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。

（注） 上書きポリシーには、必ずイベント文を指定します。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。

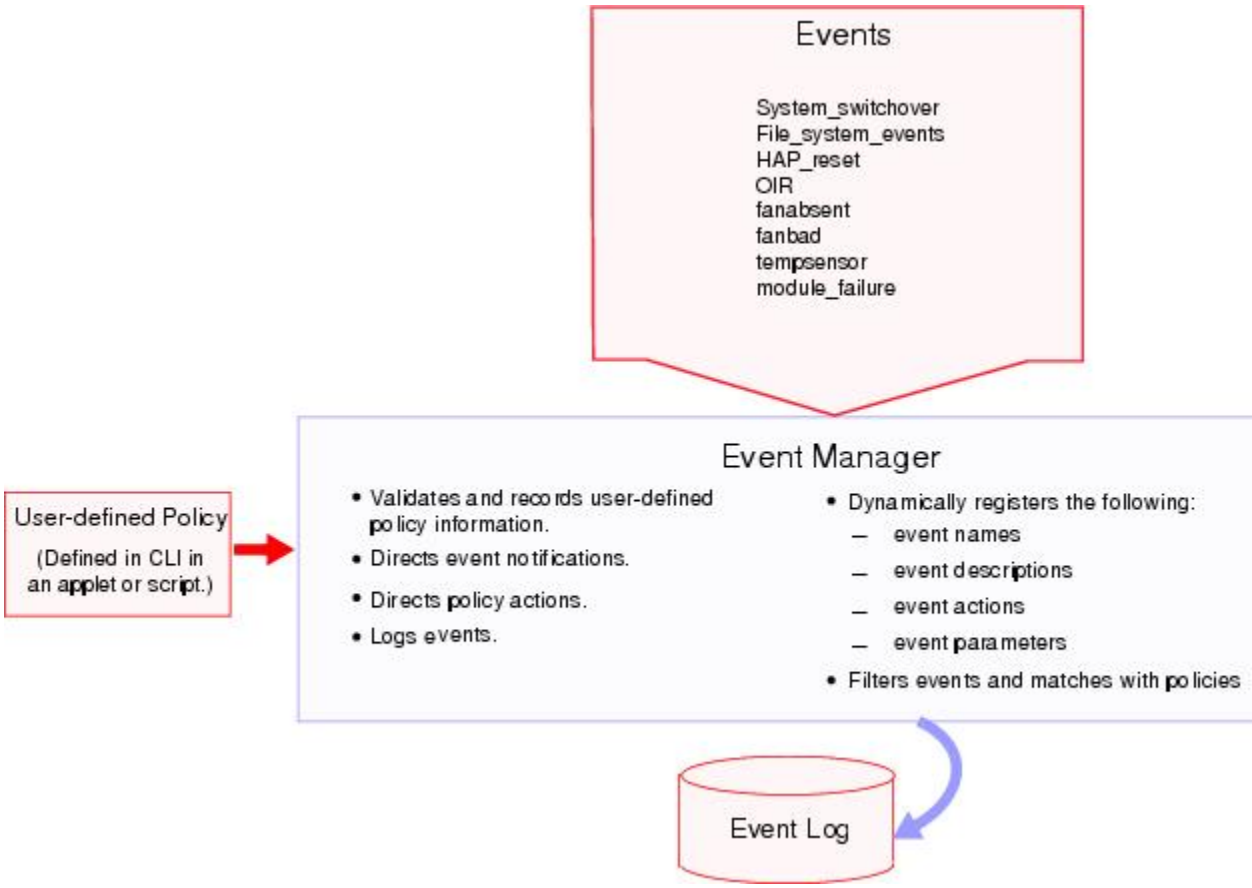
イベント文

イベントは、回避、通知など、何らかのアクションが必要なデバイス アクティビティです。これらのイベントは通常、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。

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。

この図は、EEM によって処理されたイベントを示します。

図 *2*：*EEM* の概要



イベント文では、ポリシー実行のトリガーになるイベントを指定します。複数イベント トリガーを設定できます。

EEM はイベント文に基づいてポリシーをスケジューリングし、実行します。EEM はイベントおよびアクション コマンドを検証し、定義に従ってコマンドを実行します。

（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、event-default アクション文を許可して EEM ポリシーを設定する必要があります。

