EEM システム情報の 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 システム情報コマンド（**sys_reqinfo_xxx**）では、Set _cerrno セクションが yes に設定されています。

（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す。

sys_reqinfo_cli_freq

すべてのコマンドライン インターフェイス（CLI）イベントの頻度情報を問い合わせます。

構文

```
sys_reqinfo_cli_freq
```

引数

なし

結果文字列

```
rec_list {{CLI frequency string 0},{CLI frequency str 1}, ...}
```

各 CLI の頻度の文字列は、次のとおりです。

time_sec %ld time_msec %ld match_count %u raise_count %u occurs %u period_sec %ld period_msec %ld pattern {%s}

rec_list	CLI イベント頻度リストの開始をマークします。
time_sec time_msec	この CLI イベントが発生した最後の時刻。
match count	CLI イベントによって指定されたパターンが、CLI コマンドによって照会される回数。
raise_count	この CLI イベントが発生した回数。次のフィールドは、CLI イベント指定に関する情報です。 • sync：「yes」は、イベント パブリッシュが同期的に実行される必要があることを意味します。Event Manager Server がイベントのパブリッシュを完了したときに、イベントディテクタが通知されます。Event Manager Server は、CLI コマンドが実行される必要があるかどうかを示すコードを返します。 • skip：「yes」は、sync フラグが設定されているときに、CLI コマンドは実行してはいけないことを意味します。
occurs	イベントが発生する前の発生回数。この引数が指定されない場合、イベントは 1 回目から発生します。

period_sec period_msec	イベントを発生させるには、発生回数が POSIX タイマー ユニットのこの数以内である必要があります。この引数が指定されない場合は、適用されません。
pattern	CLI コマンドのパターン マッチの実行に使用される正規表現。

_cerrno を設定

対応

sys_reqinfo_cli_history

コマンドライン インターフェイス（CLI）コマンドの履歴を問い合わせます。

構文

```
sys_reqinfo_cli_history
```

引数

なし

結果文字列

```
rec_list {{CLI history string 0}, {CLI history str 1},...}
```

各 CLI の履歴の文字列は、次のとおりです。

```
time_sec %ld time_msec %ld cmd {%s}
```

rec_list	CLI コマンド履歴リストの開始をマークします。
time_sec time_msec	CLI コマンドが実行された時刻。
cmd	CLI コマンドのテキスト。

_cerrno を設定

対応

sys_reqinfo_cpu_all

指定された期間で、指定された順序で、上位プロセスの CPU 使用率（POSIX プロセスと IOS プロセスの両方）を問い合わせます。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

```
sys_reqinfo_cpu_all order cpu_used [sec ?] [msec ?] [num ?]
```

引数

order	（必須）プロセスの CPU 使用率のソートに使用される順序。
cpu_used	（必須）指定されたウィンドウの、CPU 使用率の平均が、降順でソートされるよう、指定します。
sec msec	（任意）CPU 使用率の平均が計算される、秒単位およびミリ秒単位での時間。0 から 4294967295 の範囲の整数である必要があります。指定されない場合か、または、sec と msec の両方が 0 と指定される場合、最新の CPU サンプルが使用されます。
num	（任意）表示される、ソートされたプロセスのリストの上位からのエントリの数。1 ～ 4294967295 の範囲の整数である必要があります。デフォルト値は 5 です。

結果文字列

```
rec_list {{process CPU info string 0},{process CPU info string 1}, ...}
```

各プロセスの CPU 情報文字列は、次のとおりです。

```
pid %u name {%s} cpu_used %u
```

rec_list	プロセス CPU 情報リストの開始をマークします。
pid	プロセス ID。
name	プロセス名。
cpu_used	sec と msec が、ゼロよりも大きい数で指定される場合、平均パーセンテージは、指定された時間のプロセス CPU 使用率から計算されるよう、指定します。sec と msec が、両方ともゼロか、または指定されない場合。平均パーセンテージは、最新のサンプルのプロセス CPU 使用率から計算されます。

_cerrno を設定

対応

sys_reqinfo_crash_history

クラッシュしたすべてのプロセスのプロセス情報を問い合わせます。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

```
sys_reqinfo_crash_history
```

引数

なし

結果文字列

```
rec_list {{crash info string 0},{crash info string 1}, ...}
Where each crash info string is:
job_id %u name {%s} respawn_count %u fail_count %u dump_count %u
inst_id %d exit_status 0x%x exit_type %d proc_state {%s} component_id 0x%x
crash_time_sec %ld crash_time_msec %ld
```

job_id	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。1 ～ 4294967295 の整数。
name	プロセス名。
respawn_count	プロセスの再起動の合計回数。
fail_count	プロセスの再起動試行の回数。プロセスが正常に再起動されると、このカウントはゼロにリセットされます。
dump_count	実行されたコア ダンプの回数。
inst_id	プロセス インスタンス ID。
exit_status	プロセスの最後の終了ステータス。
exit_type	最後の終了タイプ。
proc_state	Sysmgr プロセスの状態。error、forced_stop、hold、init、ready_to_run、run、run_rnode、stop、waitEOltimer、wait_rnode、wait_spawntimer、wait_tpl の 1 つです。
component_id	プロセスが属するコンポーネントのコンポーネント ID に割り当てられているバージョン マネージャ。
crash_time_sec crash_time_msec	1970 年 1 月 1 日以降の秒およびミリ秒の単位で、プロセスがクラッシュした最後の時刻を表します。

