



EEM イベント登録の 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す。

- [event_register_appl](#) (2 ページ)
- [event_register_cli](#) (4 ページ)
- [event_register_counter](#) (8 ページ)
- [event_register_gold](#) (10 ページ)
- [event_register_identity](#) (17 ページ)
- [event_register_interface](#) (20 ページ)
- [event_register_ioswdsysmon](#) (26 ページ)
- [event_register_ipsla](#) (30 ページ)
- [event_register_mat](#) (33 ページ)
- [event_register_neighbor_discovery](#) (35 ページ)
- [event_register_nf](#) (40 ページ)
- [event_register_none](#) (43 ページ)

- [event_register_oir](#) (45 ページ)
- [event_register_process](#) (47 ページ)
- [event_register_resource](#) (51 ページ)
- [event_register_rf](#) (53 ページ)
- [event_register_routing](#) (56 ページ)
- [event_register_rpc](#) (59 ページ)
- [event_register_snmp](#) (61 ページ)
- [event_register_snmp_notification](#) (65 ページ)
- [event_register_snmp_object](#) (68 ページ)
- [event_register_syslog](#) (71 ページ)
- [event_register_timer](#) (74 ページ)
- [event_register_timer_subscriber](#) (80 ページ)
- [event_register_track](#) (82 ページ)
- [event_register_wdssystemon](#) (84 ページ)

event_register_appl

アプリケーションイベントの登録を行います。この Tcl コマンド拡張は、**event_publish** Tcl コマンド拡張の別のポリシーの実行に続いて、アプリケーションイベントがトリガされたときにポリシーを実行するために使用します。**event_publish** コマンド拡張によって、アプリケーションイベントがパブリッシュされます。

アプリケーションイベントを登録するためには、サブシステムを指定する必要があります。Tcl ポリシーまたは内部 Embedded Event Manager (EEM) API のいずれかによって、アプリケーションイベントをパブリッシュできます。イベントがポリシーによってパブリッシュされている場合、ポリシーで予約される `sub_system` 引数は 798 です。

構文

```
event_register_appl [tag ?] sub_system ? type ? [queue_priority low|normal|high|last]
[maxrun ?] [nice 0|1]
```

引数

tag	(任意) Tcl スクリプト内で複数のイベント文をサポートするため、Tcl コマンド拡張のトリガーとともに使用できるタグを指定する文字列。
sub_system	(必須) アプリケーションイベントをパブリッシュした EEM ポリシーに割り当てられる番号。他のすべての番号は Cisco での使用のために予約されているため、番号は 798 に設定されます。この引数が指定されない場合、すべてのコンポーネントが照会されます。

type	<p>(必須) 指定されたイベント内のイベント サブタイプ。sub_system 引数および type 引数によって、アプリケーション イベントが一意に識別されます。この引数が指定されない場合、すべてのタイプが照会されます。この引数を指定する場合、1 ~ 4294967295 の整数を選択する必要があります。</p> <p>パブリッシュと登録が機能するためには、event_publish コマンド拡張と event_register_appl コマンド拡張の間でコンポーネントとタイプが一致する必要があります。</p>
queue_priority	<p>(任意) 次のような、スクリプトがキューに入れられる優先度レベル。</p> <ul style="list-style-type: none"> • queue_priority low : 3 つの優先度レベルの最も低いレベルでスクリプトがキューに入れられるよう指定します。 • queue_priority normal : low プライオリティよりも高く、high プライオリティよりも低い優先度レベルでスクリプトがキューに入れられるよう指定します。 • queue_priority high : 3 つの優先度レベルの最も高いレベルで、スクリプトがキューに入れられるよう指定します。 • queue_priority last : 最も低い優先度レベルでスクリプトがキューに入れられるよう指定します。 <p>「queue_priority_last」引数が設定された状態で複数のスクリプトが登録されている場合、これらのスクリプトは、イベントのパブリッシュ順に実行されます。</p> <p>(注) queue_priority 引数によって、登録されているスクリプトの実行優先度ではなく、キューイングの優先度が指定されます。</p> <p>この引数が指定されない場合、デフォルトのキューイング プライオリティは normal です。</p>
maxrun	<p>(任意) スクリプトの最大ランタイム (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)。この引数が指定されない場合、デフォルトの 20 秒ランタイム制限が使用されます。</p>
nice	<p>(任意) ポリシーの実行時間優先度の設定。nice 引数が 1 に設定されている場合、ポリシーは、デフォルトの優先度よりも低い実行時間優先度で実行されます。デフォルト値は 0 です</p>