アクション ステートメント

アクション文では、ポリシーによって実行されるアクションを記述します。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。

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。
- 例外の記録。
- モジュールの強制的シャットダウン
- デバイスのリロード。
- 電力のバジェット超過による特定モジュールのシャットダウン。
- Syslog メッセージの生成。
- Call Home イベントの生成。
- 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。

（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、match 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 CLI コマンドを実行できません。

（注） ユーザ ポリシーまたは上書きポリシーの中に、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ようなアクション文がないかどうかを確認してください。

VSH スクリプト ポリシー

テキストエディタを使用し、VSHスクリプトでポリシーを作成することもできます。このようなポリシーにも、他のポリシーと同様、イベント文およびアクション文（複数可）を使用します。また、これらのポリシーでシステムポリシーを補うことも上書きすることもできます。VSHスクリプト ポリシーを書き込んだ後、デバイスにコピーしてアクティブにします。

環境変数

すべてのポリシーに使用できる、EEMの環境変数を定義できます。環境変数は、複数のポリシーで使用できる共通の値を設定する場合に便利です。たとえば、外部電子メールサーバのIPアドレスに対応する環境変数を作成できます。

パラメータ置換フォーマットを使用することによって、アクション文で環境変数を使用できます。

この例では、「EEM action」というリセット理由を指定し、モジュール 1 を強制的にシャットダウンするアクション文の例を示します。

```
switch (config-eem-policy)# action 1.0 forceshut module 1 reset-reson “EEM action.”
```

シャットダウンの理由に default-reason という環境変数を定義すると、次の例のように、リセット理由を環境変数に置き換えることができます。

```
switch (config-eem-policy)# action 1.0 foreshut module 1 reset-reason $default-reason
```

この環境変数は、任意のポリシーで再利用できます。

EEM イベント相関

イベントの組み合わせに基づいて EEM ポリシーをトリガーできます。まず、**tag** キーワードを使用してEEMポリシーに複数のイベントを作成し区別します。次に、一連のブール演算子（**AND**、**OR**、**ANDNOT**）を使用して、回数および時間をもとに、カスタム処理をトリガーするこれらのイベントの組み合わせを定義できます。

ハイ アベイラビリティ

Cisco NX-OS は、EEM のステートレス リスタートをサポートします。リブートまたはスーパーバイザ スイッチオーバーの後、Cisco NX-OS は実行コンフィギュレーションを適用します。

仮想化のサポート

アクションまたはイベントがすべて表示されるわけではありません。ポリシーを設定するには、network-admin の権限が必要です。

EEM のライセンス要件

製品	ライセンス要件
Cisco NX-OS	EEM にはライセンスは不要です。ライセンス パッケージに含まれていない機能は nx-os イメージにバンドルされており、無料で提供されます。Cisco NX-OS ライセンス方式の詳細については、『*Cisco NX-OS Licensing Guide*』を参照してください。

EEM の前提条件

EEM の前提条件は、次のとおりです。

- EEM を設定するには、network-admin のユーザ権限が必要です。

EEM の注意事項と制約事項

EEM に関する設定時の注意事項および制約事項は、次のとおりです。

- 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ような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、match 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 CLI コマンドを実行できません。
- 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。
- 通常のコマンド式に適用できるルールは、すべてのキーワードを拡張する必要があること、そして * 記号のみが引数の置換に使用できることです。
- EEM イベント相関はスーパーバイザ モジュールだけでサポートされます。
- EEM イベント相関は、単一ポリシー内の別のモジュール間ではサポートされません。
- 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、モジュール、モジュール障害、oir、snmp、syslog 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 イベント相関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。
- Python から EEM を呼び出すことができます。Python の詳細については、『*Cisco Nexus 9000 Series NX-OS Programmability Guide*』を参照してください。

EEM のデフォルト設定

次の表に、EEM パラメータのデフォルト設定を示します。

パラメータ（**Parameters**）	デフォルト
システム ポリシー	Active

EEM の設定

システム ポリシーに基づいて実行されるアクションを含むポリシーを作成できます。システム ポリシーに関する情報を表示するには、**show event manager system-policy** コマンドを使用します。

