



Embedded Event Manager コマンドリファレンス、Cisco IOS XE Release 3SE (Catalyst 3850 スイッチ)

初版：2013 年 01 月 22 日

最終更新：2013 年 01 月 22 日

シスコシステムズ合同会社

〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー

<http://www.cisco.com/jp>

お問い合わせ先：シスコ コンタクトセンター

0120-092-255 (フリーコール、携帯・PHS含む)

電話受付時間：平日 10:00～12:00、13:00～17:00

<http://www.cisco.com/jp/go/contactcenter/>

【注意】 シスコ製品をご使用になる前に、安全上の注意（www.cisco.com/jp/go/safety_warning/）をご確認ください。本書は、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。また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。



目次

A から action snmp までのコマンド 1

A から action snmp までのコマンド 1

action add	2
action append	4
action break	6
action cli	8
action comment	11
action context retrieve	13
action context save	15
action else	17
action end	19
action foreach	21
action gets	23
action if	25
action ifgoto	28
action increment	31
action info type interface-names	33
action info type snmp getid	35
action info type snmp inform	38
action info type snmp oid	41
action info type snmp trap	46
action info type snmp var	49
action multiply	53
action puts	56
action regexp	58
action set (EEM)	60

action string から D までのコマンド 63

action string から D までのコマンド	63
action string compare	64

action string equal	66
action string first	68
action string index	70
action string last	72
action string length	74
action string match	76
action string range	78
action string replace	80
action string tolower	82
action string toupper	84
action string trim	86
action string trimleft	88
action string trimright	90
action subtract	92
action track read	94
action track set	96
action while	99
attribute (EEM)	101
description (EEM)	103

E から event manager までのコマンド 105

E から event manager までのコマンド 105

event application	106
event identity	109
event ipsla	112
event manager applet	118
event manager detector routing	122
event manager directory user	124
event manager environment	127
event manager history size	129
event manager run	131
event manager scheduler clear	134
event manager scheduler hold	137
event manager scheduler modify	140
event manager scheduler release	143
event manager scheduler suspend	145

event mat から R までのコマンド 147**event mat から R までのコマンド 147**

- event mat 148**
- event neighbor-discovery 151**
- event nf 156**
- event none 160**
- event routing 163**
- event snmp 166**
- event snmp-notification 173**
- event snmp-object 178**
- event track 182**

S から Z までのコマンド 185**S から Z までのコマンド 185**

- show event manager directory user 186**
- show event manager environment 188**
- show event manager history events 190**
- show event manager history traps 193**
- show event manager metric processes 195**
- show event manager policy active 197**
- show event manager policy available 200**
- show event manager policy pending 203**
- show event manager scheduler 207**
- track stub-object 209**
- trigger (EEM) 211**



A から action snmp までのコマンド

- [A から action snmp までのコマンド, 1 ページ](#)

A から action snmp までのコマンド

action add

Embedded Event Manager (EEM) アプレットがトリガーされたときに 2 個の変数の値を追加するアクションを指定するには、アプレット コンフィギュレーションモードで **action add** コマンドを使用します。追加のアクションを取り消すには、このコマンドの **no** 形式を使用します。

action label add {*long-integer*|*variable-name*} {*long-integer*|*variable-name*}

no action label add

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できません）。アクションは、ラベルをソート キーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
add	2 個の変数の値を追加します。
<i>variable-name</i>	変数名として配置されるストリング値。
<i>long-integer</i>	変数に追加される長整数値。

コマンド デフォルト

デフォルトでは、EEM アプレット内に設定された変数の値は変更されません。

コマンド モード

アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン

このコマンドを使用して、2 個の変数の値を追加できます。結果は **_result** という名前の変数に保存されます。変数の値は長整数である必要があり、これ以外の場合はアクションが失敗します。

例

次に、2 つの変数の値を追加するように EEM アプレットを設定する例を示します。

```
Router(config)#event manager applet one
Router(config-applet)#action 1.0 set $var1 10
Router(config-applet)#action 1.0 set $var2 20
Router(config-applet)#action 1.0 add $var1 $var2
Router(config-applet)#
```

関連コマンド

コマンド	説明
event manager applet	Embedded Event Manager にイベント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。

action append

Embedded Event Manager (EEM) アプレットがトリガーされたときに、特定のストリング値を現在の変数値に追加するアクションを指定するには、アプレットコンフィギュレーションモードで **action append** コマンドを使用します。追加のアクションを取り消すには、このコマンドの **no** 形式を使用します。

action label append *variable-name* [*variable-value*]

no action label add

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できません）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
append	特定のストリング値を指定された変数の現在の値に追加します。
<i>variable-name</i>	変数名として配置されるストリング値。
<i>variable-value</i>	（任意）指定した変数名の値に追加される長整数値。

コマンド デフォルト

デフォルトでは、EEM アプレット内に設定された変数の値は変更されません。

コマンド モード

アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン このコマンドを使用して、特定のストリング値を変数の現在の値に追加できます。変数が存在しない場合、作成されて特定の値に設定されます。

例 次に、指定された変数の現在の値に、特定のストリング値を追加するようにEEMアプレットを設定する例を示します。

```
Router(config)#event manager applet one
Router(config-applet)#action 1.0 set $var1 10
Router(config-applet)#action 1.0 append $var1 12
Router(config-applet)#
```

関連コマンド

コマンド	説明
event manager applet	Embedded Event Manager にイベント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。

action break

Embedded Event Manager (EEM) アプレットがトリガーされたときに、アクションのループを終了するアクションを指定するには、アプレット コンフィギュレーションモードで **action break** コマンドを使用します。中断アクションをディセーブルにするには、このコマンドの **no** 形式を使用します。

action label break

no action label break

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できます）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
break	アクションのループをただちに終了させます。

コマンド デフォルト

デフォルトでは、EEM アプレット内に設定されたアクションのループは終了しません。

コマンド モード

アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン

このコマンドを使用して、関連する最後のアクションまでのすべてのアクションをスキップできます。

例

次に、アクションのループを中断するように EEM アプレットを設定する例を示します。

```
Router(config)# event manager applet loop
Router(config-applet)# event none
Router(config-applet)# action 1 while 1 eq 1
Router(config-applet)# action 2 break
Router(config-applet)# action 3 end
```

関連コマンド

コマンド	説明
event manager applet	Embedded Event Manager にイベント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。

action cli

Embedded Event Manager (EEM) アプレットがトリガーされたときに、Cisco IOS コマンドライン インターフェイス (CLI) のアクションを実行するように指定するには、アプレットコンフィギュレーション モードで **action cli** コマンドを使用します。CLI コマンドを実行するアクションを削除するには、このコマンドの **no** 形式を使用します。

action *label cli command cli-string* [**pattern** *pattern-string*]

no action *label cli command cli-string*

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できません）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
command	Cisco IOS CLI に送信するメッセージを指定します。
<i>cli-string</i>	実行する CLI コマンド。ストリングに空白が含まれる場合は二重引用符で囲みます。
pattern	（任意）コマンドストリングが入力を求める場合にだけ、 command cli-string に対する正規表現の応答パターンを指定します。
<i>pattern-string</i>	（任意） pattern キーワードで指定されるアクションを指定します。次の応答プロンプトに一致する正規表現の <i>pattern-string</i> を指定する必要があります。

コマンド デフォルト

EEM アプレットがトリガーされたときに、CLI コマンドは実行されません。

コマンド モード

アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.3(14)T	このコマンドが導入されました。

リリース	変更内容
12.2(28)SB	このコマンドが、Cisco IOS Release 12.2(28)SB に統合されました。
12.2(18)SXF4	このコマンドが Cisco IOS Release 12.2(18)SXF4 に統合され、ソフトウェアモジュラリティイメージだけをサポートするようになりました。
12.2(33)SRA	このコマンドが、Cisco IOS Release 12.2(33)SRA に統合されました。
12.2(18)SXF5	このコマンドが、Cisco IOS Release 12.2(18)SXF5 に統合されました。
12.2(33)SXH	pattern キーワードが追加されました。

使用上のガイドライン

action cli コマンドを使用して、EEM アプレットがトリガーされたときに Cisco IOS CLI コマンドを実行するアクションを指定します。**pattern** キーワードはオプションで、コマンド文字列が入力を求める場合にだけ使用します。

Cisco IOS CLI コマンドには 2 種類あります。

- 通常：通常のルータ プロンプトが続いて表示される出力を生成する Cisco IOS CLI コマンドです。**action cli** コマンドは、通常のルータ プロンプトを受け取ると終了します。
- 応答：通常のルータ プロンプトが表示される前に 1 つ以上の質問を行う Cisco IOS CLI コマンドです（たとえば、「confirm」を「yes」または「no」の入力で終える必要がある場合など）。

action cli コマンドは、オプションの **pattern** キーワードで指定された応答プロンプトを受け取ったときに終了します。次の応答プロンプトに一致する正規表現パターンを指定する必要があります。正しくないパターンを指定すると、**action cli** コマンドが、maxrun タイマー期限切れによるアプレット実行タイムアウトまで、待ち続けることとなります。

vty 行は、**line vty** CLI コンフィギュレーション コマンドを使用して設定された vty 行のプールから割り当てられます。EEM によって vty 行が使用されていない場合で、使用可能な vty 行がある場合、EEM では、vty 行が使用されます。EEM によって vty 行がすでに使用されている場合で、使用可能な 3 行以上の vty 行がある場合も、EEM では、vty 行が使用されます。3 行よりも少ない vty 行が使用可能な場合、残りの vty 行は Telnet で使用するために予約されているので、接続は失敗することに注意してください。

次の表に、**action cli** コマンドを実行したときに設定される組み込み変数を示します。

表 1: action cli コマンドの EEM 組み込み変数

組み込み変数	説明
\$_cli_result	CLI コマンドの実行結果。

例

次に、Cisco IOS **interface loopback** CLI コマンドが 3 回設定されたときに EEM アプレットが実行されるように指定する例を示します。アプレットは、ループバックインターフェイスが動作可能であることを確実にするために **no shutdown** コマンドを実行します。

```
Router(config)# event manager applet cli-match
Router(config-applet)# event cli command {.*interface loopback*} sync yes occurs 3
Router(config-applet)# action 1.0 cli command "no shutdown"
```

次に、**pattern** キーワードにより、**clear counters Ethernet0/1** コマンドに *confirm* 引数が指定されたときに EEM アプレットが実行されるように指定する例を示します。

```
Router(config)# event manager applet cli-match
Router(config-applet)# action 1.0 cli command "enable"
Router(config-applet)# action 2.0 cli command "clear counters Ethernet0/1" pattern "confirm"
Router(config-applet)# action 3.0 cli command "y"
!
```

関連コマンド

コマンド	説明
event manager applet	Embedded Event Manager にイベント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。

action comment

Embedded Event Manager (EEM) アプレットがトリガーされたときに、アプレットにコメントを追加するアクションを指定するには、アプレットコンフィギュレーションモードで **action comment** コマンドを使用します。コメントをディセーブルにするには、このコマンドの **no** 形式を使用します。

action label comment string

no action label comment

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できます）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
comment	アプレットにコメントを追加します。
<i>string</i>	コメントとして配置する一連の文字（埋め込みスペースを含む）。

コマンド デフォルト

デフォルトでは、アプレットにコメントは追加されません。

コマンド モード

アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン

このコマンドを使用して、アプレットにコメントを追加できます。この結果、アプレットが実行されたときに no-op になります。

例

次に、アプレットにコメントを追加する例を示します。

```
Router(config)#event manager applet one
Router(config-applet)#action 1.0 comment keyvalue
Router(config-applet)#
```

関連コマンド

コマンド	説明
event manager applet	Embedded Event Manager にイベント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。

action context retrieve

Embedded Event Manager (EEM) アプレットがトリガーされたときに、特定のコンテキスト名キーのセットによって識別される変数を取得するアクションを指定するには、アプレット コンフィギュレーションモードで **action context retrieve** コマンドを使用します。取得アクションを取り消すには、このコマンドの **no** 形式を使用します。

action *label* **context retrieve** **key** *key-name* **variable** *variable-name-pattern*

no **action** *label* **context retrieve**

構文の説明

<i>label</i>	一意な識別子（任意の文字列値を指定できます）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。文字列に空白が含まれる場合は二重引用符で囲みます。
context retrieve	特定のコンテキスト名キーで識別される変数を取得するために使用します。
key <i>key-name</i>	コンテキスト名キーを指定します。
variable <i>variable-name-pattern</i>	変数の説明を示します。

コマンド デフォルト

デフォルトでは、特定のコンテキスト名キーのセットで指定した変数は取得されません。

コマンド モード

アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン このコマンドを使用して、特定のコンテキスト名キーのセットによって識別される変数を取得できます。取得された情報は、自動的にコンテキスト データベースから削除されます。

コマンドで指定された変数の情報が取得されるのは、**action context save** コマンドを使用して、対応する **context save** コールで同じ名前の変数が保存された場合のみです。

例 次に、特定のコンテキスト名キーのセットによって識別される変数を取得するように EEM アプレットを設定する例を示します。

```
Router(config)#event manager applet one
Router(config-applet)#action 1.0 context retrieve key pki-72a variable var1
Router(config-applet)#
```

関連コマンド

コマンド	説明
event manager applet	Embedded Event Manager にイベント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。
action context save	このコマンドは、複数のポリシートリガーにわたって情報を保存するために使用されます。

action context save

Embedded Event Manager (EEM) アプレットがトリガーされたときに、複数のポリシー トリガーにわたって情報を保存するアクションを指定するには、アプレットコンフィギュレーションモードで **action context save** コマンドを使用します。保存された情報を削除するには、このコマンドの **no** 形式を使用します。

action label context save key key-name variable variable-name-pattern

no action label context save

構文の説明

<i>label</i>	一意な識別子（任意のSTRING値を指定できます）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。STRINGに空白が含まれる場合は二重引用符で囲みます。
context save	複数のポリシー トリガーにわたって情報を保存するために使用されます。
key key-name	コンテキスト名キーを指定します。
variable variable-name-pattern	変数の説明を示します。

コマンド デフォルト

デフォルトでは、情報は複数のポリシー トリガーにわたって保存されません。

コマンド モード

アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。

使用上のガイドライン

action context save コマンドを使用して、複数のポリシー トリガーにわたって情報を保存できます。このコマンドは、識別情報と同じコンテキスト名キーを持つ、特定のパターンに一致する変数を保存します。保存された情報は、**action context retrieve** コマンドを使用して、別のアプレットから取得できます。

保存された情報が取得されると、その情報は自動的にコンテキストデータベースから削除されます。情報を取得したアプレットにその情報を保存するには、そのアプレットで再度 **action context save** コマンドを実行する必要があります。

例

次に、複数のポリシー トリガーにわたって情報を保存するように EEM アプレットを設定する例を示します。

```
Router(config)#event manager applet one
Router(config-applet)#action 1.0 context save key pki-72a variable var1
Router(config-applet)#
```

関連コマンド

コマンド	説明
event manager applet	Embedded Event Manager にイベント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。
action context retrieve	特定のコンテキスト名キーで識別される変数を取得します。

action else

Embedded Event Manager (EEM) アプレットがトリガーされたときに、if/else 条件付きアクションブロックの else 条件付きアクションブロックの開始を識別するには、アプレット コンフィギュレーションモードで **action else** コマンドを使用します。else 条件付きアクションブロックを削除するには、このコマンドの **no** 形式を使用します。

action label else

no action label else

構文の説明

<i>label</i>	一意な識別子（任意の文字列値を指定できます）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。文字列に空白が含まれる場合は二重引用符で囲みます。
--------------	---

コマンド デフォルト

アプレットコンフィギュレーションモードでコマンドを入力しなかった場合、設定を終了したときに個別のアプレットが登録されません。

コマンド モード

アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン

action else コマンドを使用して、else 条件付きアクションブロックを識別します。このアプレットにステートメントが関連付けられていない場合、イベントは引き続きトリガーされますがアクションまたは結果は発生しません。設定の終了時に、このアプレットにステートメントが関連付けられていないことを示す警告メッセージが表示されます。

Tool Command Language (Tcl) と CLI アプレットベースの EEM ポリシーの間でカスタマー向けの一貫したユーザ インターフェイスを実現するには、次の基準に従います。

- Tcl ベースの実装では、イベント仕様基準は TCL で記述されます。
- アプレットベースの実装では、イベント仕様データは CLI アプレット サブモード コンフィギュレーション文を使用して記述されます。

アプレットベースの実装では、一部のキーワードが必要以上に長いか、ハイフンで連結されているように見えますが、これは Tcl ベースの実装が最初に開発され、導入されているためです。

アプレット コンフィギュレーション モードを開始するには、グローバル コンフィギュレーション モードを開始後、**event manager applet *applet-name*** コマンドを使用します。アプレット コンフィギュレーション モードでは、**config** プロンプトが、**(config-applet)#** に変わります。アプレット コンフィギュレーション モードでは、3 種類のコンフィギュレーション文がサポートされています。

- **event** : このアプレットの実行の原因となる、イベント基準を指定します。
- **action** : 組み込みアクションを実行します。
- **set** : アプレット変数を設定します (現在、サポートされている変数は `_exit_status` のみです)。

例

次に、else アクションブロックの開始を識別する例を示します。

```
Router(config)# event manager applet action
Router(config-applet)# action label1 if $var eq 0
Router(config-applet)# action label2 else
Router(config-applet)# end
```

関連コマンド

コマンド	説明
action elseif	EEMアプレットがトリガーされたときに、elseif条件付きアクションブロックの開始を識別します。
action if	EEMアプレットがトリガーされたときに、if条件付きアクションブロックの開始を識別します。

action end

Embedded Event Manager (EEM) アプレットがトリガーされたときに、if/else および while 条件付きアクションブロックで、条件付きアクションブロックの終了を識別するには、アプレット コンフィギュレーション モードで **action end** コマンドを使用します。end 条件付きアクションブロックを削除するには、このコマンドの **no** 形式を使用します。

action label end

no action label end

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できます）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
--------------	---

コマンド デフォルト

アプレット コンフィギュレーション モードでコマンドを入力しなかった場合、設定を終了したときに個別のアプレットが削除されます。

コマンド モード

アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。

使用上のガイドライン

action end コマンドを使用して、if/else および while 条件付きアクションブロックで、条件付きアクションブロックの終了を識別します。

Tool Command Language (Tcl) と CLI アプレットベースの EEM ポリシーの間でカスタマー向けの一貫したユーザ インターフェイスを実現するには、次の基準に従います。

- Tcl ベースの実装では、イベント仕様基準は TCL で記述されます。
- アプレットベースの実装では、イベント仕様データは CLI アプレット サブモード コンフィギュレーション文を使用して記述されます。

アプレットベースの実装では、一部のキーワードが必要以上に長い、ハイフンで連結されているように見えますが、これは Tcl ベースの実装が最初に開発され、導入されているためです。

アプレット コンフィギュレーション モードを開始するには、グローバル コンフィギュレーション モードを開始後、**event manager applet** *applet-name* コマンドを使用します。アプレット コンフィギュレーション モードでは、**config** プロンプトが、**(config-applet)#** に変わります。アプレット コンフィギュレーション モードでは、3 種類のコンフィギュレーション文がサポートされています。

- **event** : このアプレットの実行の原因となる、イベント基準を指定します。
- **action** : 組み込みアクションを実行します。
- **set** : アプレット変数を設定します (現在、サポートされている変数は `_exit_status` のみです)。

例

次に、条件付きアクションブロックの終了を識別する例を示します。

```
Router(config)# event manager applet action
Router(config-applet)# event none
Router(config-applet)# action 1.0 set x "5"
Router(config-applet)# action 2.0 if $x lt 10
Router(config-applet)# action 3.0 puts "$x is less than 10"
Router(config-applet)# action 4.0 end
```

関連コマンド

コマンド	説明
action else	EEM アプレットがトリガーされたときに、else 条件付きアクションブロックの開始を識別します。
action if	EEM アプレットがトリガーされたときに、if 条件付きアクションブロックの開始を識別します。

action foreach

デリミタをトークン化パターンとして使用して、入力文字列の反復を指定するには、アプレットコンフィギュレーションモードで **action foreach** コマンドを使用します。入力文字列の反復を削除するには、このコマンドの **no** 形式を使用します。

action label foreach [*string-iterator*] [*string-input*] [*string-delimiter*]

no action label foreach

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できます）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>string-iterator</i>	（任意）イテレータとして機能する一連の文字。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>string-input</i>	（任意）入力として機能する一連の文字。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>string-delimiter</i>	（任意）デリミタとして機能する一連の文字。ストリングに空白が含まれる場合は二重引用符で囲みます。デフォルトのデリミタは、スペースです。

コマンド デフォルト

アプレットコンフィギュレーションモードでコマンドを入力しなかった場合、設定を終了したときに個別のアプレットが削除されます。

コマンド モード

アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。

リリース	変更内容
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン

action foreach コマンドを使用して、デリミタをトークン化パターンとして使用して入力文字列を繰り返します。デリミタは、正規表現パターン文字列です。各反復で見つかったトークンは、与えられた **iterator** 変数に割り当てられます。すべての算術演算は、長整数としてオーバーフローのチェックなしで実行されます。このアプレットにステートメントが関連付けられていない場合、イベントは引き続きトリガーされますがアクションまたは結果は発生しません。設定の終了時に、このアプレットにステートメントが関連付けられていないことを示す警告メッセージが表示されます。

Tool Command Language (Tcl) と CLI アプレットベースの EEM ポリシーの間でカスタマー向けの一貫したユーザ インターフェイスを実現するには、次の基準に従います。

- Tcl ベースの実装では、イベント仕様基準は TCL で記述されます。
- アプレットベースの実装では、イベント仕様データは CLI アプレット サブモード コンフィギュレーション文を使用して記述されます。

アプレットベースの実装では、一部のキーワードが必要以上に長い、ハイフンで連結されているように見えますが、これは Tcl ベースの実装が最初に開発され、導入されているためです。

アプレット コンフィギュレーション モードを開始するには、グローバル コンフィギュレーション モードを開始後、**event manager applet *applet-name*** コマンドを使用します。アプレット コンフィギュレーション モードでは、**config** プロンプトが、**(config-applet)#** に変わります。アプレット コンフィギュレーション モードでは、3 種類のコンフィギュレーション文がサポートされています。

- **event** : このアプレットの実行の原因となる、イベント基準を指定します。
- **action** : 組み込みアクションを実行します。
- **set** : アプレット変数を設定します (現在、サポートされている変数は **_exit_status** のみです)。

例

次に、デリミタをトークン化パターンとして使用して入力文字列を繰り返す例を示します。

```
Router(config)# event manager applet action
Router(config-applet)# event none
Router(config-applet)# action 1 foreach _iterator "red blue green orange"
Router(config-applet)# action 2 puts "iterator is $_iterator"
Router(config-applet)# action 3 end
Router# event manager run action
iterator is red
iterator is blue
iterator is green
iterator is orange
Router#
```

action gets

Embedded Event Manager (EEM) アプレットがトリガーされたときに、同期アプレットのローカル tty から入力を取得して値を特定の変数に格納するには、アプレット コンフィギュレーションモードで **action gets** コマンドを使用します。ローカル tty から入力を受信するプロセスをキャンセルするには、このコマンドの **no** 形式を使用します。

action label gets variable

no action label gets

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できます）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>variable</i>	同期アプレットからの入力値を格納する可変語。

コマンド デフォルト

アプレット コンフィギュレーションモードでコマンドを入力しなかった場合、設定を終了したときに個別のアプレットが削除されます。

コマンド モード

アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン

action gets コマンドを使用して、同期アプレットのローカル tty から入力を取得し、特定の変数に値を格納します。このコマンドは非同期アプレットでは動作しません。アプレットはエラーなしで続行されますが、変数は設定されません。すべての算術演算は、長整数としてオーバーフローのチェックなしで実行されます。このアプレットにステートメントが関連付けられていない場

合、イベントは引き続きトリガーされますがアクションまたは結果は発生しません。設定の終了時に、このアプレットにステートメントが関連付けられていないことを示す警告メッセージが表示されます。

Tool Command Language (Tcl) と CLI アプレットベースの EEM ポリシーの間でカスタマー向けの一貫したユーザ インターフェイスを実現するには、次の基準に従います。

- Tcl ベースの実装では、イベント仕様基準は TCL で記述されます。
- アプレットベースの実装では、イベント仕様データは CLI アプレット サブモード コンフィギュレーション文を使用して記述されます。

アプレットベースの実装では、一部のキーワードが必要以上に長い、ハイフンで連結されているように見えますが、これは Tcl ベースの実装が最初に開発され、導入されているためです。

アプレット コンフィギュレーション モードを開始するには、グローバル コンフィギュレーション モードを開始後、**event manager applet *applet-name*** コマンドを使用します。アプレット コンフィギュレーション モードでは、**config** プロンプトが、**(config-applet)#** に変わります。アプレット コンフィギュレーション モードでは、3 種類のコンフィギュレーション文がサポートされています。

- **event** : このアプレットの実行の原因となる、イベント基準を指定します。
- **action** : 組み込みアクションを実行します。
- **set** : アプレット変数を設定します (現在、サポートされている変数は `_exit_status` のみです)。

例

次に、同期アプレットのローカル tty から入力を取得して値を格納する例を示します。

```
Router(config)# event manager applet action
Router(config-applet)# event none
Router(config-applet)# action label2 gets input
Router(config-applet)# action label3 syslog msg "Input entered was \"$input\""
```

関連コマンド

コマンド	説明
action puts	EEM アプレットがトリガーされたときに、同期アプレットのローカル tty に直接データを出力します。

action if

Embedded Event Manager (EEM) アプレットがトリガーされたときに、if条件付きブロックの開始を識別するには、アプレットコンフィギュレーションモードで **action if** コマンドを使用します。if条件付きアクションブロックを削除するには、このコマンドの **no** 形式を使用します。

action label if [*string-op-1*] { **eq** | **gt** | **ge** | **lt** | **le** | **ne** } [*string-op-2*]

no action label if

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できます）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>string-op-1</i>	（任意）ストリングの文字の範囲を置き換える一連の文字。
eq	2個の文字列を比較するために使用される Equal To キーワード。
gt	2個の文字列を比較するために使用される Greater Than キーワード。
ge	2個の文字列を比較するために使用される Greater Than or Equal To キーワード。
lt	2個の文字列を比較するために使用される Less Than キーワード。
le	2個の文字列を比較するために使用される Less Than or Equal To キーワード。
ne	2個の文字列を比較するために使用される Not Equal To キーワード。
<i>string-op-2</i>	（任意）ストリングの文字の範囲を置き換える一連の文字。

コマンド デフォルト

アプレットコンフィギュレーションモードでコマンドを入力しなかった場合、設定を終了したときに個別のアプレットが削除されます。

コマンド モード

アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン

action if コマンドを使用して、if 条件付きアクションブロックの開始を識別します。すべての算術演算は、長整数としてオーバーフローのチェックなしで実行されます。goto label オプションを使用している場合、if 機能ではアクションブロックの開始は識別されませんが、アプレットは、条件が真の場合は特定のラベルにジャンプするように指示されます。

このアプレットにステートメントが関連付けられていない場合、イベントは引き続きトリガーされますがアクションまたは結果は発生しません。設定の終了時に、このアプレットにステートメントが関連付けられていないことを示す警告メッセージが表示されます。

Tool Command Language (Tcl) と CLI アプレットベースの EEM ポリシーの間でカスタマー向けの一貫したユーザ インターフェイスを実現するには、次の基準に従います。

- Tcl ベースの実装では、イベント仕様基準は TCL で記述されます。
- アプレットベースの実装では、イベント仕様データは CLI アプレット サブモード コンフィギュレーション文を使用して記述されます。

アプレットベースの実装では、一部のキーワードが必要以上に長い、ハイフンで連結されているように見えますが、これは Tcl ベースの実装が最初に開発され、導入されているためです。

アプレット コンフィギュレーション モードを開始するには、グローバル コンフィギュレーションモードを開始後、**event manager applet applet-name** コマンドを使用します。アプレット コンフィギュレーションモードでは、config プロンプトが、(config-applet)# に変わります。アプレット コンフィギュレーションモードでは、3 種類のコンフィギュレーション文がサポートされています。

- **event** : このアプレットの実行の原因となる、イベント基準を指定します。
- **action** : 組み込みアクションを実行します。
- **set** : アプレット変数を設定します (現在、サポートされている変数は_exit_statusのみです)。

例

次に、if 条件付きブロックの開始を識別する例を示します。

```
Router(config)# event manager applet action
Router(config-applet)# event none
Router(config-applet)# action 1.0 set x "5"
Router(config-applet)# action 2.0 if $x lt 10
Router(config-applet)# action 3.0 puts "$x is less than 10"
Router(config-applet)# action 4.0 end
Router# event manager run action
5 is less than 10
Router#
```

関連コマンド

コマンド	説明
action elseif	EEM アプレットがトリガーされたときに、else/if 条件付きブロックの else 条件付きアクションブロックの開始を識別します。
action ifgoto	EEM アプレットがトリガーされたときに、条件が真の場合は特定のラベルにジャンプするようにアプレットに指示します。

action ifgoto

Embedded Event Manager (EEM) アプレットがトリガーされたときに、指定された条件が真の場合は、特定のラベルにジャンプするようにアプレットに指示するには、アプレット コンフィギュレーション モードで **action ifgoto** コマンドを使用します。アプレットジャンプのプロセスをキャンセルするには、このコマンドの **no** 形式を使用します。

action *label-1* **if** [*string-op-1*] {**eq**|**gt**|**ge**|**lt**|**le**|**ne**} [*string-op-2*] **goto** *label-2*

no action *label ifgoto*

構文の説明

<i>label-1</i>	一意な識別子（任意のストリング値を指定できます）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>string-op-1</i>	（任意）ストリングの文字の範囲を置き換える一連の文字。
eq	2 個の文字列を比較するために使用される Equal To キーワード。
gt	2 個の文字列を比較するために使用される Greater Than キーワード。
ge	2 個の文字列を比較するために使用される Greater Than Or Equal To キーワード。
lt	2 個の文字列を比較するために使用される Less Than キーワード。
le	2 個の文字列を比較するために使用される Less Than Or Equal To キーワード。
ne	2 個の文字列を比較するために使用される Not Equal To キーワード。
<i>string-op-2</i>	（任意）ストリングの文字の範囲を置き換える一連の文字。