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、アプリケーション イベントが発生します。

結果文字列

なし

_cerrno を設定

なし

Event_reqinfo

Event_reqinfo

```
"event_id %u event_type %u event_type_string {%s} event_pub_sec %u event_pub_msec %u"
"sub_system 0x%x type %u data1 {%s} data2 {%s} data3 {%s} data4 {%s}"
```

イベントタイプ	説明
event_id	パブリッシュされた該当イベントの ID を示す一意の番号。同一のイベントで複数のポリシーを実行可能であり、その場合、各ポリシーは同一の event_id を保持します。
event_type	イベントのタイプ。
event_type_string	このイベントタイプのイベントの名前を表す ASCII 文字列。
event_pub_sec event_pub_msec	イベントが Embedded Event Manager (EEM) にパブリッシュされたときの、秒単位およびミリ秒単位の時間。
sub_system	アプリケーションイベントをパブリッシュした EEM ポリシーに割り当てられる番号。他のすべての番号は Cisco での使用のために予約されているため、番号は 798 に設定されます。
type	指定されたコンポーネント内のイベントサブタイプ。
data1 data2 data3 data4	イベントがパブリッシュされるときに、アプリケーション固有のイベントに渡される、引数データ。データは、文字テキスト、環境変数、または、この 2 つの組み合わせです。

event_register_cli

CLI イベントの登録を行います。この Tcl コマンドを使用すると、拡張 CLI コマンドに対して実行されるパターンマッチに基づいて、特定パターンの CLI コマンドが入力されるときに、ポリシーが実行されます。



(注) ユーザーは、**sh mem summary** などの省略形の CLI コマンドを入力できます。パーサーによってコマンドが **show memory summary** に拡張され、照会が実行されます。



- (注) CLI イベント デテクタによる機能は、有効な IOS CLI コマンドでの正規表現パターン比較機能だけです。これには、リダイレクションが使用される場合のパイプ記号 (|) 以降のテキストは含まれません。

構文

```
event_register_cli [tag ?] sync yes|no skip yes|no
[occurs ?] [period ?] pattern ? [default ?] [enter] [questionmark] [tab] [mode]
[queue_priority low|normal|high|last] [maxrun ?] [nice 0|1]
```

引数

tag	(任意) Tcl スクリプト内で複数のイベント文をサポートするため、Tcl コマンド拡張のトリガーとともに使用できるタグを指定する文字列。
sync	(必須) 「yes」は、ポリシー (イベントパブリッシュ) が、CLI コマンドと同期的に実行することを意味します。「no」は、イベントパブリッシュが CLI コマンドと非同期に実行されることを意味します。ポリシーの実行が完了すると、イベントデテクタによって通知されます。ポリシーの終了ステータスは、CLI コマンドを実行する必要があるかどうかを示します。終了ステータスがゼロの場合は、ポリシーが正常に実行されたことを意味し、CLI コマンドは実行されません。それ以外の場合は、CLI コマンドが実行されます。
skip	sync 引数が no の場合は必須で、sync 引数が yes の場合は不要です skip 引数が yes の場合、CLI コマンドを実行する必要がないことを意味します。skip 引数が no の場合、CLI コマンドを実行する必要があることを意味します。 注意 skip 引数が yes の場合、パターンマッチがコンフィギュレーション コマンドに対して行われる場合、正規表現に一致する CLI コマンドは実行されないため、想定外の結果が生成される場合があります。
occurs	(任意) 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。この引数が指定される場合は、1～4294967295 の範囲の整数である必要があります。
period	(任意) イベントがパブリッシュされるようにするために、すべての CLI イベントが発生する必要がある (occurs 句を満たす必要がある) 逆方向検索時間ウィンドウを指定します (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)。この引数が指定されない場合は、最新のイベントが使用されます。
pattern	(必須) CLI コマンドパターンマッチの実行に使用される正規表現を指定します。