環境変数の定義

EEM ポリシーでパラメータとして機能する変数を定義できます。

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

	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 environment***variable-name variable-value* 例： `switch(config)# event manager environment emailto "admin@anyplace.com"`	EEM 用の環境変数を作成します。*variable-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。*variable-value* には最大 39 文字の英数字を引用符で囲んで使用できます。
ステップ 3	**show event manager environment** {*variable-name* | **all**} 例： `switch(config)# show event manager environment all`	（任意） 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	**copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	（任意） 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ます。

CLI によるユーザ ポリシーの定義

CLI を使用して、デバイスにユーザ ポリシーを定義できます。

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

	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***applet-name*
	
	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	**description***policy-description* 例： `switch(config-applet)# description “Monitors interface shutdown.”`	（任意） 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***event-statement* 例： `switch(config-applet)# event cli match “shutdown”`	ポリシーのイベント文を設定します。イベント文が複数ある場合、このステップを繰り返します。イベント文の設定，（10 ページ）を参照してください。
ステップ 5	**tag***tag* {**and** | **andnot** | **or**} *tag* [**and** | **andnot** | **or** {*tag*}] {**happens***occurs in seconds*} 例： `switch(config-applet)# tag one or two happens 1 in 10000`	（任意） 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***number*[*.number2*] *action-statement* 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。アクション文の設定，（15 ページ）を参照してください。
ステップ 7	**show event manager policy-state***name* [**module***module-id*] 例： `switch(config-applet)# show event manager policy-state monitorShutdown`	（任意） 設定したポリシーの状態に関する情報を表示します。
ステップ 8	**copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	（任意） 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 モードで次のいずれかのコマンドを使用します。

コマンド	目的
event application [**tag***tag*] **sub-system***sub-system-id***type***event-type* 例： `switch(config-applet)# event application sub-system 798 type 1`	イベントの指定がサブシステム ID およびアプリケーション イベント タイプに一致する場合に、イベントを発生させます。 *sub-system-id* と *event-type* の範囲は 1 ～ 4294967295 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。 （注） このコマンドを使用するには、まず **feature evmed** コマンドをイネーブルにして一般的なイベント ディテクタをイネーブルにする必要があります。
event cli [**tag***tag*] **match***expression* [**count***repeats* | **time***seconds*] 例： `switch(config-applet)# event cli match "shutdown"`	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 秒です。0 は無制限を示します。
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： `switch(config-applet)# event counter name mycounter entry-val 20 gt`	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。

コマンド	目的
event fanabsent [**fan***number*] **time***seconds* 例： `switch(config-applet)# event fanabsent time 300`	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。*repeats* の範囲は 10 ～ 64000 です。
event fanbad [**fan***number*] **time***seconds* 例： `switch(config-applet)# event fanbad time 3000`	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。*repeats* の範囲は 10 ～ 64000 です。
event fib {**adjacency extra** | **resource tcam usage** | **route** {**extra** | **inconsistent** | **missing**}} 例： `switch(config-applet)# event fib adjacency extra`	次のいずれかに対するイベントを発生させます。 • **adjacency extra**：ユニキャスト FIB に追加のルートがある場合。 • **resource tcam usage**：TCAM 使用率がいずれかの方向で 5 の倍数になるごとに。 • **route** {**extra** | **inconsistent** | **missing**}：ユニキャスト FIB でルートが追加、変更、または削除される場合。
event gold module {*slot* | **all**} **test***test-name* [**severity** {**major** | **minor** | **moderate**}] **testing-type** {**bootup** | **monitoring** | **ondemand** | **scheduled**} **consecutive-failure***count* 例： `switch(config-applet)# event gold module 2 test ASICRegisterCheck testing-type ondemand consecutive-failure 2`	名前で指定されたオンライン診断テストが、設定された回数だけ連続して、設定された重大度で失敗した場合に、イベントを発生させます。*slot* の範囲は 1 ～ 10 です。*test-name* は設定されたオンライン診断テストの名前です。*count* の範囲は 1 ～ 1000 です。
event interface [**tag***tag*] {**name***interface slot*/*port***parameter**} 例： `switch(config-applet)# event interface ethernet 2/2 parameter`	カウンタが指定のインターフェイスに対して超えた場合に、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。 （注） このコマンドを使用するには、まず **feature evmed** コマンドをイネーブルにして一般的なイベント ディテクタをイネーブルにする必要があります。
event memory {**critical** | **minor** | **severe**} 例： `switch(config-applet)# event memory critical`	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。メモリのしきい値の設定，（20 ページ）も参照してください。