<i>label-2</i>	一意な識別子（任意のストリング値を指定できます）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>label</i>	一意な識別子（任意のストリング値を指定できます）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。

コマンド デフォルト

アプレットコンフィギュレーションモードでコマンドを入力しなかった場合、設定を終了したときに個別のアプレットが削除されます。

コマンド モード

アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン

action ifgoto コマンドを使用して、指定された条件が真の場合は、特定のラベルにジャンプするようにアプレットに指示します。 **goto label** オプションを使用する場合、**action if** コマンドではアクションブロックの開始は識別されません。 **Goto** アクションは、**if/goto** フォーマット内でのみサポートされます。 **if** を使用しないで **goto** をシミュレーションするには、常に真のテストを使用します。すべての算術演算は、長整数としてオーバーフローのチェックなしで実行されます。このアプレットにステートメントが関連付けられていない場合、イベントは引き続きトリガーされますがアクションまたは結果は発生しません。設定の終了時に、このアプレットにステートメントが関連付けられていないことを示す警告メッセージが表示されます。

Tool Command Language (Tcl) と CLI アプレットベースの EEM ポリシーの間でカスタマー向けの一貫したユーザインターフェイスを実現するには、次の基準に従います。

- Tcl ベースの実装では、イベント仕様基準は TCL で記述されます。

- アプレットベースの実装では、イベント仕様データは CLI アプレット サブモード コンフィギュレーション文を使用して記述されます。

アプレットベースの実装では、一部のキーワードが必要以上に長い、ハイフンで連結されているように見えますが、これは Tcl ベースの実装が最初に開発され、導入されているためです。

アプレット コンフィギュレーション モードを開始するには、グローバル コンフィギュレーション モードで **event manager applet applet-name** コマンドを使用します。アプレット コンフィギュレーション モードでは、**config** プロンプトが、**(config-applet)#** に変わります。アプレット コンフィギュレーション モードでは、3 種類のコンフィギュレーション文がサポートされています。

- **event** : このアプレットの実行の原因となる、イベント基準を指定します。
- **action** : 組み込みアクションを実行します。
- **set** : アプレット変数を設定します (現在、サポートされている変数は `_exit_status` のみです)。

例

次に、特定のラベルにジャンプすることをアプレットに指示する例を示します。

```
Router(config)# event manager applet action
Router(config-applet)# event none
Router(config-applet)# action 1 set x "5"
Router(config-applet)# action 2 if $x lt 10 goto 4
Router(config-applet)# action 3 puts "skipping this"
Router(config-applet)# action 4 puts "jumped to action 4"
Router(config-applet)# action 5 end
Router# event manager run action
jumped to action 4
```

関連コマンド

コマンド	説明
action else	EEM アプレットがトリガーされたときに、else 条件付きアクションブロックの開始を識別します。
action if	EEM アプレットがトリガーされたときに、if 条件付きアクションブロックの開始を識別します。

action increment

Embedded Event Manager (EEM) アプレットがトリガーされたときに、変数の値を増分するアクションを指定するには、アプレット コンフィギュレーション モードで **action increment** コマンドを使用します。アプレットからアクションを削除するには、このコマンドの **no** 形式を使用します。

action *label* **increment** *variable-name* *long-integer*

no **action** *label* **increment**

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できます）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
increment	指定された長整数で、変数の値を増分します。
<i>variable-name</i>	変数名として配置されるストリング値。
<i>long-integer</i>	変数が増分する長整数値。

コマンド デフォルト

デフォルトでは、EEM アプレット内に設定された変数の値は変更されません。

コマンド モード

アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン

このコマンドを使用して、変数の値を増分できます。変数の値は長整数である必要があり、これ以外の場合はアクションが失敗します。

例

次に、変数の値を増分するように EEM アプレットを設定する例を示します。

```
Router(config)#event manager applet one
Router(config-applet)#action 1.0 set varname 20
Router(config-applet)#action 1.0 increment varname 12
Router(config-applet)#
```

関連コマンド

コマンド	説明
event manager applet	Embedded Event Manager にイベント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。

action info type interface-names

Embedded Event Manager (EEM) アプレットがトリガーされたときに、インターフェイス名を取得するには、アプレット コンフィギュレーション モードで **action info type interface-names** コマンドを使用します。インターフェイス名を取得するアクションをディセーブルにするには、このコマンドの **no** 形式を使用します。

action label info type interface-names [*include string-operator*| *exclude string-operator*| *regexp regular-expression*]

no action label info type

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できます）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
include	（任意）ストリングパターンを含むすべてのインターフェイス名が含まれます。
exclude	（任意）ストリングパターンを含むすべてのインターフェイス名が除外されます。
<i>string-operator</i>	（任意）インターフェイス名を含めるまたは除外するためのストリングパターン。
regexp	（任意）指定された正規表現に一致するすべてのインターフェイスを取得します。
<i>regular-expression</i>	（任意）正規表現パターン。たとえば、 <code>[^abc]</code> とします。

コマンド デフォルト

現在のすべてのインターフェイス名がデータベースから取得されます。

コマンド モード

アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン

action info type interface-names コマンドは、現在のインターフェイス名を取得し、`$_info_interface_names` 組み込み変数にスペースで区切ったリストとして格納します。

例

次に、インターフェイス名に「eth」が含まれるものを取得するように指定する例を示します。

```
Router# configure terminal
Router(config)# event manager applet interface-app
Router(config-applet)# action 1.2 info type interface-names include eth
```

関連コマンド

コマンド	説明
event manager applet	EEMにイベント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。

action info type snmp getid

SNMP `get` 操作中に簡易ネットワーク管理プロトコル (SNMP) エンティティから個別の変数を取得するには、アプレット コンフィギュレーション モードで **action info type snmp getid** コマンドを使用します。SNMP からの個別の変数の取得をディセーブルにするには、このコマンドの **no** 形式を使用します。

action label info type snmp getid *oid-value* [**community** *community-string*] [**ipaddr** *ip-address*]

no action label info type

構文の説明

<i>label</i>	一意な識別子 (任意のストリング値を指定できます)。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
getid	SNMP 変数を取得します。
<i>oid-value</i>	SNMP ドット付き表記でのデータエレメントのオブジェクト ID 値。オブジェクト ID は、一連の整数値またはテキスト文字列で表されます。たとえば、インターフェイス MIB のオブジェクト名は 1.3.6.1.2.1.2 または <code>iso.internet.mgmt.mib-2.interfaces</code> と表すことができます。 OID は、関連する MIB (CISCO-EMBEDDED-EVENT-MGR-MIB) 内にタイプとして定義され、各タイプはオブジェクト値を保持します。一部の OID 型のモニタリングが可能です。次のタイプが有効です。 <ul style="list-style-type: none"> • INTEGER_TYPE • COUNTER_TYPE • GAUGE_TYPE • TIME_TICKS_TYPE • COUNTER_64_TYPE • OCTET_PRIM_TYPE • OPAQUE_PRIM_TYPE

community	(任意) SNMP エンティティにアクセスするコミュニティ ストリングを指定します。
<i>community-string</i>	(任意) SNMP コミュニティ ストリング。コミュニティ ストリングは、SNMP エンティティにアクセスするためのパスワードと同様に機能します。ストリングは 1 ~ 32 文字の英数字で構成され、次のタイプのコミュニティ ストリングのいずれかに設定できます。 <ul style="list-style-type: none"> • ro : SNMP エンティティへの読み取り専用アクセスを設定します。このコミュニティ ストリングのデフォルト値は public です。 • rw : SNMP エンティティへの読み取り/書き込みアクセスを設定します。このコミュニティ ストリングのデフォルト値は private です。
ipaddr	(任意) SNMP エンティティの IP アドレスを指定します。
<i>ip-address</i>	(任意) SNMP get および set 操作のためにオブジェクトを取得する、ネットワーク管理システム (NMS) の IP アドレス。

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン 次の表に、SNMP get 操作から取得された変数が保存される組み込み変数を示します。

表 2 : action info コマンドの EEM 組み込み変数

組み込み変数	説明
<code>\$_info_snmp_sysname_oid</code>	sysName 変数の OID 値。

組み込み変数	説明
\$_info_snmp_sysname_value	sysName 変数の値文字列。
\$_info_snmp_syslocation_oid	sysLocation 変数の OID 値。
\$_info_snmp_syslocation_value	sysLocation 変数の値文字列。
\$_info_snmp_sysdescr_oid	sysDescr 変数の OID 値。
\$_info_snmp_sysdescr_value	sysDescr 変数の値文字列。
\$_info_snmp_sysobjectid_oid	sysObjectID 変数の OID 値。
\$_info_snmp_sysobjectid_value	sysObjectID 変数の値文字列。
\$_info_snmp_sysuptime_oid	sysUptime 変数の OID 値。
\$_info_snmp_sysuptime_value	sysUptime 変数の値文字列。
\$_info_snmp_syscontact_oid	sysContact 変数の OID 値。
\$_info_snmp_syscontact_value	sysContact 変数の値文字列。

例

次に、SNMP エンティティから sysDescr.0 変数を取得する例を示します。

```
Router(config)# event manager applet
Router(config-applet)# action 1.3 info type snmp getid 1.3.6.1.2.1.1.1.0 community public
ipaddr 172.17.16.69
Router(config-applet)#
```

関連コマンド

コマンド	説明
event manager applet	EEM にイベント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。
snmp-server community	SNMP エンティティへのアクセスをイネーブルにするコミュニティ アクセス スtring を設定します。

action info type snmp inform

Embedded Event Manager (EEM) アプレットがトリガーされたときに、簡易ネットワーク管理プロトコル (SNMP) インフォーム要求を送信するには、アプレット コンフィギュレーション モードで **action info type snmp inform** コマンドを使用します。SNMP インフォーム要求の送信をディセーブルにするには、このコマンドの **no** 形式を使用します。

action *label* **info type snmp inform** **trap-oid** *trap-oid-value* **trap-var** *trap-variable* **community** *community-string* **ipaddr** *ip-address*

no **action** *label* **info type**

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できます）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>trap-oid</i>	SNMP トラップを生成するオブジェクトのオブジェクト ID を指定します。
<i>trap-oid-value</i>	SNMP トラップを生成するオブジェクトの OID 値。
<i>trap-var</i>	トラップを生成するオブジェクトのインスタンスに関連付けられた変数を指定します。
<i>trap-variable</i>	SNMP トラップを生成するオブジェクトの変数の値。
community	SNMP エンティティにアクセスするコミュニティ ストリングを指定します。

<i>community-string</i>	<p>SNMP コミュニティストリング。コミュニティストリングは、SNMP エンティティにアクセスするためのパスワードと同様に機能します。ストリングは 1 ～ 32 文字の英数字で構成され、次のいずれかに設定できます。</p> <ul style="list-style-type: none"> • ro : SNMP エンティティへの読み取り専用アクセスを設定します。このコミュニティストリングのデフォルト値は public です。 • rw : SNMP エンティティへの読み取り/書き込みアクセスを設定します。このコミュニティストリングのデフォルト値は private です。
ipaddr	SNMP エンティティの IP アドレスを指定します。
<i>ip-address</i>	SNMP get および set 操作のためにオブジェクトを取得する、ネットワーク管理システム (NMS) の IP アドレス。

コマンド デフォルト SNMP インフォーム要求はデフォルトで送信されません。

コマンド モード アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン SNMP インフォーム要求は、SNMP マネージャにネットワーク状態を警告する SNMP 通知で、SNMP マネージャからの受信の確認を要求します。

例

次に、SNMP インフォーム要求を送信する例を示します。

```
Router(config)# event manager applet
Router(config-applet)# action 1.4 info type snmp inform trap-oid 1.3.6.1.4.1.1.226.0.2.1
trap-var sysUpTime community public ipaddr 172.69.16.2
Router(config-applet)#
```

関連コマンド

コマンド	説明
event manager applet	EEMにイベント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。
snmp-server community	SNMP エンティティへのアクセスをイネーブルにするコミュニティ アクセス ストリングを設定します。
snmp-server enable traps	システムで使用可能なすべての SNMP 通知タイプをイネーブルにします。

action info type snmp oid

Embedded Event Manager (EEM) アプレットがトリガーされたときに、SNMP set 操作中に取得されるオブジェクトと簡易ネットワーク管理プロトコル (SNMP) のタイプを指定するには、アプレットコンフィギュレーションモードで **action info type snmp oid** コマンドを使用します。この機能をディセーブルにするには、このコマンドの **no** 形式を使用します。

action *label* **info type snmp oid** *oid-value* {**get-type** {**exact**|**next**} [**community** *community-string*]} **set-type** *oid-type* *oid-type-value* **community** *community-string*] [**ipaddr** *ip-address*]

no **action** *label* **info type**

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できます）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
oid	SNMP オブジェクト ID (OID) で指定された SNMP オブジェクトの値を要求します。
<i>oid-value</i>	SNMP ドット付き表記でのデータエレメントのオブジェクト ID 値。オブジェクト ID は、一連の整数値またはテキスト文字列で表されます。たとえば、インターフェイス MIB のオブジェクト名は 1.3.6.1.2.1.2 または iso.internet.mgmt.mib-2.interfaces と表すことができます。 OID は、関連する MIB (CISCO-EMBEDDED-EVENT-MGR-MIB) 内にタイプとして定義され、各タイプはオブジェクト値を保持します。一部の OID 型のモニタリングが可能です。次のタイプが有効です。 <ul style="list-style-type: none"> • INTEGER_TYPE • COUNTER_TYPE • GAUGE_TYPE • TIME_TICKS_TYPE • COUNTER_64_TYPE • OCTET_PRIM_TYPE • OPAQUE_PRIM_TYPE

get-type	<p><i>oid-value</i> 引数で指定したオブジェクト ID に適用される SNMP <i>get</i> 操作のタイプを指定します。</p> <ul style="list-style-type: none"> • exact : (任意) <i>oid-value</i> 引数で指定したオブジェクト ID を取得します。 • next : (任意) <i>oid-value</i> 引数で指定したオブジェクト ID の英数字順でオブジェクト ID を取得します。
community	SNMP エンティティにアクセスするコミュニティストリングを指定します。
<i>community-string</i>	<p>SNMP コミュニティストリング。コミュニティストリングは、SNMP エンティティにアクセスするためのパスワードと同様に機能します。ストリングは 1 ~ 32 文字の英数字で構成され、次のいずれかに設定できます。</p> <ul style="list-style-type: none"> • ro : SNMP エンティティへの読み取り専用アクセスを設定します。このコミュニティストリングのデフォルト値は public です。 • rw : SNMP エンティティへの読み取り/書き込みアクセスを設定します。このコミュニティストリングのデフォルト値は private です。
set-type	SNMP <i>set</i> 操作中に取得するオブジェクトのタイプを指定します。 <i>set</i> 操作を実行するには、OID、OID タイプ、および値を指定する必要があります。

<p><i>oid-type</i></p>	<p>OID のタイプ。次の値が有効です。</p> <ul style="list-style-type: none"> • counter32 : 最小値が 0 の 32 ビットの数値。最大値に到達すると、カウンタが 0 にリセットされます。 • gauge : 最小値が 0 の 32 ビットの数値。たとえば、gauge オブジェクトタイプを使用して、ルータ上のインターフェイスの速度を測定できます。 • integer : 管理対象オブジェクトのコンテキスト内の番号が付けられたタイプを指定する場合は、32 ビットの数字が使用されます。たとえば、ルータ インターフェイスの動作ステータスを 1 に設定した場合はアップ、2 に設定した場合はダウンを示します。 • ipv4 : IP バージョン 4 アドレス。 • octet string : 物理アドレスを表すために使用される、16 進表記のオクテット文字列。 • string : テキスト文字列を表すために使用される、テキスト表記のオクテット文字列。 • unsigned32 : 10 進数の値を表すために使用される、32 ビット番号。
<p><i>oid-type-value</i></p>	<p>SNMP set 操作用に指定された OID タイプの整数またはテキストストリング値。各 OID タイプで有効な値は次のとおりです。</p> <ul style="list-style-type: none"> • counter : 0 ~ 4294967295 の範囲の整数値。 • gauge : 0 ~ 4294967295 の範囲の整数値。 • integer : 0 ~ 4294967295 の範囲の整数値。 • ipv4 : ドット付き 10 進表記の IPv4 アドレス。 • octet string : テキストストリング。 • string : テキストストリング。 • unassigned32 : 0 ~ 4294967295 の範囲の符号なし整数値。

ipaddr	(任意) SNMP エンティティの IP アドレスを指定します。
<i>ip-address</i>	(任意) SNMP get および set 操作のためにオブジェクトを取得する、ネットワーク管理システム (NMS) の IP アドレス。

コマンド デフォルト EEM アプレットがトリガーされたときに、SNMP set または get 操作の要求は送信されません。

コマンド モード アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.3(14)T	このコマンドが導入されました。
12.2(28)SB	このコマンドが、Cisco IOS Release 12.2(28)SB に統合されました。
12.2(18)SXF4	このコマンドが Cisco IOS Release 12.2(18)SXF4 に統合され、ソフトウェア モジュラリティ イメージだけをサポートするようになりました。
12.2(33)SRA	このコマンドが、Cisco IOS Release 12.2(33)SRA に統合されました。
12.2(18)SXF5	このコマンドが、Cisco IOS Release 12.2(18)SXF5 に統合されました。
12.4(22)T	set-type 、 community 、および ipaddr キーワードが追加されました。

使用上のガイドライン SNMP set 操作は個々の変数を SNMP エンティティに設定し、SNMP get 操作は SNMP エンティティから個々の変数を取得します。

次の表に、SNMP get および set 操作の結果が保存される組み込み変数を示します。

表 3 : action info コマンドの EEM 組み込み変数

組み込み変数	説明
<code>\$_info_snmp_oid</code>	SNMP オブジェクト ID。

組み込み変数	説明
\$_info_snmp_value	割り当てられた SNMP データ エLEMENT の値文字列。

例

次に、SNMP エンティティからオブジェクトの個々の変数を取得する例を示します。

```
Router(config)# event manager applet
Router(config-applet)# action 1.3 info type snmp oid 1.3.6.1.4.1.9.9.48.1.1.1.6.1 get-type
exact community public ipaddr 172.17.16.69
Router(config-applet)#
```

次に、SNMP エンティティに個々の変数を設定する例を示します。

```
Router(config)# event manager applet
Router(config-applet)# action 1.4 info type snmp oid 1.3.6.1.4.1.9.9.48.1.1.1.6.1 set-type
integer 42220 sysName.0 community public ipaddr 172.17.16.69
Router(config-applet)#
```

関連コマンド

コマンド	説明
event manager applet	EEM にイベント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。
snmp-server community	SNMP エンティティへのアクセスをイネーブルにするコミュニティ アクセス ストリングを設定します。

action info type snmp trap

Embedded Event Manager (EEM) アプレットがトリガーされたときに、簡易ネットワーク管理プロトコル (SNMP) トラップ要求を送信するには、アプレット コンフィギュレーション モードで **action info type snmp trap** コマンドを使用します。SNMP トラップ要求の送信をディセーブルにするには、このコマンドの **no** 形式を使用します。

action *label* **info type snmp trap enterprise-oid** *enterprise-oid-value* **generic-trapnum** *generic-trap-number* **specific-trapnum** *specific-trap-number* **trap-oid** *trap-oid-value* **trap-var** *trap-variable*

no action *label* **info type**

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できません）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
trap	SNMP トラップの要求を送信します。
enterprise-oid	オブジェクトのエンタープライズOID値を指定します。
<i>enterprise-oid-value</i>	SNMP トラップを生成するオブジェクトのエンタープライズOID値。OID値はエンタープライズ固有で、一連の整数またはテキストストリングで表されます。
generic-trapnum	汎用 SNMP トラップ番号を指定します。
<i>generic-trap-number</i>	汎用トラップ番号。次の汎用トラップおよびトラップ番号が有効です。 <ul style="list-style-type: none"> • coldStart (0) • warmStart (1) • linkDown (2) • linkUp (3) • authenticationFailure (4) • egpNeighborLoss (5) • enterpriseSpecific (6)

specific-trapnum	汎用トラップ番号が 6 に設定されていない場合に、エンタープライズ固有のトラップを指定します。
<i>specific-trap-number</i>	エンタープライズイベントに固有のトラップに関連付けられた番号。
trap-oid	SNMP トラップを生成するオブジェクトのオブジェクト ID を指定します。
<i>trap-oid-value</i>	SNMP トラップを生成するオブジェクトの OID 値。
trap-var	トラップを生成するオブジェクトのインスタンスに関連付けられた変数を指定します。
<i>trap-variable</i>	SNMP トラップを生成するオブジェクトの変数の値。

コマンド デフォルト SNMP トラップ要求はデフォルトで送信されません。

コマンド モード アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン

トラップは、SNMP マネージャまたは NMS にネットワーク状態を警告する SNMP 通知です。SNMP インフォーム要求とは異なり、トラップは SNMP マネージャに受信確認を要求しません。

例

次に、SNMP トラップ要求を送信する例を示します。

```
Router(config)# event manager applet
Router(config-applet)# action 1.4 info type snmp trap enterprise-oid 1.3.6.1.4.1.1
```

```
generic-trapnum 4 specific-trapnum 7 trap-oid 1.3.6.1.4.1.1.226.0.2.1 trap-var sysUpTime
Router(config-applet)#
```

関連コマンド

コマンド	説明
event manager applet	EEMにイベント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。
snmp-server enable traps	システムで使用可能なすべてのSNMP通知タイプをイネーブルにします。

action info type snmp var

Embedded Event Manager (EEM) アプレットから簡易ネットワーク管理プロトコル (SNMP) オブジェクト ID (OID) の変数およびその値を作成するには、アプレット コンフィギュレーション モードで `action info type snmp var` コマンドを使用します。変数を削除するには、このコマンドの `no` 形式を使用します。

`action label info type snmp var variable-name oid oid-value oid-type oid-type-value`

`no action label info type`

構文の説明

<i>label</i>	一意な識別子（任意の文字列値を指定できます）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。文字列に空白が含まれる場合は二重引用符で囲みます。
var	SNMP MIB オブジェクトの SNMP 変数またはオブジェクトインスタンスを指定します。
<i>variable-name</i>	SNMP 変数の名前。たとえば、 <code>sysDescr.0</code> 。
oid	SNMP OID で指定された SNMP オブジェクトの値を要求します。

<i>oid-value</i>	<p>SNMP ドット付き表記でのデータ要素のオブジェクト ID 値。オブジェクト ID は、一連の整数値またはテキスト文字列で表されます。たとえば、インターフェイス MIB のオブジェクト名は 1.3.6.1.2.1.2 または iso.internet.mgmt.mib-2.interfaces と表すことができます。</p> <p>OID は、関連する MIB (CISCO-EMBEDDED-EVENT-MGR-MIB) 内にタイプとして定義され、各タイプはオブジェクト値を保持します。一部の OID 型のモニタリングが可能です。次のタイプが有効です。</p> <ul style="list-style-type: none">• INTEGER_TYPE• COUNTER_TYPE• GAUGE_TYPE• TIME_TICKS_TYPE• COUNTER_64_TYPE• OCTET_PRIM_TYPE• OPAQUE_PRIM_TYPE
------------------	---

<p><i>oid-type</i></p>	<p>OID のタイプ。次の値が有効です。</p> <ul style="list-style-type: none"> • counter32 : 最小値が 0 の 32 ビットの数値。最大値に到達すると、カウンタが 0 にリセットされます。 • gauge : 最小値が 0 の 32 ビットの数値。たとえば、gauge オブジェクトタイプを使用して、ルータ上のインターフェイスの速度を測定できます。 • integer : 管理対象オブジェクトのコンテキスト内の番号が付けられたタイプを指定する場合は、32 ビットの数字が使用されます。たとえば、ルータ インターフェイスの動作ステータスを 1 に設定した場合はアップ、2 に設定した場合はダウンを示します。 • ipv4 : IP バージョン 4 アドレス。 • octet string : 物理アドレスを表すために使用される、16 進表記のオクテット文字列。 • string : テキスト文字列を表すために使用される、テキスト表記のオクテット文字列。 • unsigned32 : 10 進数の値を表すために使用される、32 ビット番号。
<p><i>oid-type-value</i></p>	<p>変数を作成するために指定された OID タイプの整数またはテキスト ストリング値。各 OID タイプで有効な値は次のとおりです。</p> <ul style="list-style-type: none"> • counter : 0 ~ 4294967295 の範囲の整数値。 • gauge : 0 ~ 4294967295 の範囲の整数値。 • integer : 0 ~ 4294967295 の範囲の整数値。 • ipv4 : ドット付き 10 進表記の IPv4 アドレス。 • octet string : テキスト ストリング。 • string : テキスト ストリング。 • unassigned32 : 0 ~ 4294967295 の範囲の符号なし整数値。

コマンド デフォルト デフォルトでは、EEM アプレットがトリガーされたときに変数は作成されません。

コマンド モード アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン 変数は、OID とインスタンスによって識別されます。一般的に、インスタンスは、OID に .0 を追加することで指定されます。たとえば、sysDescr.0。

例 次に、オブジェクト ID の変数を作成する例を示します。

```
Router(config)# event manager applet
Router(config-applet)# action 1.3 info type snmp var sysDescr.0 oid
1.3.6.1.4.1.9.9.48.1.1.1.6.1 integer 4220
Router(config-applet)#
```

関連コマンド

コマンド	説明
event manager applet	EEM にイベント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。

action multiply

Embedded Event Manager (EEM) アプレットがトリガーされたときに、変数の値を指定された特定の整数の値で乗算するアクションを指定するには、アプレットコンフィギュレーションモードで **action multiply** コマンドを使用します。計算プロセスを削除するには、このコマンドの **no** 形式を使用します。

action label multiply [*long-integer-1* | *variable-name-1*] [*long-integer-2* | *variable-name-2*]

no action label multiply

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できます）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>long-integer-1</i>	（任意）乗算の最初の整数値。
<i>variable-name-1</i>	（任意）乗算の最初の変数名。乗数の変数名に格納されている値は長整数値である必要があります。そうしないと、アクションは失敗します。
<i>long-integer-2</i>	（任意）乗算の2番目の整数値。
<i>variable-name-2</i>	（任意）乗算の2番目の変数名。乗数の変数名に格納されている値は長整数値である必要があります。そうしないと、アクションは失敗します。

コマンド デフォルト

アプレットコンフィギュレーションモードでコマンドを入力しなかった場合、設定を終了したときに個別のアプレットが登録されません。

コマンド モード

アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。

リリース	変更内容
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン

action multiply コマンドを使用して、変数の値を特定の整数値で乗算します。すべての算術演算は、長整数としてオーバーフローのチェックなしで実行されます。このアプレットにステートメントが関連付けられていない場合、イベントは引き続きトリガーされますがアクションまたは結果は発生しません。設定の終了時に、このアプレットにステートメントが関連付けられていないことを示す警告メッセージが表示されます。**action multiply** コマンドの結果は、すべて `$_result` に格納されます。

Tool Command Language (Tcl) と CLI アプレットベースの EEM ポリシーの間でカスタマー向けの一貫したユーザ インターフェイスを実現するには、次の基準に従います。

- Tcl ベースの実装では、イベント仕様基準は TCL で記述されます。
- アプレットベースの実装では、イベント仕様データは CLI アプレット サブモード コンフィギュレーション文を使用して記述されます。

アプレットベースの実装では、一部のキーワードが必要以上に長い、ハイフンで連結されているように見えますが、これは Tcl ベースの実装が最初に開発され、導入されているためです。

アプレット コンフィギュレーション モードを開始するには、グローバル コンフィギュレーション モードを開始後、**event manager applet** *applet-name* コマンドを使用します。アプレット コンフィギュレーション モードでは、**config** プロンプトが、(config-applet)# に変わります。アプレット コンフィギュレーション モードでは、3 種類のコンフィギュレーション文がサポートされています。

- **event** : このアプレットの実行の原因となる、イベント基準を指定します。
- **action** : 組み込みアクションを実行します。
- **set** : アプレット変数を設定します (現在、サポートされている変数は `_exit_status` のみです)。

例

次に、保存された変数の値を乗算する例を示します。

```
Router(config)# event manager applet action
Router(config-applet)# action label2 multiply 23 25
```

関連コマンド

コマンド	説明
action add	EEM アプレットがトリガーされたときに、変数の値に特定の値を加算します。

コマンド	説明
<code>action divide</code>	EEM アプレットがトリガーされたときに、変数の値を特定の値で除算します。
<code>action subtract</code>	EEM アプレットがトリガーされたときに、変数の値から特定の値を減算します。

action puts

Embedded Event Manager (EEM) アプレットがトリガーされたときに、ローカル tty に直接データを出力するアクションを指定するには、アプレットコンフィギュレーションモードで **action puts** コマンドを使用します。この機能をディセーブルにするには、このコマンドの **no** 形式を使用します。

action *label* **puts** [**nonewline**] *string*

no **action** *label* **puts**

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できます）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
nonewline	（任意）改行文字の表示を抑制します。
<i>string</i>	文字のシーケンス。ストリングに空白が含まれる場合は二重引用符で囲みます。

コマンド デフォルト

データはローカル TTY に出力されません。

コマンド モード

アプレットコンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン

action puts コマンドは、同期イベントに適用されます。同期アプレットに対するこのコマンドの出力は、syslog をバイパスして tty に直接表示されます。非同期イベントの場合、このコマンドの

デフォルトは syslog です。 **nonewline** キーワードは、改行文字の表示を抑制します。 非同期アプレットの **action puts** コマンドの出力は、ログャーに向けられます。

例

次に、ローカル tty に直接データを出力する例を示します。

```
Router(config-applet)# event manager applet puts
Router(config-applet)# event none
Router(config-applet)# action 1 regexp "(.*) (.*) (.*)" "one two three" _match _sub1
Router(config-applet)# action 2 puts "match is $_match"
Router(config-applet)# action 3 puts "submatch 1 is $_sub1"
Router# event manager run puts
match is one two three
submatch 1 is one
Router#
```

