



EEM ユーティリティの 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。

- [appl_read](#) (2 ページ)
- [appl_reqinfo](#) (3 ページ)
- [appl_setinfo](#) (3 ページ)
- [counter_modify](#) (4 ページ)
- [description](#) (6 ページ)
- [fts_get_stamp](#) (7 ページ)
- [register_counter](#) (7 ページ)
- [register_timer](#) (9 ページ)
- [timer_arm](#) (11 ページ)
- [timer_cancel](#) (13 ページ)
- [unregister_counter](#) (14 ページ)

appl_read

Embedded Event Manager (EEM) アプリケーションの揮発性データを読み取ります。この Tcl コマンド拡張では、EEM アプリケーションの揮発性データの読み取りがサポートされます。EEM アプリケーションの揮発性データは、API をパブリッシュする EEM アプリケーションが使用される Cisco ソフトウェア プロセスによってパブリッシュすることができます。EEM アプリケーションの揮発性データは、EEM ポリシーによってパブリッシュできません。



(注) 現在、アプリケーション揮発性データをパブリッシュする Cisco ソフトウェアはありません。

構文

```
appl_read name ? length ?
```

引数

name	(必須) アプリケーションによってパブリッシュされる文字列データの名前。
length	(必須) 読み取る文字列データの長さ。1 ~ 4294967295 の範囲の整数である必要があります。

結果文字列

```
data %s
```

data は、読み取られる、アプリケーションによってパブリッシュされた文字列データです。

_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 = 7)    FH_ENOSUCHKEY    (could not find key)
```

このエラーは、アプリケーションイベントディテクタ情報キーまたはその他の ID が見つからなかったことを意味します。

```
(_cerr_sub_err = 9)    FH_EMEMORY    (insufficient memory for request)
```

このエラーは、メモリの内部 EEM 要求に障害が発生したことを意味します。

appl_reqinfo

Embedded Event Manager (EEM) から、前に保存された情報が取得されます。この Tcl コマンド拡張によって、一意のキーで前に保存された EEM からの情報の取得がサポートされます。これは、情報を取得するために指定する必要があります。情報の取得によって、その情報が EEM から削除されることに、注意してください。再度取得できるようにするには、再保存する必要があります。

構文

```
appl_reqinfo key ?
```

引数

キー	(必須) データの文字列キー。
----	-----------------

結果文字列

```
data %s
```

data は、取得されるアプリケーション文字列データです。

_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 = 7)    FH_ENOSUCHKEY    (could not find key)
```

このエラーは、アプリケーションイベントディテクタ情報キーまたはその他の ID が見つからなかったことを意味します。

appl_setinfo

Embedded Event Manager (EEM) に情報を保存します。この Tcl コマンド拡張によって、同じポリシーまたは別のポリシーによって、後で取得できる Embedded Event Manager への情報の保存がサポートされます。一意のキーを指定する必要があります。このキーを使用すると、情報を後で取得することができます。

構文

```
appl_setinfo key ? data ?
```

引数

キー	(必須) データの文字列キー。
data	(必須) 保存するアプリケーション文字列データ。

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 = 8)    FH_EDUPLICATEKEY    (duplicate appl info key)
```

このエラーは、アプリケーションイベントディテクタ情報キーまたはその他の ID が重複していたことを意味します。

```
(_cerr_sub_err = 9)    FH_EMEMORY    (insufficient memory for request)
```

このエラーは、メモリの内部 EEM 要求に障害が発生したことを意味します。

```
(_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。

```
(_cerr_sub_err = 43)   FH_EBADLENGTH    (bad API length)
```

