# EEM の設定

この章の内容は、次のとおりです。


- Embedded Event Manager について，1 ページ
- EEM ポリシー，2 ページ
- EEM イベント文，3 ページ
- EEM アクション文，4 ページ
- VSH スクリプト ポリシー，4 ページ
- EEM イベント相関，5 ページ
- EEM 仮想化サポート，5 ページ
- EEM のライセンス要件，5 ページ
- EEM の前提条件，5 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，6 ページ
- EEM の設定，7 ページ
- EEM 設定の確認，20 ページ
- EEM のコンフィギュレーション例，21 ページ


## Embedded Event Manager について

Embedded Event Manager（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。

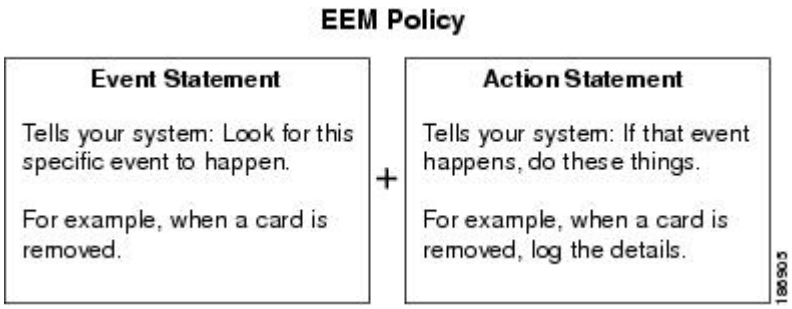
- アクション文：電子メールの送信、インターフェイスのディセーブル化など、イベントから回復するために EEM が実行できるアクション。
- ポリシー：イベントのトラブルシューティングまたはイベントからの回復を目的とした 1 つまたは複数のアクションとペアになったイベント。

# EEM 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。

次の図に、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**




EEM ポリシーを設定するには、CLI または VSH スクリプトを使用します。

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。ユーザポリシーを設定する場合には、CLI によるユーザ ポリシーの定義，（7 ページ）を参照してください。