関連コマンド

コマンド	説明
action gets	ローカル tty から入力を取得し、特定の変数に値を格納します。
event manager applet	EEMにイベント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。

action regexp

Embedded Event Manager (EEM) アプレットがトリガーされたときに、入力ストリングで正規表現パターンをマッチングするには、アプレット コンフィギュレーションモードで **action regexp** コマンドを使用します。この機能をディセーブルにするには、このコマンドの **no** 形式を使用します。

action label regexp *string-pattern string-input* [*string-match* [*string-submatch1*] [*string-submatch2*] [*string-submatch3*]]

no action label regexp

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できます）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>string-pattern</i>	正規表現パターンマッチングに使用する文字のシーケンス。
<i>string-input</i>	入力として使用する文字のシーケンス。
<i>string-match</i>	(任意) 全体一致を格納する変数名。
<i>string-submatch</i>	(任意) 存在するすべてのサブマッチを格納する変数名。最大3個のサブマッチストリングを指定できます。

コマンド デフォルト 正規表現パターンのマッチングは行われません。

コマンド モード アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン *string-pattern* 引数は正規表現です。ストリングの一部がパターンに一致する場合は1、それ以外の場合は0を返します。オプションの *string-match* および *string-submatch* 引数は、マッチングの結果を格納します。

次の表に、**action regexp** コマンドの結果が保存される組み込み変数を示します。

表 4: **action regexp** コマンドの **EEM** 組み込み変数

組み込み変数	説明
\$_regexp_result	正規表現パターンマッチングの結果がこの変数に格納されます。

例

次に、正規表現マッチングを定義する例を示します。

```
Router(config-applet)# event manager applet regexp
Router(config-applet)# event none
Router(config-applet)# action 1 regexp "(.*) (.*) (.*)" "one two three" _match _sub1
Router(config-applet)# action 2 puts "match is $_match"
Router(config-applet)# action 3 puts "submatch 1 is $_sub1"
Router# event manager run regexp
match is one two three
submatch 1 is one
Router#
```

関連コマンド

コマンド	説明
event manager applet	EEMにイベント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。

action set (EEM)

Embedded Event Manager (EEM) アプレットがトリガーされたときに、変数の値を設定するには、アプレットコンフィギュレーションモードで **action set** コマンドを使用します。EEM アプレット変数の値を削除するには、このコマンドの **no** 形式を使用します。

action label set variable-name variable-value

no action label set

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できます）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>variable-name</i>	設定する変数に割り当てられた名前。
<i>variable-value</i>	変数の値。

コマンド デフォルト

変数の値は設定されません。

コマンド モード

アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。 set (EEM) コマンドが、このコマンドに置き換えられました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン

action set コマンドを使用して、EEM アプレットがトリガーされたときに、変数の値を設定します。

例

次に、変数の値を設定する例を示します。

```
Router(config-applet)# event manager applet set
Router(config-applet)# event none
Router(config-applet)# action 1 set str "this is some text"
Router(config-applet)# action 2 string range "$str" 0 6
Router(config-applet)# action 3 puts "$_string_result"
Router# event manager run set
"this is"
Router#
```

関連コマンド

コマンド	説明
event manager applet	EEMにイベント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。



action string から D までのコマンド

- [action string から D までのコマンド](#), 63 ページ

action string から D までのコマンド

action string compare

Embedded Event Manager (EEM) アプレットがトリガーされたときに、2つの等しくないストリングを比較するには、アプレット コンフィギュレーション モードで **action string compare** コマンドを使用します。この機能をディセーブルにするには、このコマンドの **no** 形式を使用します。

action label string compare [*nocase*] [*length integer*] *string1 string2*

no action label string compare

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できません）。アクションは、ラベルをソート キーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
nocase	（任意）大文字と小文字を区別しない比較を指定します。
length	（任意）比較を最初の整数文字に制限します。
<i>integer</i>	（任意） length 引数の有効な値の範囲は 1 ～ 4294967295 です。
<i>string1</i>	文字のシーケンス。
<i>string2</i>	文字のシーケンス。

コマンド デフォルト 等しくないストリングは比較されません。

コマンド モード アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン スtringの比較は、左から右に向かってバイトごとに行われます。Stringの長さが等しくない場合、長いStringのほうが短いStringよりも多く比較されます。**action string compare** コマンドは、2つの等しくないStringを強制的に比較し、続けてString比較の結果で整数比較を行います。

2つの等しい同じ文字列を比較した場合の結果は0、一方のStringが他方のStringの前にソートされる場合の結果は-1になります。その他のすべての比較の結果は1です。比較されている文字列が整数に変換された場合は、**strcmp** コマンドを使用して結果同士の比較が行われます。

次の表に、**action string compare** コマンドの結果が保存される組み込み変数を示します。

表 5: **action string compare** コマンドの **EEM** 組み込み変数

組み込み変数	説明
<code>\$_string_result</code>	action string compare コマンドの結果がこの変数に格納されます。

例

次に、2つの等しくないStringを比較する例を示します。

```
Router(config-applet)# event manager applet compare
Router(config-applet)# event none
Router(config-applet)# action 1 set str "this contains some $str"
Router(config-applet)# action 2 string compare nocase length 3 "contains" "$str"
```

関連コマンド

コマンド	説明
event manager applet	EEMにイベント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。

action string equal

Embedded Event Manager (EEM) アプレットがトリガーされたときに、2つのストリングが等しいかどうかを確認するには、アプレット コンフィギュレーション モードで **action string equal** コマンドを使用します。この機能をディセーブルにするには、このコマンドの **no** 形式を使用します。

action *label* **string equal** [**nocase**] [**length** *integer*] *string1* *string2*

no **action** *label* **string equal**

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できません）。アクションは、ラベルをソート キーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
nocase	（任意）大文字と小文字を区別しない比較を指定します。
length	（任意）値の長さを指定して比較を制限します。
<i>integer</i>	（任意） length 引数の有効な値の範囲は 1 ～ 4294967295 です。
<i>string1</i>	文字のシーケンス。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>string2</i>	文字のシーケンス。ストリングに空白が含まれる場合は二重引用符で囲みます。

コマンド デフォルト ストリングが等しいかどうかは確認されません。

コマンド モード アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。

リリース	変更内容
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン

action string equal コマンドは、2つのストリングを比較し、ストリングが等しい場合は 1 を返します。大文字と小文字を区別しないで比較する場合は、**nocase** を使用します。

次の表に、**action string equal** コマンドの結果が保存される組み込み変数を示します。

表 6 : **action string equal** コマンドの **EEM** 組み込み変数

組み込み変数	説明
\$_string_result	action string equal コマンドの結果がこの変数に格納されます。

例

次に、2つのストリングが等しいかどうかを確認する例を示します。

```
Router(config-applet)# event manager applet equal
Router(config-applet)# event none
Router(config-applet)# action 1 set str "this contains some data"
Router(config-applet)# action 2 string equal "contains" "data"
```

関連コマンド

コマンド	説明
event manager applet	EEMにイベント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。

action string first

Embedded Event Manager (EEM) アプレットがトリガーされたときに、*string1* の最初の出現で、インデックスを *string2* に返すには、アプレット コンフィギュレーション モードで **action string first** コマンドを使用します。この機能をディセーブルにするには、このコマンドの **no** 形式を使用します。

action *label* **string first** *string1* *string2* [*index-value*]

no **action** *label* **string first**

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できます）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>string1</i>	文字のシーケンス。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>string2</i>	文字のシーケンス。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>index-value</i>	（任意）最初のテストを開始するインデックス値。番号の範囲は 0 ~ 4294967295 です。

コマンド デフォルト *string1* の最初の出現で、インデックスは *string2* に返されません。

コマンド モード アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン *string1* の最初の出現で、*string2* にインデックスが配置されます。*string1* が見つからない場合は、-1 を返します。

次の表に、**action string first** コマンドの結果が保存される組み込み変数を示します。

表 7: **action string first** コマンドの EEM 組み込み変数

組み込み変数	説明
<code>\$_string_result</code>	action string first コマンドの結果がこの変数に格納されます。

例 次に、*string1* の最初の出現で、インデックスを *string2* に返す例を示します。

```
Router(config-applet)# event manager applet first
Router(config-applet)# event none
Router(config-applet)# action 1 set str "this contains some data"
Router(config-applet)# action 2 string first "contains" "$str"
Router(config-applet)# action 3 puts "$_string_result"
Router# event manager run first
5
Router#
```

関連コマンド

コマンド	説明
action string last	<i>string1</i> の最後の出現で、 <i>string2</i> にインデックスを返します。
event manager applet	EEM にイベント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。

action string index

Embedded Event Manager (EEM) アプレットがトリガーされたときに、特定のインデックス値に指定された文字を返すには、アプレット コンフィギュレーションモードで **action string index** コマンドを使用します。この機能をディセーブルにするには、コマンドの **no**形式を使用します。

action label string index *string* [*value*| *end*]

no action label string index

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できません）。アクションは、ラベルをソート キーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>string</i>	文字のシーケンス。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>value</i>	（任意）インデックス値。番号の範囲は 0 ～ 4294967295 です。カウントは 0 から始まります。
<i>end</i>	（任意）ストリングの最後の文字。

コマンド デフォルト

特定のインデックス値で指定された文字は返されません。

コマンド モード

アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン

インデックス番号はゼロから始まります。 *end* 引数は、ストリングの最後の文字に使用します。

次の表に、**action string index** コマンドが文字を保存する組み込み変数を示します。

表 8 : **action string index** コマンドの **EEM** 組み込み変数

組み込み変数	説明
<code>\$_string_result</code>	action string index コマンドは、この変数に文字を格納します。

例

次に、特定のインデックス値で指定された文字を返す例を示します。

```
Router(config-applet)# event manager applet index
Router(config-applet)# event none
Router(config-applet)# action 1 set str "this is text"
Router(config-applet)# action 2 string index "$str" 8
Router(config-applet)# action 3 puts "$_string_result"
Router# event manager run index
t
Router#
```

関連コマンド

コマンド	説明
event manager applet	EEMにイベント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。

action string last

Embedded Event Manager (EEM) アプレットがトリガーされたときに、*string1* の最後の出現で、インデックスを *string2* に返すには、アプレット コンフィギュレーション モードで **action string last** コマンドを使用します。この機能をディセーブルにするには、このコマンドの **no** 形式を使用します。

action label string last string1 string2 [index-value]

no action label string last

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できません）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>string1</i>	文字のシーケンス。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>string2</i>	文字のシーケンス。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>index-value</i>	（任意）最後のテストを開始するインデックス値。番号の範囲は 0 ~ 4294967295 です。

コマンド デフォルト *string1* の最後の出現で、インデックスは *string2* に返されません。

コマンド モード アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン *string1* の最初の出現で、*string2* にインデックスが配置されます。*string1* が見つからない場合は、-1 を返します。

次の表に、**action string last** コマンドの結果が保存される組み込み変数を示します。

表 9: **action string last** コマンドの **EEM** 組み込み変数

組み込み変数	説明
\$_string_result	action string last コマンドの結果がこの変数に格納されます。

例 次に、*string1* の最後の出現で、インデックスを *string2* に返す例を示します。

```
Router(config-applet)# event manager applet last
Router(config-applet)# event none
Router(config-applet)# action 1 set str "this contains some data"
Router(config-applet)# action 2 string last "contains" "$str"
Router(config-applet)# action 3 puts "$_string_result"
Router# event manager run last
5
Router#
```

関連コマンド

コマンド	説明
action string first	<i>string1</i> の最初の出現で、 <i>string2</i> にインデックスを返します。
event manager applet	EEM にイベント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。

action string length

Embedded Event Manager (EEM) アプレットがトリガーされたときに、ストリング内の文字数を返すには、アプレット コンフィギュレーション モードで **action string length** コマンドを使用します。この機能をディセーブルにするには、このコマンドの **no** 形式を使用します。

action label string length string

no action label string length

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できます）。アクションは、ラベルをソート キーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>string</i>	文字のシーケンス。ストリングに空白が含まれる場合は二重引用符で囲みます。

コマンド デフォルト

ストリングの文字数は返されません。

コマンド モード

アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン

action string length コマンドを使用して、EEM アプレットがトリガーされたときにストリング内の文字数を返すアクションを指定します。

次の表に、**action string length** コマンドの結果が保存される組み込み変数を示します。

表 10: action string length コマンドの EEM 組み込み変数

組み込み変数	説明
\$_string_result	action string length コマンドの結果がこの変数に格納されます。

例

次に、ストリング内の文字数を返す例を示します。

```
Router(config-applet)# event manager applet length
Router(config-applet)# event none
Router(config-applet)# action 1 set str "this contains some data"
Router(config-applet)# action 2 string length "contains"
Router(config-applet)# action 3 puts "$_string_result"
Router# event manager run length
8
Router#
```

関連コマンド

コマンド	説明
event manager applet	EEMにイベント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。

action string match

Embedded Event Manager (EEM) アプレットがトリガーされたときに、ストリングがパターンに一致する場合は `$_string_result` に 1 を返すには、アプレット コンフィギュレーション モードで **action string match** コマンドを使用します。このアクションをディセーブルにするには、このコマンドの **no** 形式を使用します。

action *label* **string match** [**nocase**] *string-pattern* *string*

no **action** *label* **string match**

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できません）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
nocase	（任意）大文字と小文字を区別しない比較を指定します。
<i>string-pattern</i>	大文字と小文字を区別しない比較のパターン。
<i>string</i>	文字のシーケンス。ストリングに空白が含まれる場合は二重引用符で囲みます。

コマンド デフォルト

ストリングのパターン マッチングの結果は、`$_string_result` に返されません。

コマンド モード

アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン スtringが指定されたパターンに一致する場合、結果は1です。パターンが一致しない場合は、結果は0です。

次の表に、**action string match** コマンドの結果が保存される組み込み変数を示します。

表 11 : **action string match** コマンドの **EEM** 組み込み変数

組み込み変数	説明
<code>\$_string_result</code>	action string match コマンドの結果がこの変数に格納されます。

例 次に、Stringがパターンに一致した場合に `$_string_result` に 1 に返す例を示します。

```
Router(config-applet)# event manager applet match
Router(config-applet)# event none
Router(config-applet)# action 1 set str "this is some text"
Router(config-applet)# action 2 string match "$str" "this is"
Router(config-applet)# action 3 puts "$_string_result"
Router# event manager run match
1
Router#
```

関連コマンド

コマンド	説明
event manager applet	EEMにイベント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。

action string range

Embedded Event Manager (EEM) アプレットがトリガーされたときに、文字の範囲をストリングに保存するには、アプレット コンフィギュレーション モードで **action string range** コマンドを使用します。この機能をディセーブルにするには、このコマンドの **no** 形式を使用します。

action *label* **string range** *string start-index end-index*

no **action** *label* **string range**

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できます）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>string</i>	最大 4294967295 文字の文字のシーケンス。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>start-index</i>	開始インデックスのストリング値。指定できる範囲は 0 ~ 4294967295 です。
<i>end-index</i>	終了インデックスのストリング値。指定できる範囲は 0 ~ 4294967295 です。

コマンド デフォルト ストリングは保存されません。

コマンド モード アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン **action string range** コマンドを使用して、EEM アプレットがトリガーされたときに文字の範囲をストリングに保存するアクションを指定します。 *start-index* および *end-index* 引数は、操作対象のストリングの範囲を指定します。

次の表に、**action string range** コマンドの結果が保存される組み込み変数を示します。

表 12: **action string range** コマンドの **EEM** 組み込み変数

組み込み変数	説明
<code>\$_string_result</code>	action string range コマンドの結果がこの変数に格納されます。

例 次に、指定したストリングに文字の範囲を保存する例を示します。

```
Router(config)# event manager applet store
Router(config-applet)#
action 1.0 set string "This is some text"
Router(config-applet)# action 2.0 string range "$string" 0 6
Router(config-applet)# action 3.0 puts "$_string_result"
Router(config-applet)# end
Router# event manager run store
this is
Router#
```

関連コマンド

コマンド	説明
event manager applet	EEM にイベント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。

action string replace

Embedded Event Manager (EEM) アプレットがトリガーされたときに、指定したストリングの文字の範囲を置き換えることで新しいストリングを保存するには、アプレット コンフィギュレーションモードで **action string replace** コマンドを使用します。この機能をディセーブルにするには、このコマンドの **no** 形式を使用します。

action label string replace string start-index end-index [new-string]

no action label string replace

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できます）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>string</i>	最大 4294967295 文字の文字のシーケンス。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>start-index</i>	開始インデックスのストリング値。指定できる範囲は 0 ~ 4294967295 です。
<i>end-index</i>	終了インデックスのストリング値。指定できる範囲は 0 ~ 4294967295 です。
<i>new-string</i>	(任意) ストリングの文字の範囲を置き換える一連の文字。

コマンド デフォルト ストリングは保存されません。

コマンド モード アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。

リリース	変更内容
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン

action string replace コマンドを使用して、特定のストリングの指定された文字を置き換えることで新しいストリングを取得します。 *new-string* 引数の値を指定しない場合、文字はスペースに置き換えられます。

次の表に、**action string replace** コマンドの結果が保存される組み込み変数を示します。

表 13 : **action string replace** コマンドの **EEM** 組み込み変数

組み込み変数	説明
<code>\$_string_result</code>	action string replace コマンドの結果がこの変数に格納されます。

例

次に、ストリング内の特定の文字を置換することで作成された新しいストリングを保存する例を示します。

```
Router(config)# event manager applet replace
Router(config-applet)# event none
Router(config-applet)# action 1.0 set string "This is some text"
Router(config-applet)# action 2.0 string replace "$string" 0 6 "that was"
Router(config-applet)# action 3.0 puts "$_string_result"
Router (config-applet)# end
Router# event manager run replace
that was some text
Router#
```

関連コマンド

コマンド	説明
event manager applet	EEMにイベント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。

action string tolower

Embedded Event Manager (EEM) アプレットがトリガーされたときに、ストリングの特定の範囲の文字を小文字で保存するには、アプレットコンフィギュレーションモードで **action string tolower** コマンドを使用します。この機能をディセーブルにするには、このコマンドの **no** 形式を使用します。

action label string tolower *string* [*start-index*] [*end-index*]

no action label string tolower

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できます）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>string</i>	置換する必要がある文字のシーケンス。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>start-index</i>	（任意）開始インデックスのストリング値。指定できる範囲は 0 ～ 4294967295 です。
<i>end-index</i>	（任意）終了インデックスのストリング値。指定できる範囲は 0 ～ 4294967295 です。

コマンド デフォルト

ストリングは保存されません。

コマンド モード

アプレットコンフィギュレーション (applet-config)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン **action string tolower** コマンドを使用して、ストリングの特定の範囲の文字を小文字で保存します。*start-index* および *end-index* 引数は、操作対象のストリングの範囲を指定します。

次の表に、**action string tolower** コマンドの結果が保存される組み込み変数を示します。

表 14 : **action string tolower** コマンドの **EEM** 組み込み変数

組み込み変数	説明
<code>\$_string_result</code>	action string tolower コマンドの結果がこの変数に格納されます。

例 次に、特定のストリングの文字の範囲を小文字で保存する例を示します。

```
Router(config)# event manager applet lowercase
Router(config-applet)# action 1.0 set string "This is a STRING"
Router(config-applet)# action 2.0 string tolower "$string" 11 16
Router(config-applet)# action 3.0 puts "$_string_result"
Router(config-applet)# end
Router# event manager run lowercase
string
Router#
```

関連コマンド

コマンド	説明
action string toupper	ストリングの特定の範囲の文字を大文字で保存します。
event manager applet	EEMにイベント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。

action string toupper

Embedded Event Manager (EEM) アプレットがトリガーされたときに、ストリングの特定の範囲の文字を大文字で保存するには、アプレットコンフィギュレーションモードで **action string toupper** コマンドを使用します。この機能をディセーブルにするには、このコマンドの **no** 形式を使用します。

action *label string toupper* *string* [*start-index*] [*end-index*]

no *action label string toupper*

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できません）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>string</i>	置換する必要がある文字のシーケンスを指定します。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>start-index</i>	（任意）開始インデックスのストリング値。指定できる範囲は 0 ~ 4294967295 です。
<i>end-index</i>	（任意）終了インデックスのストリング値。指定できる範囲は 0 ~ 4294967295 です。

コマンド デフォルト

ストリングは保存されません。

コマンド モード

アプレットコンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン **action string toupper** コマンドを使用して、ストリングの特定の範囲の文字を大文字で保存します。 *start-index* および *end-index* 引数は、操作対象のストリングの範囲を指定します。

次の表に、**action string toupper** コマンドの結果が保存される組み込み変数を示します。

表 15: **action string toupper** コマンドの **EEM** 組み込み変数

組み込み変数	説明
<code>\$_string_result</code>	action string toupper コマンドの結果がこの変数に格納されます。

例 次に、特定のストリングの文字の範囲を大文字で保存する例を示します。

```
Router(config)# event manager applet uppercase
Router(config-applet)# action 1.0 set string "This is a string"
Router(config-applet)# action 2.0 string toupper "$string" 11 16
Router(config-applet)# action 3.0 puts "$_string_result"
Router(config-applet)# end
Router# event manager run uppercase
STRING
Router#
```

関連コマンド

コマンド	説明
action string tolower	ストリングの特定の範囲の文字を小文字で保存します。
event manager applet	EEMにイベント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。

action string trim

Embedded Event Manager (EEM) アプレットがトリガーされたときに、文字列をトリムするには、アプレット コンフィギュレーション モードで **action string trim** コマンドを使用します。この機能をディセーブルにするには、このコマンドの **no** 形式を使用します。

action label string trim string1 [string2]

no action label string trim

構文の説明

<i>label</i>	一意な識別子（任意の文字列値を指定できます）。アクションは、ラベルをソート キーとして使用して、英数字のキーの昇順にソートされ、実行されます。文字列に空白が含まれる場合は二重引用符で囲みます。
<i>string1</i>	文字のシーケンス。文字列に空白が含まれる場合は二重引用符で囲みます。
<i>string2</i>	(任意) 文字のシーケンス。文字列に空白が含まれる場合は二重引用符で囲みます。

コマンド デフォルト

デフォルトでは、文字列をトリムするアクションはありません。

コマンド モード

アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン

action string trim コマンドを使用して、文字列の文字をトリムします。このコマンドは、*string2* の文字を、*string1* の両端からトリムします。デフォルトでは、*string2* はスペースに対応します。

次の表に、**action string trim** コマンドの結果が保存される組み込み変数を示します。

表 16: **action string trim** コマンドの **EEM** 組み込み変数

組み込み変数	説明
<code>\$_string_result</code>	action string trim コマンドの結果がこの変数に格納されます。

例

次に、文字列をトリムする例を示します。

```
Router(config)# event manager applet trim
Router(config-applet)# action 1.0 set string "Hello How are you?Hello"
Router(config-applet)# action 2.0 string trim "$string" "Hello "
Router(config-applet)# action 3.0 puts "$_string_result"
Router(config-applet)# end
Router# event manager run trim
How are you?
Router#
```

関連コマンド

コマンド	説明
action string trimleft	1つの文字列分の文字を、別の文字列の左端からトリムします。
action string trimright	1つの文字列分の文字を、別の文字列の右端からトリムします。
event manager applet	EEMにイベント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。

action string trimleft

Embedded Event Manager (EEM) アプレットがトリガーされたときに、1つのストリング分の文字を、別のストリングの左端からトリムするには、アプレット コンフィギュレーション モードで **action string trimleft** コマンドを使用します。この機能をディセーブルにするには、このコマンドの **no** 形式を使用します。

action label string trimleft *string1* [*string2*]

no action label string trimleft

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できます）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>string1</i>	文字のシーケンス。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>string2</i>	（任意）文字のシーケンス。ストリングに空白が含まれる場合は二重引用符で囲みます。

コマンド デフォルト

デフォルトでは、ストリングをトリムするアクションはありません。

コマンド モード

アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン **action string trimleft** コマンドを使用して、文字列を別の文字列の左端からトリムします。このコマンドは、*string2* で指定された文字を、*string1* の左端からトリムします。デフォルトでは、*string2* はスペースに対応します。

次の表に、**action string trimleft** コマンドの結果が保存される組み込み変数を示します。

表 17: **action string trimleft** コマンドの EEM 組み込み変数

組み込み変数	説明
<code>\$_string_result</code>	action string trimleft コマンドの結果がこの変数に格納されます。

例 次に、1つの文字列を、別の文字列の左側からトリムする例を示します。

```
Router(config)# event manager applet trimleft
Router(config-applet)# action 1.0 set string "Hello How are you?"
Router(config-applet)# action 2.0 string trimleft "$string" "Hello "
Router(config-applet)# action 3.0 puts "$_string_result"
Router(config-applet)# end
Router# event manager run trimleft
How are you?
Router#
```

関連コマンド

コマンド	説明
action string trim	文字列をトリムします。
action string trimright	1つの文字列分の文字を、別の文字列の右端からトリムします。
event manager applet	EEMにイベント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。

action string trimright

Embedded Event Manager (EEM) アプレットがトリガーされたときに、1つのストリング分の文字を、別のストリングの右端からトリムするには、アプレット コンフィギュレーション モードで **action string trimright** コマンドを使用します。この機能をディセーブルにするには、このコマンドの **no** 形式を使用します。

action label string trimright string1 [string2]

no action label string trimright

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できます）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>string1</i>	文字のシーケンス。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>string2</i>	（任意）文字のシーケンス。ストリングに空白が含まれる場合は二重引用符で囲みます。

コマンド デフォルト

デフォルトでは、ストリングをトリムするアクションはありません。

コマンド モード

アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン

action string trimright コマンドを使用して、文字列を別の文字列の右端からトリムします。このコマンドは、*string2* で指定された文字を、*string1* の右端からトリムします。デフォルトでは、*string2* はスペースに対応します。

次の表に、**action string trimright** コマンドの結果が保存される組み込み変数を示します。

表 18: **action string trimright** コマンドの **EEM** 組み込み変数

組み込み変数	説明
<code>\$_string_result</code>	action string trimright コマンドの結果がこの変数に格納されます。

例

次に、1つの文字列を、別の文字列の右側からトリムする例を示します。

```
Router(config)# event manager applet trimright
Router(config-applet)# action 1.0 set string "How are you? Hello"
Router(config-applet)# action 2.0 string trim "$string" " Hello"
Router(config-applet)# action 3.0 puts "$_string_result"
Router(config-applet)# end
Router# event manager run trimright
How are you?
Router#
```

関連コマンド

コマンド	説明
action string trim	文字列をトリムします。
action string trimleft	1つの文字列分の文字を、別の文字列の左端からトリムします。
event manager applet	EEMにイベント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。

action subtract

Embedded Event Manager (EEM) アプレットがトリガーされたときに、変数の値を別の値から減算するアクションを指定するには、アプレット コンフィギュレーションモードで **action subtract** コマンドを使用します。減算のアクションを取り消すには、このコマンドの **no** 形式を使用します。

action *label* **subtract** {*variable-name*| *long-integer*} {*variable-name*| *long-integer*}

no **action** *label* **subtract**

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できます）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
subtract	変数の値を別の変数の値から引きます。
<i>variable-name</i>	変数名として配置されるストリング値。
<i>long-integer</i>	別の値を減算する長整数値。

コマンド デフォルト

デフォルトでは、EEM アプレット内に設定された変数の値は変更されません。

コマンド モード

アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン このアクションを使用して、変数の値を別の変数の値から減算できます。結果は `$_result` という名前の変数に保存されます。変数の値は長整数である必要があり、これ以外の場合はアクションが失敗します。

例 次に、変数の値を別の値から減算するように EEM アプレットを設定する例を示します。

```
Router(config)#event manager applet one
Router(config-applet)#action 1.0 set $var1 20
Router(config-applet)#action 1.0 set $var2 10
Router(config-applet)#action 1.0 subtract $var1 $var2
Router(config-applet)#
```

関連コマンド

コマンド	説明
event manager applet	Embedded Event Manager にイベント 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。

action track read

Embedded Event Manager (EEM) アプレットがトリガーされたときに、トラッキング対象オブジェクトの状態を読み取るアクションを指定するには、アプレットコンフィギュレーションモードで **action track read** コマンドを使用します。コンフィギュレーションから **action track read** コマンドを削除するには、このコマンドの **no** 形式を使用します。

action label track read object-number

no action label track read object-number

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できます）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>object-number</i>	1 から 500 の範囲のトラッキング対象オブジェクトの番号。番号は、 track stub コマンドを使用して定義されます。

コマンド デフォルト

トラッキング対象オブジェクトの状態は読み取られません。

コマンド モード

アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(2)T	このコマンドが導入されました。
12.2(31)SB3	このコマンドが、Cisco IOS Release 12.2(31)SB3 に統合されました。
12.2(33)SRB	このコマンドが、Cisco IOS Release 12.2(33)SRB に統合されました。
12.2(33)SXI	このコマンドが、Cisco IOS Release 12.2(33)SXI に統合されました。

使用上のガイドライン

このコマンドは、次の結果の変数を生成します。

- **_track_state** : 指定されたトラッキング対象オブジェクトの状態。返されるテキストストリングは、**up** または **down** のいずれかです。状態が **up** の場合、オブジェクトが存在し、アップ状態であることを意味します。状態が **down** の場合、オブジェクトが存在しないか、またはダウン状態であることを意味します。

このコマンドは、EEMを使用したオブジェクトのトラッキングを支援するために使用されます。各トラッキング対象オブジェクトは、トラッキング コマンドライン インターフェイス (CLI) で指定された一意の番号で識別されます。EEM などのクライアント プロセスは、この番号を使用して特定のオブジェクトをトラッキングします。トラッキングプロセスは定期的に、トラッキング対象オブジェクトをポーリングし、値の変更を確認します。トラッキング対象オブジェクトの変更は、すぐに、または指定された遅延後に、対象のクライアントプロセスに通知されます。オブジェクトの値は、アップまたはダウンとして報告されます。拡張されたオブジェクトトラッキング イベント デテクタは、トラッキング対象オブジェクトが変化したときに EEM イベントをパブリッシュします。

例

次に、トラッキング対象オブジェクトに基づいてイベント基準を指定する例を示します。

```
event manager applet track-ten
event track 10 state any
action 1.0 track set 10 state up
action 2.0 track read 10
```