デフォルト	<p>(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 (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)。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。</p>
queue_priority	<p>(任意) 次のような、スクリプトがキューに入れられる優先度レベル。</p> <ul style="list-style-type: none"> • queue_priority low : 3 つの優先度レベルの最も低いレベルでスクリプトがキューに入れられるよう指定します。 • queue_priority normal : low プライオリティよりも高く、high プライオリティよりも低い優先度レベルでスクリプトがキューに入れられるよう指定します。 • queue_priority high : 3 つの優先度レベルの最も高いレベルで、スクリプトがキューに入れられるよう指定します。 • queue_priority last : 最も低い優先度レベルでスクリプトがキューに入れられるよう指定します。 <p>「queue_priority_last」引数が設定された状態で複数のスクリプトが登録されている場合、これらのスクリプトは、イベントのパブリッシュ順に実行されます。</p> <p>(注) queue_priority 引数によって、登録されているスクリプトの実行優先度ではなく、キューイングの優先度が指定されます。</p> <p>この引数が指定されない場合、デフォルトのキューイング プライオリティは normal です。</p>
enter	<p>(任意) ユーザーが Enter キーを押したときにイベント照会が実行されるよう、指定します。このパラメータが使用されると、照会前には入力文字列は拡張されません。</p>
questionmark	<p>(任意) ユーザーが ? キーを押したときにイベント照会が実行されるよう、指定します。このパラメータが使用されると、照会前には入力文字列は拡張されません。</p>
タブ	<p>(任意) ユーザーが Tab キーを押したときにイベント照会が実行されるよう、指定します。このパラメータが使用されると、照会前には入力文字列は拡張されません。</p>
mode	<p>(任意) パーサーが指定されたパーサー モードの場合のみ、イベントが生成されます。使用可能なモードのリストは、show parser dump CLI コマンドを使用して表示できます。オプションパラメータの enter、questionmark、または tab のいずれか 1 つが指定されている場合、mode パラメータが確認されます。</p>

maxrun	(任意) スクリプトの最大ランタイム (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)。この引数が指定されない場合、デフォルトの 20 秒ランタイム制限が使用されます。
nice	(任意) ポリシーの実行時間優先度の設定。nice 引数が 1 に設定されている場合、ポリシーは、デフォルトの優先度よりも低い実行時間優先度で実行されます。デフォルト値は 0 です

複数の条件が存在する場合、すべての条件が一致したときに、CLI イベントが発生します。

結果文字列

なし

_cerrno を設定

なし



- (注) このポリシーは、CLI コマンドの実行前に実行されます。たとえば、**copy** コマンドが入力されると、policy_CLI が実行のために登録されるとします。**copy** コマンドが入力されると、CLI イベントディテクタがパターン的一致を検出し、このポリシーの実行がトリガーされます。ポリシーの実行が終了すると、CLI イベントディテクタは、「sync」、「skip」(ポリシーで設定)、および、必要に応じてポリシー実行の終了ステータスに従って、**copy** コマンドを実行する必要があるかどうかを判断します。

Event_reqinfo

```
"event_id %u event_type %u event_type_string {%s} event_pub_sec %u event_pub_msec %u
event_severity %u msg {%s} msg_count %d line %u key %u tty %u error_code %u"
```