このエラーは、API メッセージ長が無効であったことを意味します。

counter_modify

カウンタの値を変更します。

構文

```
counter_modify event_id ? val ? op nop|set|inc|dec
```

引数

event_id	(必須) register_counter Tcl コマンド拡張によって返されるカウンタイベント ID。0 ~ 4294967295 の範囲の整数である必要があります。
val	(必須) (注) <code>op nop</code> 引数値の組み合わせが指定されている以外は必須です。 <ul style="list-style-type: none"> • <code>op</code> が設定されている場合、この引数は、設定されるカウンタ値を表します。 • <code>op</code> が <code>inc</code> の場合、この引数は、カウンタを増やすために使用される値です。 • <code>op</code> が <code>dec</code> の場合、この引数は、カウンタを減らすために使用される値です。
op	(必須) <ul style="list-style-type: none"> • <code>nop</code> : 現在のカウンタの値を取得します。 • <code>set</code> : カウンタの値を指定値に設定します。 • <code>inc</code> : カウンタの値を指定値分増やします。 • <code>dec</code> : カウンタの値を指定値分減らします。

結果文字列

```
val_remain %d
```

val_remain は、カウンタの現在の値です。

_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 = 11)   FH_ENOSUCHESID (unknown event specification ID)
```

このエラーは、イベントが登録されたときか、またはイベントディテクタの内部イベント構造が破損したときに、イベント指定 ID を照会できなかったことを意味します。

```
(_cerr_sub_err = 22)   FH_ENULLPTR    (event detector internal error - ptr is null)
```

このエラーは、内部 EEM イベントディテクタ ポインタに値が含まれている必要があったときに、ヌルであったことを意味します。

```
(_cerr_sub_err = 30)   FH_ECTBADOPER (bad counter threshold operator)
```

このエラーは、カウンタ イベント ディテクタの設定演算子または変更演算子が、無効であったことを意味します。

description

記録されたポリシーの簡単な説明を記述します。

構文

```
description ?
```

引数

line	(任意) 1 文字から 240 文字で構成されるポリシーの簡単な説明。
------	-------------------------------------

結果文字列

なし

_cerno を設定

対応

使用例

説明文は、ポリシーの作成者によって入力されます。Tcl のイベント登録文の前または後に表示できます。ポリシーには、1 つの説明のみ使用できます。



(注) 1 つのポリシーに複数の説明文を登録した場合、障害が発生します。

次に、**event_register_syslog** ポリシーに簡単な説明が指定される例を示します。

```
::cisco::eem::description "This Tcl command looks for the word count in syslog messages."
::cisco::eem::event_register_syslog tag 1 ...
::cisco::eem::event_register_snmp_object tag 2 ...
::cisco::eem::trigger {
    ::cisco::eem::correlate event 1 and event 2
    ::cisco::eem::attribute tag 1 occurs 1
    ::cisco::eem::attribute tag 2 occurs 1
}
```

fts_get_stamp

最後にソフトウェアがブートされて以来の経過時間を返します。この Tcl コマンド拡張を使用すると、配列「nsec nnnn」に、ブート以降のナノ秒数が返されます。ここで、nnnn はナノ秒数です。

構文

```
fts_get_stamp
```

引数

なし

結果文字列

```
nsec %d
```

nsec は、ブート以降のナノ秒数です。

_cernno を設定

なし

register_counter

カウンタを登録し、カウンタ イベント ID を返します。この Tcl コマンド拡張は、カウンタのパブリッシャによって使用され、イベント ID を使用してカウンタを操作する前に、この登録が実行されます。

構文

```
register_counter name ?
```

引数

name	(必須) 操作されるカウンタの名前。
------	--------------------

結果文字列

```
event_id %d  
event_spec_id %d
```

event_id は、指定されたカウンタのカウントイベント ID です。unregister_counter または counter_modify Tcl コマンド拡張によって、カウンタの操作に使用されます。event_spec_id 引数は、指定されたカウンタのイベント指定 ID です。

_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 = 4)    FH_EINITONCE  (Init() is not yet done, or done twice.)
```

このエラーは、EEM イベント デテクタがその初期化を完了する前に、特定のイベントを登録する要求が行われたことを意味します。

```
(_cerr_sub_err = 6)    FH_EBADEVENTTYPE (unknown EEM event type)
```

このエラーは、内部イベント指定で指定されたイベントタイプが無効であったことを意味します。

```
(_cerr_sub_err = 9)    FH_EMEMORY    (insufficient memory for request)
```