関連コマンド

コマンド	説明
action track set	EEM アプレットがトリガーされたときに、トラッキング対象オブジェクトの状態を設定するアクションを指定します。
event manager applet	Embedded Event Manager にイベント 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。
show track	トラッキング情報を表示します。
track stub	トラッキング対象のスタブオブジェクトを作成します。

action track set

Embedded Event Manager (EEM) アプレットがトリガーされたときに、トラッキング対象オブジェクトの状態を設定するアクションを指定するには、アプレットコンフィギュレーションモードで **action track set** コマンドを使用します。コンフィギュレーションから **action track set** コマンドを削除するには、このコマンドの **no** 形式を使用します。

action label track set object-number state {up| down}

no action label track set object-number state {up| down}

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できます）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>object-number</i>	1 から 500 の範囲のトラッキング対象オブジェクトの番号。番号は、 track stub コマンドを使用して定義されます。
state	トラッキング対象オブジェクトが設定される状態を指定します。
up	トラッキング対象オブジェクトの状態が up に設定されるよう指定します。
down	トラッキング対象オブジェクトの状態が down に設定されるよう指定します。

コマンド デフォルト

トラッキング対象オブジェクトの状態は設定されません。

コマンド モード

アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(2)T	このコマンドが導入されました。

リリース	変更内容
12.2(31)SB3	このコマンドが、Cisco IOS Release 12.2(31)SB3 に統合されました。
12.2(33)SRB	このコマンドが、Cisco IOS Release 12.2(33)SRB に統合されました。
12.2(33)SXI	このコマンドが、Cisco IOS Release 12.2(33)SXI に統合されました。

使用上のガイドライン このコマンドは、次の結果の変数を生成します。

- **_track_state** : 指定されたトラッキング対象オブジェクトの状態。返されるテキストストリングは、**up** または **down** のいずれかです。状態が **up** の場合、オブジェクトが存在し、アップ状態であることを意味します。状態が **down** の場合、オブジェクトが存在しないか、またはダウン状態であることを意味します。

このコマンドは、EEMを使用したオブジェクトのトラッキングを支援するために使用されます。各トラッキング対象オブジェクトは、トラッキング コマンドライン インターフェイス (CLI) で指定された一意の番号で識別されます。EEM などのクライアント プロセスは、この番号を使用して特定のオブジェクトをトラッキングします。トラッキングプロセスは定期的に、トラッキング対象オブジェクトをポーリングし、値の変更を確認します。トラッキング対象オブジェクトの変更は、すぐに、または指定された遅延後に、対象のクライアントプロセスに通知されます。オブジェクトの値は、アップまたはダウンとして報告されます。拡張されたオブジェクトトラッキング イベント デテクタは、トラッキング対象オブジェクトが変化したときに EEM イベントをパブリッシュします。

例 次に、トラッキング対象オブジェクトに基づいてイベント基準を指定する例を示します。

```
event manager applet track-ten
event track 10 state any
action 1.0 track set 10 state up
action 2.0 track read 10
```

関連コマンド

コマンド	説明
action track read	EEM アプレットがトリガーされたときに、トラッキング対象オブジェクトの状態を読み取るアクションを指定します。

コマンド	説明
event manager applet	Embedded Event Manager にイベント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。
show track	トラッキング情報を表示します。
track stub	トラッキング対象のスタブオブジェクトを作成します。

action while

Embedded Event Manager (EEM) アプレットがトリガーされたときに、条件付きブロックのループの開始を識別するには、アプレット コンフィギュレーション モードで **action while** コマンドを使用します。この機能をディセーブルにするには、このコマンドの **no** 形式を使用します。

action label while string-op1 operator string-op2

no action label while

構文の説明

<i>label</i>	一意な識別子（任意のストリング値を指定できます）。アクションは、ラベルをソートキーとして使用して、英数字のキーの昇順にソートされ、実行されます。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>string-op1</i>	第 1 オペランドを指定します。
<i>operator</i>	<i>string-op1</i> および <i>string-op2</i> オペランドと一緒に使用される値で、現在のカウンタの値をどのように開始値または終了値と比較するかを決定します。次の値が有効です。 <ul style="list-style-type: none"> • gt : より大きい。 • ge : 以上。 • eq : 等しい。 • ne : 等しくない。 • lt : より小さい。 • le : 以下。
<i>string-op2</i>	第 2 オペランド。

コマンド デフォルト 条件付きブロックは指定されません。

コマンド モード アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン

action while コマンドを使用して、ループ条件付きブロックの開始を識別します。ストリング内に `$_variable` が存在する場合は、式がテストされる前に代入されます。

例

次に、EEM アプレットがトリガーされたときに、条件付きブロックのループの開始を識別する例を示します。

```
Router(config-applet)# action 1 set _i 2
Router(config-applet)# action 2 while $_i lt 10
Router(config-applet)# action 3 action syslog msg "i is $_i"
Router(config-applet)# action 4 end
```

関連コマンド

コマンド	説明
action else	if/else 条件付きブロックの else ブロックの開始を識別します。
action elseif	if/else 条件付きブロックの開始を識別します。
action if	if 条件付きブロックの開始を識別します。
event manager applet	EEM にイベント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。

attribute (EEM)

Embedded Event Manager (EEM) アプレットの複合イベントを指定するには、トリガー アプレット コンフィギュレーション モードで **attribute** コマンドを使用します。属性を削除するには、このコマンドの **no** 形式を使用します。

attribute tag event-tag [occurs occurs-value]

no attribute tag event-tag [occurs occurs-value]

構文の説明

tag	イベントを関連付けるために attribute コマンドで使用できる <i>event-tag</i> 引数を使用して、タグを指定します。
<i>event-tag</i>	タグを識別するストリング。
occurs	(任意) EEM イベントがトリガーされる前の発生数を指定します。指定されない場合、EEM イベントは 1 回目から発生します。
<i>occurs-value</i>	(任意) 番号の範囲は 1 ~ 4294967295 です。

コマンド デフォルト

EEM アプレットには、複合イベントは指定されていません。

コマンド モード

トリガー アプレット コンフィギュレーション (config-applet-trigger)

コマンド履歴

リリース	変更内容
12.4(20)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン

トリガー アプレット コンフィギュレーション モードでは、複合イベントを作成するために、最大 8 個の属性文を指定できます。属性文が指定されていない場合、トリガー文のオプションは、アプレットで定義された最初のイベントに適用されます。

例

次に、**attributes** コマンドを使用して EEM アプレットに複合イベントを指定する例を示します。この例では、60 秒の期間内に **show bgp all** コマンドと「COUNT」というストリングを含む syslog メッセージが出現した場合にアプレットが実行されます。

```
Router(config)# event manager applet delay_50
Router(config-applet)# event
  tag 1.0 cli pattern "show bgp all" sync yes occurs 32 period 60 maxrun 60
Router(config-applet)# event
  tag 2.0 syslog pattern "COUNT"
Router(config-applet)# trigger occurs 1 delay 50
Router(config-applet-trigger)# correlate event 1.0 or event 2.0
Router(config-applet-trigger)# attribute tag 1.0 occurs 1
Router(config-applet-trigger)# attribute tag 2.0 occurs 1
Router(config-applet-trigger)# action 1.0 cli command "show memory"
Router(config-applet)# action 2.0 cli command "enable"
Router(config-applet)# action 3.0 cli command "config terminal"
Router(config-applet)# action 4.0 cli command " ip route 192.0.2.0 255.255.255.224 192.0.2.12"
Router(config-applet)# action 91.0 cli command "exit"
Router(config-applet)# action 99.0 cli command "show ip route | incl 192.0.2.5"
```

関連コマンド

コマンド	説明
correlate	1つの複合イベントを構築します。
trigger (EEM)	トリガー アプレット コンフィギュレーション モードを開始し、EEM アプレットに対して複数 イベントのコンフィギュレーション文を指定します。

description (EEM)

組み込みイベントマネージャ (EEM) アプレットが実行する内容を説明するには、アプレットコンフィギュレーションモードで **description** (EEM) コマンドを使用します。アプレットの説明を削除するには、このコマンドの **no** 形式を使用します。

description line

no description

構文の説明

line

240 文字までポリシーの簡単な説明。

コマンド デフォルト

デフォルトでは、アプレットに説明は指定されていません。

コマンド モード

アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース

変更内容

15.0(1)M

このコマンドが導入されました。

使用上のガイドライン

このコマンドを使用して、EEM アプレットが実行する内容を説明します。アプレットに説明を付けなくても有効です。アプレットの説明は、他のアプレット設定の前でも後でも、任意の順序で追加できます。すでに説明があるアプレットに新しい説明を設定すると、現在の説明が上書きされます。

例

次に、EEM の説明を追加または変更する例を示します。

```
Router(config)# event manager applet one
Router(config-applet)# description "This applet looks for the word count in syslog messages"
Router(config-applet)# event syslog pattern
"count"
Router(config-applet)# action 1 syslog msg hi
```

関連コマンド

コマンド	説明
show event manager policy active	実行される EEM ポリシーを表示します。
show event manager policy available	登録可能な EEM ポリシーを表示します。



E から event manager までのコマンド

- [E から event manager までのコマンド, 105 ページ](#)

E から event manager までのコマンド

event application

EEM イベントパブリッシュアプリケーションプログラミングインターフェイス (API) 経由で発生したイベント単位で起動される Embedded Event Manager (EEM) アプレットのイベント条件を指定するには、アプレットコンフィギュレーションモードで **event application** コマンドを使用します。アプリケーションイベント条件を削除するには、このコマンドの **no** 形式を使用します。

event [**tag** *event-tag*] **application** **subsystem** *subsystem-id* **type** *event-type* [**maxrun** *maxruntime-number*]
no [**tag** *event-tag*] **event application** **subsystem** *subsystem-id* **type** *event-type* [**maxrun** *maxruntime-number*]

構文の説明

tag	(任意) trigger コマンドで使用できる <i>event-tag</i> 引数を使用してタグを指定し、アプレットの複数のイベント文をサポートします。
<i>event-tag</i>	(任意) タグを識別するストリング。
2subsystem	アプリケーションイベントをパブリッシュするサブシステムの ID を指定します。
<i>subsystem-id</i>	サブシステムを識別する 1 ~ 4294967295 の範囲の番号。EEM ポリシーによりイベントがパブリッシュされる場合、ポリシー用に予約されている <i>subsystem-id</i> は 798 です。
type	指定されたイベント内のイベントタイプを指定します。
<i>event-type</i>	1 ~ 4294967295 の範囲の整数。
maxrun	(任意) アプレットの最大実行時間を指定します。 maxrun キーワードが指定されている場合、 <i>maxruntime-number</i> 値を指定する必要があります。 maxrun キーワードが指定されていない場合、デフォルトのアプレットランタイムは 20 秒です。
<i>maxruntime-number</i>	(任意) <i>sssssss[.mmm]</i> 形式で指定される秒数。 <i>sssssss</i> は、0 ~ 31536000 の秒数を表す整数で、 <i>mmm</i> は 0 ~ 999 のミリ秒数を表す整数である必要があります。

コマンド モデル **EEM** イベント標準は指定されなければ (**config-applet**)

コマンド履歴

リリース	変更内容
12.2(25)S	このコマンドが導入されました。
12.3(14)T	このコマンドが、Cisco IOS Release 12.3(14)T に統合されました。
12.2(28)SB	このコマンドが、Cisco IOS Release 12.2(28)SB に統合されました。
12.2(18)SXF4	このコマンドが Cisco IOS Release 12.2(18)SXF4 に統合され、ソフトウェア モジュラリティ イメージだけをサポートするようになりました。
12.2(33)SRA	このコマンドが、Cisco IOS Release 12.2(33)SRA に統合されました。
12.2(18)SXF5	このコマンドが、Cisco IOS Release 12.2(18)SXF5 に統合されました。
12.2SX	このコマンドは、Cisco IOS Release 12.2SX トレインでサポートされます。このトレインの特定の 12.2SX リリースにおけるサポートは、フィチャセット、プラットフォーム、およびプラットフォームハードウェアによって異なります。
12.4(20)T	tag および maxrun キーワードが、アプレット内の複数のイベント文をサポートするために追加されました。

使用上のガイドライン

EEM イベントは、アプリケーションがサブシステム ID およびアプリケーション イベントタイプに一致するイベント仕様で EEM イベント パブリッシュ API をコールしたときにトリガーされます。

例

次に、EventPublish_A という名前のポリシーが 20 秒ごとに実行され、番号が 1 の既知の EEM イベントタイプにイベントをパブリッシュする例を示します。EventPublish_B という名前の別のポリシーは、既知の EEM イベントタイプ 1 が発生したときに実行されるように登録されます。EventPublish_B ポリシーが実行されると、このポリシーにより EventPublish_A の引数として渡されたデータを含む syslog に、メッセージが出力されます。

```
Router(config)# event manager applet EventPublish_A
Router(config-applet)# event timer watchdog time 20.0
Router(config-applet)# action 1.0 syslog msg "Applet EventPublish_A"
Router(config-applet)# action 2.0 publish-event sub-system 798 type 1 arg1 twenty
Router(config-applet)# exit
Router(config)# event manager applet EventPublish_B
Router(config-applet)# event application subsystem 798 type 1
Router(config-applet)# action 1.0 syslog msg "Applet EventPublish_B arg1
$_application_data1"
```

関連コマンド

コマンド	説明
event manager applet	Embedded Event Manager にイベント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。

event identity

認証、許可、またはインターフェイス上で通常のトラフィックが流れ始めた後にイベントをパブリッシュするには、アプレット コンフィギュレーション モードで **event identity** コマンドを使用します。 イベントのパブリッシュをディセーブルにするには、このコマンドの **no** 形式を使用します。

```
event [tag event-tag] identity interface {type number|regexp interface-name} [maxrun maxruntime-number]
[aaa-attribute attribute-name] [authc {all|fail|success}] [authz {all|fail|success}] [authc-complete]
[mac-address mac-address]
```

```
no event identity
```

構文の説明

tag	(任意) trigger コマンドで使用できる event-tag 引数を使用してタグを指定し、アプレットの複数のイベント文をサポートします。
<i>event-tag</i>	(任意) タグを識別するストリング。
interface	インターフェイスを指定します。
<i>type number</i>	インターフェイスのタイプと番号。
regexp <i>interface-name</i>	インターフェイス名と照合する正規表現パターンを指定します。
maxrun	(任意) アプレットの最大実行時間を指定します。 maxrun キーワードが指定されている場合、maxruntime-number 値を指定する必要があります。 maxrun キーワードが指定されていない場合、デフォルトのアプレットランタイムは 20 秒です。
<i>maxruntime-number</i>	(任意) ssssssss[.mmm] 形式で指定される秒数。 ssssssss は、0 ~ 31536000 の秒数を表す整数で、mmm は 0 ~ 999 のミリ秒数を表す整数である必要があります。
aaa-attribute	(任意) AAA 属性用の正規表現パターンを指定します。
<i>attribute-name</i>	(任意) AAA 属性名。

authc	<p>(任意) 成功した認証、失敗した認証、または成功と失敗の両方の認証で、イベントをトリガーします。次のいずれかを指定する必要があります。</p> <ul style="list-style-type: none"> • all : すべてのケースの認証でイベントをトリガーします。 • fail : 認証が失敗した場合にイベントをトリガーします。 • success : 認証が成功した場合にイベントをトリガーします。
authz	<p>(任意) 成功した許可、失敗した許可、または成功と失敗の両方の許可で、イベントをトリガーします。次のいずれかを指定する必要があります。</p> <ul style="list-style-type: none"> • all : すべてのケースの許可でイベントをトリガーします。 • fail : 許可が失敗した場合にイベントをトリガーします。 • success : 許可が成功した場合にイベントをトリガーします。
authz-complete	<p>(任意) インターフェイスに接続されたデバイスが完全に認証、許可され、通常のトラフィックがそのインターフェイスで流れ始めたときにイベントをトリガーします。</p>
mac-address	<p>(任意) MAC アドレスを指定します。</p>
<i>mac-address</i>	<p>(任意) MAC アドレス。</p>

コマンド デフォルト デフォルトでは、イベントはパブリッシュされません。

コマンド モード アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.2(52)SE	このコマンドが導入されました。
12.2(54)SG	このコマンドが、Cisco IOS Release 12.2(54)SG に統合されました。

使用上のガイドライン

インターフェイスを指定する必要があります。任意またはすべての他のキーワードを指定できません。キーワードは任意の組み合わせで使用できます。

例

次に、認証が成功または失敗し、インターフェイスに接続されているデバイスが完全に認証、許可され、そのインターフェイスで通常のトラフィックが流れ始めたときにイベントをパブリッシュする例を示します。

```
Router(config)# event manager applet identity
Router(config-applet)# event identity interface fastethernet0 authz all athuz-complete
Router(config-applet)#
```

関連コマンド

コマンド	説明
event manager applet	Embedded Event Manager にイベント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。

event ipsla

Embedded Event Manager (EEM) のアプレットに対して IP SLA 操作がトリガーされたときにイベントをパブリッシュするには、アプレットコンフィギュレーションモードで **event ipsla** コマンドを使用します。IP SLA の応答がトリガーされたときのイベントのパブリッシュをディセーブルにするには、このコマンドの **no** 形式を使用します。

event [**tag** *event-tag*] **ipsla** {**group-name** *name* [**operation-id** *operation-id-value*]} **operation-id** *operation-id-value* [**group-name** *name*] [**dest-ip-address** *ip-address*] [**reaction-type** *type*] [**maxrun** *maxruntime-number*]

no event [**tag** *event-tag*] **ipsla**

構文の説明

tag	(任意) trigger コマンドで使用できる <i>event-tag</i> 引数を使用してタグを指定し、アプレットの複数のイベント文をサポートします。
<i>event-tag</i>	(任意) タグを識別するストリング。
group-name	IP SLA グループ ID を指定します。
<i>name</i>	IP SLA グループの名前。
operation-id	IP SLA 動作 ID を指定します。
<i>operation-id-value</i>	番号の範囲は 1 ~ 2147483647 です。
dest-ip-address	(任意) IP SLA イベントがモニタされる宛先 IP アドレスを指定します。
<i>ip-address</i>	(任意) 宛先ポートの IP アドレスを指定します。
reaction-type	(任意) 指定した IP SLA 動作に対する応答を指定します。

<i>type</i>	<p>(任意) IPSLA の応答のタイプ。次のキーワードの 1 つを指定できます。</p> <ul style="list-style-type: none">• connectionLoss : 監視対象動作で一方向接続損失があった場合に、応答が発生することを指定します。• icpif : 一方向の Calculated Planning Impairment Factor (ICPIF) の値が上限しきい値または下限しきい値に違反すると、応答が発生することを指定します。• jitterAvg : 平均ラウンドトリップジッター値が、上限しきい値または下限しきい値に違反すると、応答が発生することを指定します。• jitterDSAvg : 平均の一方向の (宛先から送信元) ジッター値が、上限しきい値または下限しきい値に違反すると、応答が発生することを指定します。• jitterSDAvg : 平均の一方向の (送信元から宛先) ジッター値が、上限しきい値または下限しきい値に違反すると、応答が発生することを指定します。• maxOfNegativeDS : 一方向の (宛先から送信元) 最大の負のジッターしきい値に違反すると、応答が発生することを指定します。• maxOfNegativeSD : 一方向の (送信元から宛先) 最大の負のジッターしきい値に違反すると、応答が発生することを指定します。
-------------	--

event ipsla



- **maxOfPositiveDS** : 一方向の（宛先から送信元）最大の正のジッターしきい値に違反すると、応答が発生することを指定します。
- **maxOfPositiveSD** : 一方向の（送信元から宛先）最大の正のジッターしきい値に違反すると、応答が発生することを指定します。
- **mos** : 一方向の平均オピニオン評点（MOS）値が、上限しきい値または下限しきい値に違反すると、応答が発生することを指定します。
- **packetLateArrival** : 一方向の遅延パケット数が、上限しきい値または下限しきい値に違反すると、応答が発生することを指定します。
- **packetLossDS** : 一方向の（宛先から送信元）パケット損失値が、上限しきい値または下限しきい値に違反すると、応答が発生することを指定します。
- **packetLossSD** : 一方向の（送信元から宛先）パケット損失値が、上限しきい値または下限しきい値に違反すると、応答が発生することを指定します。
- **packetMIA** : 一方向の不明パケット数が、上限しきい値または下限しきい値に違反すると、応答が発生することを指定します。
- **packetOutOfSequence** : 一方向のシーケンスを外れたパケット数が、上限しきい値または下限しきい値に違反すると、応答が発生することを指定します。
- **rtt** : ラウンドトリップ時間が、上限しきい値または下限しきい値に違反すると、応答が発生することを指定します。
- **timeout** : 監視対象動作で一方向タイムアウトがあった場合に、応答が発生することを指定します。
- **verifyError** : 一方向のエラー検証違反があった場合に、応答が発生することを指定

	します。
maxrun	(任意) アプレットの最大実行時間を指定します。 maxrun キーワードが指定されている場合、 <i>maxruntime-number</i> 値を指定する必要があります。 maxrun キーワードが指定されていない場合、デフォルトのアプレットランタイムは 20 秒です。
<i>maxruntime-number</i>	(任意) <i>sssssss[.mmm]</i> 形式で指定される秒数。 <i>sssssss</i> は、0～31536000 の秒数を表す整数で、 <i>mmm</i> は 0～999 のミリ秒数を表す整数である必要があります。

コマンド デフォルト

イベントは IP SLA 動作がトリガーされたときにパブリッシュされません。

コマンド モード

アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン

EEM イベントは、IP SLA 応答がトリガーされるときにパブリッシュされます。**group-name** または **operation-id** のいずれかを指定する必要があります。残りのパラメータはオプションです。

例

次に、IP SLA 動作がトリガーされたときにイベントをパブリッシュする例を示します。この例では、**grp1** という名前のグループが、3 秒ごとに現在のインターフェイスを経由して宛先サーバ 209.165.200.221 に ping を送信します。応答がない場合、アクションはタイムアウトになります。

```
Router# configure terminal
Router(config)# event manager applet EventIPSLA
Router(config-applet)# event ipsla group-name grp1 dest-ip-address 209.165.200.221
reaction-type timeout maxrun 3
```

関連コマンド

コマンド	説明
event manager applet	EEM にイベント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。

event manager applet

Embedded Event Manager (EEM) にアプレットを登録してアプレット コンフィギュレーション モードを開始するには、グローバル コンフィギュレーション モードで **event manager applet** コマンドを使用します。アプレットを登録解除するには、このコマンドの **no** 形式を使用します。

event manager applet *applet-name* [**authorization bypass**] [**class class-options**] [**trap**]

no event manager applet *applet-name* [**authorization bypass**] [**class class-options**] [**trap**]

構文の説明

<i>applet-name</i>	アプレット ファイルの名前。
authorization	(任意) アプレットの AAA 許可タイプを指定します。
bypass	(任意) EEM の AAA 許可タイプのバイパスを指定します。
class	(任意) EEM ポリシー クラスを指定します。
<i>class-options</i>	(任意) EEM ポリシー クラス。次のいずれかを指定できます： <ul style="list-style-type: none"> • <i>class-letter</i> : 各ポリシー クラスを識別する A ~ Z の文字。任意の <i>class-letter</i> を 1 つ 指定できます。 • default : デフォルトクラスに登録されたポリシーを指定します。
trap	(任意) ポリシーがトリガーされたときに簡易 ネットワーク管理プロトコル (SNMP) トラップを生成します。

コマンド デフォルト EEM アプレットは登録されません。

コマンド モード グローバル コンフィギュレーション (config)

コマンド履歴

リリース	変更内容
12.0(26)S	このコマンドが導入されました。
12.3(4)T	このコマンドが Cisco IOS Release 12.3(4)T に統合されました。
12.3(2)XE	このコマンドが、Cisco IOS Release 12.3(2)XE に統合されました。
12.2(25)S	このコマンドが、Cisco IOS Release 12.2(25)S に統合されました。
12.2(28)SB	このコマンドが、Cisco IOS Release 12.2(28)SB に統合されました。
12.2(18)SXF4	このコマンドが Cisco IOS Release 12.2(18)SXF4 に統合され、ソフトウェアモジュラリティイメージだけをサポートするようになりました。
12.2(33)SRA	このコマンドが、Cisco IOS Release 12.2(33)SRA に統合されました。
12.2(18)SXF5	このコマンドが、Cisco IOS Release 12.2(18)SXF5 に統合されました。
12.2SX	このコマンドは、Cisco IOS Release 12.2SX トレインでサポートされません。このトレインの特定の 12.2SX リリースにおけるサポートは、フィーチャセット、プラットフォーム、およびプラットフォームハードウェアによって異なります。
12.4(22)T	class および trap キーワードと <i>class-options</i> 引数が追加されました。
15.0(1)M	コマンドが変更されました。 authorization および bypass キーワードが追加されました。

使用上のガイドライン

EEM アプレットは、イベントスクリーニング基準とイベント発生時に実行するアクションを定義する簡潔な方法です。

アプレット コンフィギュレーションでは、**event** コンフィギュレーション コマンドを 1 つだけ使用できます。アプレット コンフィギュレーション サブモードが終了し、**event** コマンドが存在しない場合は、アプレットにイベントが関連付けられていないことを示す警告が表示されます。イベントが指定されていない場合、このアプレットは登録されたと判断されないため、アプレットは表示されません。このアプレットにアクションが割り当てられない場合、イベントはトリガーされますが、アクションは実行されません。1 つのアプレット コンフィギュレーション内で複数の **action** アプレット コンフィギュレーション コマンドが使用できます。 **show event manager policy registered** コマンドを使用して、登録されたアプレットの一覧を表示します。

アプレット コンフィギュレーション モードを終了しないと既存のアプレットが置き換えられないため、EEM アプレットを変更する前に、このコマンドの **no** 形式を使用して登録を解除します。

アプレットコンフィギュレーションモードでアプレットを修正中であっても、既存のアプレットを実行できます。アプレットコンフィギュレーションモードを終了すると、古いアプレットが登録解除され、新しいバージョンが登録されます。

action コンフィギュレーションコマンドは、*label* 引数を使用することで一意に識別できます。*label* 引数には任意の文字列値が使用できます。アクションは、*label* 引数をソートキーとして、英数字のキーの昇順にソートされ、この順序で実行されます。

EEMは、ポリシー自体に含まれているイベントの指定内容に基づいて、ポリシーをスケジューリングおよび実行します。アプレットコンフィギュレーションモードが終了するとき、EEMは、入力された **event** コマンドと **action** コマンドを検査し、指定されたイベントの発生時に実行されるようにアプレットを登録します。

EEM ポリシーは、登録されたときに **class class-letter** が指定されている場合はクラスに割り当てられます。クラスなしで登録された EEM ポリシーは、**default** クラスに割り当てられます。**default** をクラスとして保持するスレッドは、スレッドが作業に利用可能であるとき、デフォルトクラスにサービスを提供します。特定のクラス文字に割り当てられたスレッドは、スレッドが作業に利用可能であるとき、クラス文字が一致する任意のポリシーをサービスします。

EEM実行スレッドが、指定されたクラスのポリシー実行に利用可能でない場合で、クラスのスケジューラールールが設定されている場合は、ポリシーは該当クラスのスレッドが実行可能になるまで待ちます。同じ入力イベントからトリガーされた同期ポリシーは、同一の実行スレッドにスケジューラールールされなければなりません。ポリシーは、**queue_priority** をキューイング順序として使用し、各クラスの別々のキューにキューイングされます。

ポリシーがトリガーされると、AAA が設定されている場合は、許可のために AAA サーバに接続します。**authorization bypass** キーワードの組み合わせを使用して、AAA サーバへの接続をスキップし、ポリシーをただちに実行することができます。EEMは、AAA バイパスポリシー名をリストに保存します。このリストは、ポリシーがトリガーされたときに検査されます。一致が見つかった場合、AAA 許可はバイパスされます。

EEM ポリシーによって設定されたコマンドの許可を避けるために、EEMは AAA が提供する名前付き方式リストを使用します。これらの名前付き方式リストは、コマンド許可を持たないように設定できます。

次に、AAA の設定例を示します。

この設定は、192.168.10.1 のポート 10000 に TACACS+ サーバを想定しています。TACACS+ サーバがイネーブルでない場合、コンフィギュレーションコマンドは、コンソールで許可されます。ただし、EEM ポリシーとアプレット CLI の相互作用は失敗します。

```
enable password lab
aaa new-model
tacacs-server host 128.107.164.152 port 10000
tacacs-server key cisco
aaa authentication login consoleline none
aaa authorization exec consoleline none
aaa authorization commands 1 consoleline none
aaa authorization commands 15 consoleline none
line con 0
  exec-timeout 0 0
  login authentication consoleline
aaa authentication login default group tacacs+ enable
aaa authorization exec default group tacacs+
aaa authorization commands 1 default group tacacs+
aaa authorization commands 15 default group tacacs+
```

authorization、**class** および **trap** キーワードは任意の組み合わせで使用できます。

例

次に、IPSLAping1 という名前の EEM アプレットが登録され、指定された SNMP オブジェクト ID の値と完全一致する（正常な IP SLA ICMP エコー動作を表す）場合に実行される例を示します（これは **ping** コマンドに相当します）。エコー操作が失敗した場合は 4 つのアクションがトリガーされ、イベントモニタリングは 2 回目の失敗後までディセーブルにされます。サーバへの ICMP エコー動作が失敗したことを示すメッセージが syslog に送信され、SNMP トラップが生成され、EEM はアプリケーション固有のイベントをパブリッシュし、IPSLA1F というカウンタが値 1 で増分されます。

```
Router(config)# event manager applet IPSLAping1
Router(config-applet)# event snmp oid 1.3.6.1.4.1.9.9.42.1.2.9.1.6.4 get-type exact
entry-op eq entry-val 1 exit-op eq exit-val 2 poll-interval 5
Router(config-applet)# action 1.0 syslog priority critical msg "Server IP echo failed:
OID=${_snmp_oid_val}"
Router(config-applet)# action 1.1 snmp-trap strdata "EEM detected server reachability
failure to 10.1.88.9"
Router(config-applet)# action 1.2 publish-event sub-system 88000101 type 1 arg1 10.1.88.9
arg2 IPSLAEcho arg3 fail
Router(config-applet)# action 1.3 counter name _IPSLA1F value 1 op inc
```

次に、名前 one、クラス A でアプレットを登録し、タイマーイベントディテクタが 10 秒ごとにイベントをトリガーするアプレット コンフィギュレーションモードを開始する例を示します。イベントがトリガーされると、**action syslog** コマンドにより、syslog にメッセージ「hello world」が書き込まれます。

```
Router(config)# event manager applet one class A
Router(config-applet)# event timer watchdog time 10
Router(config-applet)# action syslog syslog msg "hello world"
Router(config-applet)# exit
```