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。

上書きポリシーを設定する場合は、ポリシーの上書き，（14 ページ）を参照してください。

（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。

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。

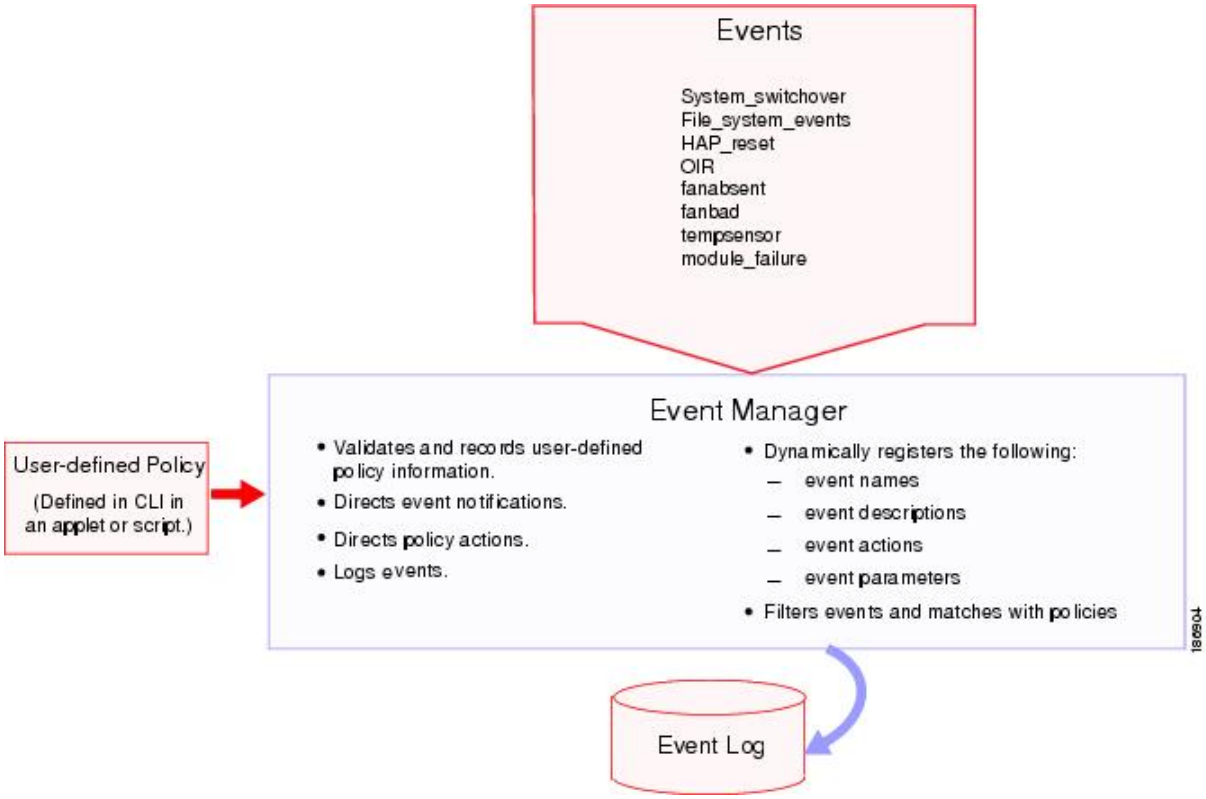
# EEM イベント文

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。

次の図は、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。複数イベントの設定の詳細については、EEM イベント相関, （5 ページ）を参照してください。

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。

# EEM アクション文

アクション文は、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。
- 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。

（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。

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。VHS スクリプトでポリシーを設定する場合は、VSH スクリプトによるポリシーの定義, （13 ページ）を参照してください。

# EEM イベント相関

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、not）を使用し、カウントと時間を合わせて、カスタム アクションをトリガーするためのこれらイベントの組み合わせを定義できます。

（注） EEM イベント相関の設定方法の詳細については、CLI によるユーザ ポリシーの定義, （7 ページ）を参照してください。

# EEM 仮想化サポート

ログインした仮想デバイス コンテキスト（VDC）で、EEM を設定します。デフォルトでは、Cisco NX-OS はデフォルト VDC に配置します。モジュールベースのイベントに対応するポリシーを設定する場合は、この VDC を使用する必要があります。

すべての VDC ですべてのアクションまたはイベントを確認できるわけではありません。ポリシーを設定するには、network-admin または vdc-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すべて Cisco NX-OS システム イメージにバンドルされており、追加費用は一切発生しません。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 には次の前提条件があります。

- EEM を設定するには、network-admin または vdc-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、モジュール、モジュール障害、oir、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。

# 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。

**表 *1*：デフォルトの *EEM* パラメータ**

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

# EEM の設定

## 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。

### はじめる前に

正しい VDC 内にいることを確認します。VDC を変更するには、**switchto vdc** コマンドを使用します。

### 手順

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | switch# **configure terminal** | 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 | switch(config)# **event manager environment** *variable-name variable-value* | 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 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 | switch(config)# **show event manager environment** {*variable-name* \| **all**} | （任意）<br>設定した環境変数に関する情報を表示します。ストリングは引用符で囲みます。 |
| ステップ 4 | switch(config)# **copy running-config startup-config** | （任意）<br>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。 |

次に、環境変数を定義する例を示します。

```
switch# configure terminal
switch(config)# event manager environment emailto "admin@anyplace.com"
switch(config)# show event manager environment all
switch(config)# copy running-config startup-config
```