イベントタイプ	説明
event_id	パブリッシュされた該当イベントの ID を示す一意の番号。同一のイベントで複数のポリシーを実行可能であり、その場合、各ポリシーは同一の event_id を保持します。
event_type	イベントのタイプ。
event_type_string	このイベントタイプのイベントの名前を表す ASCII 文字列。
event_pub_sec event_pub_msec	EEM に対してイベントがパブリッシュされた、秒単位およびミリ秒単位の時間。
event_severity	イベントの重大度。
msg	CLI プロンプトで入力されるテキスト。

イベントタイプ	説明
msg_count	イベントがトリガーされる前にパターン マッチされた回数。
line	一致したキーが入力されたポイントまで、パーサーによって拡張できたテキスト。
key	Enter キー、疑問符、または Tab キー。
tty	ユーザーがコマンドを実行する行番号に対応します。
error_code	CLI のエラー コード。 0 : パーサーからキーが入力されたポイントまで、エラーはありません。 1 : キーが入力されたポイントまで、コマンドはあいまいです。 4 : キーが入力されたポイントまで、未知のコマンドです。

event_register_counter

パブリッシャとサブスクリイバの両方として、カウンタ イベントの登録を行います。この Tcl コマンド拡張を使用すると、しきい値に近くなった名前付きカウンタに基づいて、ポリシーが実行されます。サブスクリイバとして、このイベントカウンタによって、登録に必要なカウンタの名前が指定され、別のポリシーまたは別のプロセスに依存して、カウンタが実際に操作されます。たとえば、**policyB** をカウンタポリシーとして動作させる一方、**policyA**（カウンタポリシーである必要はない）では、**register_counter**、**counter_modify**、または **unregister_counter** の各 Tcl コマンド拡張を使用して、**policyB** で定義されているカウンタを操作します。

構文

```
event_register_counter [tag ?] name ? entry_op gt|ge|eq|ne|lt|le entry_val ?
exit_op gt|ge|eq|ne|lt|le exit_val ? [queue_priority low|normal|high|last]
[maxrun ?] [nice 0|1]
```

引数

tag	(任意) Tcl スクリプト内で複数のイベント文をサポートするため、Tcl コマンド拡張のトリガーとともに使用できるタグを指定する文字列。
name	(必須) カウンタの名前。
entry_op	(必須) 現在のカウンタの値を開始値と比較するために使用される開始比較演算子。真の場合、イベントが発生し、終了基準を満たすまでイベントモニタリングがディセーブルにされます。

entry_val	(必須) カウンタ イベントを発生させる必要があるかどうかを判断するために、現在のカウンタの値を比較する必要がある値。
exit_op	(必須) 現在のカウンタの値を終了値と比較するために使用される終了比較演算子。真の場合、このイベントのイベントモニターリングが再度イネーブルにされます。
exit_val	(必須) 終了基準を満たすかどうかを判断するために、現在のカウンタの値を比較する必要がある値。
queue_priority	<p>(任意) 次のような、スクリプトがキューに入れられる優先度レベル。</p> <ul style="list-style-type: none"> • queue_priority low : 3つの優先度レベルの最も低いレベルでスクリプトがキューに入れられるよう指定します。 • queue_priority normal : low プライオリティよりも高く、high プライオリティよりも低い優先度レベルでスクリプトがキューに入れられるよう指定します。 • queue_priority high : 3つの優先度レベルの最も高いレベルで、スクリプトがキューに入れられるよう指定します。 • queue_priority last : 最も低い優先度レベルでスクリプトがキューに入れられるよう指定します。 <p>「queue_priority_last」引数が設定された状態で複数のスクリプトが登録されている場合、これらのスクリプトは、イベントのパブリッシュ順に実行されます。</p> <p>(注) queue_priority 引数によって、登録されているスクリプトの実行優先度ではなく、キューイングの優先度が指定されます。</p> <p>この引数が指定されない場合、デフォルトのキューイング プライオリティは normal です。</p>
maxrun	(任意) スクリプトの最大ランタイム (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)。この引数が指定されない場合、デフォルトの 20 秒ランタイム制限が使用されます。
nice	(任意) ポリシーの実行時間優先度の設定。nice 引数が 1 に設定されている場合、ポリシーは、デフォルトの優先度よりも低い実行時間優先度で実行されません。デフォルト値は 0 です