_cerrno を設定

対応

sys_reqinfo_mem_all

指定された期間で、指定された順序で、上位プロセスのメモリの使用状況（POSIX と IOS の両方）を問い合わせます。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

```
sys_reqinfo_mem_all order allocates|increase|used [sec ?] [msec ?] [num ?]
```

引数

order	（必須）プロセスのメモリの使用状況のソートに使用される順序。
allocates	（必須）指定された時間ウィンドウの期間に、メモリの使用状況が、プロセス割り当ての数によって降順でソートされるよう、指定します。
increase	（必須）指定された時間ウィンドウの期間に、メモリの使用状況が、プロセスで増えたメモリのパーセンテージによって降順でソートされるよう、指定します。
used	（必須）メモリが、プロセスによって使用される現在のメモリによってソートされるよう、指定します。
sec msec	（任意）プロセスでのメモリの使用状況が計算される、秒単位およびミリ秒単位での時間。0 から 4294967295 の範囲の整数である必要があります。sec と msec の両方が指定され、ゼロではない場合、割り当て数は、該当する時間で収集された最も古いサンプルと最新のサンプルでの、割り当て数の差です。パーセンテージは、該当する時間で収集された最も古いサンプルと最新のサンプルとの、パーセンテージの差分として計算されます。指定されない場合か、または、sec と msec の両方が 0 と指定される場合、収集された最初のサンプルが、最も古いサンプルとして使用されます。つまり、時間は、起動から現在時までの時間で設定されます。
num	（任意）表示される、ソートされたプロセスのリストの上位からのエントリの数。1 ～ 4294967295 の範囲の整数である必要があります。デフォルト値は 5 です。

結果文字列

```
rec_list {{process mem info string 0},{process mem info string 1}, ...}
```

各プロセスのメモリ情報文字列は、次のとおりです。

```
pid %u name {%s} delta_allocs %d initial_alloc %u current_alloc %u percent_increase %d
```

rec_list	プロセスのメモリの使用状況情報リストの開始をマークします。
pid	プロセス ID。

name	プロセス名。
delta_allocs	該当する期間で収集された、最も古いサンプルと最新のサンプルでの、割り当て数の差として、差を指定します。
initial_alloc	時間の開始時にプロセスによって使用される、キロバイト単位での、メモリの容量を指定します。
current_alloc	プロセスによって使用される、キロバイト単位での、メモリの容量を指定します。
percent_increase	該当する時間で収集された最も古いサンプルと最新のサンプルとの、使用メモリのパーセンテージの差分を指定します。パーセンテージの差は、current_alloc から initial_alloc の 100 を差し引いた数として、および、initial_alloc で割った数として、表すことができます。

_cerrno を設定

対応

sys_reqinfo_proc

1 つの POSIX プロセスに関する情報を問い合わせます。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

```
sys_reqinfo_proc job_id ?
```

引数

job_id	（必須）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。1 ～ 4294967295 の範囲の整数である必要があります。

結果文字列

```
job_id %u component_id 0x%x name {%s} helper_name {%s} helper_path {%s} path {%s}
node_name {%s} is_respawn %u is_mandatory %u is_hold %u dump_option %d
max_dump_count %u respawn_count %u fail_count %u dump_count %u
last_respawn_sec %ld last_respawn_msec %ld inst_id %u proc_state %s
level %d exit_status 0x%x exit_type %d
```

job_id	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。1 ～ 4294967295 の整数。
component_id	プロセスが属するコンポーネントのコンポーネント ID に割り当てられているバージョン マネージャ。

name	プロセス名。
helper_name	ヘルパー プロセスの名前。
helper_path	ヘルパー プロセスの実行可能パス。
path	プロセスの実行可能パス。
node_name	プロセスが属するノードのノード名に割り当てられているシステム マネージャ。
is_respawn	プロセスが再生成できることを指定するフラグ。
is_mandatory	プロセスが実行され続ける必要があることを指定するフラグ。
is_hold	APIによって呼び出されるまでプロセスが再生成されることを指定するフラグ。
dump_option	コア ダンプのオプション。
max_dump_count	許可されるコア ダンプの最大数。
respawn_count	プロセスの再起動の合計回数。
fail_count	プロセスの再起動試行の回数。プロセスが正常に再起動されると、このカウントはゼロにリセットされます。
dump_count	実行されたコア ダンプの回数。
last_respawn_sec last_respawn_msec	1970 年 1 月 1 日以降の POSIX タイマー ユニットでの秒およびミリ秒の単位で、プロセスが開始された最後の時刻を表します。
inst_id	プロセス インスタンス ID。
proc_state	Sysmgr プロセスの状態。error、forced_stop、hold、init、ready_to_run、run、run_rnode、stop、waitEOltimer、wait_rnode、wait_spawntimer、wait_tpl の 1 つです。
level	プロセス実行レベル。
exit_status	プロセスの最後の終了ステータス。
exit_type	最後の終了タイプ。

