



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 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、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)#** になります。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。
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文字列値。
<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tring を指定します。
<i>community-string</i>	(任意) SNMP コミュニティ String。コミュニティ String 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。String 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 String 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 String 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 String 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) • authenticationFaliure (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、sysDescr.0。
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、ロガーに向けられます。

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-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。