コマンド	目的
event module [**tag***tag*] **status** {**online** | **offline** | **any**} **module** {**all** | *module-num*} 例： `switch(config-applet)# event module status offline module all`	指定したモジュールが選択された状態になったとき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。
event module-failure [**tag***tag*] **type***failure-type***module** {*slot* | **all**} **count***repeats* [**time***seconds*] 例： `switch(config-applet)# event module-failure type lc-failed module 3 count 1`	モジュールが設定された障害タイプになった場合に、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。 *repeats* の範囲は 0 ～ 4294967295 です。*seconds* の範囲は 0 ～ 4294967295 秒です。0 は無制限を示します。
event none 例： `switch(config-applet)# event none`	手動で指定されたイベントがないポリシーイベントを実行します。 （注） このコマンドを使用するには、まず **feature evmed** コマンドをイネーブルにして一般的なイベント ディテクタをイネーブルにする必要があります。
event oir [**tag***tag*] {**fan** | **module** | **powersupply**} {**anyoir** | **insert** | **remove**} [*number*] 例： `switch(config-applet)# event oir fan remove 4`	設定されたデバイス構成要素（ファン、モジュール、または電源モジュール）がデバイスに取り付けられた場合、またはデバイスから取り外された場合に、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。 任意で、ファン、モジュール、または電源モジュールの具体的な番号を設定できます。*number* の範囲は次のとおりです。 • ファン番号：モジュール依存 • モジュール番号：デバイス依存 • 電源モジュール番号：範囲は 1 ～ 3 です。

コマンド	目的
event policy-default count*repeats* [**time***seconds*] 例： `switch(config-applet)# event policy-default count 3`	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。 *repeats* の範囲は 1 ～ 65000 です。*seconds* の範囲は 0 ～ 4294967295 秒です。0 は無制限を示します。
event poweroverbudget 例： `switch(config-applet)# event poweroverbudget`	電力バジェットが設定された電源モジュールの容量を超えた場合に、イベントを発生させます。
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： `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`	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。 *entry* および *exit* の値の範囲は 0 ～ 18446744073709551615 です。time の範囲は 0 ～ 2147483647 秒です。interval の範囲は 1 ～ 2147483647 秒です。
event storm-control 例： `switch(config-applet)# event storm-control`	ポート上のトラフィックが設定されたストーム制御しきい値を超えた場合に、イベントを発生させます。
event syslog [**occurs***count*] {**pattern***string* | **period***time* | **priority***level* | **tag***tag*} 例： `switch(config-applet)# event syslog period 500`	指定した syslog のしきい値を超えた場合にイベントを発生させます。カウントの範囲は 1 ～ 65000 で、時間の範囲は 1 ～ 4294967295 です。プライオリティの範囲は 0 ～ 7 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。
event sysmgr memory [**module***module-num*] **major***major-percent***minor***minor-percent***clear***clear-percent* 例： `switch(config-applet)# event sysmgr memory minor 80`	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。パーセンテージの範囲は 1 ～ 99 です。

コマンド	目的
event sysmgr switchover count*count***time***interval* 例： `switch(config-applet)# event sysmgr switchover count 10 time 1000`	指定した switchover count が、指定した time interval を超えた場合にイベントを発生させます。switchover count の範囲は 1 ～ 65000 です。time interval の範囲は 0 ～ 2147483647 です。
event temperature [**module***slot*] [*sensor-number*] **threshold** {**any** | **major** | **minor**} 例： `switch(config-applet)# event temperature module 2 threshold any`	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。sensor の範囲は 1 ～ 18 です。
event timer {**absolute time***time***name***name* | **countdown time***time***name***name* | **cron cronentry***string* | **tag***tag* | **watchdog time***time***name***name*} 例： `switch(config-applet)# event timer absolute time 100 name abtimer`	指定した時間に到達した場合に、イベントを発生させます。時間の範囲は 1 ～ 4294967295 です。 • **absolute time**：指定された絶対時刻が発生した場合に、イベントを発生させます。 • **countdown time**：指定された時間がゼロにカウントダウンされたときに、イベントを発生させます。タイマーはリセットされません。 • **cron cronentry**：CRON 文字列の指定が現在時刻に一致する場合に、イベントを発生させます。 • **watchdog time**：指定された時間がゼロにカウントダウンされたときに、イベントを発生させます。タイマーは、初期値に自動的にリセットされ、カウント ダウンが続行されます。 **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。 （注） このコマンドを使用するには、まず **feature evmed** コマンドをイネーブルにして一般的なイベント ディテクタをイネーブルにする必要があります。