_cerrno を設定

対応

sys_reqinfo_proc_all

すべての POSIX プロセスの情報を問い合わせます。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

```
sys_reqinfo_proc_all
```

引数

なし

結果文字列

```
rec_list {{process info string 0}, {process info string 1},...}
```

各プロセスの情報文字列は、**sysreq_info_proc** Tcl コマンド拡張の結果文字列と同じです。

_cerrno を設定

対応

sys_reqinfo_routername

デバイス名を問い合わせます。

構文

```
sys_reqinfo_routername
```

引数

なし

結果文字列

```
routername %s
```

この場合、routername がデバイスの名前です。

_cerrno を設定

対応

sys_reqinfo_snmp

簡易ネットワーク管理プロトコル（SNMP）オブジェクトIDによって指定されたエンティティの値を問い合わせます。

構文

```
sys_reqinfo_snmp oid ? get_type exact|next
```

引数

oid	（必須）ドット付き表記での SNMP OID（たとえば、1.3.6.1.2.1.2.1.0）。
get_type	（必須）指定された OID に適用する必要がある SNMP 取得操作のタイプ。get_type が「exact」の場合、指定された OID の値が取得されます。get_type が「next」の場合、指定された OID の辞書順での後続値が取得されます。

結果文字列

```
oid {%s} value {%s}
```

oid	SNMP OID。
value	割り当てられたSNMPデータエレメントの値文字列。

_cerrno を設定

可

```
(_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 = 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 = 37)    FH_ENOSNMPDATA  (can't retrieve data from SNMP)
```

このエラーは、SNMP オブジェクト タイプのデータがなかったことを意味します。

```
(_cerr_sub_err = 51)    FH_ESTATSTYP  (invalid statistics data type)
```

このエラーは、SNMP 統計データ タイプ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。

sys_reqinfo_syslog_freq

すべての Syslog イベントの頻度情報を問い合わせます。

構文

```
sys_reqinfo_syslog_freq
```

引数

なし

結果文字列

```
rec_list {{event frequency string 0}, {log freq str 1}, ...}
```

各イベントの頻度の文字列は、次のとおりです。

```
time_sec %ld time_msec %ld match_count %u raise_count %u occurs %u
period_sec %ld period_msec %ld pattern {%s}
```

time_sec time_msec	1970 年 1 月 1 日以降の POSIX タイマー ユニットでの秒およびミリ秒の単位で、最後のイベントが発生した時刻を表します。
match_count	イベントの登録以降、この Syslog イベント指定によって指定されたパターンが、Syslog メッセージによって照会される回数。
raise_count	この Syslog イベントが発生した回数。
occurs	イベントを発生させるために必要な発生回数。指定されない場合、イベントは 1 回目から発生します。
period_sec period_msec	イベントを発生させるには、発生回数が POSIX タイマーユニットのこの数以内である必要があります。この引数が指定されない場合、時間のチェックは適用されません。
pattern	Syslog メッセージのパターン マッチの実行に使用される正規表現。

_cerrno を設定

可

```
(_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 = 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 = 45)    FH_ESEQNUM  (sequence or workset number out of sync)
```

このエラーは、イベント ディテクタ シーケンスまたは作業セット番号が無効であったことを意味します。

```
(_cerr_sub_err = 46)    FH_EREGEMPTY  (registration list is empty)
```

このエラーは、イベント ディテクタ登録リストが空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。

sys_reqinfo_syslog_history

指定された Syslog メッセージの履歴を問い合わせます。

構文

```
sys_reqinfo_syslog_history
```

引数

なし

結果文字列

```
rec_list {{log hist string 0}, {log hist str 1}, ...}
```

各記録の履歴の文字列は、次のとおりです。

```
time_sec %ld time_msec %ld msg {%s}
```

time_sec time_msec	1970 年 1 月 1 日以降の秒およびミリ秒の単位で、メッセージが記録された時刻を表します。
msg	Syslog メッセージ。

_cerrno を設定

可

```
(_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 = 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 = 44)    FH_EHISTEMPTY  (history list is empty)
```

このエラーは、履歴のリストが空であったことを意味します。

```
(_cerr_sub_err = 45)    FH_ESEQNUM  (sequence or workset number out of sync)
```

このエラーは、イベント ディテクタ シーケンスまたは作業セット番号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。

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。