次に、名前 one、クラス A でアプレットを登録するときに、AAA 許可をバイパスする例を示します。

```
Router(config)# event manager applet one class A authorization bypass
Router(config-applet)#
```

関連コマンド

コマンド	説明
show event manager policy registered	登録されている EEM ポリシーを表示します。

event manager detector routing

ルーティング イベント ディテクタがモニタリング イベントを開始する際の遅延時間を設定するには、グローバル コンフィギュレーション モードで **event manager detector routing** コマンドを使用します。遅延時間をディセーブルにするには、このコマンドの **no** 形式を使用します。

event manager detector routing bootup-delay *delay-time*

no event manager detector routing

構文の説明

bootup-delay	起動後にモニタリングをオンにする時間遅延を指定します。
<i>delay-time</i>	秒数およびオプションのミリ秒数を ssssssssss [.mmm] 形式で表した数字。秒数の範囲は 0 ～ 4294967295 です。ミリ秒の範囲は 0 ～ 999 です。ミリ秒だけを使用する場合は、0.mmm の形式でミリ秒を指定します。

コマンド デフォルト

ルーティング イベント ディテクタ コマンドは設定されません。

コマンド モード

グローバル コンフィギュレーション (config)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン

起動後に遅延時間が経過してからルーティング アップデートをオンにするように設定するには、**event manager detector routing** コマンドを使用します。設定された場合、ルーティング イベント ディテクタは、起動遅延時間後にイベントのモニタリングを開始します。起動遅延時間に到達すると、ルーティング アップデートがオンになり、ポリシーはトリガーを開始します。

例

次に、ルーティング アップデートをオンにための遅延時間を設定する例を示します。

```
Router(config)# event manager detector routing bootup-delay 800
```

関連コマンド

コマンド	説明
event manager detector rpc	RPC イベント デテクタ コマンドを使用して EEM アプレットを受け入れるようにルータを設定します。

event manager directory user

ユーザライブラリファイルまたはユーザ定義の Embedded Event Manager (EEM) ポリシーを保存するディレクトリを指定するには、グローバルコンフィギュレーションコマンドで **event manager directory user** コマンドを使用します。ユーザライブラリファイルまたはユーザ定義の EEM ポリシーを保存するディレクトリの使用をディセーブルにするには、このコマンドの **no** 形式を使用します。

event manager directory user {*library path*| *policy path*}

no event manager directory user {*library path*| *policy path*}

構文の説明

library	ディレクトリを使用してユーザライブラリファイルを保存するように指定します。
policy	ディレクトリを使用してユーザ定義の EEM ポリシーを保存するように指定します。
<i>path</i>	フラッシュ デバイスのユーザ ディレクトリの絶対パス名。

コマンド デフォルト

ユーザライブラリファイルまたはユーザ定義の EEM ポリシーを保存するディレクトリは指定されていません。

コマンド モード

グローバル コンフィギュレーション

コマンド履歴

リリース	変更内容
12.3(14)T	このコマンドが導入されました。
12.2(28)SB	このコマンドが、Cisco IOS Release 12.2(28)SB に統合されました。
12.2(18)SXF4	このコマンドが Cisco IOS Release 12.2(18)SXF4 に統合され、ソフトウェアモジュラリティ イメージだけをサポートするようになりました。
12.2(33)SRA	このコマンドが、Cisco IOS Release 12.2(33)SRA に統合されました。
12.2(18)SXF5	このコマンドが、Cisco IOS Release 12.2(18)SXF5 に統合されました。

リリース	変更内容
12.2SX	このコマンドは、Cisco IOS Release 12.2SX トレインでサポートされます。このトレインの特定の12.2SXリリースにおけるサポートは、フィーチャセット、プラットフォーム、およびプラットフォームハードウェアによって異なります。

使用上のガイドライン

ユーザライブラリディレクトリは、EEMポリシーの作成に関連するユーザライブラリファイルを保存するために必要です。EEMポリシーを記述する予定がない場合は、ユーザライブラリディレクトリを作成する必要はありません。

Cisco IOS Release 12.3(14)T以降のリリースでは、ソフトウェアは Tool Command Language (Tcl) のスクリプト言語を使用して作成されたポリシーファイルをサポートします。Tclは、EEMをネットワークデバイスにインストールする際に、Cisco IOS ソフトウェアイメージで提供されます。.tcl 拡張子を持つファイルは、EEMポリシー、Tclライブラリファイル、または「tclindex」という名前の特別なTclライブラリインデックスファイルのいずれかです。tclindexファイルには、ユーザ関数名のリストと、ユーザ関数を含むライブラリファイルが含まれます。EEMは、Tclがtclindexファイルの処理を開始すると、ユーザライブラリディレクトリを検索します。

EEMに対して識別する前にユーザライブラリディレクトリを作成するには、特権EXECモードで **mkdir** コマンドを使用します。ユーザライブラリディレクトリを作成したら、**copy** コマンドを使用して.tclライブラリファイルをユーザライブラリディレクトリにコピーできます。

ユーザポリシーディレクトリは、ユーザ定義のポリシーファイルを保存するために必要です。EEMポリシーを記述する予定がない場合は、ユーザポリシーディレクトリを作成する必要はありません。EEMは、**event manager policy policy-filename type user** コマンドが入力されたときにユーザポリシーディレクトリを検索します。

EEMに対して識別する前にユーザポリシーディレクトリを作成するには、特権EXECモードで **mkdir** コマンドを使用します。ユーザポリシーディレクトリを作成したら、**copy** コマンドを使用して、ポリシーファイルをユーザポリシーディレクトリにコピーできます。

例

次に、ユーザライブラリファイルを保存するディレクトリとして disk0:/user_library を指定する例を示します。

```
Router(config)# event manager directory user library disk0:/user_library
```

関連コマンド

コマンド	説明
copy	コピー元からコピー先に任意のファイルをコピーします。
event manager policy	EEMポリシーをEEMに登録します。

コマンド	説明
mkdir	クラス C フラッシュ ファイル システムに新しいディレクトリを作成します。

event manager environment

Embedded Event Manager (EEM) 環境変数を設定するには、グローバル コンフィギュレーション モードで **event manager environment** コマンドを使用します。EEM 環境変数をディセーブルにするには、このコマンドの **no** 形式を使用します。

event manager environment *variable-name string*

no event manager environment *variable-name*

構文の説明

<i>variable-name</i>	EEM 環境変数に割り当てる名前。
<i>string</i>	環境変数 <i>variable-name</i> に配置する一連の文字列 (埋め込みスペースを含む)。

コマンド デフォルト

EEM 環境変数は設定されていません。

コマンド モード

グローバル コンフィギュレーション

コマンド履歴

リリース	変更内容
12.2(25)S	このコマンドが導入されました。
12.3(14)T	このコマンドが、Cisco IOS Release 12.3(14)T に統合されました。
12.2(28)SB	このコマンドが、Cisco IOS Release 12.2(28)SB に統合されました。
12.2(18)SXF4	このコマンドが Cisco IOS Release 12.2(18)SXF4 に統合され、ソフトウェア モジュラリティ イメージだけをサポートするようになりました。
12.2(33)SRA	このコマンドが、Cisco IOS Release 12.2(33)SRA に統合されました。
12.2(18)SXF5	このコマンドが、Cisco IOS Release 12.2(18)SXF5 に統合されました。
12.2SX	このコマンドは、Cisco IOS Release 12.2SX トレインでサポートされません。このトレインの特定の 12.2SX リリースにおけるサポートは、フィーチャセット、プラットフォーム、およびプラットフォームハードウェアによって異なります。

使用上のガイドライン 通例として、シスコで定義されているすべての環境変数の名前は、他の変数と区別するためにアンダースコア文字で始まります（_show_cmd など）。

string 引数で埋め込みスペースをサポートするために、このコマンドでは *variable-name* 引数の後ろから行の最後までを、*string* 引数の一部と解釈します。

設定後、EEM 環境変数の名前と値を表示するには、**show event manager environment** コマンドを使用します。

例

次に、**event manager environment** コマンドが EEM 環境変数のセットを定義する例を示します。

```
Router(config)# event manager environment _cron_entry 0-59/2 0-23/1 * * 0-7
Router(config)# event manager environment _show_cmd show version
```

関連コマンド

コマンド	説明
show event manager environment	すべての EEM 環境変数の名前および値を表示します。

event manager history size

Embedded Event Manager (EEM) 履歴テーブルのサイズを変更するには、グローバルコンフィギュレーションモードで **event manager history size** コマンドを使用します。デフォルトの履歴テーブルのサイズに戻すには、このコマンドの **no** 形式を使用します。

event manager history size {events| traps} [size]

no event manager history size {events| traps}

構文の説明

events	EEM イベント履歴テーブルのサイズを変更します。
traps	EEM 簡易ネットワーク管理プロトコル (SNMP) トラップ履歴テーブルのサイズを変更します。
<i>size</i>	(任意) 履歴テーブルのエントリ数を指定する 1 ~ 50 の範囲の整数。デフォルトは 50 です。

コマンド デフォルト

履歴テーブルのサイズは 50 エントリです。

コマンド モード

グローバル コンフィギュレーション

コマンド履歴

リリース	変更内容
12.2(25)S	このコマンドが導入されました。
12.3(14)T	このコマンドが、Cisco IOS Release 12.3(14)T に統合されました。
12.2(28)SB	このコマンドが、Cisco IOS Release 12.2(28)SB に統合されました。
12.2(18)SXF4	このコマンドが Cisco IOS Release 12.2(18)SXF4 に統合され、ソフトウェア モジュールリティ イメージだけをサポートするようになりました。
12.2(33)SRA	このコマンドが、Cisco IOS Release 12.2(33)SRA に統合されました。
12.2(18)SXF5	このコマンドが、Cisco IOS Release 12.2(18)SXF5 に統合されました。

リリース	変更内容
12.2SX	このコマンドは、Cisco IOS Release 12.2SX トレインでサポートされます。このトレインの特定の 12.2SX リリースにおけるサポートは、フィーチャセット、プラットフォーム、およびプラットフォームハードウェアによって異なります。

例

次に、**event manager history size** コマンドが SNMP トラップ履歴テーブルのサイズを 30 エントリに変更する例を示します。

```
Router(config)# event manager history size traps 30
```

関連コマンド

コマンド	説明
show event manager history events	トリガーされた EEM イベントを表示します。
show event manager history traps	送信された EEM SNMP トラップを表示します。

event manager run

登録済みの Embedded Event Manager (EEM) ポリシーを手動で実行するには、特権 EXEC モードで **event manager run** コマンドを使用します。

```
event manager run policy-filename [[ parameter1 ] [ parameter2 ]... [ parameter15 ]]
```

構文の説明

<i>policy-filename</i>	ポリシー ファイルの名前。
<i>parameter</i>	(任意) スクリプトに渡されるパラメータ。最大 15 個のパラメータを指定できます。パラメータは、英数字のストリングである必要があります。引用符、埋め込みスペースや特殊文字は使用できません。

コマンド デフォルト

登録済みの EEM ポリシーは実行されません。

コマンド モード

特権 EXEC (#)

コマンド履歴

リリース	変更内容
12.3(14)T	このコマンドが導入されました。
12.2(28)SB	このコマンドが、Cisco IOS Release 12.2(28)SB に統合されました。
12.2(18)SXF4	このコマンドが Cisco IOS Release 12.2(18)SXF4 に統合され、ソフトウェア モジュラリティ イメージだけをサポートするようになりました。
12.2(33)SRA	このコマンドが、Cisco IOS Release 12.2(33)SRA に統合されました。
12.2(18)SXF5	このコマンドが、Cisco IOS Release 12.2(18)SXF5 に統合されました。
12.2SX	このコマンドは、Cisco IOS Release 12.2SX トレインでサポートされました。このトレインの特定の 12.2SX リリースにおけるサポートは、フィーチャセット、プラットフォーム、およびプラットフォーム ハードウェアによって異なります。
12.4(20)T	parameter 引数が追加されました。最大 15 のパラメータ値を指定でき、引数はレジストリ コールで指定できます。

使用上のガイドライン また、このコマンドを使用することでパラメータをイベントポリシーで使用し、レジストリコールで引数を指定できます。

EEMは、通常、ポリシーそのものに含まれるイベント仕様に基づいてポリシーをスケジューリングし、実行します。 **event manager run** コマンドを使用することで、ポリシーを手動で実行できます。手動でポリシーを実行するには、最初に **event none** コマンドを設定する必要があります。None イベントディテクタは、None イベントをパブリッシュする場合は引数を含みます。このコマンドには、**no** 形式はありません。

例 次に、policy-manual.tcl という名前の EEM ポリシーを手動で実行する例を示します。

```
Router# event manager run policy-manual.tcl
```

各パラメータは組み込みの総数 (`$_none_argc`) と、続けて組み込みのリスト (`$_none_arg1`、`$_none_arg2`、および `$_none_arg3`) で構成されます。次に、アプレットおよび Tool Tcl スクリプトの例を示します。

例

```
event manager applet none_parameter_test
 event none
 action 1 syslog msg "Number of Arguments is $_none_argc"
 action 2 syslog msg "Argument 1 is $_none_arg1"
 action 3 syslog msg "Argument 2 is $_none_arg2"
 action 4 syslog msg "Argument 3 is $_none_arg3"
end
```

```
Router# event manager run none_parameter_test 11 22 33
01:26:03: %HA_EM-6-LOG: none_parameter_test: Number of Arguments is 3
01:26:03: %HA_EM-6-LOG: none_parameter_test: Argument 1 is 11
01:26:03: %HA_EM-6-LOG: none_parameter_test: Argument 2 is 22
01:26:03: %HA_EM-6-LOG: none_parameter_test: Argument 3 is 33
```

ポリシーの場合、**event_reqinfo** はオプションパラメータを文字列で返し、その文字列はその後ポリシーによって処理されます。

例

```
none_paramter_test.tcl
::cisco::eem::event_register_none
namespace import ::cisco::eem::*
namespace import ::cisco::lib::*
# query the event info
array set arr_einfo [event_reqinfo]
if {$_cerrno != 0} {
    set result [format "component=%s; subsys err=%s; posix err=%s;\n%s" \
        $_cerr_sub_num $_cerr_sub_err $_cerr_posix_err $_cerr_str]
    error $result
}
action_syslog priority info msg "Number of Arguments is $arr_einfo(argc)"
if {$_cerrno != 0} {
    set result [format \
        "component=%s; subsys err=%s; posix err=%s;\n%s" \
        $_cerr_sub_num $_cerr_sub_err $_cerr_posix_err $_cerr_str]
    error $result
}
```

```

action_syslog priority info msg "Argument 1 is $arr_einfo(arg1)"
if {$_cerrno != 0} {
  set result [format \
    "component=%s; subsys err=%s; posix err=%s;\n%s" \
    $_cerr_sub_num $_cerr_sub_err $_cerr_posix_err $_cerr_str]
  error $result
}
action_syslog priority info msg "Argument 2 is $arr_einfo(arg2)"
if {$_cerrno != 0} {
  set result [format \
    "component=%s; subsys err=%s; posix err=%s;\n%s" \
    $_cerr_sub_num $_cerr_sub_err $_cerr_posix_err $_cerr_str]
  error $result
}
action_syslog priority info msg "Argument 3 is $arr_einfo(arg3)"
if {$_cerrno != 0} {
  set result [format \
    "component=%s; subsys err=%s; posix err=%s;\n%s" \
    $_cerr_sub_num $_cerr_sub_err $_cerr_posix_err $_cerr_str]
  error $result
}
}
jubjub#event manager run none_parameter_test.tcl 1 2 3
01:26:03: %HA_EM-6-LOG: tmpsys:/eem_policy/none_parameter_test.tcl: Number of Arguments is
3
01:26:03: %HA_EM-6-LOG: tmpsys:/eem_policy/none_parameter_test.tcl: Argument 1 is 1
01:26:03: %HA_EM-6-LOG: tmpsys:/eem_policy/none_parameter_test.tcl: Argument 2 is 2
01:26:03: %HA_EM-6-LOG: tmpsys:/eem_policy/none_parameter_test.tcl: Argument 3 is 3

```

関連コマンド

コマンド	説明
event manager applet	EEM に EEM 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。
event manager policy	EEM ポリシーを EEM に登録します。
event none	EEM に登録して手動で起動される EEM ポリ シーを指定します。
show event manager policy registered	登録済みの EEM ポリシーを表示します。

event manager scheduler clear

実行中または実行を保留中の Embedded Event Manager (EEM) ポリシーをクリアするには、特権 EXEC モードで **event manager scheduler clear** コマンドを使用します。

```
event manager scheduler clear {all| policy job-id| queue-type {applet| call-home| axp| script} [class
class-options]} [processor {rp_primary| rp_standby}]
```

構文の説明

all	現在実行中または実行保留中のキューにあるすべてのポリシーをクリアします。
policy	ジョブ ID で指定された EEM ポリシーをクリアします。
<i>job-id</i>	キューにある各ポリシーを識別する 1 ～ 4294967295 の範囲の番号。
queue-type	EEM ポリシーのキュータイプをクリアします。
applet	EEM キュータイプのアプレットを指定します。
call-home	EEM キュータイプ の Call Home ポリシーを指定します。
axp	EEM キュータイプ のアプリケーション拡張プラットフォームを指定します。
script	EEM 実行スレッドで Tcl スクリプトを実行するように指定します。
class	指定したクラスの EEM ポリシーをクリアします。

<i>class-options</i>	<p>EEM ポリシー クラスを指定します。次のうちの 1 つまたはすべてを指定できます。</p> <ul style="list-style-type: none"> • class-letter : 各ポリシー クラスを識別する A ~ Z の文字。 <i>class-letter</i> の複数のインスタンスを指定できます。 • default : デフォルト クラス。 クラスなしで登録された EEM ポリシーは、デフォルト クラスに割り当てられます。 • range class-range : EEM ポリシー クラスを範囲で指定します。 A ~ Z. の範囲を指定できます。 range class-range の複数のインスタンスを指定できます。
processor	(任意) プロセッサがコマンドを実行するように指定します。
rp_primary	(任意) デフォルトの RP を示します。 ポリシーは、イベント相関によりポリシーがスケジュールされる場合は、プライマリ RP で実行されます。
rp_standby	(任意) スタンバイ RP を示します。 ポリシーは、イベント相関によりポリシーがスケジュールされる場合は、スタンバイ RP で実行されます。

コマンド モード

特権 EXEC (#)

コマンド履歴

リリース	変更内容
12.4(20)T	このコマンドが導入されました。
12.4(22)T	queue-type および processor キーワードと、 <i>class-letter</i> および <i>class-range</i> 引数が追加されました。
12.2(33)SRE	このコマンドが、 Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン **show event manager policy pending** コマンドを使用して、サーバ実行キューで保留されているポリシーを表示します。

show event manager policy active コマンドを使用して、実行中のポリシーを表示します。

event manager scheduler clear コマンドを使用して、サーバのポリシーまたはポリシー キューをクリアします。

class キーワードの場合、*class-letter*、**default**、または **range class-range** のいずれかのオプションを指定する必要があります。同じ CLI 文でこれらのオプションをすべて指定できます。

例 次に、実行保留中の EEM ポリシーをクリアする例を示します。 **show** コマンドは、ポリシーをクリアする前後の出力例を表示します。

```
Router# show event manager policy pending
no. job id status time of event          event type      name
1   1      pend  Thu Sep 7 02:54:04 2006  syslog         applet: one
2   2      pend  Thu Sep 7 02:54:04 2006  syslog         applet: two
3   3      pend  Thu Sep 7 02:54:04 2006  syslog         applet: three
```

```
Router# event manager scheduler clear policy 2
Router# show event manager policy pending

no. job id status time of event          event type      name
1   1      pend  Thu Sep 7 02:54:04 2006  syslog         applet: one
3   3      pend  Thu Sep 7 02:54:04 2006  syslog         applet: three
```

関連コマンド

コマンド	説明
event manager policy	EEM ポリシーを EEM に登録します。
show event manager policy pending	実行保留中の EEM ポリシーを表示します。

event manager scheduler hold

EEM スケジューラのスケジュールされた Embedded Event Manager (EEM) ポリシー イベントまたは イベント キューをホールドするには、特権 EXEC モードで **event manager scheduler hold** コマンドを使用します。ポリシー イベントまたは イベント キューを再開するには、**event manager scheduler release** コマンドを使用します。

```
event manager scheduler hold {all| policy job-id| queue-type {applet| call-home| axp| script} [class class-options]} [processor {rp_primary| rp_standby}]
```

構文の説明

all	EEM スケジューラのすべての EEM ポリシー イベントまたは イベント キューをホールドします。
policy	ジョブ ID で指定された EEM スケジューラの EEM ポリシー イベントまたは イベント キューをホールドします。
<i>job-id</i>	キューにある各ポリシーを識別する 1 ~ 4294967295 の範囲の番号。
queue-type	EEM キュータイプに基づいて EEM ポリシー イベントまたは イベント キューをホールドします。
applet	EEM キュータイプの アプレットを指定します。
call-home	EEM キュータイプの Call Home ポリシーを指定します。
axp	EEM キュータイプの アプリケーション拡張プラットフォームを指定します。
script	EEM 実行スレッドで Tcl スクリプトを実行するように指定します。
class	EEM ポリシーのクラスを指定します。

<i>class-options</i>	EEM ポリシー クラスを指定します。次のうちの 1 つまたはすべてを指定できます。 <ul style="list-style-type: none"> • class-letter : 各ポリシー クラスを識別する A ~ Z の文字。 <i>class-letter</i> の複数のインスタンスを指定できます。 • default : デフォルトクラス。クラスなしで登録された EEM ポリシーは、デフォルトクラスに割り当てられます。 • range class-range : EEM ポリシー クラスを範囲で指定します。A ~ Z の範囲を指定できます。 range class-range の複数のインスタンスを指定できます。
processor	(任意) プロセッサがコマンドを実行するように指定します。
rp_primary	(任意) デフォルトの RP を示します。ポリシーは、イベント相関によりポリシーがスケジュールされる場合は、プライマリ RP で実行されます。
rp_standby	(任意) スタンバイ RP を示します。ポリシーは、イベント相関によりポリシーがスケジュールされる場合は、スタンバイ RP で実行されます。

コマンド モード

特権 EXEC (#)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン

show event manager policy pending コマンドを使用して、サーバ実行キューで保留されているポリシーを表示します。

event manager scheduler hold コマンドを使用して、サーバのポリシーまたはポリシー キューをホールドします。

class キーワードの場合、*class-letter*、**default**、または **range class-range** のいずれかのオプションを指定する必要があります。同じ CLI 文でこれらのオプションをすべて指定できます。

例

次に、EEM スケジューラのスケジュールされたポリシーイベントをホールドする例を示します。**show** コマンドは、ポリシー イベントをホールドする前後の出力例を表示します。

```
Router# show event manager policy pending
no. job id status time of event      event type      name
1 1      pend  Thu Sep 7 02:54:04 2006  syslog         applet: one
2 2      pend  Thu Sep 7 02:54:04 2006  syslog         applet: two
3 3      pend  Thu Sep 7 02:54:04 2006  syslog         applet: three
Router# event manager scheduler hold policy 2
Router# show event manager policy pending

no. job id status time of event      event type      name
1 1      pend  Thu Sep 7 02:54:04 2006  syslog         applet: one
2 2      held  Thu Sep 7 02:54:04 2006  syslog         applet: two
3 3      pend  Thu Sep 7 02:54:04 2006  syslog         applet: three
```

関連コマンド

コマンド	説明
event manager policy	EEM ポリシーを EEM に登録します。
event manager scheduler release	ポリシー イベントまたはイベント キューを再開します。
show event manager policy pending	実行保留中の EEM ポリシーを表示します。

event manager scheduler modify

Embedded Event Manager (EEM) ポリシーのスケジューリング パラメータを変更するには、特権 EXEC モードで **event manager scheduler modify** コマンドを使用します。

```
event manager scheduler modify {all| policy job-id queue-type {applet| call-home| axp| script}} {class
class-options [queue-priority {high| last| low| normal}]} [queue-priority {high| last| low| normal}] [class
class-options]} [processor {rp_primary| rp_standby}]
```

構文の説明

all	現在実行中または実行保留中のキューにあるすべてのポリシーを変更します。
policy	ジョブ ID で指定された EEM ポリシーを変更します。
<i>job-id</i>	キューにある各ポリシーを識別する 1 ～ 4294967295 の範囲の番号。
queue-type	EEM ポリシーのキュー タイプを変更します。
applet	EEM キュータイプのアプレットを指定します。
call-home	EEM キュータイプ の Call Home ポリシーを指定します。
axp	EEM キュータイプ のアプリケーション拡張プラットフォームを指定します。
script	EEM 実行スレッドで Tcl スクリプトを実行するように指定します。
class	EEM ポリシーのクラスを変更します。
<i>class-options</i>	EEM ポリシー クラスを指定します。次のうちの 1 つまたはすべてを指定できます。 <ul style="list-style-type: none"> • class-letter : 各ポリシー クラスを識別する A ～ Z の文字。 <i>class-letter</i> の複数のインスタンスを指定できます。 • default : デフォルトクラス。 クラスなしで登録された EEM ポリシーは、デフォルトクラスに割り当てられます。

queue-priority	(任意) EEM ポリシーのキューイング順序のプライオリティを変更します。
high	(任意) キューのプライオリティを高に指定します。
last	(任意) キューのプライオリティを最後に指定します。
low	(任意) キューのプライオリティを低に指定します。
normal	(任意) キューのプライオリティを通常に指定します。
processor	(任意) プロセッサがコマンドを実行するように指定します。
rp_primary	(任意) デフォルトの RP を示します。ポリシーは、イベント関連によりポリシーがスケジュールされる場合は、プライマリ RP で実行されます。
rp_standby	(任意) スタンバイ RP を示します。ポリシーは、イベント関連によりポリシーがスケジュールされる場合は、スタンバイ RP で実行されます。

コマンドモード

特権 EXEC (#)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン

show event manager policy pending コマンドを使用して、サーバ実行キューで保留されているポリシーを表示します。

event manager scheduler modify コマンドを使用して、ポリシーのスケジューリングパラメータを変更します。

class キーワードの場合、*class-letter* または **default** のいずれかのオプションを指定する必要があります。同じ CLI 文で両方のオプションを指定できます。

例

次に、EEM ポリシーのスケジューリングパラメータを変更する例を示します。**show** コマンドは、スケジューリングパラメータを変更する前後の出力例を表示します。

```
Router# show event manager policy pending
no. class status time of event event type name
1 default pend Thu Sep 7 02:54:04 2006 syslog applet: one
2 default pend Thu Sep 7 02:54:04 2006 syslog applet: two
3 B pend Thu Sep 7 02:54:04 2006 syslog applet: three

Router# event manager scheduler modify all class A
Router# show event manager policy pending
no. class status time of event event type name
1 A pend Thu Sep 7 02:54:04 2006 syslog applet: one
2 A pend Thu Sep 7 02:54:04 2006 syslog applet: two
3 A pend Thu Sep 7 02:54:04 2006 syslog applet: three
```

関連コマンド

コマンド	説明
event manager policy	EEM ポリシーを EEM に登録します。
show event manager policy pending	実行保留中の EEM ポリシーを表示します。

event manager scheduler release

Embedded Event Manager (EEM) ポリシーの実行を再開するには、特権 EXEC モードで **event manager scheduler release** コマンドを使用します。

```
event manager scheduler release {all| policy policy-id| queue-type {applet| call-home| axp| script} [class class-options]} [processor {rp_primary| rp_standby}]
```

構文の説明

all	すべての EEM ポリシーの実行を再開します。
policy	ポリシー ID で指定された EEM ポリシーを再開します。
<i>policy-id</i>	キューにある各ポリシーを識別する 1 ～ 4294967295 の範囲の番号。
queue-type	EEM キュー タイプに基づいてポリシーの実行を再開します。
applet	EEM アプレット ポリシーを指定します。
call-home	Call Home ポリシーを指定します。
axp	Application Extension Platform (AXP) ポリシーを指定します。
script	EEM スクリプト ポリシーを指定します。
class	EEM ポリシー クラスを指定します。
<i>class-options</i>	EEM ポリシー クラス。次のうちの 1 つまたはすべてを指定できます。 <ul style="list-style-type: none"> • class-letter : 各ポリシー クラスを識別する A ～ Z の文字。 class-letter の複数のインスタンスを指定できます。 • default : デフォルト クラスに登録されたポリシーを指定します。 • range class-letter-range : EEM ポリシー クラスを範囲で指定します。 range class-letter-range の複数のインスタンスを指定できます。 class-letter-range で使用する文字は大文字にする必要があります。

processor	プロセッサがコマンドを実行するように指定します。デフォルト値はプライマリ RP です。
rp_primary	プライマリ RP を示します。
rp_standby	スタンバイ RP を示します。

コマンド デフォルト ディセーブル

コマンド モード 特権 EXEC (#)

コマンド履歴	リリース	変更内容
	12.4(22)T	このコマンドが導入されました。

使用上のガイドライン **event manager scheduler hold** コマンドを使用してホールドした EEM ポリシーを解放するには、**event manager scheduler release** コマンドを使用します。

class-letter、**default**、および **range class-letter-range** のいずれかのオプションを指定する必要があります。同じ CLI 文でこれらのオプションをすべて指定できます。

例 次に、すべての EEM ポリシーの実行を再開する例を示します。

```
Router# event manager scheduler release all
```

次に、クラス A ~ E のポリシーの実行を再開する例を示します。

```
Router# event manager scheduler release queue-type script class range A-E
```