このエラーは、メモリの内部 EEM 要求に障害が発生したことを意味します。

```
(_cerr_sub_err = 10)   FH_ECORRUPT  (internal EEM API context is corrupt)
```

このエラーは、内部 EEM API コンテキスト構造が破損したことを意味します。

```
(_cerr_sub_err = 11)   FH_ENOSUCHESID (unknown event specification ID)
```

このエラーは、イベントが登録されたときか、またはイベントデテクタの内部イベント構造が破損したときに、イベント指定 ID を照会できなかったことを意味します。

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

このエラーは、イベントが登録されたときか、またはイベントデテクタの内部イベント構造が破損したときに、イベント ID を照会できなかったことを意味します。

```
(_cerr_sub_err = 16)   FH_EBADFMPPTR  (bad ptr to fh_p data structure)
```

このエラーは、各 EEM API コールで使用されるコンテキスト ポインタが不正確であったことを意味します。

```
(_cerr_sub_err = 17)   FH_EBADADDRESS (bad API control block address)
```

このエラーは、EEM API に渡された制御ブロック アドレスが不正確であったことを意味します。


```
(_cerr_sub_err = 22)    FH_ENULLPTR    (event detector internal error - ptr is null)
```

このエラーは、内部 EEM イベントディテクタポインタに値が含まれている必要があったときに、ヌルであったことを意味します。

```
(_cerr_sub_err = 25)    FH_ESUBSEXCEED    (number of subscribers exceeded)
```

このエラーは、タイマーまたはカウンタのサブスクリイバの数が、最大値を超えたことを意味します。

```
(_cerr_sub_err = 26)    FH_ESUBSIDXINV    (invalid subscriber index)
```

これは、サブスクリイバの索引が無効であったことを意味します。

```
(_cerr_sub_err = 54)    FH_EFDUNAVAIL    (connection to event detector unavailable)
```

このエラーは、イベントディテクタが使用できなかったことを意味します。

```
(_cerr_sub_err = 56)    FH_EFDCONNERR    (event detector connection error)
```

このエラーは、この要求を処理する EEM イベントディテクタは使用できないことを意味します。

register_timer

タイマーを登録し、タイマー イベント ID を返します。この Tcl コマンド拡張は、カウンタのパブリッシャによって使用され、パブリッシャまたはサブスクリイバとしての登録に、**event_register_timer** コマンド拡張が使用されなかった場合に、イベント ID を使用してタイマーを操作する前に、この登録が実行されます。

構文

```
register_timer watchdog|countdown|absolute|cron name ?
```

引数

name	(必須) 操作されるタイマーの名前。
------	--------------------

結果文字列

```
event_id %u
```

event_id は指定したタイマーのタイマーイベント ID です (これを使用して、**timer_arm** または **timer_cancel** コマンド拡張によってタイマーを操作するために使用されます)。

_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 = 4)    FH_EINITONCE  (Init() is not yet done, or done twice.)
```

このエラーは、EEM イベント デテクタがその初期化を完了する前に、特定のイベントを登録する要求が行われたことを意味します。

```
(_cerr_sub_err = 6)    FH_EBADEVENTTYPE (unknown EEM event type)
```

このエラーは、内部イベント指定で指定されたイベントタイプが無効であったことを意味します。

```
(_cerr_sub_err = 9)    FH_EMEMORY    (insufficient memory for request)
```

このエラーは、メモリの内部 EEM 要求に障害が発生したことを意味します。

```
(_cerr_sub_err = 10)   FH_ECORRUPT  (internal EEM API context is corrupt)
```

このエラーは、内部 EEM API コンテキスト構造が破損したことを意味します。

```
(_cerr_sub_err = 11)   FH_ENOSUCHESID (unknown event specification ID)
```

このエラーは、イベントが登録されたときか、またはイベントデテクタの内部イベント構造が破損したときに、イベント指定 ID を照会できなかったことを意味します。

```
(_cerr_sub_err = 16)   FH_EBADFMPPTR  (bad ptr to fh_p data structure)
```

このエラーは、各 EEM API コールで使用されるコンテキスト ポインタが不正確であったことを意味します。