コマンド	目的
event track [**tag***tag*] *object-number***state** {**any** | **down** | **up**} 例： `switch(config-applet)# event track 1 state down`	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 *object-number* の範囲は 1 ～ 500 です。

アクション文の設定

アクション文を設定するには、EEM コンフィギュレーション モードで次のいずれかのコマンドを使用します。

コマンド	目的
action*number*[.*number2*] **cli***command1* [*command2...*][**local**] 例： `switch(config-applet)# action 1.0 cli "show interface e 3/1"`	設定された CLI 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 number1.number2 です。 *number* は 16 桁までの任意の数値にできます。*number2* の範囲は 0 ～ 9 です。
action*number*[.*number2*] **counter name***counter***value***val***op** {**dec** | **inc** | **nop** | **set**} 例： `switch(config-applet)# action 2.0 counter name mycounter value 20 op inc`	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 number1.number2 です。 *number* は 16 桁までの任意の数値にできます。*number2* の範囲は 0 ～ 9 です。 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。*val* には 0 ～ 2147483647 の整数または置換パラメータを指定できます。
action*number*[.*number2*] **event-default** 例： `switch(config-applet)# action 1.0 event-default`	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 number1.number2 です。 *number* は 16 桁までの任意の数値にできます。*number2* の範囲は 0 ～ 9 です。

コマンド	目的
action*number*[.*number2*] **forceshut** [**module***slot* | **xbar***xbar-number*] **reset-reason***seconds* 例： `switch(config-applet)# action 1.0 forceshut module 2 reset-reason "flapping links"`	モジュール、クロスバー、またはシステム全体を強制的にシャットダウンします。アクション ラベルのフォーマットは number1.number2 です。 *number* は 16 桁までの任意の数値にできます。*number2* の範囲は 0 ～ 9 です。 リセット理由は、引用符で囲んだ最大 80 文字の英数字ストリングです。
action*number*[.*number2*] **overbudgetshut** [**module***slot*[-*slot*]] 例： `switch(config-applet)# action 1.0 overbudgetshut module 3-5`	電力バジェット超過の問題により、1 つまたは複数のモジュールまたはシステム全体を強制的にシャットダウンします。 *number* は 16 桁までの任意の数値にできます。*number2* の範囲は 0 ～ 9 です。
action*number*[.*number2*] **policy-default** 例： `switch(config-applet)# action 1.0 policy-default`	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 number1.number2 です。 *number* は 16 桁までの任意の数値にできます。*number2* の範囲は 0 ～ 9 です。
action*number*[.*number2*] **publish-event** 例： `switch(config-applet)# action 1.0 publish-event`	アプリケーション固有のイベントの発行を強制します。アクション ラベルのフォーマットは number1.number2 です。 *number* は 16 桁までの任意の数値にできます。*number2* の範囲は 0 ～ 9 です。
action*number*[.*number2*] **reload** [**module***slot*[-*slot*]] 例： `switch(config-applet)# action 1.0 reload module 3-5`	1 つまたは複数のモジュールまたはシステム全体を強制的にリロードします。 *number* は 16 桁までの任意の数値にできます。*number2* の範囲は 0 ～ 9 です。
action*number*[.*number2*] **snmp-trap** {[**intdata1***data* [**intdata2***data*]] [**strdata***string*]} 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。*number* は 16 桁までの任意の数値にできます。*number2* の範囲は 0 ～ 9 です。 *data* 引数には、最大 80 桁の任意の数を指定できます。*string* には最大 80 文字の英数字を使用できます。

コマンド	目的
action*number*[.*number2*] **syslog** [**priority***prio-val*] **msg***error-message* 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。*number* は 16 桁までの任意の数値にできます。*number2* の範囲は 0 ～ 9 です。 *error-message* 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。

（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、match 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 CLI 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、CLI が一致するすべての EEM ポリシーで、CLI コマンドを実行できます。