関連コマンド

コマンド	説明
event manager scheduler hold	EEM ポリシーのスケジューリング実行をホールドします。

event manager scheduler suspend

Embedded Event Manager (EEM) ポリシーのスケジューリング実行をただちに停止するには、グローバル コンフィギュレーション モードで **event manager scheduler suspend** コマンドを使用します。EEM ポリシーのスケジューリングを再開するには、このコマンドの **no** 形式を使用します。

event manager scheduler suspend

no event manager scheduler suspend

構文の説明

このコマンドには引数またはキーワードはありません。

コマンド デフォルト

ポリシー スケジューリングはアクティブです。

コマンド モード

グローバル コンフィギュレーション

コマンド履歴

リリース	変更内容
12.2(25)S	このコマンドが導入されました。
12.3(14)T	このコマンドが、Cisco IOS Release 12.3(14)T に統合されました。
12.2(28)SB	このコマンドが、Cisco IOS Release 12.2(28)SB に統合されました。
12.2(18)SXF4	このコマンドが Cisco IOS Release 12.2(18)SXF4 に統合され、ソフトウェア モジュールリティ イメージだけをサポートするようになりました。
12.2(33)SRA	このコマンドが、Cisco IOS Release 12.2(33)SRA に統合されました。
12.2(18)SXF5	このコマンドが、Cisco IOS Release 12.2(18)SXF5 に統合されました。
12.2SX	このコマンドは、Cisco IOS Release 12.2SX トレインでサポートされません。このトレインの特定の 12.2SX リリースにおけるサポートは、フィーチャセット、プラットフォーム、およびプラットフォームハードウェアによって異なります。

使用上のガイドライン

event manager scheduler suspend コマンドを使用して、すべてのポリシーのスケジューリング要求を停止します。このコマンドの **no** 形式を入力するまで、スケジューリングは行われません。こ

のコマンドの **no** 形式を使用すると、ポリシーのスケジューリングが再開され、保留中のポリシーが実行されます。

ポリシーを1つずつ登録解除せずに、ポリシーの実行をただちに停止する必要がある状況として、次のものが挙げられます。

- セキュリティ上の理由：システムのセキュリティが侵害されていると思われる場合。
- パフォーマンス上の理由：他の機能に多くのCPUサイクルを割り当てるため、ポリシーの実行を一時的に停止する場合。

例

次の **event manager scheduler suspend** コマンドの例では、ポリシーのスケジュールがディセーブルになります。

```
Router(config)# event manager scheduler suspend
May 19 14:31:22.439: fm_server[12330]: %HA_EM-6-FMS_POLICY_EXEC: fh_io_msg: Policy execution
has been suspended
```

次の **event manager scheduler suspend** コマンドの例では、ポリシーのスケジュールがイネーブルになります。

```
Router(config)# no event manager scheduler suspend
May 19 14:31:40.449: fm_server[12330]: %HA_EM-6-FMS_POLICY_EXEC: fh_io_msg: Policy execution
has been resumed
```

関連コマンド

コマンド	説明
event manager policy	EEM ポリシーを EEM に登録します。



event mat から R までのコマンド

- [event mat から R までのコマンド](#), 147 ページ

event mat から R までのコマンド

event mat

MACアドレスがMACアドレステーブルで学習されたときにイベントをパブリッシュするには、アプレットコンフィギュレーションモードで **event mat** コマンドを使用します。イベントのパブリッシュをディセーブルにするには、このコマンドの **no** 形式を使用します。

```
event [tag event-tag] mat {interface {type number| regexp interface-name} [mac-address mac-address]
mac-address mac-address [interface {type number| regexp interface-name}]} [maxrun maxruntime-number]
[hold-down seconds] [type {add| delete}]
```

```
no event mat
```

構文の説明

tag	(任意) trigger コマンドで使用できる event-tag 引数を使用してタグを指定し、アプレットの複数のイベント文をサポートします。
<i>event-tag</i>	(任意) タグを識別するストリング。
interface	インターフェイスを指定します。
<i>type number</i>	インターフェイスのタイプと番号。
regexp interface-name	インターフェイス名と照合する正規表現パターンを指定します。
mac-address	MACアドレスを指定します。
<i>mac-address</i>	MACアドレス。
maxrun	(任意) アプレットの最大実行時間を指定します。maxrun キーワードが指定されている場合、maxruntime-number 値を指定する必要があります。maxrun キーワードが指定されていない場合、デフォルトのアプレットランタイムは20秒です。
<i>maxruntime-number</i>	(任意) ssssss[.mmm]形式で指定される秒数。sssssss は、0～31536000の秒数を表す整数で、mmm は0～999のミリ秒数を表す整数である必要があります。
hold-down	(任意) イベント処理を遅延する時間を指定します。

<i>seconds</i>	(任意) 秒数およびオプションのミリ秒数を ssssssssss [.mmm] 形式で表した数字。秒数の範囲は 1 ~ 4294967295 です。ミリ秒の範囲は 0 ~ 999 です。ミリ秒だけを使用する場合は、0.mmm の形式でミリ秒を指定します。
type	(任意) MAC アドレス テーブルのイベントをモニタします。次のオプションのいずれかのオプションを指定する必要があります。 <ul style="list-style-type: none"> • add : MAC アドレス テーブルの追加イベントだけをモニタします。 • delete : MAC アドレス テーブルの削除イベントだけをモニタします。

コマンド デフォルト デフォルトでは、イベントはパブリッシュされません。

コマンド モード アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.2(52)SE	このコマンドが導入されました。
12.2(54)SG	このコマンドが、Cisco IOS Release 12.2(54)SG に統合されました。

使用上のガイドライン インターフェイスまたは MAC アドレスを指定する必要があります。そのいずれかが指定されている場合、残り 1 つは任意です。すべてのキーワードは任意の組み合わせで使用できます。

例 次に、MAC アドレスが MAC アドレス テーブルで学習されたときにイベントをパブリッシュする例を示します。

```
Router(config)# event manager applet mat
Router(config-applet)# event mat interface fastethernet0 hold-down 34 type delete
Router(config-applet)#
```

関連コマンド

コマンド	説明
event manager applet	Embedded Event Manager にイベント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。

event neighbor-discovery

Cisco Discovery Protocol (CDP) またはリンク層検出プロトコル (LLDP) のキャッシュ エントリが変更されるか、Embedded Event Manager (EEM) アプレットでインターフェイス リンク ステータスが変更されたときにイベントをパブリッシュするには、アプレット コンフィギュレーション モードで **event neighbor-discovery** コマンドを使用します。 イベントをパブリッシュするアクションをディセーブルにするには、このコマンドの **no** 形式を使用します。

event [*tag event-tag*] **neighbor-discovery interface** {*type number*|**regex** *interface-name*} [**maxrun** *maxruntime-number*] *event-to-monitor*

no event neighbor-discovery

構文の説明

tag	(任意) trigger コマンドで使用できる event-tag 引数を使用してタグを指定し、アプレットの複数のイベント文をサポートします。
<i>event-tag</i>	(任意) タグを識別するストリング。
interface	インターフェイスを指定します。
<i>type number</i>	インターフェイスのタイプと番号。
regex <i>interface-name</i>	インターフェイス名と照合する正規表現パターンを指定します。
maxrun	(任意) アプレットの最大実行時間を指定します。 maxrun キーワードが指定されている場合、 maxruntime-number 値を指定する必要があります。 maxrun キーワードが指定されていない場合、デフォルトのアプレット ランタイムは 20 秒です。
<i>maxruntime-number</i>	(任意) sssssss[mmm] 形式で指定される秒数。 sssssss は、0 ~ 31536000 の秒数を表す整数で、mmm は 0 ~ 999 のミリ秒数を表す整数である必要があります。

<i>event-to-monitor</i>	
-------------------------	--

インターフェイス上でモニタするイベントを指定します。次のいずれかの値を指定できます。複数の値を指定できます。

- **cdp** : CDP のマッピング イベント発生時にイベントをトリガーします。次のオプションのいずれかのオプションを指定する必要があります。
 - **add** : 新しい CDP キャッシュ エントリが CDP テーブルに作成された場合にイベントをトリガーします。
 - **all** : CDP キャッシュ エントリが CDP キャッシュ テーブルに追加された場合、または削除された場合、および CDP キャッシュ エントリをアップデートするためにリモート CDP デバイスがキープアライブを送信した場合にイベントをトリガーします。
 - **delete** : CDP キャッシュ エントリが CDP テーブルから削除された場合だけイベントをトリガーします。
 - **update** : CDP キャッシュ エントリが CDP キャッシュ テーブルに追加された場合、または CDP キャッシュ エントリをアップデートするためにリモート CDP デバイスが CDP キープアライブを送信した場合にイベントをトリガーします。
- **lldp** : LLDP のマッピング イベント発生時にイベントをトリガーします。次のオプションのいずれかのオプションを指定する必要があります。
 - **add** : 新しい CDP キャッシュ エントリが CDP テーブルに作成された場合にイベントをトリガーします。
 - **all** : CDP キャッシュ エントリが CDP キャッシュ テーブルに追加された場合、または削除された場合、および CDP キャッシュ エントリをアップデートするためにリモート CDP デバイスがキープアライブを送信した場

合にイベントをトリガーします。

- **delete** : CDP キャッシュ エントリが CDP テーブルから削除された場合だけイベントをトリガーします。
- **update** : CDP キャッシュ エントリが CDP キャッシュ テーブルに追加された場合、または CDP キャッシュ エントリをアップデートするためにリモート CDP デバイスが CDP キープアライブを送信した場合にイベントをトリガーします。
- **line-event** : インターフェイス回線プロトコルのステータスが変った場合にイベントをトリガーします。
- **link-event** : インターフェイス リンクのステータスが変った場合にイベントをトリガーします。 次のオプションのいずれかのオプションを指定する必要があります。
 - **admindown** : リンクの admin-down イベントをモニタします。
 - **all** : すべてのリンク イベントをモニタします。
 - **deleted** : リンク削除イベントをモニタします。
 - **down** : リンク ダウンイベントをモニタします。
 - **goingdown** : リンクの going-down イベントをモニタします。
 - **init** : リンクの初期化イベントをモニタします。
 - **reset** : リンクのリセット イベントをモニタします。
 - **testing** : リンクのテスト イベントをモニタします。
 - **up** : リンクのアップ イベントをモニタします。

コマンド デフォルト デフォルトでは、イベントはパブリッシュされません。

コマンド モード アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.2(52)SE	このコマンドが導入されました。
12.2(54)SG	このコマンドが、Cisco IOS Release 12.2(54)SG に統合されました。

使用上のガイドライン イベントの指定が受け入れられるようにするには、インターフェイスと、`cdp`、`lldp`、`link-event`、`line-event` から 1 つ以上を指定する必要があります。 `interface` および `maxrun` キーワードと、`event-trigger-criteria` 引数は任意の順序で使用できます。

例 次に、CDP キャッシュ エントリが変更されたときにイベントをパブリッシュする例を示します。

```
Router(config)# event manager applet discovery
Router(config-applet)# event neighbor-discovery interface fastethernet0 cdp all
Router(config-applet)#
```

関連コマンド

コマンド	説明
event manager applet	Embedded Event Manager にイベント 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。

event nf

Embedded Event Manager (EEM) のアプレットに対して NetFlow 操作がトリガーされたときにイベントをパブリッシュするには、アプレット コンフィギュレーション モードで **event nf** コマンドを使用します。NetFlow 操作がトリガーされたときにイベントをパブリッシュするアクションをディセーブルにするには、このコマンドの **no** 形式を使用します。

```
event [tag event-tag] nf monitor-name name event-type {create| delete| update} [exit-event-type] {create| delete| update} subevent field field-type entry-value value-string [exit-value value-string] entry-op operator-value [exit-op operator-value] [rate-interval seconds] [exit-rate-interval seconds] [maxrun maxruntime-number]
```

```
no event [tag event-tag] nf
```

構文の説明

tag	(任意) trigger コマンドで使用できる <i>event-tag</i> 引数を使用してタグを指定し、アプレットの複数のイベント文をサポートします。
<i>event-tag</i>	(任意) タグを識別するストリング。
monitor-name name	NetFlow モニタの名前を指定します。
event-type	モニタするイベントのタイプを指定します (キャッシュまたはフィールド)。
create	NetFlow イベントを作成します。
delete	NetFlow イベントを削除します。
update	NetFlow イベントを更新します。
exit-event-type	監視のためにイベントが再準備されるイベントタイプ (create、delete、update)。
<i>subevent</i>	監視するイベントとその属性を指定します。有効な値は、 event1 、 event2 、 event3 、 event4 です。 (注) サブイベント キーワードは、単独でも、一緒でも、それぞれの任意の組み合わせでも使用できますが、各キーワードは 1 回のみ使用できます。

field <i>field-type</i>	<p>監視されるキャッシュまたはフィールド属性を指定します。次の属性の1つを指定できます。</p> <ul style="list-style-type: none"> • counter {bytes packets} : counter フィールドを指定します。 • datalink {dot1q mac} : datalink (layer2) フィールドを指定します。 • flow {direction sampler} : flow 識別フィールドを指定します。 • interface {input output} : interface フィールドを指定します。 • ipv4 <i>field-type</i> : IPv4 フィールドを指定します。 • ipv6 <i>field-type</i> : IPv6 フィールド • routing <i>routing-attribute</i> : ルーティング属性を指定します。 • timestamp sysuptime {first last} : timestamp フィールドを指定します。 • transport <i>field-type</i> : Transport layer フィールドを指定します。 <p>詳細については、疑問符 (?) オンラインヘルプ機能を使用します。</p>
entry-value <i>value-string</i>	比較する開始値を指定します。
exit-value <i>string</i>	(任意) イベントが再度モニタされるように設定される値。
rate-interval <i>sec</i>	レートインターバル値を秒単位で指定します。有効範囲は 1 ~ 4294967295 です。
exit-rate-interval <i>sec</i>	(任意) キャッシュ レートとキャッシュ エントリのインターバル値を指定します。有効値は 0 ~ 4294967295 です。
entry-op	収集された使用率サンプルと指定した値との比較で使用する演算子を指定します。有効な値は次のとおりです。

<i>operator-value</i>	比較演算子。次の値が有効です。 <ul style="list-style-type: none"> • eq : 等しい • ge : 以上 • gt : より大きい • le : 以下 • lt : より小さい • wc : ワイルドカード
exit-op	(任意) 現在のイベント属性値と終了値を比較するために使用する演算子。
maxrun	(任意) アプレットの最大実行時間を指定します。 maxrun キーワードが指定されている場合、 <i>maxruntime-number</i> 値を指定する必要があります。 maxrun キーワードが指定されていない場合、デフォルトのアプレットランタイムは 20 秒です。
<i>maxruntime-number</i>	(任意) <i>sssssss[.mmm]</i> 形式で指定される秒数。 <i>sssssss</i> は、0～31536000 の秒数を表す整数で、 <i>mmm</i> は 0～999 のミリ秒数を表す整数である必要があります。

コマンド デフォルト デフォルトでは、イベントは NetFlow 動作がトリガーされたときにパブリッシュされません。

コマンド モード アプレット コンフィギュレーション (config-applet)

コマンド履歴	リリース	変更内容
	12.4(22)T	このコマンドが導入されました。
	12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン **event nf** コマンドを使用して、NetFlow イベントをモニタします。複数のイベントを一緒に指定して、複数のイベントに対し追加のフィルタリングを行うことができます。

例 次に、NetFlow イベントをモニタするようにアプレットを設定する例を示します。

```
Router(config)# event manager applet EventNF
Router(config-applet)# event nf event-type create monitor-name mon1 event1 entry-op eq
entry-val val1 field counter bytes long rate-interval 12
Router(config-applet)#
```

関連コマンド

コマンド	説明
event manager applet	EEMにイベント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。

event none

Embedded Event Manager (EEM) ポリシーを EEM に登録し、手動で実行できるように指定するには、アプレット コンフィギュレーション モードで **event none** コマンドを使用します。コンフィギュレーション ファイルから **event none** コマンドを削除するには、このコマンドの **no** 形式を使用します。

event [**tag** *event-tag*] **none** [**sync** {**yes**| **no**}] [**default**] [**maxrun** *maxruntime-number*]

no event none

構文の説明

tag	(任意) trigger コマンドで使用できる <i>event-tag</i> 引数を使用してタグを指定し、アプレットの複数のイベント文をサポートします。
<i>event-tag</i>	(任意) タグを識別するストリング。
synch	CLI コマンドを実行する前にポリシーを同期的に実行するかどうかを示します。 <ul style="list-style-type: none"> • yes キーワードが指定されている場合、ポリシーは、CLI コマンドと同期的に実行されます。 • no キーワードが指定されている場合、ポリシーは CLI コマンドと非同期的に実行されます。
default	(任意) CLI イベントディテクタがポリシーの終了を待つ時間 (sssssssss[.mmm] 形式で指定します。sssssssss は、0 ~ 4294967295 の秒数を表す整数で、mmm は 0 ~ 999 のミリ秒数を表す整数である必要があります)。ポリシーが終了する前にデフォルトの時間の期限が切れると、デフォルトアクションが実行されます。デフォルトアクションによって、コマンドが実行されます。この引数が指定されない場合、デフォルトの時間は 30 秒に設定されます。

maxrun	(任意) アプレットの最大実行時間を指定します。 maxrun キーワードが指定されている場合、 <i>maxruntime-number</i> 値を指定する必要があります。 maxrun キーワードが指定されていない場合、デフォルトのアプレットランタイムは 20 秒です。
<i>maxruntime-number</i>	(任意) sssssss[.mmm] 形式で指定される秒数。sssssss は、0 ~ 31536000 の秒数を表す整数で、mmm は 0 ~ 999 のミリ秒数を表す整数である必要があります。

コマンド デフォルト EEM イベントは、Cisco IOS システム モニタ カウンタに基づいてトリガーされません。

コマンド モード アプレット コンフィギュレーション (config-applet)。

コマンド履歴

リリース	変更内容
12.3(14)T	このコマンドが導入されました。
12.2(28)SB	このコマンドが、Cisco IOS Release 12.2(28)SB に統合されました。
12.2(18)SXF4	このコマンドが Cisco IOS Release 12.2(18)SXF4 に統合され、ソフトウェア モジュールリティ イメージだけをサポートするようになりました。
12.2(33)SRA	このコマンドが、Cisco IOS Release 12.2(33)SRA に統合されました。
12.2(18)SXF5	このコマンドが、Cisco IOS Release 12.2(18)SXF5 に統合されました。
12.2SX	このコマンドは、Cisco IOS Release 12.2SX トレインでサポートされます。このトレインの特定の 12.2SX リリースにおけるサポートは、フィーチャセット、プラットフォーム、およびプラットフォーム ハードウェアによって異なります。
12.4(20)T	tag および maxrun キーワードが、アプレット内の複数のイベント文をサポートするために追加されました。

使用上のガイドライン

EEMは、通常、ポリシーそのものに含まれるイベント仕様に基づいてポリシーをスケジューリングし、実行します。**event none** コマンドを使用することで、EEMポリシーが手動で実行できるものか、またはEEMアプレットがトリガーされたときに実行されるものかをEEMが識別できるようになります。ポリシーを実行するには、アプレットコンフィギュレーションモードで**action policy** コマンドを使用するか、またはグローバルコンフィギュレーションモードで**event manager run** コマンドを使用します。

例

次に、手動で実行する **manual-policy** という名前のポリシーを登録し、ポリシーを実行する例を示します。

```
Router(config)# event manager applet manual-policy
Router(config-applet)# event none
Router(config-applet)# exit
Router(config)# event manager run manual-policy
```

関連コマンド

コマンド	説明
action policy	EEM ポリシーを EEM に登録します。
event manager applet	EEM に EEM 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。
event manager run	登録された EEM ポリシーを手動で実行します。
show event manager policy registered	登録されている EEM ポリシーを表示します。

event routing

ルート エントリがルーティング情報ベース (RIB) インフラストラクチャで変更されたときにイベントをパブリッシュするには、アプレット コンフィギュレーション モードで **event routing** コマンドを使用します。ルート エントリが RIB で変更されたときのイベントのパブリッシュを停止するには、このコマンドの **no** 形式を使用します。

```
event[tag event-tag]routing network ip-address/length[ge ge-length][le le-length][protocol
protocol-value][type{add| all| modify| remove}][maxrun maxruntime-number]
```

```
no event [tag event-tag] routing
```

構文の説明

tag	(任意) trigger コマンドで使用できる <i>event-tag</i> 引数を使用してタグを指定し、アプレットの複数のイベント文をサポートします。
<i>event-tag</i>	(任意) タグを識別するストリング。
network	ルートをモニタするネットワーク IP アドレスと長さを指定します。
<i>ip-address / length</i>	モニタするネットワークの IP アドレスと長さ。たとえば、192.0.2.4/8 です。
ge <i>ge-length</i>	(任意) 照会されるプレフィックスの最小の長さを指定します。
le <i>le-length</i>	(任意) 照会されるプレフィックスの最大の長さを指定します。
ne <i>ne-length</i>	(任意) プレフィックスの長さを照会しない指定をします。
protocol	(任意) 監視されているネットワークのプロトコルの値を指定します。
<i>protocol-value</i>	ネットワーク プロトコル値。 all 、 bgp 、 connected 、 eigrp 、 isis 、 iso-igrp 、 mobile 、 odr 、 ospf 、 rip 、および static のプロトコルの 1 つを使用できます。デフォルトは、 all です。
type	(任意) 必要なポリシーのトリガーを指定します。デフォルトは、 all です。

add	エントリがルーティングテーブルに追加されることを指定します。
all	ルーティングテーブルエントリが追加、削除または変更されることを指定します。
modify	ルーティングテーブルのエントリが変更されることを指定します。
remove	エントリがルーティングテーブルから削除されることを指定します
maxrun	(任意) アプレットの最大実行時間を指定します。 maxrun キーワードが指定されている場合、 <i>maxruntime-number</i> 値を指定する必要があります。 maxrun キーワードが指定されていない場合、デフォルトのアプレットランタイムは 20 秒です。
<i>maxruntime-number</i>	(任意) <i>sssssss[.mmm]</i> 形式で指定される秒数。 <i>sssssss</i> は、0～31536000 の秒数を表す整数で、 <i>mmm</i> は 0～999 のミリ秒数を表す整数である必要があります。

コマンド デフォルト

デフォルトでは、ルートエントリが RIB インフラストラクチャで変更されたときにイベントはパブリッシュされません。

コマンド モード

アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン RIB インフラストラクチャでルートエントリの変更が検出されると EEM イベントがパブリッシュされます。監視されるルートのネットワーク IP アドレスを指定する必要があります。照合されるネットワークプレフィックス、プロトコル値、およびタイプはオプションのパラメータです。



(注) 既存のスタティックルートの変更は、Cisco IOS リリースに応じて、**remove** イベントとそれに続く古い API (v1.0) の **add** イベントか、または新しい API (v2.0) の **modify** イベントという結果になります。

例 次に、多数のパラメータがモニタされている場合に、特定のルートエントリがどのように変わるかの例を示します。

```
Router(config)
)# event manager applet EventRouting
Router(config-applet)# event routing 192.0.2.4/8 protocol static type add ge 5 maxrun 56
Router(config-applet)#
```

次に、古いルーティング API (v1.0) を使用する Cisco IOS バージョンの出力例を示します。

```
Router# show event manager detector routing
No. Name      Version      Node         Type
1 routing     01.00       node0/0      RP
```

次に、新しいルーティング API (v2.0) を使用する Cisco IOS バージョンの出力例を示します。

```
Router# show event manager detector routing
No. Name      Version      Node         Type
1 routing     02.00       node0/0      RP
```

関連コマンド

コマンド	説明
event manager applet	EEM にイベント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。

event snmp

簡易ネットワーク管理プロトコル（SNMP）オブジェクト識別子の値をサンプリングすることによって起動される Embedded Event Manager（EEM）アプレットのイベント条件を指定するには、アプレットコンフィギュレーションモードで **event snmp** コマンドを使用します。SNMP イベント条件を削除するには、このコマンドの **no** 形式を使用します。

event [**tag** *event-tag*] **snmp oid** *oid-value* **get-type** {**exact**|**next**} **entry-op** *operator* **entry-val** *entry-value* **entry-type** {**value**|**increment**|**rate**} [**exit-comb** {**or**|**and**}] [**exit-op** *operator*] [**exit-val** *exit-value*] [**exit-type** {**value**|**increment**|**rate**}] [**exit-time** *exit-time-value*] [**exit-event** {**true**|**false**}] [**average-factor** *average-factor-value*] **poll-interval** *poll-int-value* [**maxrun** *maxruntime-number*]

no event [**tag** *event-tag*] **snmp oid** *oid-value* **get-type** {**exact**|**next**} **entry-op** *operator* **entry-val** *entry-value* **entry-type** {**value**|**increment**|**rate**} [**exit-comb** {**or**|**and**}] [**exit-op** *operator*] [**exit-val** *exit-value*] [**exit-type** {**value**|**increment**|**rate**}] [**exit-time** *exit-time-value*] [**exit-event** {**true**|**false**}] [**average-factor** *average-factor-value*] **poll-interval** *poll-int-value* [**maxrun** *maxruntime-number*]

構文の説明

tag	(任意) trigger コマンドで使用できる <i>event-tag</i> 引数を使用してタグを指定し、アプレットの複数のイベント文をサポートします。
<i>event-tag</i>	(任意) タグを識別するストリング。
oid	<i>oid-value</i> 引数に SNMP オブジェクト識別子 (オブジェクト ID) をイベント条件として指定します。
<i>oid-value</i>	SNMP ドット付き表記でのデータエレメントのオブジェクト ID 値。OID は、関連する MIB (CISCO-EMBEDDED-EVENT-MGR-MIB) 内にタイプとして定義され、各タイプはオブジェクト値を保持します。一部の OID 型のモニタリングが可能です。 oid キーワードを使用すると、OID が次のいずれでもない場合にエラーメッセージが返されます。 <ul style="list-style-type: none"> • INTEGER_TYPE • COUNTER_TYPE • GAUGE_TYPE • TIME_TICKS_TYPE • COUNTER_64_TYPE • OCTET_PRIM_TYPE • OPAQUE_PRIM_TYPE

get-type	<i>oid-value</i> 引数で指定したオブジェクト ID に適用する SNMP get 操作のタイプを指定します。
exact	<i>oid-value</i> 引数で指定したオブジェクト ID を検索します。
next	<i>oid-value</i> 引数で指定したオブジェクト ID の英数字順で取得します。
entry-op	指定した演算子を使用して、現在のオブジェクト ID の内容と開始値を比較します。 マッチが見つかったら、イベントがトリガーされ、終了条件が満たされるまでイベント モニタリングがディセーブルになります。
<i>operator</i>	2 文字のストリング。 <i>operator</i> 引数には、次のいずれかの値を指定できます。 <ul style="list-style-type: none"> • gt : より大きい。 • ge : 以上。 • eq : 等しい。 • ne : 等しくない。 • lt : より小さい。 • le : 以下。
entry-val	SNMP イベントを発生させるかどうかを決定するために、現在のオブジェクト ID の内容と比較する値を指定します。
<i>entry-value</i>	データ要素のエントリ オブジェクト ID 値。
entry-type	<i>entry-value</i> 引数によって指定されたオブジェクト ID に適用される操作のタイプを指定します。
value	値は、 <i>entry-value</i> または <i>exit-value</i> 引数の実際の値として定義されます。

increment	増分では、 <i>entry-value</i> または <i>exit-value</i> フィールドは増分差異として使用され、 <i>entry-value</i> または <i>exit-value</i> は、現在のカウンタの値と、イベントが最後にトリガーされたとき（またはこれが新しいイベントの場合は最初にポーリングされたサンプル）の値との間の差と比較されます。負の値によって、減少しているカウンタの増分差異がチェックされます。
rate	レートは、ある期間の変更の平均レートとして定義されます。期間は、 <i>average-factor-value</i> に、 <i>poll-int-value</i> を乗じたものです。ポーリング間隔ごとに、現在のサンプルと前のサンプルとの間の差が取得され、絶対値として記録されます。前の <i>average-factor-value</i> サンプルの平均は、変更のレートとして取得されます。
exit-comb	(任意) イベントモニタリングを再度イネーブルにするために満たす必要のある終了条件の組み合わせを示します。
or	(任意) 終了比較演算子、終了オブジェクト ID 値、または終了時間値を指定する必要があります。
and	(任意) 終了比較演算子、終了オブジェクト ID 値、および終了時間値を指定する必要があります。
exit-op	(任意) 指定された演算子を使用して、現在のオブジェクト ID の内容と終了値を比較します。マッチが見つかりるとイベントが起動され、イベントモニタリングが再イネーブルされます。
exit-val	(任意) 終了条件が満たされているかどうかを決定するために、現在のオブジェクト ID の内容と比較する値を指定します。
<i>exit-value</i>	(任意) データ要素の終了オブジェクト ID 値。
exit-type	(任意) <i>exit-value</i> 引数によって指定されたオブジェクト ID に適用される操作のタイプを指定します。指定されない場合、値が仮定されます。

exit-time	(任意) この時間が経過した後に、イベントモニタリングが再度イネーブルになる時間を指定します。タイミングは、イベントがトリガーされた後に開始します。
<i>exit-time-value</i>	(任意) 秒数およびオプションのミリ秒数を ssssss[.mmm] 形式で表した数字。秒数の範囲は 0 ~ 4294967295 です。ミリ秒の範囲は 0 ~ 999 です。ミリ秒だけを使用する場合、形式は 0.mmm です。
exit-event	(任意) 最初のイベントがトリガーされた後に、イベントモニタリングがイネーブルになった場合、別の終了イベントがトリガーされるかどうかを示します。
true	(任意) 別の終了イベントがトリガーされることを指定します。
false	(任意) 別の終了イベントがトリガーされないことを指定します。これはデフォルトです。
average-factor	(任意) レートベースの計算に使用される期間の計算に使用される数値を指定します。 <i>average-factor-value</i> に <i>poll-int-value</i> を乗じて、ミリ秒単位で期間を算出します。
<i>average-factor-value</i>	(任意) 1 ~ 64 の範囲の番号。最少平均係数値は 1 です。
poll-interval	連続するポーリング操作間の時間間隔を指定します。
<i>poll-int-value</i>	秒数およびオプションのミリ秒数を ssssss[.mmm] 形式で表した数字。秒数の範囲は 1 ~ 4294967295 です。ミリ秒の範囲は 0 ~ 999 です。最小ポーリング間隔は 1 秒です。
maxrun	(任意) アプレットの最大実行時間を指定します。 maxrun キーワードが指定されている場合、 <i>maxruntime-number</i> 値を指定する必要があります。 maxrun キーワードが指定されていない場合、デフォルトのアプレットランタイムは 20 秒です。

<i>maxruntime-number</i>	(任意) SSSSSSSSS[.MMM] 形式で指定される秒数。SSSSSSSSS は、0 ~ 4294967295 の秒数を表す整数で、MMM は 0 ~ 999 のミリ秒数を表す整数である必要があります。
--------------------------	--