結果文字列

なし

_cerno を設定

なし

Event_reqinfo

```
"event_id %u event_type %u event_type_string {%s} %u event_pub_sec %u event_pub_msec %u"
"name {%s}"
```

イベントタイプ	説明
event_id	パブリッシュされた該当イベントの ID を示す一意の番号。同一のイベントで複数のポリシーを実行可能であり、その場合、各ポリシーは同一の event_id を保持します。
event_type	イベントのタイプ。
event_type_string	このイベントタイプのイベントの名前を表す ASCII 文字列。
event_pub_sec event_pub_msec	EEM に対してイベントがパブリッシュされた、秒単位およびミリ秒単位の時間。
name	カウンタ名。

event_register_gold

Generic Online Diagnostic (GOLD) 障害イベントの登録を行います。この Tcl コマンド拡張を使用すると、指定されたカードおよびサブカードの Generic Online Diagnostic (GOLD) 障害イベントに基づいて、ポリシーが実行されます。

構文

```
event_register_gold card all|card_number
[subcard all|subcard_number]
[new_failure TRUE|FALSE]
[severity_major TRUE]
[severity_minor TRUE]
[severity_normal TRUE]
[action_notify TRUE|FALSE]
[testing_type [bootup|ondemand|schedule|monitoring]]
[test_name [testname]]
[test_id [testnumber]]
[consecutive_failure consecutive_failure_number]
[platform_action [action_flag]]
[maxrun ?]
[queue_priority low|normal|high|last]
[nice 0|1]
```

引数

card	<p>(必須) すべてのカードまたは1つのカードがモニターされるよう指定します。</p> <ul style="list-style-type: none"> • card all : すべてのカードを監視対象に指定します。これはデフォルトです。 • card-number : card-number の番号によって指定されたカードを監視対象に指定します。 <p>event_register_gold Tcl コマンド拡張を完了させるには、この引数を指定する必要があります。</p>
subcard	<p>(任意) 1つまたは複数のサブカードがモニターされるよう指定します。</p> <ul style="list-style-type: none"> • subcard all : すべてのサブカードを監視対象に指定します。 • subcard-number : subcard-number の番号によって指定されたサブカードを監視対象に指定します。 <p>この引数が指定されない場合、すべてのサブカードがデフォルトでモニターされます。</p>
new_failure	<p>(任意) GOLD からの新しいテスト障害情報に基づいて、イベント基準を指定します。</p> <ul style="list-style-type: none"> • new_failure TRUE : GOLD からの新しいテスト障害のイベント基準が真であると指定します。 • new_failure FALSE : GOLD からの新しいテスト障害のイベント基準が偽であると指定します。 <p>この引数が指定されない場合、GOLD からの新しいテスト障害情報は、イベント基準で考慮されません。</p>
severity_major	<p>(任意) 診断結果のイベント基準が、GOLD からの診断 (メジャーエラー) と合致するよう指定します。</p>
severity_minor	<p>(任意) 診断結果のイベント基準が、GOLD からの診断 (マイナーエラー) と合致するよう指定します。</p>
severity_normal	<p>(任意) 診断結果のイベント基準が、GOLD からの診断 (通常) と合致するよう指定します。これはデフォルトです。</p>

