



EEM アクションの Tcl コマンド拡張

次の表記法が、Tcl コマンド拡張ページで説明されている構文に使用されます。

- 任意の引数は、たとえば次の例のように、角カッコ内に示されます。

[type ?]

- 疑問符 (?) は入力する変数を表します。
- 引数間の選択肢は、たとえば次の例のように、パイプ文字で示されます。

priority low|normal|high



(注) すべての EEM Tcl コマンド拡張について、エラーがあった場合、戻される Tcl 結果文字列には、エラー情報が含まれます。



(注) 数値範囲が指定されていない引数は、-2147483648 から 2147483647 までの整数から取得されません。

- [action_policy](#) (2 ページ)
- [action_process](#) (2 ページ)
- [action_program](#) (4 ページ)
- [action_reload](#) (5 ページ)
- [action_script](#) (5 ページ)
- [action_snmp_trap](#) (6 ページ)
- [action_snmp_object_value](#) (7 ページ)
- [action_switch](#) (8 ページ)
- [action_syslog](#) (8 ページ)
- [action_track_read](#) (9 ページ)
- [action_track_set](#) (10 ページ)

action_policy

Tcl スクリプトで、None イベントディテクタで登録された Embedded Event Manager (EEM) ポリシーを実行できるようにします。EEM ポリシーを実行するアクションは、**event manager run** コマンドを使用して実行することもできます。

構文

```
action_policy ?
```

引数

?(文字列を表す)	(必須) 実行がスケジュールされる EEM ポリシーの名前。ポリシーは、None イベントディテクタを使用して事前に登録しておく必要があります。
-----------	--

なし

結果文字列

なし

_cerrno を設定

対応

```
(_cerr_sub_err = 2) FH_ESYSERR (generic/unknown error from OS/system)
```

このエラーは、オペレーティングシステムによってレポートされたエラーを意味します。エラーとともにレポートされる POSIX `errno` 値を使用して、オペレーティングシステムエラーの原因を調べます。

```
(_cerr_sub_err = 12) FH_ENOSUCHEID (unknown event ID)
```

このエラーは、ポリシーが登録されていないため、未知であることを意味します。

```
(_cerr_sub_err = 14) FH_ENOSUCHACTION (unknown action type)
```

このエラーは、要求されたアクション コマンドが未知であることを示します。

action_process

ソフトウェア モジュール方式プロセスを起動、再起動、または停止します。この Tcl コマンド拡張は、ソフトウェア モジュール方式イメージでのみサポートされます。

構文

```
action_process start|restart|kill [job_id ?]
[process_name ?] [instance ?]
```

引数

start	(必須) プロセスが起動されるよう指定します。
restart	(必須) プロセスが再起動されるよう指定します。
kill	(必須) プロセスが停止されるよう指定します。
job_id	(任意) システムマネージャによってプロセスに割り当てられるジョブ ID。この引数を指定する場合、1 ~ 4294967295 の範囲の整数である必要があります。
process_name	(任意) プロセス名。job_idを指定するか、または、process_name および instance を指定する必要があります。
instance	(任意) プロセスインスタンス ID。この引数を指定する場合、1 ~ 4294967295 の範囲の整数である必要があります。

結果文字列

なし

_cerno を設定

対応

```
(_cerr_sub_err = 14) FH_ENOSUCHACTION (unknown action type)
```

このエラーは、要求されたアクション コマンドが未知であることを示します。

```
(_cerr_sub_num = 425, _cerr_sub_err = 1) SYSMGR_ERROR_INVALID_ARGS (Invalid arguments passed)
```

このエラーは、渡された引数が無効であったことを意味します。

```
(_cerr_sub_num = 425, _cerr_sub_err = 2) SYSMGR_ERROR_NO_MEMORY (Could not allocate required memory)
```

このエラーは、メモリの内部 SYSMGR 要求に障害が発生したことを意味します。

```
(_cerr_sub_num = 425, _cerr_sub_err = 5) SYSMGR_ERROR_NO_MATCH (This process is not known to sysmgr)
```