コマンド デフォルト EEM イベントは SNMP オブジェクト ID 値に基づいてトリガーされません。

コマンド モード アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.0(26)S	このコマンドが導入されました。
12.3(4)T	このコマンドが Cisco IOS Release 12.3(4)T に統合されました。
12.3(2)XE	このコマンドが、Cisco IOS Release 12.3(2)XE に統合されました。
12.2(25)S	このコマンドが、Cisco IOS Release 12.2(25)S に統合されました。
12.3(14)T	SNMP レートベースのイベントをサポートするオプションのキーワードが追加されました。
12.2(28)SB	このコマンドが、Cisco IOS Release 12.2(28)SB に統合されました。
12.2(18)SXF4	このコマンドが Cisco IOS Release 12.2(18)SXF4 に統合され、ソフトウェア モジュラリティ イメージだけをサポートするようになりました。
12.2(33)SRA	このコマンドが、Cisco IOS Release 12.2(33)SRA に統合されました。
12.2(18)SXF5	このコマンドが、Cisco IOS Release 12.2(18)SXF5 に統合されました。
12.2SX	このコマンドは、Cisco IOS Release 12.2SX トレインでサポートされません。このトレインの特定の 12.2SX リリースにおけるサポートは、フィーチャセット、プラットフォーム、およびプラットフォームハードウェアによって異なります。
12.4(20)T	tag および maxrun キーワードと、関連する引数が追加されました。

使用上のガイドライン EEM イベントは、SNMP オブジェクト ID によって指定されたフィールドのいずれか 1 つが定義されたしきい値を超えたときにトリガーされます。複数の条件が存在する場合、すべての条件が満たされたときに、SNMP イベントがトリガーされます。

終了基準は任意です。終了基準を指定しない場合、イベントモニタリングがすぐに再イネーブル化されます。終了基準が（値または期間に基づいて）指定されている場合、イベントモニタリングは条件が満たされるまで再イネーブル化されません。

entry-op キーワードが使用されている場合に一致が見つかり、イベントがトリガーされ、終了条件が満たされるまでイベントモニタリングがディセーブルになります。

exit-op キーワードが使用されている場合に一致が見つかり、イベントがトリガーされ、イベントモニタリングが再イネーブル化されます。

entry-type キーワードは、次のいずれかのアクションをトリガーします。

- **value** キーワードが指定されている場合、*entry-value* は実際の値で、絶対値が発生するたびに SNMP イベントが発生します。
- **increment** キーワードが指定されている場合、*entry-value* は増分で、増分値に到達するたびに SNMP イベントが発生します。
- **rate** キーワードが指定されている場合、*entry-value* は変化率で、変化率の値に到達するたびに SNMP イベントが発生します。

オプションの **exit-type** キーワードが使用される場合、次が発生します。

- **value** キーワードが指定されている場合、*exit-value* は実際の値で、絶対値が発生するたびにイベントモニタリングが再イネーブル化されます。これはデフォルトです。
- **increment** キーワードが指定されている場合、*exit-value* は増分で、増分値に到達するたびにイベントモニタリングが再イネーブル化されます。
- **rate** キーワードが指定されている場合、*exit-value* は変化率で、変化率の値に到達するたびにイベントモニタリングが再イネーブル化されます。

increment および **rate** タイプは、INTEGER_TYPE、COUNTER_TYPE、および COUNTER_64_TYPE の OID タイプに対してだけサポートされます。

例

次に、現在のプロセスメモリ量を表す、指定された SNMP オブジェクト ID の値との完全一致がある場合に、**memory-fail** と呼ばれる EEM アプレットがどのように実行されるかの例を示します。プロセスメモリが不足していること、および現在使用可能なメモリを示すメッセージが syslog に送信されます。

```
Router(config)# event manager applet memory-fail
Router(config-applet)# event snmp oid 1.3.6.1.4.1.9.9.48.1.1.1.6.1 get-type exact entry-op
lt entry-val 5120000 poll-interval 10
Router(config-applet)# action 1.0 syslog msg "Memory exhausted; current available memory
is $_snmp_oid_val bytes"
```

次に、IPSLAping1 という名前の EEM アプレットが登録され、指定された SNMP オブジェクト ID の値と完全一致する（正常な IP SLA ICMP エコー動作を表す）場合に実行される例を示します（これは ping コマンドに相当します）。エコー操作が失敗した場合は 4 つのアクションがトリガーされ、イベント モニタリングは 2 回目の失敗後までディセーブルにされます。

サーバへの ICMP エコー動作が失敗したことを示すメッセージが syslog に送信され、SNMP トラップが生成され、EEM はアプリケーション固有のイベントをパブリッシュし、IPSLA1F というカウンタが値 1 で増分されます。

```
Router(config)# event manager applet IPSLAping1
Router(config-applet)# event snmp oid 1.3.6.1.4.1.9.9.42.1.2.9.1.6.4 get-type exact
entry-op eq entry-val 1 exit-op eq exit-val 2 poll-interval 5
Router(config-applet)# action 1.0 syslog priority critical msg "Server IP echo failed:
OID=$_snmp_oid_val"
Router(config-applet)# action 1.1 snmp-trap strdata "EEM detected server reachability
failure to 10.1.88.9"
Router(config-applet)# action 1.2 publish-event sub-system 88000101 type 1 arg1 10.1.88.9
arg2 IPSLAEcho arg3 fail
```

関連コマンド

コマンド	説明
event manager applet	Embedded Event Manager にイベント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。

event snmp-notification

簡易ネットワーク管理プロトコル（SNMP）通知をサンプリングすることによって実行される Embedded Event Manager（EEM）アプレットのイベント条件を登録するには、アプレットコンフィギュレーションモードで **event snmp-notification** コマンドを使用します。SNMP 通知イベント条件を削除するには、このコマンドの **no** 形式を使用します。

event [**tag** *event-tag*] **snmp-notification** **oid** *oid-string* **oid-val** *comparison-value* **op** *operator* [**maxrun** *maxruntime-number*] [**src-ip-address** *ip-address*] [**dest-ip-address** *ip-address*] [**default** *seconds*] [**direction** {**incoming**| **outgoing**}] [**msg-op** {**drop**| **send**}]

no event [**tag** *event-tag*] **snmp-notification**

構文の説明

tag	（任意） trigger コマンドで使用できる <i>event-tag</i> 引数を使用してタグを指定し、アプレットの複数のイベント文をサポートします。
<i>event-tag</i>	（任意） タグを識別するストリング。
oid	<i>oid-val</i> 引数に SNMP オブジェクト識別子（オブジェクト ID）値をイベント条件として指定します。
<i>oid-string</i>	SNMP ドット付き表記でのデータエレメントのオブジェクト ID 値。OID は、関連する MIB（CISCO-EMBEDDED-EVENT-MGR-MIB）内にタイプとして定義され、各タイプはオブジェクト値を保持します。一部の OID 型のモニタリングが可能です。 oid キーワードを使用すると、OID が次のいずれでもない場合にエラーメッセージが返されます。 <ul style="list-style-type: none"> • COUNTER_TYPE • COUNTER_64_TYPE • GAUGE_TYPE • INTEGER_TYPE • OCTET_PRIM_TYPE • OPAQUE_PRIM_TYPE • TIME_TICKS_TYPE
oid-val <i>comparison-value</i>	OID の比較値を指定します。

op	指定した演算子を使用して、現在のオブジェクト ID の内容と SNMP プロトコルデータユニット (PDU) の開始値を比較します。マッチが見つかり、イベントがトリガーされ、終了条件が満たされるまでイベントモニタリングがディセーブルになります。
<i>operator</i>	2 文字のストリング。 <i>operator</i> 引数には、次のいずれかの値を指定できます。 <ul style="list-style-type: none"> • gt : より大きい。 • ge : 以上。 • eq : 等しい。 • ne : 等しくない。 • lt : より小さい。 • le : 以下。
maxrun	(任意) アプレットの最大実行時間を指定します。 maxrun キーワードが指定されていない場合、デフォルトのアプレットランタイムは 20 秒です。
<i>maxruntime-number</i>	(任意) ssssssss[.mmm] 形式で指定される秒数。 ssssssss は、0 ~ 31536000 の秒数を表す整数で、 mmm は 0 ~ 999 のミリ秒数を表す整数である必要があります。 デフォルト値は 20 秒です。
src-ip-address	(任意) SNMP 通知トラップが発信される送信元 IP アドレスを指定します。 デフォルトは all です。すべての IP アドレスから SNMP 通知トラップを受信するよう、設定されます。
<i>ip-address</i>	(任意) 送信元 IP アドレス。
dest-ip-address	(任意) SNMP 通知トラップを送信する宛先 IP アドレスを指定します。 デフォルトは all です。すべての宛先 IP アドレスから SNMP トラップを受信するよう、設定されます。
<i>dest-ip-address</i>	(任意) 宛先 IP アドレス。

default <i>seconds</i>	(任意) SNMP 通知イベントディテクタがポリシーの終了を待つ時間を指定します。time 時間は、sssssssss[.mmm] 形式で指定します。sssssssss は、0 ~ 4294967295 の秒数を表す整数で、mmm は 0 ~ 999 のミリ秒数を表す整数である必要があります。
direction	(任意) SNMP トラップの方向、または通知 PDU がフィルタリングする方向を決定します。デフォルトは incoming です。 incoming : SNMP トラップの着信方向、または通知 PDU がフィルタリングする方向を決定します。 outgoing : SNMP トラップの発信方向、または通知 PDU がフィルタリングする方向を決定します。
msg-op	(任意) イベントがトリガーされたときに、SNMP PDU で行われるアクション (ドロップまたは送信) を示します。 drop : メッセージをドロップするように指定します。 send : メッセージを送信するように指定します。

コマンド デフォルト EEM イベントは SNMP 通知オブジェクト ID 値に基づいてトリガーされません。

コマンド モード アプレット コンフィギュレーション (config-applet)

コマンド履歴	リリース	変更内容
	12.4(20)T	このコマンドが導入されました。
	15.0(1)M	このコマンドが変更されました。次のキーワードと引数が追加されました。 default 、 seconds 、 direction 、 incoming 、 outgoing 、 msg-op 、 drop 、および send 。

使用上のガイドライン SNMP 通知イベント ディテクタには、ルータが受信した SNMP トラップおよび SNMP インフォームメッセージを代行受信する機能があります。SNMP 通知イベントは、受信 SNMP トラップまたは SNMP インフォームメッセージが指定された値に一致するか、指定されたしきい値を超えたときに生成されます。

SNMP および SNMP サーバ マネージャは SNMP 通知イベント ディテクタの使用前に設定され、イネーブルにされている必要があります。

EEM イベントは、SNMP 通知オブジェクト ID によって指定されたフィールドのいずれか 1 つが定義されたしきい値を超えたときにトリガーされます。複数の条件が存在する場合、すべての条件が満たされたときに、SNMP 通知イベントがトリガーされます。

OID は、関連する MIB (CISCO-EMBEDDED-EVENT-MGR-MIB) 内にタイプとして定義され、各タイプはオブジェクト値を保持します。一部の OID 型のモニタリングが可能です。oid キーワードを使用すると、OID が次のいずれでもない場合にエラーメッセージが返されます。

- INTEGER_TYPE
- COUNTER_TYPE
- GAUGE_TYPE
- TIME_TICKS_TYPE
- COUNTER_64_TYPE
- OCTET_PRIM_TYPE
- OPAQUE_PRIM_TYPE

op キーワードが使用されている場合に一致が見つかったら、イベントがトリガーされ、終了条件が満たされるまでイベント モニタリングがディセーブルになります。

operator 引数には、次のいずれかの値を指定できます。

- gt : より大きい。
- ge : 以上。
- eq : 等しい。
- ne : 等しくない。
- lt : より小さい。
- le : 以下。

例

次に、event snmp-notification が設定される前に、snmp-server community public RW コマンド、および snmp-server manager コマンドを設定する例を示します。

```
Router(config)# snmp-server community public RW
Router(config)# snmp-server manager
```

次に、値が 10 であるオブジェクト OID 1 の宛先 IP アドレス 192.168.1.1 でルータが SNMP 通知を受け取ったときに、EEM スクリプトを実行するように SNMP_Notification という名前の EEM アプレットを登録する例を示します。

```
Router(config)# event manager applet SNMP_Notification
Router(config-applet)# event snmp-notification dest-ip-address 192.168.1.1 oid 1 op eq
oid-val 10
Router(config-applet)# action 1 policy eem_script
```

次に、OID が 1.3.6.1.4.1.318.2.3.3 で OID 値が「UPS: Returned from battery backup power」の SNMP トラップを代行受信し、メッセージをドロップして別のメッセージを送信する例を示します。

```
Router(config)# event manager applet SNMP_Notification
Router(config-applet)# event snmp-notification dest_ip_address 192.168.1.1 oid
1.3.6.1.4.1.318.2.3.3 op eq oid-value "UPS: Returned from battery backup power" direction
outgoing msg-op drop
```

関連コマンド

コマンド	説明
event manager applet	EEM にイベント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。

event snmp-object

SNMP オブジェクトのサンプリングによって実行される Embedded Event Manager (EEM) アプレットの簡易ネットワーク管理プロトコル (SNMP) オブジェクトイベントを登録するには、アプレットコンフィギュレーションモードで **event snmp-object** コマンドを使用します。SNMP オブジェクトイベント条件を削除するには、このコマンドの **no** 形式を使用します。

```
event snmp-object oid oid-value type value sync {yes|no} skip {yes|no} istable {yes|no} [default seconds]
[maxrun maxruntime-number]
```

```
no event snmp-object
```

構文の説明

oid	SNMP オブジェクト識別子 (オブジェクト ID) を指定します。
<i>oid-value</i>	SNMP ドット付き表記でのデータエレメントのオブジェクト ID 値。OID は、関連する MIB (CISCO-EMBEDDED-EVENT-MGR-MIB) 内にタイプとして定義され、各タイプはオブジェクト値を保持します。

<p>type <i>value</i></p>	<p>オブジェクトのタイプを指定します。次の値が有効です。</p> <ul style="list-style-type: none"> • counter : 最小値が 0 の 32 ビットの数値。最大値に到達すると、カウンタが 0 にリセットされます。 • counter64 : 最小値が 0 の 64 ビットの数値。最大値に到達すると、カウンタが 0 にリセットされます。 • gauge : 最小値が 0 の 32 ビットの数値。たとえば、gauge オブジェクトタイプを使用して、ルータ上のインターフェイスの速度を測定できます。 • int : 管理対象オブジェクトのコンテキスト内の番号が付けられたタイプを指定する場合は、32 ビットの数字が使用されます。たとえば、ルータ インターフェイスの動作ステータスを 1 に設定した場合はアップ、2 に設定した場合はダウンを示します。 • ipv4 : IP バージョン 4 アドレス。 • octet : 物理アドレスを表すために使用される、16 進表記のオクテット文字列。 • oid : オブジェクト ID 値。 • string : テキスト文字列を表すために使用される、テキスト表記のオクテット文字列。 • uint : 10 進数の値を表すために使用される、32 ビット番号。
<p>sync</p>	<p>SNMP および EEM ポリシー実行を指定します。</p> <ul style="list-style-type: none"> • no : ポリシーと SNMP は非同期で実行されます。 • yes : ポリシーを実行し、その結果により SNMP 要求を実行するかどうかが決まります。

skip	<p>sync が no に設定されている場合は必須で、sync が yes の場合は使用できません。CLI コマンドの実行をスキップするかどうかを指定します。</p> <ul style="list-style-type: none"> • no : CLI コマンドを実行する必要があります。 • yes : CLI コマンドを実行しません。
istable	<p>(任意) OID が SNMP テーブルかどうかを指定します。</p> <ul style="list-style-type: none"> • yes : OID は、SNMP テーブルです。 • no : IOD は SNMP テーブルではありません。
default	<p>(任意) SNMP オブジェクト イベント デテクタがポリシーの終了を待つ時間。</p>
<i>seconds</i>	<p>(任意) 秒数およびオプションのミリ秒数を ssssssss [.mmm] 形式で表した数字。秒数の範囲は 0 ~ 4294967295 です。ミリ秒の範囲は 0 ~ 999 です。ミリ秒だけを使用する場合は、0.mmm の形式でミリ秒を指定します。</p>
maxrun	<p>(任意) アプレットの最大実行時間を指定します。</p>
<i>maxruntime-number</i>	<p>(任意) ssssssss[.mmm] 形式で指定される秒数。sssssss は、0 ~ 31536000 の秒数を表す整数で、mmm は 0 ~ 999 のミリ秒数を表す整数である必要があります。デフォルト値は 20 秒です。</p>

コマンドモード

アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
15.0(1)M	このコマンドが導入されました。
15.0(1)M1	このコマンドが変更されました。 counter64 および oid 値が type キーワードに追加されました。

使用上のガイドライン **event snmp-object** コマンドを使用して、SNMP オブジェクトをサンプリングすることで実行される EEM アプレットの SNMP オブジェクト イベントを登録します。

例 次に、**event snmp-object** コマンドの使用例を示します。

```
Router(config)# event manager applet test
Router(config-applet)# event snmp-object
```

関連コマンド

コマンド	説明
action syslog	EEM アプレットがトリガーされたときに、syslog にメッセージを書き込むアクションを指定します。
event manager applet	EEM にイベント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。

event track

指定したオブジェクト番号の Cisco IOS Object Tracking サブシステム レポートに基づいて起動される Embedded Event Manager (EEM) アプレットのイベント条件を指定するには、アプレット コンフィギュレーション モードで **event track** コマンドを使用します。 レポート イベント条件を削除するには、このコマンドの **no** 形式を使用します。

```
event [ label ] [tag event-tag] track object-number [state {up| down| any}] [maxrun maxruntime-number]
no event [ label ] [tag event-tag] track object-number [state {up| down| any}] [maxrun maxruntime-number]
```

構文の説明

tag	(任意) trigger コマンドで使用できる <i>event-tag</i> 引数を使用してタグを指定し、アプレットの複数のイベント文をサポートします。
<i>event-tag</i>	(任意) タグを識別するストリング。
<i>label</i>	(任意) 一意な識別子 (任意のストリングを指定できます)。ストリングに空白が含まれる場合は二重引用符で囲みます。
<i>object-number</i>	1 から 500 の範囲のトラッキング対象オブジェクトの番号。番号は、 track stub コマンドを使用して定義されます。
state	(任意) トラックされるオブジェクトの状態遷移によってイベントが発生するよう、指定します。
up	(任意) トラッキング対象オブジェクトがダウン状態からアップ状態に移行したときにイベントが発生するよう、指定します。
down	(任意) トラッキング対象オブジェクトがアップ状態からダウン状態に移行したときにイベントが発生するよう、指定します。
any	(任意) トラッキング対象オブジェクトが任意の状態から任意の状態に移行したときにイベントが発生するよう、指定します。これはデフォルトです。

maxrun	(任意) アプレットの最大実行時間を指定します。 maxrun キーワードが指定されている場合、 <i>maxruntime-number</i> 値を指定する必要があります。 maxrun キーワードが指定されていない場合、デフォルトのアプレットランタイムは 20 秒です。
<i>maxruntime-number</i>	(任意) sssssss[.mmm] 形式で指定される秒数。sssssss は、0 ~ 31536000 の秒数を表す整数で、mmm は 0 ~ 999 のミリ秒数を表す整数である必要があります。

コマンド デフォルト EEM イベント基準は指定されていません。

コマンド モード アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(2)T	このコマンドが導入されました。
12.2(31)SB3	このコマンドが、Cisco IOS Release 12.2(31)SB3 に統合されました。
12.2(33)SRB	このコマンドが、Cisco IOS Release 12.2(33)SRB に統合されました。
12.4(20)T	tag および maxrun キーワードが、アプレット内の複数のイベント文をサポートするために追加されました。
12.2(33)SXI	このコマンドが、Cisco IOS Release 12.2(33)SXI に統合されました。

使用上のガイドライン このコマンドに関連付けられた 2 つのエントリ変数があります。

- **_track_number** : イベントがトリガーされる原因となるトラックされるオブジェクトの番号。
- **_track_state** : イベントがトリガーされたときのトラックされるオブジェクトの状態。有効な値は「up」または「down」です。

このコマンドは、EEM を使用したオブジェクトのトラッキングを支援するために使用されます。各トラッキング対象オブジェクトは、トラッキング コマンドライン インターフェイス (CLI) で指定された一意の番号で識別されます。EEM などのクライアント プロセスは、この番号を使用

して特定のオブジェクトをトラッキングします。トラッキングプロセスは定期的に、トラッキング対象オブジェクトをポーリングし、値の変更を確認します。トラッキング対象オブジェクトの変更は、すぐに、または指定された遅延後に、対象のクライアントプロセスに通知されます。オブジェクトの値は、アップまたはダウンとして報告されます。

例

次に、トラッキング対象オブジェクトに基づいてイベント基準を指定する例を示します。

```
event manager applet track-ten
  event track 10 state any
  action 1.0 track set 10 state up
  action 2.0 track read 10
```

関連コマンド

コマンド	説明
action track read	EEM アプレットがトリガーされたときに、トラッキング対象オブジェクトの状態を読み取るアクションを指定します。
action track set	EEM アプレットがトリガーされたときに、トラッキング対象オブジェクトの状態を設定するアクションを指定します。
event manager applet	Embedded Event Manager にイベント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。
show track	トラッキング情報を表示します。
track stub	トラッキング対象のスタブオブジェクトを作成します。



S から Z までのコマンド

- [S から Z までのコマンド, 185 ページ](#)

S から Z までのコマンド

show event manager directory user

ユーザライブラリファイルまたはユーザ定義の Embedded Event Manager (EEM) ポリシーを保存するディレクトリを表示するには、特権 EXEC モードで **show event manager directory user** コマンドを使用します。

show event manager directory user [library| policy]

構文の説明

library	(任意) ユーザライブラリファイル。
policy	(任意) ユーザ定義 EEM ポリシー。

コマンドデフォルト

ユーザライブラリおよびユーザポリシーファイルの両方のディレクトリが表示されます。

コマンドモード

特権 EXEC

コマンド履歴

リリース	変更内容
12.3(14)T	このコマンドが導入されました。
12.2(28)SB	このコマンドが、Cisco IOS Release 12.2(28)SB に統合されました。
12.2(18)SXF4	このコマンドが Cisco IOS Release 12.2(18)SXF4 に統合され、ソフトウェアモジュラリティイメージだけをサポートするようになりました。
12.2(33)SRA	このコマンドが、Cisco IOS Release 12.2(33)SRA に統合されました。
12.2(18)SXF5	このコマンドが、Cisco IOS Release 12.2(18)SXF5 に統合されました。
12.2SX	このコマンドは、Cisco IOS Release 12.2SX トレインでサポートされます。このトレインの特定の 12.2SX リリースにおけるサポートは、フィチャーセット、プラットフォーム、およびプラットフォームハードウェアによって異なります。

使用上のガイドライン

event manager directory user コマンドを使用して、ユーザライブラリまたはユーザポリシーファイルを保存するために使用するディレクトリを指定します。

例

次に、ディスク 0 の /usr/fm_policies フォルダを EEM ユーザ ライブラリ ファイルの保存に使用するディレクトリとする例を示します。

```
Router# show event manager directory user library  
disk0:/usr/fm_policies
```

関連コマンド

コマンド	説明
event manager directory user	ユーザ ライブラリ ファイルまたはユーザ定義 EEM ポリシーの保存に使用するディレクトリを指定します。

show event manager environment

Embedded Event Manager (EEM) 環境変数の名前と値を表示するには、特権 EXEC モードで **show event manager environment** コマンドを使用します。

show event manager environment [**all** | *variable-name*]

構文の説明

all	(任意) すべての環境変数の情報を表示します。これはデフォルトです。
<i>variable-name</i>	(任意) 指定された環境変数に関する情報を表示します。

コマンド デフォルト

引数またはキーワードを指定しない場合、すべての環境変数の情報が表示されます。

コマンド モード

特権 EXEC

コマンド履歴

リリース	変更内容
12.2(25)S	このコマンドが導入されました。
12.3(14)T	このコマンドが、Cisco IOS Release 12.3(14)T に統合されました。
12.2(28)SB	このコマンドが、Cisco IOS Release 12.2(28)SB に統合されました。
12.2(18)SXF4	このコマンドが Cisco IOS Release 12.2(18)SXF4 に統合され、ソフトウェアモジュラリティイメージだけをサポートするようになりました。
12.2(33)SRA	このコマンドが、Cisco IOS Release 12.2(33)SRA に統合されました。
12.2(18)SXF5	このコマンドが、Cisco IOS Release 12.2(18)SXF5 に統合されました。
12.2SX	このコマンドは、Cisco IOS Release 12.2SX トレインでサポートされます。このトレインの特定の 12.2SX リリースにおけるサポートは、フィーチャセット、プラットフォーム、およびプラットフォームハードウェアによって異なります。

例

show event manager environment コマンドの出力例は、次のとおりです。

```
Router# show event manager environment
No.  Name          Value
1    _cron_entry    0-59/1 0-23/1 * * 0-7
2    _show_cmd     show version
3    _syslog_pattern .*UPDOWN.*Ethernet1/0.*
4    _config_cmd1 interface Ethernet1/0
5    _config_cmd2 no shutdown
```

下の表で、この出力で表示される重要なフィールドについて説明しています。

表 19 : *show event manager environment* のフィールドの説明

フィールド	説明
No.	EEM 環境変数に割り当てられたインデックス番号。
Name	EEM 環境変数が作成されたときに付けられた名前。
Value	EEM 環境変数が作成されたときに定義されたテキスト コンテンツ。

関連コマンド

コマンド	説明
event manager environment	EEM 環境変数を設定します。

show event manager history events

トリガーされた Embedded Event Manager (EEM) イベントを表示するには、特権 EXEC モードで **show event manager history events** コマンドを使用します。

show event manager history events [detailed] [maximum number]

構文の説明

detailed	(任意) 各 EEM イベントの詳細情報を表示します。
maximum	(任意) 表示するイベントの最大数を指定します。
<i>number</i>	(任意) 1 ~ 50 の範囲の番号。デフォルトは 50 です。

コマンドモード

特権 EXEC (#)

コマンド履歴

リリース	変更内容
12.2(25)S	このコマンドが導入されました。
12.3(14)T	このコマンドが、Cisco IOS Release 12.3(14)T に統合されました。
12.2(28)SB	このコマンドが、Cisco IOS Release 12.2(28)SB に統合されました。
12.2(18)SXF4	このコマンドが Cisco IOS Release 12.2(18)SXF4 に統合され、ソフトウェアモジュラリティイメージだけをサポートするようになりました。
12.2(33)SRA	このコマンドが、Cisco IOS Release 12.2(33)SRA に統合されました。
12.2(18)SXF5	このコマンドが、Cisco IOS Release 12.2(18)SXF5 に統合されました。
12.2SX	このコマンドは、Cisco IOS Release 12.2SX トレインでサポートされません。このトレインの特定の 12.2SX リリースにおけるサポートは、フィーチャセット、プラットフォーム、およびプラットフォームハードウェアによって異なります。
12.4(20)T	出力に Job ID および Status フィールドが含まれるように変更されました。

使用上のガイドライン `show event manager history events` コマンドを使用して、トリガーされた EEM イベントに関する情報をトラッキングします。

例 次に、トリガーされた 2 つのタイプのイベント（簡易ネットワーク管理プロトコル（SNMP）およびアプリケーション）を示す、`show event manager history events` コマンドの出力例を示します。

```
Router# show event manager history events
No.   Time of Event           Event Type           Name
1     Fri Aug13 21:42:57 2004 snmp                 applet: SAAPing1
2     Fri Aug13 22:20:29 2004 snmp                 applet: SAAPing1
3     Wed Aug18 21:54:48 2004 snmp                 applet: SAAPing1
4     Wed Aug18 22:06:38 2004 snmp                 applet: SAAPing1
5     Wed Aug18 22:30:58 2004 snmp                 applet: SAAPing1
6     Wed Aug18 22:34:58 2004 snmp                 applet: SAAPing1
7     Wed Aug18 22:51:18 2004 snmp                 applet: SAAPing1
8     Wed Aug18 22:51:18 2004 application         applet: CustApp1
```

次に、**Job ID** および **Status** フィールドが含まれる、`show event manager history events` コマンドの出力例を示します。

```
Router# show event manager history events
No. Job ID Status Time of Event           Event Type Name
1   1      success Thu Sep 7 02:54:04 2006 syslog   applet: two
2   2      success Thu Sep 7 02:54:04 2006 syslog   applet: three
3   3      success Thu Sep 7 02:54:04 2006 syslog   applet: four
4   4      abort   Thu Sep 7 02:54:04 2006 syslog   applet: five
5   5      abort   Thu Sep 7 02:54:04 2006 syslog   applet: six
6   6      abort   Thu Sep 7 02:54:04 2006 syslog   applet: seven
7   7      abort   Thu Sep 7 02:54:04 2006 syslog   applet: eight
8   8      cleared Thu Sep 7 02:54:04 2006 syslog   applet: nine
9   9      cleared Thu Sep 7 02:54:04 2006 syslog   applet: ten
10  10     cleared Thu Sep 7 02:54:04 2006 syslog   applet: eleven
```

次に、**detailed** キーワードを使用した `show event manager history events` コマンドの出力例を示します。

```
Router# show event manager history events detailed
No. Job ID Status Time of Event           Event Type Name
1   1      success Thu Sep 7 02:54:04 2006 syslog   applet: two
msg {23:13:29: %CLEAR-5-COUNTERS: Clear counter on all interfaces by console}
2   2      success Thu Sep 7 02:54:04 2006 syslog   applet: three
msg {23:13:29: %CLEAR-5-COUNTERS: Clear counter on all interfaces by console}
3   3      success Thu Sep 7 02:54:04 2006 syslog   applet: four
msg {23:13:29: %CLEAR-5-COUNTERS: Clear counter on all interfaces by console}
4   4      abort   Thu Sep 7 02:54:04 2006 syslog   applet: five
msg {23:13:29: %CLEAR-5-COUNTERS: Clear counter on all interfaces by console}
5   5      abort   Thu Sep 7 02:54:04 2006 syslog   applet: six
msg {23:13:29: %CLEAR-5-COUNTERS: Clear counter on all interfaces by console}
6   6      abort   Thu Sep 7 02:54:04 2006 syslog   applet: seven
msg {23:13:29: %CLEAR-5-COUNTERS: Clear counter on all interfaces by console}
7   7      cleared Thu Sep 7 02:54:04 2006 syslog   applet: eight
msg {23:13:29: %CLEAR-5-COUNTERS: Clear counter on all interfaces by console}
8   8      cleared Thu Sep 7 02:54:04 2006 syslog   applet: nine
msg {23:13:29: %CLEAR-5-COUNTERS: Clear counter on all interfaces by console}
9   9      cleared Thu Sep 7 02:54:04 2006 syslog   applet: ten
msg {23:13:29: %CLEAR-5-COUNTERS: Clear counter on all interfaces by console}
10  10     success Thu Sep 7 02:54:04 2006 syslog   applet: eleven
msg {23:13:29: %CLEAR-5-COUNTERS: Clear counter on all interfaces by console}
```