VSH スクリプトによるポリシーの定義

VSH スクリプトを使用してポリシーを定義できます。

はじめる前に

管理者の権限でログインしていることを確認します。

スクリプト名がスクリプト ファイル名と同じ名前であることを確認します。

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

VSH スクリプト ポリシーの登録およびアクティブ化

VSH スクリプトで定義したポリシーを登録してアクティブにできます。

手順の概要

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

手順の詳細

	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***policy-script* 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。*policy-script* 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 3	**copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	（任意） 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ます。

ポリシーの上書き

システム ポリシーは上書き可能です。

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. （任意） **description***policy-description*
5. **event***event-statement*
6. **action***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

	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	**show event manager policy-state***system-policy*
	
	例：
	`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`
		（任意）
		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。システム ポリシー名を突き止めるには、**show event manager system-policy** コマンドを使用します。システム ポリシーについては、Embedded Event Manager システム イベントおよびコンフィギュレーション例を参照してください。

ステップ 3	**event manager applet***applet-name***override***system-policy*
	
	例：
	`switch(config)# event manager applet ethport override __ethpm_link_flap`
	`switch(config-applet)#`
		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 29 の英数字を使用できます。*system-policy* は、既存のシステム 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： `description "Overrides link flap policy."`	（任意） 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 5	**event***event-statement* 例： `switch(config-applet)# event policy-default count 2 time 1000`	ポリシーのイベント文を設定します。
ステップ 6	**action***number action-statement* 例： `switch(config-applet)# action 1.0 syslog priority warnings msg "Link is flapping."`	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	**show event manager policy-state***name* 例： `switch(config-applet)# show event manager policy-state ethport`	（任意） 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	**copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	（任意） 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ます。

メモリのしきい値の設定

イベントを発生させるメモリしきい値を設定し、オペレーティング システムがメモリを割り当てられない場合にプロセスを終了させるかどうかを設定できます。

はじめる前に

管理者の権限でログインしていることを確認します。

手順の概要

1. **configure terminal**
2. **system memory-thresholds minor***minor***severe***severe***critical***critical*
3. （任意） **system memory-thresholds threshold critical no-process-kill**
4. （任意） **show running-config | include "system memory"**
5. （任意） **copy running-config startup-config**

手順の詳細

	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	**system memory-thresholds minor***minor***severe***severe***critical***critical* 例： `switch(config)# system memory-thresholds minor 60 severe 70 critical 80`	EEM メモリ イベントを生成するシステム メモリしきい値を設定します。デフォルト値は次のとおりです。 • Minor-85 • Severe-90 • Critical-95 これらのメモリのしきい値を超えた場合、システムは次の syslog を生成します。 • 2013 May 7 17:06:30 switch %$ %PLATFORM-2-MEMORY_ALERT: Memory Status Alert : MINOR • 2013 May 7 17:06:30 switch %$ %PLATFORM-2-MEMORY_ALERT: Memory Status Alert : SEVERE

	コマンドまたはアクション	目的
		• 2013 May 7 17:06:30 switch %$ %PLATFORM-2-MEMORY_ALERT: Memory Status Alert : CRITICAL • 2013 May 7 17:06:35 switch %$ %PLATFORM-2-MEMORY_ALERT: Memory Status Alert : MINOR ALERT RECOVERED • 2013 May 7 17:06:35 switch %$ %PLATFORM-2-MEMORY_ALERT: Memory Status Alert : SEVERE ALERT RECOVERED • 2013 May 7 17:06:35 switch %$ %PLATFORM-2-MEMORY_ALERT: Memory Status Alert : CRITICAL ALERT RECOVERED
ステップ 3	**system memory-thresholds threshold critical no-process-kill** 例： `switch(config)# system memory-thresholds threshold critical no-process-kill`	（任意） メモリを割り当てることができない場合にプロセスを停止しないようにシステムを設定します。デフォルト値では、最もメモリを消費するプロセスから終了できます。
ステップ 4	**show running-config | include "system memory"** 例： `switch(config-applet)# show running-config | include "system memory"`	（任意） システム メモリ設定に関する情報を表示します。
ステップ 5	**copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	（任意） 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ます。