```
(_cerr_sub_err = 17)   FH_EBADADDRESS (bad API control block address)
```

このエラーは、EEM API に渡された制御ブロック アドレスが不正確であったことを意味します。

```
(_cerr_sub_err = 22)   FH_ENULLPTR   (event detector internal error - ptr is null)
```

このエラーは、内部 EEM イベント デテクタ ポインタに値が含まれている必要があったときに、ヌルであったことを意味します。

```
(_cerr_sub_err = 25)   FH_ESUBSEXCEED (number of subscribers exceeded)
```

このエラーは、タイマーまたはカウンタのサブスクリイバの数が、最大値を超えたことを意味します。

```
(_cerr_sub_err = 26)   FH_ESUBSIDXINV (invalid subscriber index)
```

これは、サブスクリイバの索引が無効であったことを意味します。

```
(_cerr_sub_err = 54)    FH_EFDUNAVAIL (connection to event detector unavailable)
```

このエラーは、イベント デテクタが使用できなかったことを意味します。

```
(_cerr_sub_err = 56)    FH_EFDCONNERR (event detector connection error)
```

このエラーは、この要求を処理する EEM イベント デテクタは使用できないことを意味します。

timer_arm

タイマーを搭載します。タイプは、CRON、ウォッチドッグ、カウントダウン、または絶対の場合があります。

構文

```
timer_arm event_id ? cron_entry ?|time ?
```

引数

event_id	(必須) register_timer Tcl コマンド拡張によって返されるタイマー イベント ID。0 ~ 4294967295 の範囲の整数である必要があります。
cron_entry	(必須) タイマー タイプが CRON の場合に存在する必要があります。他のタイプのタイマーの場合には、存在させることはできません。CRON タイマー指定によって、CRON テーブル エントリの形式が使用されます。
time	(必須) タイマー タイプが CRON ではない場合に存在する必要があります。タイマー タイプが CRON の場合には、存在できません。ウォッチドッグ タイマーおよびカウントダウン タイマーでは、タイマーの期限が切れるまでの秒数およびミリ秒数です。絶対タイマーでは、期限切れ時刻のカレンダー時間です (SSSSSSSSSS[.MMM] 形式で指定します。SSSSSSSSSS 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)。期限の絶対日付は、1970 年 1 月 1 日以降の秒およびミリ秒の単位での数です。指定された日付がすでに過ぎた場合、タイマーの期限はただちに切れます。

結果文字列

```
sec_remain %ld msec_remain %ld
```

sec_remain および msec_remain は、タイマーの次の期限切れまでの残り時間です。



(注) タイマー タイプが CRON の場合、sec_remain 引数および msec_remain 引数には 0 が返されません。

_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 = 6)    FH_EBADEVENTTYPE (unknown EEM event type)
```

このエラーは、内部イベント指定で指定されたイベントタイプが無効であったことを意味します。

```
(_cerr_sub_err = 9)    FH_EMEMORY    (insufficient memory for request)
```

このエラーは、メモリの内部 EEM 要求に障害が発生したことを意味します。

```
(_cerr_sub_err = 11)   FH_ENOSUCHESID    (unknown event specification ID)
```

このエラーは、イベントが登録されたときか、またはイベントディテクタの内部イベント構造が破損したときに、イベント指定 ID を照会できなかったことを意味します。

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

このエラーは、イベントが登録されたときか、またはイベントディテクタの内部イベント構造が破損したときに、イベント ID を照会できなかったことを意味します。

```
(_cerr_sub_err = 22)   FH_ENULLPTR    (event detector internal error - ptr is null)
```

このエラーは、内部 EEM イベントディテクタポインタに値が含まれている必要があったときに、ヌルであったことを意味します。

```
(_cerr_sub_err = 27)   FH_ETMDELAYZR    (zero delay time)
```

このエラーは、タイマーの搭載に指定された時間がゼロであったことを意味します。

```
(_cerr_sub_err = 42)   FH_ENOTREGISTERED (request for event spec that is unregistered)
```

このエラーは、イベント検出が登録できなかったことを意味します。

```
(_cerr_sub_err = 54)   FH_EFDUNAVAIL    (connection to event detector unavailable)
```