このエラーは、プロセス名が未知であったことを意味します。

```
(_cerr_sub_num = 425, _cerr_sub_err = 14) SYSMGR_ERROR_TOO_BIG (outside the valid limit)
```

このエラーは、オブジェクトサイズがその最大値を超えたことを意味します。

```
(_cerr_sub_num = 425, _cerr_sub_err = 15) SYSMGR_ERROR_INVALID_OP (Invalid operation for this process)
```

このエラーは、その動作がプロセスに対して無効であったことを意味します。

action_program

Tcl スクリプトで、POSIX プロセス（プログラム）を実行できるようにします。任意で、該当する引数文字列、環境文字列、標準入力（stdin）パス名、標準出力（stdout）パス名、または標準エラー（stderr）パス名を使用します。この Tcl コマンド拡張は、ソフトウェア モジュール方式イメージでのみサポートされます。

構文

```
action_program path ? [argv ?] [envp ?] [stdin ?] [stdout ?] [stderr ?]
```

引数

path	(必須) 実行するプログラムのパス名。
argv	(任意) プログラムの引数文字列。
envp	(任意) プログラムの環境文字列。
stdin	(任意) stdin のパス名。
stdout	(任意) stdout のパス名。
stderr	(任意) stderr のパス名。

結果文字列

なし

_cerrno を設定

対応

```
(_cerr_sub_err = 2) FH_ESYSERR (generic/unknown error from OS/system)
```

このエラーは、オペレーティングシステムによってレポートされたエラーを意味します。エラーとともにレポートされる POSIX `errno` 値を使用して、オペレーティングシステムエラーの原因を調べます。

```
(_cerr_sub_err = 14) FH_ENOSUCHACTION (unknown action type)
```

このエラーは、要求されたアクション コマンドが未知であることを示します。

```
(_cerr_sub_err = 34)    FH_EMAXLEN (maximum length exceeded)
```

このエラーは、オブジェクト長またはオブジェクト数が、最大値を超えたことを意味します。

action_reload

デバイスがリロードされます。

構文

```
action_reload
```

引数

なし

結果文字列

なし

_cerno を設定

対応

```
(_cerr_sub_err = 2)    FH_ESYSERR (generic/unknown error from OS/system)
```

このエラーは、オペレーティングシステムによってレポートされたエラーを意味します。エラーとともにレポートされる POSIX `errno` 値を使用して、オペレーティングシステムエラーの原因を調べます。

```
(_cerr_sub_err = 14)   FH_ENOSUCHACTION (unknown action type)
```

このエラーは、要求されたアクション コマンドが未知であることを示します。

action_script

Tcl スクリプトで、すべての Tcl スクリプトの実行をイネーブルまたはディセーブルにします (スクリプト スケジューラをイネーブルまたはディセーブルにします)。

構文

```
action_script [status enable|disable]
```

引数

status	(任意) スクリプト実行ステータスを示すフラグ。この引数がイネーブルに設定されている場合、スクリプト実行がイネーブルにされます。この引数がディセーブルに設定されている場合、スクリプト実行がディセーブルにされます。
--------	--

結果文字列

なし

_cerrno を設定

対応

```
(_cerr_sub_err = 2)    FH_ESYSERR (generic/unknown error from OS/system)
```

このエラーは、オペレーティングシステムによってレポートされたエラーを意味します。エラーとともにレポートされる POSIX errno 値を使用して、オペレーティングシステムエラーの原因を調べます。

```
(_cerr_sub_err = 14)   FH_ENOSUCHACTION (unknown action type)
```

このエラーは、要求されたアクション コマンドが未知であることを示します。

```
(_cerr_sub_err = 52)   FH_ECONFIG (configuration error)
```

このエラーは、設定エラーが発生したことを意味します。

action_snmp_trap

Embedded Event Manager Notification MIB を使用して簡易ネットワーク管理プロトコル (SNMP) トラップを送信します。

構文

```
action_snmp_trap [intdata1 ?] [intdata2 ?] [strdata ?]
```

引数

intdata1	(任意) トラップで送信される任意の整数。
intdata2	(任意) トラップで送信される任意の整数。
strdata	(任意) トラップで送信される任意の文字列データ。