下の表で、この出力で表示される重要なフィールドについて説明します。

表 20 : show event manager history events のフィールドの説明

フィールド	説明
No.	イベント番号。
Job ID	一意の内部 EEM スケジューラ ジョブの ID 番号です。
Status	このイベントに対してスケジュールされたポリシーの実行完了ステータス。3つのステータス値があります。 <ul style="list-style-type: none"> • Success : このイベントのポリシーが正常に完了したことを示します。 • Abort : このイベントのポリシーがに異常終了したことを示します。 • Cleared : このイベントのポリシーが event manager scheduler clear コマンドを使用して実行から削除されたことを示します。
Time of Event	イベントがトリガーされた曜日、日付、時刻。
Event Type	イベントのタイプ。
Name	トリガーされたポリシーの名前。

関連コマンド

コマンド	説明
event manager history size	EEM 履歴テーブルのサイズを変更します。
event manager scheduler clear	実行中または実行を保留中の EEM ポリシーをクリアします。

show event manager history traps

送信された Embedded Event Manager (EEM) の簡易ネットワーク管理プロトコル (SNMP) トラップを表示するには、特権 EXEC モードで **show event manager history traps** コマンドを使用します。

show event manager history traps [server| policy]

構文の説明

server	(任意) EEM サーバからトリガーされた SNMP トラップを表示します。
policy	(任意) EEM ポリシー内からトリガーされた SNMP トラップを表示します。

コマンドモード

特権 EXEC

コマンド履歴

リリース	変更内容
12.2(25)S	このコマンドが導入されました。
12.3(14)T	このコマンドが、Cisco IOS Release 12.3(14)T に統合されました。
12.2(28)SB	このコマンドが、Cisco IOS Release 12.2(28)SB に統合されました。
12.2(18)SXF4	このコマンドが Cisco IOS Release 12.2(18)SXF4 に統合され、ソフトウェア モジュールリティ イメージだけをサポートするようになりました。
12.2(33)SRA	このコマンドが、Cisco IOS Release 12.2(33)SRA に統合されました。
12.2(18)SXF5	このコマンドが、Cisco IOS Release 12.2(18)SXF5 に統合されました。
12.2SX	このコマンドは、Cisco IOS Release 12.2SX トレインでサポートされません。このトレインの特定の 12.2SX リリースにおけるサポートは、フィーチャセット、プラットフォーム、およびプラットフォームハードウェアによって異なります。

使用上のガイドライン

show event manager history traps コマンドを使用して、SNMP ポリシーが EEM サーバから実行されたか、EEM ポリシーから実行されたかを識別します。

例

次に、**show event manager history traps** コマンドの出力例を示します。

```
Router# show event manager history traps policy
No.  Time                Trap Type                Name
1    Wed Aug18  22:30:58 2004  policy                EEM Policy Director
2    Wed Aug18  22:34:58 2004  policy                EEM Policy Director
3    Wed Aug18  22:51:18 2004  policy                EEM Policy Director
```

下の表で、この出力で表示される重要なフィールドについて説明しています。

表 21 : **show event manager history traps** のフィールドの説明

フィールド	説明
No.	トラップ番号。
Time	SNMP トラップが実行された日時。
Trap Type	SNMP トラップのタイプ。
Name	実行された SNMP トラップの名前。

関連コマンド

コマンド	説明
event manager history size	EEM 履歴テーブルのサイズを変更します。

show event manager metric processes

Cisco IOS ソフトウェア モジュールリティ プロセスに関する Embedded Event Manager (EEM) の信頼性メトリック データを表示するには、特権 EXEC モードで **show event manager metric processes** コマンドを使用します。

show event manager metric processes {**all**|*process-name*}

構文の説明

all	すべての Cisco IOS ソフトウェア モジュールリティ プロセスのプロセスメトリック データを表示します。
<i>process-name</i>	特定のプロセス名。

コマンドモード

特権 EXEC

コマンド履歴

リリース	変更内容
12.2(18)SXF4	このコマンドが導入されました。

使用上のガイドライン

このコマンドを使用して、Cisco IOS ソフトウェア モジュールリティ プロセスの信頼性メトリック データを表示します。システムでは、プロセスの開始時と終了時にレコードが保存され、このデータが、信頼性分析の基本データとして使用されます。

このコマンドによって提供される情報を使用することで、プロセスまたはプロセス グループの可用性に関する情報を取得できます。プロセスは、実行している場合に利用可能と見なされます。

例

show event manager metric processes コマンドの出力例の一部は、次のとおりです。この部分の例では、システムに挿入されているすべてのカード上でのプロセスのメトリック データを示す、最初と最後のエントリが表示されます。

```
Router# show event manager metric processes all
=====
node name: node0
process name: devc-pty, instance: 1
sub_system: 0, version: 00.00.0000
-----
last event type: process start
recent start time: Fri Oct10 20:34:40 2003
```

show event manager metric processes

```

recent normal end time: n/a
recent abnormal end time: n/a
number of times started: 1
number of times ended normally: 0
number of times ended abnormally: 0
most recent 10 process start times:
-----
Fri Oct10 20:34:40 2003
-----
most recent 10 process end times and types:
cumulative process available time: 6 hours 30 minutes 7 seconds 378 milliseconds
cumulative process unavailable time: 0 hours 0 minutes 0 seconds 0 milliseconds
process availability: 0.100000000
number of abnormal ends within the past 60 minutes (since reload): 0
number of abnormal ends within the past 24 hours (since reload): 0
number of abnormal ends within the past 30 days (since reload): 0
.
.
.
=====
node name: node0
process name: cdp2.iosproc, instance: 1
sub_system: 0, version: 00.00.0000
-----
last event type: process start
recent start time: Fri Oct10 20:35:02 2003
recent normal end time: n/a
recent abnormal end time: n/a
number of times started: 1
number of times ended normally: 0
number of times ended abnormally: 0
most recent 10 process start times:
-----
Fri Oct10 20:35:02 2003
-----
most recent 10 process end times and types:

cumulative process available time: 6 hours 29 minutes 45 seconds 506 milliseconds
cumulative process unavailable time: 0 hours 0 minutes 0 seconds 0 milliseconds
process availability: 0.100000000
number of abnormal ends within the past 60 minutes (since reload): 0
number of abnormal ends within the past 24 hours (since reload): 0
number of abnormal ends within the past 30 days (since reload): 0

```

下の表で、この出力で表示される重要なフィールドについて説明しています。

表 22 : show event manager metric processes のフィールドの説明

フィールド	説明
node name	ノード名。
process name	ソフトウェア モジュールリティ プロセス名。
instance	ソフトウェア モジュールリティ プロセスのインスタンス番号。
sub_system	サブシステム番号。
version	バージョン番号。

show event manager policy active

実行中の Embedded Event Manager (EEM) ポリシーを表示するには、特権 EXEC モードで **show event manager policy active** コマンドを使用します。

show event manager policy active [*queue-type* {*applet*|*call-home*|*axp*|*script*}] [*class class-options*]
detailed]

構文の説明

queue-type	(任意) EEM ポリシーのキュー タイプを指定します。
applet	(任意) EEM アプレット ポリシーを指定します。
call-home	(任意) EEM Call Home ポリシーを指定します。
axp	(任意) EEM axp ポリシーを指定します。
script	(任意) EEM スクリプト ポリシーを指定します。
class	(任意) EEM クラス ポリシーを指定します。
<i>class-options</i>	EEM クラス ポリシーを指定します。 次のうちの 1 つまたはすべてを指定できます。 <ul style="list-style-type: none"> • class-letter : EEM ポリシーに割り当てられるクラス文字。 A ~ Z の範囲の文字。 複数のインスタンスのクラス文字を指定できます。 • default : デフォルト クラスに登録されたポリシーを指定します。 • range class-letter-range : EEM ポリシー クラスを範囲で指定します。 range class-letter-range の複数のインスタンスを指定できます。 <i>class-letter-range</i> で使用する文字は大文字にする必要があります。
detailed	(任意) EEM ポリシーの詳細内容を指定します。

コマンドモード

特権 EXEC (#)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン

show event manager policy active コマンドを使用して、実行中のポリシーを表示します。

例

次に、priority、scheduler node、および event type フィールドが含まれた **show event manager policy active** コマンドの出力例を示します。

```
Router# show event manager policy active
no. job id  p s  status  time of event  event type  name
1      1      N A  wait    Wed Oct8 21:45:10 2008  syslog     continue.tcl
2      12609  N A  running Mon Oct29 20:49:42 2007  timer watchdog  loop.tcl
```

下の表で、この出力で表示される重要なフィールドについて説明します。

表 23 : **show event manager policy active** のフィールドの説明

フィールド	説明
no.	自動的にポリシーに割り当てられたインデックス番号。
job id	一意の内部 EEM スケジューラ ジョブの ID 番号です。
p	<p>ポリシーのプライオリティ。4つのプライオリティがあります。</p> <ul style="list-style-type: none"> • L : ポリシーが低いプライオリティであることを示します。 • H : ポリシーが高いプライオリティであることを示します。 • N : ポリシーが標準プライオリティであることを示します。 • Z : ポリシーが最も低いプライオリティであることを示します。

フィールド	説明
s	<p>ポリシーのスケジューラノード。2つのノードがあります。</p> <ul style="list-style-type: none"> • A : このポリシーのスケジューラノードがアクティブであることを示します。 • S : このポリシーのスケジューラノードがスタンバイであることを示します。
status	<p>ポリシーのスケジューリングステータス。6つのステータス値があります。</p> <ul style="list-style-type: none"> • pend : ポリシーの実行を待機しています。 • runn : ポリシーが実行中であることを示します。 • exec : ポリシーの実行が完了し、スケジューラクリーンアップタスクを待機しています。 • hold : ポリシーはホールド中であることを示します。 • wait : ポリシーが新しいイベントを待機していることを示します。 • continue : ポリシーが新しいイベントを受信し、実行可能であることを示します。
time of event	<p>ポリシーが EEM サーバの実行キューにキューイングされた日時。</p>
event type	<p>イベントのタイプ。</p>
name	<p>EEM ポリシー ファイルの名前。</p>

関連コマンド

コマンド	説明
show event manager	<p>EEM ポリシーのイベント マネージャの詳細を示します。</p>

show event manager policy available

登録が可能な Embedded Event Manager (EEM) ポリシーを表示するには、特権 EXEC モードで **show event manager policy available** コマンドを使用します。

show event manager policy available [**description** [*policy-name*]] [**detailed** *policy-filename*] [**system**|**user**]]

構文の説明

description	(任意) 使用可能なポリシーの簡単な説明を入力します。
<i>policy-name</i>	(任意) ポリシーの名前。
detailed	(任意) 指定された <i>policy-filename</i> の実際のサンプルポリシーを表示します。
<i>policy-filename</i>	(任意) 表示されるサンプルポリシーの名前。
system	(任意) 使用可能なすべてのシステムポリシーを表示します。
user	(任意) 使用可能なすべてのユーザポリシーを表示します。

コマンド デフォルト

キーワードを指定しない場合、使用可能なすべてのシステムおよびユーザポリシーの情報が表示されます。

コマンド モード

特権 EXEC (#)

コマンド履歴

リリース	変更内容
12.2(25)S	このコマンドが導入されました。
12.3(14)T	user キーワードが追加され、このコマンドが Cisco IOS Release 12.3(14)T に統合されました。
12.2(28)SB	このコマンドが、Cisco IOS Release 12.2(28)SB に統合されました。

リリース	変更内容
12.2(18)SXF4	detailed キーワードおよび <i>policy-filename</i> 引数が追加され、このコマンドは、Cisco IOS Release 12.2(18)SXF4 に統合され、ソフトウェア モジュラリティ イメージだけをサポートするようになりました。
12.2(33)SRA	このコマンドが、Cisco IOS Release 12.2(33)SRA に統合されました。
12.2(18)SXF5	このコマンドが、Cisco IOS Release 12.2(18)SXF5 に統合されました。
12.2SX	このコマンドは、Cisco IOS Release 12.2SX トレインでサポートされます。このトレインの特定の 12.2SX リリースにおけるサポートは、フィーチャセット、プラットフォーム、およびプラットフォーム ハードウェアによって異なります。
12.4(20)T	.tbc ファイル拡張子が付いたバイトコードスクリプトを表示するように出力が変更されました。
15.0(1)M	コマンドが変更されました。 description キーワードと <i>policy-name</i> 引数が追加されました。

使用上のガイドライン

このコマンドは、**event manager policy** コマンドに必要なポリシーの正確な名前がわからない場合に便利です。

detailed キーワードは、実際の指定されたサンプルポリシーを表示します。 **description** *policy-name* を使用して、ポリシーを説明します。 *policy-name* を指定しない場合、**show** コマンドの出力は、すべての利用可能なポリシーの説明を表示します。

Cisco IOS Release 12.4(20)T では、EEM 2.4 はバイトコードのサポートを導入して Tcl スクリプトをバイトコード形式で保存できるようになりました。また、Tcl スクリプトの通常の .tcl 拡張子に加えて、.tbc 拡張子のファイルも表示するようにこのコマンドの出力が変更されました。

例

show event manager policy available コマンドの出力例は、次のとおりです。

```
Router# show event manager policy available
No.  Type      Time Created          Name
1    system    Tue Sep 12 09:41:32 2002  sl_intf_down.tcl
2    system    Tue Sep 12 09:41:32 2002  tm_cli_cmd.tcl
```

次の表に、この出力で表示されるフィールドについて説明します。

表 24 : **show event manager policy available** のフィールドの説明

フィールド	説明
No.	自動的にポリシーに割り当てられたインデックス番号。

show event manager policy available

フィールド	説明
Type	ポリシーがシステムポリシーであるかどうかを示します。
Time Created	ポリシーファイルが作成された日時を示すタイムスタンプ。
Name	EEM ポリシー ファイルの名前。

次に、**detailed** キーワードおよびポリシー名を指定した **show event manager policy available** コマンドの出力例を示します。

```
Router# show event manager policy available detailed tm_cli_cmd.tcl
::cisco::eem::event_register_timer cron name crontimer2 cron_entry $_cron_entry maxrun 240
#-----
# EEM policy that will periodically execute a cli command and email the
# results to a user.
#
# July 2005, Cisco EEM team
#
# Copyright (c) 2005 by cisco Systems, Inc.
# All rights reserved.
#-----
### The following EEM environment variables are used:
###
### _cron_entry (mandatory)           - A CRON specification that determines
###                                   when the policy will run. See the
###                                   IOS Embedded Event Manager
###                                   documentation for more information
###                                   on how to specify a cron entry.
### Example: _cron_entry              0-59/1 0-23/1 * * 0-7
###
### _email_server (mandatory)        - A Simple Mail Transfer Protocol (SMTP)
###                                   mail server used to send e-mail.
### Example: _email_server            mailserver.customer.com
###
```

次は、.tcl ファイル名拡張子が付いた Tcl スクリプトと .tbc ファイル名拡張子が付いたバイトコードスクリプトを表示する **show event manager policy available** コマンドの出力例を示します。この例は、Cisco IOS Release 12.4(20)T 以降のイメージ用です。

```
Router# show event manager policy available
No.  Type      Time Created          Name
1    system    Tue Jun 10 09:41:32 2008  sl_intf_down.tcl
2    system    Tue Jun 10 09:41:32 2008  tm_cli_cmd.tbc
```

関連コマンド

コマンド	説明
event manager policy	EEM ポリシーを EEM に登録します。

show event manager policy pending

実行保留中の Embedded Event Manager (EEM) ポリシーを表示するには、特権 EXEC モードで **show event manager policy pending** コマンドを使用します。

show event manager policy pending [**queue-type** {**applet**| **call-home**| **axp**| **script**}] [**class** *class-options*] **detailed**]

構文の説明

queue-type	(任意) EEM ポリシーのキュー タイプを指定します。
applet	(任意) EEM アプレット ポリシーを指定します。
call-home	(任意) EEM Call Home ポリシーを指定します。
axp	(任意) EEM axp ポリシーを指定します。
script	(任意) EEM スクリプト ポリシーを指定します。
class	(任意) EEM クラス ポリシーを指定します。
<i>class-options</i>	(任意) EEM ポリシー クラスを指定します。次のうちの1つまたはすべてを指定できます。 <ul style="list-style-type: none"> • class-letter : EEM ポリシーに割り当てられるクラス文字。A ~ Z の範囲の文字。複数のインスタンスのクラス文字を指定できます。 • default : デフォルト クラスに登録されたポリシーを指定します。 • range class-letter-range : EEM ポリシー クラスを範囲で指定します。 range class-letter-range の複数のインスタンスを指定できます。 <i>class-letter-range</i> で使用する文字は大文字にする必要があります。
detailed	(任意) EEM ポリシーの詳細内容を指定します。

コマンドモード

特権 EXEC (#)

コマンド履歴

リリース	変更内容
12.2(25)S	このコマンドが導入されました。
12.3(14)T	このコマンドが、Cisco IOS Release 12.3(14)T に統合されました。
12.2(28)SB	このコマンドが、Cisco IOS Release 12.2(28)SB に統合されました。
12.2(18)SXF4	このコマンドが Cisco IOS Release 12.2(18)SXF4 に統合され、ソフトウェア モジュール イメージだけをサポートするようになりました。
12.2(33)SRA	このコマンドが、Cisco IOS Release 12.2(33)SRA に統合されました。
12.2(18)SXF5	このコマンドが、Cisco IOS Release 12.2(18)SXF5 に統合されました。
12.2SX	このコマンドは、Cisco IOS Release 12.2SX トレインでサポートされます。このトレインの特定の 12.2SX リリースにおけるサポートは、フィチャセット、プラットフォーム、およびプラットフォーム ハードウェアによって異なります。
12.4(20)T	出力に Job ID および Status フィールドが含まれるように変更されました。
12.4(22)T	このコマンドは、出力表示でレポートされるポリシー キューの適格性を確認するために新しいオプションと一緒にサポートされ、詳細なポリシー情報を提供します。

使用上のガイドライン

保留中のポリシーは、EEM サーバの実行キューにある実行保留中のポリシーです。イベントがトリガーされると、イベントを処理するために登録されたポリシーは、EEM サーバのキューに入れます。**show event manager policy pending** コマンドを使用して、このキューのポリシーを表示し、ポリシーの詳細を確認します。

例

show event manager policy pending コマンドの出力例は、次のとおりです。

```
Router# show event manager policy pending
no. job id  p s  status  time of event  event type  name
1   12851  N A  pend   Mon Oct29   20:51:18 2007  timer watchdog  loop.tcl
2   12868  N A  pend   Mon Oct29   20:51:24 2007  timer watchdog  loop.tcl
3   12873  N A  pend   Mon Oct29   20:51:27 2007  timer watchdog  loop.tcl
4   12907  N A  pend   Mon Oct29   20:51:41 2007  timer watchdog  loop.tcl
5   13100  N A  pend   Mon Oct29   20:52:55 2007  timer watchdog  loop.tcl
```

下の表で、この出力で表示される重要なフィールドについて説明します。

表 25 : show event manager policy pending のフィールドの説明

フィールド	説明
no.	自動的にポリシーに割り当てられたインデックス番号。
job id	一意的内部 EEM スケジューラ ジョブの ID 番号です。
p	ポリシーのプライオリティ。4つのプライオリティがあります。 <ul style="list-style-type: none">• L : ポリシーが低いプライオリティであることを示します。• H : ポリシーが高いプライオリティであることを示します。• N : ポリシーが標準プライオリティであることを示します。• Z : ポリシーが最も低いプライオリティであることを示します。
s	ポリシーのスケジューラ ノード。2つのノードがあります。 <ul style="list-style-type: none">• A : このポリシーのスケジューラ ノードがアクティブであることを示します。• S : このポリシーのスケジューラ ノードがスタンバイであることを示します。

show event manager policy pending

フィールド	説明
status	<p>ポリシーのスケジューリングステータス。6つのステータス値があります。</p> <ul style="list-style-type: none"> • pend : ポリシーの実行を待機しています。 • runn : ポリシーが実行中であることを示します。 • exec : ポリシーの実行が完了し、スケジューラ クリーンアップ タスクを待機しています。 • hold : ポリシーはホールド中であることを示します。 • wait : ポリシーが新しいイベントを待機していることを示します。 • continue : ポリシーが新しいイベントを受信し、実行可能であることを示します。
time of event	ポリシーが EEM サーバの実行キューにキューイングされた日時。
event type	イベントのタイプ。
name	EEM ポリシー ファイルの名前。

関連コマンド

コマンド	説明
show event manager	EEM ポリシーのイベント マネージャの詳細を示します。

show event manager scheduler

スケジュールされた Embedded Event Manager (EEM) ポリシーのスケジュールアクティビティを表示するには、特権 EXEC モードで **show event manager scheduler** コマンドを使用します。

show event manager scheduler thread [queue-type {applet| call-home| axp| script}] [detailed]]

構文の説明

thread	スケジューラにスレッドを指定します。
queue-type	(任意) EEM ポリシーのキュー タイプを指定します。
applet	(任意) EEM アプレット ポリシーを指定します。
call-home	(任意) EEM Call Home ポリシーを指定します。
axp	(任意) EEM axp ポリシーを指定します。
script	(任意) EEM スクリプト ポリシーを指定します。
detailed	(任意) EEM ポリシーの詳細内容を指定します。

コマンドモード

特権 EXEC (#)

コマンド履歴

リリース	変更内容
12.4(22)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン

show event manager scheduler コマンドを使用して EEM のスケジューラ アクティビティを表示します。このコマンドは、スケジューラの視点からのすべての EEM 実行スレッドと実行中ポリシーの詳細を表示します。

次のオプションの 1 つまたはすべてを指定できます。 **applet**、**call-home**、**axp**、**script**、および **detailed**。

例

show event manager scheduler コマンドの出力例は、次のとおりです。

```
Router# show event manager scheduler thread
1 Script threads service class default
  total: 1 running: 1 idle: 0
2 Script threads service class range A-D
  total: 3 running: 0 idle: 3
3 Applet threads service class default
  total: 32 running: 0 idle: 32
4 Applet threads service class W X
  total: 5 running: 0 idle: 5
Router# show event manager scheduler script thread detailed
1 Script threads service class default
  total: 1 running: 1 idle: 0
1 job id: 1, pid: 215, name: continue.tcl
2 Script threads service class range A-D
  total: 3 running: 0 idle: 3
3 Applet threads service class default
  total: 32 running: 0 idle: 32
4 Applet threads service class W X
  total: 5 running: 0 idle: 5
```

関連コマンド

コマンド	説明
show event manager	EEM ポリシーのイベント マネージャの詳細を示します。

track stub-object

Embedded Event Manager (EEM) によってトラックできるスタブ オブジェクトを作成し、トラッキング コンフィギュレーションモードを開始するには、グローバル コンフィギュレーションモードで **track stub-object** コマンドを使用します。スタブ オブジェクトを削除するには、このコマンドの **no** 形式を使用します。

track object-number stub-object

no track object-number stub-object

構文の説明

<i>object-number</i>	トラッキング対象オブジェクトを表すオブジェクト番号。指定できる範囲は 1 ~ 1000 です。
----------------------	---

コマンド デフォルト

スタブ オブジェクトは作成されません。

コマンド モード

グローバル コンフィギュレーション (config)

コマンド履歴

リリース	変更内容
12.4(2)T	このコマンドが導入されました。
12.2(31)SB3	このコマンドが、Cisco IOS Release 12.2(31)SB3 に統合されました。
12.2(33)SRB	このコマンドが、Cisco IOS Release 12.2(33)SRB に統合されました。
Cisco IOS XE Release 2.1	このコマンドが、Cisco IOS XE Release 2.1 に統合されました。
12.2(33)SXI	このコマンドが、Cisco IOS Release 12.2(33)SXI に統合されました。
15.1(3)T	このコマンドが変更されました。 <i>object-number</i> 引数の有効範囲が 1000 に増えました。
15.1(1)S	このコマンドが変更されました。 <i>object-number</i> 引数の有効範囲が 1000 に増えました。

使用上のガイドライン **track stub-object** コマンドを使用して、外部プロセス (EEM) によりトラッキングおよび操作できるオブジェクトである、スタブ オブジェクトを作成します。スタブ オブジェクトの作成後、**default-state** コマンドを使用してスタブ オブジェクトのデフォルト状態を設定できます。

EEM は、イベント検出と回復を Cisco IOS 内部で直接行うための分散型でスケーラブルな、カスタマイズされた手法です。EEM では、イベントをモニタし、イベント発生が検出されたとき、およびしきい値を超えたときに情報通知や是正アクションを実施できます。EEM ポリシーは、イベントおよびイベントが発生した場合に行う処理を定義するエンティティです。

Cisco IOS Release 15.1(3)T 以降は、最大 1000 個のオブジェクトをトラッキングすることができます。トラッキング対象オブジェクトは 1000 個設定できますが、各トラッキング対象オブジェクトは CPU リソースを使用します。ルータで使用可能な CPU リソースの合計は、トラフィック負荷などの変数や、他のプロトコルがどのように設定され実行されているかに応じて異なります。1000 個のトラッキング対象オブジェクトが使用できるかどうかは、使用可能な CPU によって異なります。特定のサイトトラフィック条件下でサービスが機能することを保証するには、サイト上でテストを実施する必要があります。

例

次に、デフォルト状態がアップのスタブ オブジェクト 1 を作成し、設定する例を示します。

```
Router(config)#
track 1 stub-object
Router(config-track)#
default-state up
```

関連コマンド

コマンド	説明
default-state	スタブオブジェクトのデフォルト状態を設定します。
show track	トラッキング情報を表示します。

trigger (EEM)

トリガーアプレット コンフィギュレーション モードを開始し、Embedded Event Manager (EEM) アプレットに複数のイベント コンフィギュレーション文を指定するには、アプレット コンフィギュレーション モードで **trigger** コマンドを使用します。複数のイベント コンフィギュレーション文をディセーブルにするには、このコマンドの **no** 形式を使用します。

trigger [*occurs occurs-value*] [*period period-value*] [*period-start period-start-value*] [*delay delay-value*]
no trigger [*occurs occurs-value*] [*period period-value*] [*period-start period-start-value*] [*delay delay-value*]

構文の説明

occurs	(任意) EEM イベントが発生する前に発生した合計相関回数を指定します。数が指定されない場合、EEM イベントは1回目から発生します。
<i>occurs-value</i>	(任意) 番号の範囲は 1 ~ 4294967295 です。
period	(任意) 1 回以上のマッチを行う必要がある場合の実行間隔を指定します。指定しない場合、時間のチェックは適用されません。
<i>period-value</i>	(任意) 秒数およびオプションのミリ秒数を sssssssss [.mmm] 形式で表した数字。秒数の範囲は 0 ~ 4294967295 です。ミリ秒の範囲は 0 ~ 999 です。ミリ秒だけを使用する場合は、0.mmm の形式でミリ秒を指定します。
period-start	(任意) イベント相関ウィンドウの開始を指定します。指定されない場合、最初の CRON 期間の発生後、イベント監視はイネーブルにされます。
<i>period-start-value</i>	(任意) イベント相関ウィンドウの開始を指定するストリング。
delay	(任意) この秒数が経過した後に、すべての条件が真の場合はイベントが発生する秒数を指定します。指定しない場合、イベントはただちに発生します。

<i>delay-value</i>	(任意) 秒数およびオプションのミリ秒数を ssssssssss [.mmm] 形式で表した数字。秒数の範囲は 0 ~ 4294967295 です。ミリ秒の範囲は 0 ~ 999 です。ミリ秒だけを使用する場合は、0.mmm の形式でミリ秒を指定します。
--------------------	---

コマンド デフォルト ディセーブル

コマンド モード アプレット コンフィギュレーション (config-applet)

コマンド履歴

リリース	変更内容
12.4(20)T	このコマンドが導入されました。
12.2(33)SRE	このコマンドが、Cisco IOS Release 12.2(33)SRE に統合されました。

使用上のガイドライン **trigger** コマンドは、オプションの **tag** キーワードと各イベント文で指定される *event-tag* 引数を使用して、複数のイベント文を関連付けます。

例

次に、**trigger** コマンドを使用してトリガーアプレットコンフィギュレーションモードを開始し、EEM アプレットに対して複数のイベント コンフィギュレーション文を指定する例を示します。この例では、60 秒の期間内に **show bgp all** コマンドと「COUNT」というストリングを含む syslog メッセージが出現した場合にアプレットが実行されます。

```
Router(config)# event manager applet delay_50
Router(config-applet)# event
tag 1.0 cli pattern "show bgp all" sync yes occurs 32 period 60 maxrun 60
Router(config-applet)# event
tag 2.0 syslog pattern "COUNT"
Router(config-applet)# trigger occurs 1 delay 50
Router(config-applet-trigger)# correlate event 1.0 or event 2.0
Router(config-applet-trigger)# attribute tag 1.0 occurs 1
Router(config-applet-trigger)# attribute tag 2.0 occurs 1
Router(config-applet-trigger)# action 1.0 cli command "show memory"
Router(config-applet)# action 2.0 cli command "enable"
Router(config-applet)# action 3.0 cli command "config terminal"
Router(config-applet)# action 4.0 cli command " ip route 192.0.2.0 255.255.255.224 192.0.2.12"
Router(config-applet)# action 91.0 cli command "exit"
Router(config-applet)# action 99.0 cli command "show ip route | incl 192.0.2.5"
```

関連コマンド

コマンド	説明
attribute (EEM)	EEM アプレットの複合イベントを指定します。
correlate	1 つの複合イベントを構築します。
event manager applet	Embedded Event Manager にイベント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。

trigger (EEM)