このエラーは、イベントディテクタが使用できなかったことを意味します。

```
(_cerr_sub_err = 56)   FH_EFDCONNERR    (event detector connection error)
```

このエラーは、この要求を処理する EEM イベントディテクタは使用できないことを意味します。

timer_cancel

タイマーを取り消します。

構文

```
timer_cancel event_id ?
```

引数

event_id	(必須) register_timer Tcl コマンド拡張によって返されるタイマー イベント ID。0 ~ 4294967295 の範囲の整数である必要があります。
----------	--

結果文字列

```
sec_remain %ld msec_remain %ld
```

sec_remain および msec_remain は、タイマーの次の期限切れまでの残り時間です。



(注) タイマー タイプが CRON の場合、sec_remain および msec_remain には 0 が返されます。

_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 = 6)    FH_EBADEVENTTYPE (unknown EEM event type)
```

このエラーは、内部イベント指定で指定されたイベントタイプが無効であったことを意味します。

```
(_cerr_sub_err = 7)    FH_ENOSUCHKEY (could not find key)
```

このエラーは、アプリケーション イベント デテクタ情報キーまたはその他の ID が見つからなかったことを意味します。

```
(_cerr_sub_err = 11)   FH_ENOSUCHESID (unknown event specification ID)
```

このエラーは、イベントが登録されたときか、またはイベント デテクタの内部イベント構造が破損したときに、イベント指定 ID を照会できなかったことを意味します。

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

このエラーは、イベントが登録されたときか、またはイベントディテクタの内部イベント構造が破損したときに、イベント ID を照会できなかったことを意味します。

```
(_cerr_sub_err = 22)    FH_ENULLPTR (event detector internal error - ptr is null)
```

このエラーは、内部 EEM イベントディテクタ ポインタに値が含まれている必要があったときに、ヌルであったことを意味します。

```
(_cerr_sub_err = 54)    FH_EFDUNAVAIL (connection to event detector unavailable)
```

このエラーは、イベントディテクタが使用できなかったことを意味します。

```
(_cerr_sub_err = 56)    FH_EFDCONNERR (event detector connection error)
```

このエラーは、この要求を処理する EEM イベントディテクタは使用できないことを意味します。

unregister_counter

カウンタの登録を解除します。この Tcl コマンド拡張は、以前に **register_counter** Tcl コマンド拡張に登録されていたカウンタの登録を解除するために、カウンタパブリッシャによって使用されます。

構文

```
unregister_counter event_id ? event_spec_id ?
```

引数

event_id	(必須) register_counter コマンド拡張によって返されるカウンタイベント ID。0 ~ 4294967295 の範囲の整数である必要があります。
event_spec_id	(必須) register_counter コマンド拡張によって返された、指定されたカウンタのカウンタイベント指定 ID。0 ~ 4294967295 の範囲の整数である必要があります。

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 = 9)    FH_EMEMORY (insufficient memory for request)
```

このエラーは、メモリの内部 EEM 要求に障害が発生したことを意味します。

```
(_cerr_sub_err = 11)   FH_ENOSUCHESID (unknown event specification ID)
```

このエラーは、イベントが登録されたときか、またはイベントディテクタの内部イベント構造が破損したときに、イベント指定 ID を照会できなかったことを意味します。

```
(_cerr_sub_err = 22)   FH_ENULLPTR (event detector internal error - ptr is null)
```

このエラーは、内部 EEM イベントディテクタ ポインタに値が含まれている必要があったときに、ヌルであったことを意味します。

```
(_cerr_sub_err = 26)   FH_ESUBSIDXINV (invalid subscriber index)
```

これは、サブスクリバの索引が無効であったことを意味します。

```
(_cerr_sub_err = 54)   FH_EFDUNAVAIL (connection to event detector unavailable)
```

このエラーは、イベントディテクタが使用できなかったことを意味します。

```
(_cerr_sub_err = 56)   FH_EFDCONNERR (event detector connection error)
```

このエラーは、この要求を処理する EEM イベントディテクタは使用できないことを意味します。

unregister_counter

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。