EEM パブリッシャとしての syslog の設定

スイッチからの syslog メッセージをモニタできます。

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

はじめる前に

EEM は、Syslog による登録に使用可能である必要があります。

Syslog デーモンが設定され、実行される必要があります。

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

	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***applet-name*
	
	例：
	`switch(config)# event manager applet abc`
	`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。

ステップ 3	**event syslog** [**tag***tag*] {**occurs***number* | **period***seconds* | **pattern***msg-text* | **priority***priority*} 例： `switch(config-applet)# event syslog occurs 10`	syslog メッセージを監視し、ポリシーの検索文字列に基づいてポリシーを呼び出します。 • **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。 • **occurs***number* のキーワードと引数のペアは、発生回数を指定します。指定できる範囲は 1 ～ 65000 です。 • **period***seconds* のキーワードと引数のペアは、イベントの発生間隔を指定します。範囲は 1 ～ 4294967295 です。 • **pattern***msg-text* のキーワードと引数のペアは、一致する正規表現を指定します。パターンには、文字テキスト、環境変数、またはこの 2 つの組み合わせを含めることができます。文字列に空白が含まれる場合は引用符で囲みます。 • **priority***priority* のキーワードと引数のペアは、syslog メッセージのプライオリティを指定します。このキーワードを指定しないと、すべての Syslog メッセージのプライオリティ レベルが「情報レベル」に設定されます。

	コマンドまたはアクション	目的
ステップ 4	**copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	（任意） 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ます。

EEM 設定の確認

EEM のコンフィギュレーション情報を表示するには、次の作業のいずれかを行います。

コマンド	目的
show event manager environment [*variable-name* | **all**]	イベントマネージャの環境変数に関する情報を表示します。
show event manager event-types [*event* | **all** | **module***slot*]	イベント マネージャのイベント タイプに関する情報を表示します。
show event manager history events [**detail**] [**maximum***num-events*] [**severity** {**catastrophic** | **minor** | **moderate** | **severe**}]	すべてのポリシーについて、イベント履歴を表示します。
show event manager policy-state*policy-name*	しきい値を含め、ポリシーの状態に関する情報を表示します。
show event manager script system [*policy-name* | **all**]	スクリプト ポリシーに関する情報を表示します。
show event manager system-policy [**all**]	定義済みシステム ポリシーに関する情報を表示します。
show running-config eem	EEMの実行コンフィギュレーションに関する情報を表示します。
show startup-config eem	EEM のスタートアップ コンフィギュレーションに関する情報を表示します。

EEM のコンフィギュレーション例

モジュール 3 の中断のないアップグレード エラーのしきい値だけを変更することによって、__lcm_module_failure システム ポリシーを上書きする方法の例を示します。この例では、syslog メッセージも送信されます。その他のすべての場合、システム ポリシー __lcm_module_failure の設定値が適用されます。

```
event manager applet example2 override __lcm_module_failure
event module-failure type hitless-upgrade-failure module 3 count 2
action 1 syslog priority errors msg module 3 “upgrade is not a hitless upgrade!”
action 2 policy-default
```

次に、__ethpm_link_flap システム ポリシーを上書きし、インターフェイスをシャット ダウンする例を示します。

```
event manager applet ethport override __ethpm_link_flap
event policy-default count 2 time 1000
action 1 cli conf t
action 2 cli int et1/1
action 3 cli no shut
```

次に、ユーザがデバイスでコンフィギュレーション モードを開始すると、CLI コマンドを実行できるが、SNMP 通知をトリガーする EEM ポリシーを作成する例を示します。

```
event manager applet TEST
event cli match "conf t"
action 1.0 snmp-trap strdata "Configuration change"
action 2.0 event-default
```

（注） 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 CLI コマンドを実行できません。

次に、EEM ポリシーの複数イベントを関連付け、イベント トリガーの組み合わせに基づいてポリシーを実行する例を示します。この例では、EEM ポリシーは、指定された syslog パターンのいずれかが 120 秒以内に発生したときにトリガーされます。

```
event manager applet eem-correlate
event syslog tag one pattern "copy bootflash:.* running-config.*”
event syslog tag two pattern “copy run start”
event syslog tag three pattern “hello”
tag one or two or three happens 1 in 120
action 1.0 reload module 1
```

（注） 追加の EEM の設定例については、Embedded Event Manager システム イベントおよびコンフィギュレーション例を参照してください。