結果文字列

なし

`_cerrno` を設定

対応

```
(_cerr_sub_err = 2)    FH_ESYSERR    (generic/unknown error from OS/system)
```

このエラーは、オペレーティングシステムによってレポートされたエラーを意味します。エラーとともにレポートされる POSIX `errno` 値を使用して、オペレーティングシステムエラーの原因を調べます。

```
(_cerr_sub_err = 14)   FH_ENOSUCHACTION (unknown action type)
```

このエラーは、要求されたアクションコマンドが未知であることを示します。

action_snmp_object_value

SNMP `get` 要求で返される簡易ネットワーク管理プロトコル (SNMP) オブジェクト ID および値を設定します。

構文

```
action_snmp_object_value {int|uint|counter|gauge|ipv4|octet|counter64|string} ?  
[next_oid ?]
```

引数

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。
uint	10 進数の値を表す 32 ビット番号。
counter	最小値が 0 の 32 ビットの数値。
gauge	最小値が 0 の 32 ビットの数値。
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。
counter64	最小値が 0 の 64 ビットの数値。
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。
next_oid	テーブルにある次のオブジェクトの OID。テーブルの最後のオブジェクトの場合は NULL です。

結果文字列

なし

_cerrno を設定

対応

action_switch

完全冗長環境でセカンダリ プロセッサで処理するよう切り替えます。**action_switch** Tcl コマンド拡張を使用する前に、デバイスでバックアッププロセッサをインストールする必要があります。ハードウェアが完全冗長ではない場合、切り替えアクションは実行されません。

構文

```
action_switch
```

引数

なし

結果文字列

なし

_cerrno を設定

対応

```
(_cerr_sub_err = 2) FH_ESYSERR (generic/unknown error from OS/system)
```

このエラーは、オペレーティングシステムによってレポートされたエラーを意味します。エラーとともにレポートされる POSIX **errno** 値を使用して、オペレーティングシステムエラーの原因を調べます。

```
(_cerr_sub_err = 14) FH_ENOSUCHACTION (unknown action type)
```

このエラーは、要求されたアクション コマンドが未知であることを示します。

action_syslog

EEM スクリプトがトリガーされるときに、指定された機能を使用して定期的な Syslog メッセージを生成します。

構文

```
action_syslog [priority emerg|alert|crit|err|warning|notice|info|debug]
[msg ?] [facility ?]
```

引数

priority	(任意) action_syslog メッセージ ファシリティ レベル。この引数が指定されない場合、デフォルトのプライオリティは LOG_INFO です。
msg	(任意) 記録されるメッセージ。
facility	(任意) Syslog の機能。

結果文字列

なし

_cerno を設定

対応

action_track_read

Embedded Event Manager (EEM) スクリプトがトリガーされるときにトラックされるオブジェクトの状態を読み取ります。

構文

```
action_track_read ?
```

引数

? (番号を表す)	(必須) 1 から 500 の範囲でトラックされるオブジェクト番号。
-----------	------------------------------------

結果文字列

```
number {%u}
state {%s}
```

_cerno を設定

対応

```
FH_ENOTRACK
```

このエラーは、トラックされるオブジェクト番号が見つからなかったことを意味します。

action_track_set

Embedded Event Manager (EEM) スクリプトがトリガーされるときにトラックされるオブジェクトの状態を設定します。

構文

```
action_track_set ? state up|down
```

引数

? (番号を表す)	(必須) 1 から 500 の範囲でトラックされるオブジェクト番号。
state	(必須) トラックされるオブジェクトの状態が設定されるよう指定します。 up と指定されている場合、トラックされているオブジェクトの状態はアップです。down と指定されている場合、トラックされているオブジェクトの状態はダウンです。

結果文字列

なし

_cerno を設定

対応

FH_ENOTRACK

このエラーは、トラックされるオブジェクト番号が見つからなかったことを意味します。

翻訳について

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。