## CLI によるユーザ ポリシーの定義

CLI を使用したユーザ ポリシーを定義できます。

### はじめる前に

正しい VDC 内にいることを確認します。VDC を変更するには、**switchto vdc** コマンドを使用します。

### 手順

| | コマンドまたはアクション | 目的 |
|---|---|---|
| **ステップ 1** | switch# **configure terminal** | 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** | switch(config)# **event manager applet** *applet-name* | 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** | switch(config-applet)# **description** *policy-description* | （任意）<br>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** | switch(config-applet)# **event** *event-statement* | ポリシーのイベント文を設定します。イベント文の設定，（9 ページ）を参照してください。<br>イベント文が複数の場合は、ステップ 4 を繰り返します。 |
| **ステップ 5** | switch(config-applet)# **tag** *tag* {**and** \| **andnot** \| **or**} *tag* [**and** \| **andnot** \| or {*tag*}] {**happens** *occurs* **in** *seconds*} | （任意）<br>ポリシー内の複数のイベントを相互に関連付けます。<br>*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** | switch(config-applet)# **action** *number*[*.number2*] *action-statement* | ポリシーのアクション文を設定します。アクション文の設定，（11 ページ）を参照してください。<br>アクション文が複数の場合は、ステップ 6 を繰り返します。 |
| **ステップ 7** | switch(config-applet)# **show event manager policy-state** *name* [**module** *module-id*] | 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** | switch(config-applet)# **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。 |

次に、CLI を使用してユーザ ポリシーを定義する例を示します。

```
switch# configure terminal
switch(config)# event manager applet monitorShutdown
switch(config-applet)# description "Monitors interface shutdown."
switch(config-applet)# event cli match "shutdown"
switch(config-applet)# tag one or two happens 1 in 10000
switch(config-applet)# action 1.0 cli local show interface e 3/1
switch(config-applet)# show event manager policy-state monitorShutdown
switch(config-applet)# copy running-config startup-config
```

# 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 cli** [**tag** *tag*] **match** *expression* [**count** *repeats* \| **time** *seconds*]<br>例：<br>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。<br>**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。<br>*repeats* 引数の範囲は 1 ～ 65000 です。*seconds* 引数の範囲は 0 ～ 4294967295 秒です。0 は無制限を示します。<br>ストリングは引用符で囲みます。 |
| **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**}]<br>例：<br>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。<br>**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。<br>*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。 |
| **event fanabsent** [**fan** *number*] **time** *seconds*<br>例：<br>switch(config-applet)# event fanabsent time 3000 | 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。<br>*seconds* の範囲は 10 ～ 64000 です。 |

| コマンド | 目的 |
|---|---|
| **event fanbad** [**fan** *number*] **time** *seconds*<br>例：<br>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。*seconds* の範囲は 10 ～ 64000 です。 |
| **event oir** [**tag** *tag*] {**fan** \| **module** \| **powersupply**} {**anyoir** \| **insert** \| **remove**} [*number*]<br>例：<br>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。<br>**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。<br>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。<br>• ファン番号：モジュール依存<br>• モジュール番号：デバイス依存<br>• 電源モジュール番号：範囲は 1 ～ 3 |
| **event policy-default count** *repeats* [**time** *seconds*]<br>例：<br>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。<br>*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 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*<br>例：<br>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。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。<br>**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。<br>*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。 |

| コマンド | 目的 |
|---|---|
| **event storm-control**<br>例：<br>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 sysmgr memory** [**module** *module-num*] **major** *major-percent* **minor** *minor-percent* **clear** *clear-percent*<br>例：<br>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。 |
| **event temperature** [**module** *slot*] [**sensor** *number*] **threshold** {**any** \| **major** \| **minor**}<br>例：<br>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 track** [**tag** *tag*] *object-number* **state** {**any** \| **down** \| **up**}<br>例：<br>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。<br>**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。<br>*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**]<br>例：<br>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。<br>*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。<br>ストリングは引用符で囲みます。 |

| コマンド | 目的 |
|---|---|
| **action** *number*[*.number2*] **counter name** *counter* **value** *val* **op** {**dec** \| **inc** \| **nop** \| **set**}<br>例：<br>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。<br>*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。<br>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**<br>例：<br>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。<br>*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**<br>例：<br>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。<br>*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*]]<br>例：<br>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。<br>*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*]}<br>例：<br>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。<br>*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。<br>ストリングは引用符で囲みます。 |

| コマンド | 目的 |
|---|---|
| **action** *number*[*.number2*] **syslog** [**priority** *prio-val*] **msg** *error-message*<br>例：<br>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。<br>*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

正しい VDC 内にいることを確認します。VDC を変更するには、**switchto vdc** コマンドを使用します。

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。

### 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** ファイルを次のシステム ディレクトリにコピーします。
`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。

### はじめる前に

正しい VDC 内にいることを確認します。VDC を変更するには、**switchto vdc** コマンドを使用します。

### 手順

| | コマンドまたはアクション | 目的 |
|---|---|---|
| **ステップ 1** | switch# **configure terminal** | 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** | switch(config)# **event manager policy** *policy-script* | 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 文字の英数字を使用できます。 |
| **ステップ 3** | switch(config)# **show event manager policy internal** *name* | （任意）<br>設定したポリシーに関する情報を表示します。 |
| **ステップ 4** | switch(config)# **copy running-config startup-config** | （任意）<br>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。 |

次に、VSH スクリプト ポリシーを登録してアクティブにする例を示します。

```
switch# configure terminal
switch(config)# event manager policy moduleScript
switch(config)# show event manager policy internal moduleScript
switch(config)# copy running-config startup-config
```

## ポリシーの上書き

システム ポリシーは上書き可能です。

### はじめる前に

正しい VDC 内にいることを確認します。VDC を変更するには、**switchto vdc** コマンドを使用します。

### 手順

| | コマンドまたはアクション | 目的 |
|---|---|---|
| **ステップ 1** | switch# **configure terminal** | 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 | switch(config)# **show event manager policy-state** *system-policy* | 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。 |
| ステップ 3 | switch(config)# **event manager applet** *applet-name* **override** *system-policy* | 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 文字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 | switch(config-applet)# **description** *policy-description* | （任意）<br>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 | switch(config-applet)# **event** *event-statement* | ポリシーのイベント文を設定します。ページ 14-10 の「イベント文の設定」を参照してください。 |
| ステップ 6 | switch(config-applet)# **action** *number action-statement* | ポリシーのアクション文を設定します。ページ 14-13 の「アクション文の設定」を参照してください。<br>アクション文が複数の場合は、ステップ 6 を繰り返します。 |
| ステップ 7 | switch(config-applet)# **show event manager policy-state** *name* | （任意）<br>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 | switch(config-applet)# **copy running-config startup-config** | この設定変更を保存します。 |

次の例に、ポリシーの上書き方法を示します。

```
switch# configure terminal
switch(config)# 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
switch(config)# event manager applet ethport override _ethpm_link_flap
switch(config-applet)# description "Overrides link flap policy"
switch(config-applet)# event policy-default count 2 time 1000
switch(config-applet)# action 1.0 syslog priority warnings msg "Link is flapping."
switch(config-applet)# show event manager policy-state ethport
switch(config-applet)# copy running-config startup-config
```

# 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 が syslog による登録に利用可能であることを確認します。

syslog デーモンを設定して実行します。

### 手順

| | コマンドまたはアクション | 目的 |
|---|---|---|
| **ステップ 1** | switch# **configure terminal** | 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** | switch(config)# **event manager applet** *applet-name* | 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** | switch(config)# **event syslog** [**tag** *tag*] {**occurs** *number* \| **period** *seconds* \| **pattern** *msg-text* \| **priority** *priority*} | 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。<br>• **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。<br>• **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。<br>• **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。<br>• **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。<br>• **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 | switch(config)# **copy running-config startup-config** | （任意）<br>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。 |

次に、EEM パブリッシャとして syslog を設定する例を示します。

```
switch# configure terminal
switch(config)# event manager applet abc
switch(config-applet)# event syslog occurs 10
switch(config-applet)# copy running-config startup-config
```

# Tcl スクリプト トリガーのための CLI を使用したユーザ ポリシーの定義

### はじめる前に

EEM ポリシーを介してトリガーされる Tcl スクリプトをスイッチのブートフラッシュにコピーします。

### 手順

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | switch# **configure terminal** | 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 | switch(config)# **event manager applet** *applet-name* | 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 | switch(config-applet)# **description** *policy-description* | （任意）<br>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 | switch(config-applet)# **event** *event-statement* | ポリシーのイベント文を設定します。イベント文の設定，（9 ページ）を参照してください。<br>イベント文が複数の場合は、ステップ 4 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 5 | switch(config-applet)# **tag** *tag* {**and** \| **andnot** \| **or**} *tag* [**and** \| **andnot** \| or {*tag*}] {**happens** *occurs* **in** *seconds*} | （任意）<br>ポリシー内の複数のイベントを相互に関連付けます。<br>*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 | switch(config-applet)# **action** *number*[.*number2*] *action-statementtcl-filename* | ポリシーのアクション文を設定します。アクション文の設定，（11 ページ）を参照してください。<br>アクション文が複数の場合は、ステップ 6 を繰り返します。 |
| ステップ 7 | switch(config-applet)# **show event manager policy-state** *name* [**module** *module-id*] | 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 | switch(config-applet)# **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。 |

Tcl サンプル ファイル（Vlan.tcl）。ブートフラッシュにこのファイルをコピーします。ファイルを実行すると、VLAN 99 を作成し、その名前を指定します。

```
set i 1
while {$i<100} {
cli configure terminal
cli vlan $i
cli name VLAN$i
cli no shutdown
cli exit
incr i
}
```

次に、CLI を使用してポリシーを定義し、イベントがトリガーされた場合アクション ステートメントを使用して Tcl スクリプトを起動する例を示します。

```
switch# configure terminal
switch(config)# event manager applet TCL
switch(config-applet)# description "Triggers TCL Script"
switch(config-applet)# event cli match "shutdown"
switch(config-applet)# action 1.0 cli local tclsh VLAN.tcl
switch(config-applet)# copy running-config startup-config
```

# Python スクリプト トリガーのための CLI を使用したユーザ ポリシーの定義

## はじめる前に

EEM ポリシーを介してトリガーされる Python スクリプトをスイッチのブートフラッシュにコピーします。

## 手順

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | switch# **configure terminal** | 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 | switch(config)# **event manager applet** *applet-name* | 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 | switch(config-applet)# **description** *policy-description* | （任意）<br>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 | switch(config-applet)# **event** *event-statement* | ポリシーのイベント文を設定します。イベント文の設定，（9 ページ）を参照してください。<br>イベント文が複数の場合は、ステップ 4 を繰り返します。 |
| ステップ 5 | switch(config-applet)# **tag** *tag* {**and** \| **andnot** \| **or**} *tag* [**and** \| **andnot** \| or {*tag*}] {**happens** *occurs* **in** *seconds*} | （任意）<br>ポリシー内の複数のイベントを相互に関連付けます。<br>*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 | switch(config-applet)# **action** *number*[*.number2*] *action-statementpython-filename* | ポリシーのアクション文を設定します。アクション文の設定，（11 ページ）を参照してください。<br>アクション文が複数の場合は、ステップ 6 を繰り返します。 |
| ステップ 7 | switch(config-applet)# **show event manager policy-state** *name* [**module** *module-id*] | 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 | switch(config-applet)# **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。 |

サンプル Python ファイル（Python.py）。ブートフラッシュにこのファイルをコピーします。

```
import re
import cisco
cisco.cli ("show interface eth 1/1-32 transceiver detail >> bootflash:link_flap.txt")
```

次に、CLI を使用してポリシーを定義し、イベントがトリガーされた場合アクション ステートメントを使用して Python スクリプトを起動する例を示します。

```
switch# configure terminal
switch(config)# event manager applet PYTHON
switch(config-applet)# description "Triggers PYTHON Script"
switch(config-applet)# event cli match "shutdown"
switch(config-applet)# action 1.0 cli local python bootflash:Python.py
switch(config-applet)# copy running-config startup-config
```

# EEM 設定の確認

EEM の設定情報を表示するには、次のコマンドのいずれかを入力します。

| 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 internal** *[policy-name]* [**inactive**] | 設定したポリシーに関する情報を表示します。 |
| **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。 |

| コマンド | 目的 |
|---|---|
| **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

__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方法の例を示します。

```
switch# configure terminal
switch(config)# event manager applet ethport override __ethpm_link_flap
switch(config-applet)# event policy-default count 2 time 1000
switch(config-applet)# action 1 cli conf t
switch(config-applet)# action 2 cli int et1/1
switch(config-applet)# 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。

```
switch# configure terminal
switch(config)# event manager applet TEST
switch(config-applet)# event cli match "conf t"
switch(config-applet)# action 1.0 snmp-trap strdata "Configuration change"
switch(config-applet)# 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。

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