action_notify	<p>(任意) GOLD からのアクション通知情報に基づいて、イベント基準を指定します。</p> <ul style="list-style-type: none"> • action_notify TRUE : GOLD からのアクション通知のイベント基準が真であると指定します。 • action_notify FALSE : GOLD からのアクション通知のイベント基準が偽であると指定します。 <p>この引数が指定されない場合、GOLD からのアクション通知情報は、イベント基準で考慮されません。</p>
testing_type	<p>(任意) GOLD からの診断のテストタイプに基づいて、イベント基準を指定します。</p> <ul style="list-style-type: none"> • testing_type bootup : システム ブート時に実行される診断テストを指定します。 • testing_type ondemand : カードがオンライン後に CLI から実行される診断テストを指定します。 • testing_type schedule : スケジュールされる診断テストを指定します。 • testing_type monitoring : システムの状態を監視するためにバックグラウンドで定期的に行われる診断テストを指定します。 <p>この引数が指定されない場合、GOLD からのテストタイプ情報は、イベント基準で考慮されず、ポリシーは、すべての診断テストタイプに適用されます。</p>
test_name	<p>(任意) 名前 test-name でのテストに基づいて、イベント基準を指定します。</p> <ul style="list-style-type: none"> • test_name test-name : 名前 test-name でのテストに基づいて、イベント基準を指定します。 <p>この引数が指定されない場合、GOLD からのテスト名情報は、イベント基準で考慮されません。</p>

test_id	<p>(任意) テスト ID に基づいて、イベント基準を指定します。</p> <ul style="list-style-type: none"> • test_id test-id : ID 番号 test-id のテストに基づいてイベント基準を指定します。test-id の最大値は 65535 です。 <p>(注) テスト ID は、異なるラインカード上での同じテストについて、異なる可能性があるため、通常は、代わりに test_name キーワードを使用する必要があります。テスト ID が指定され、指定されたテスト名と矛盾する場合、テスト名によって、テスト ID が上書きされます。</p> <p>この引数が指定されない場合、GOLD からのテスト ID 情報は、イベント基準で考慮されません。</p>
consecutive_failure	<p>(任意) GOLD からの連続テスト障害情報に基づいて、イベント基準を指定します。</p> <ul style="list-style-type: none"> • consecutive_failure consecutive-failure-number : イベント障害が、consecutive-failure-number 連続テスト障害の発生に基づくよう、指定します。 <p>この引数が指定されない場合、GOLD からの連続テスト障害情報は、イベント基準で考慮されません。</p>
platform_action	<p>(任意) すべてのイベント基準が一致した場合に、プラットフォームへのコールバックが必要かどうかを指定します。コールバックが必要な場合、プラットフォームでは、指定されたレジストリを介してコールバック機能を登録する必要があります。</p> <ul style="list-style-type: none"> • platform_action action-flag-number : プラットフォームへのコールバックが必要な場合に、特定の情報がプラットフォーム特有の action-flag-number の値によって指定されるよう、指定します。action-flag-number の最大値は 65535 です。 <p>(注) プラットフォームにより、フラグに基づいて行われる必要があるアクションが判断されます。</p> <p>この引数が指定されない場合、コールバックはありません。</p>
maxrun	<p>(任意) スクリプトの最大実行時間を指定します。</p> <ul style="list-style-type: none"> • maxrun max-run-time-number : スクリプトの最大実行時間を、max-run-time-number 秒と指定します。max-run-time-number の最大値は 4294967295 秒です。 <p>この引数が指定されない場合、デフォルトの実行時間は 20 秒です。</p>

queue_priority	<p>(任意) 次のような、スクリプトがキューに入れられる優先度レベル。</p> <ul style="list-style-type: none"> • queue_priority low : 3つの優先度レベルの最も低いレベルでスクリプトがキューに入れられるよう指定します。 • queue_priority normal : low プライオリティよりも高く、high プライオリティよりも低い優先度レベルでスクリプトがキューに入れられるよう指定します。 • queue_priority high : 3つの優先度レベルの最も高いレベルで、スクリプトがキューに入れられるよう指定します。 • queue_priority last : 最も低い優先度レベルでスクリプトがキューに入れられるよう指定します。 <p>「queue_priority_last」引数が設定された状態で複数のスクリプトが登録されている場合、これらのスクリプトは、イベントのパブリッシュ順に実行されます。</p> <p>(注) queue_priority 引数によって、登録されているスクリプトの実行優先度ではなく、キューイングの優先度が指定されます。</p> <p>この引数が指定されない場合、デフォルトのキューイング プライオリティは normal です。</p>
nice	<p>(任意) 次のような、ポリシー実行時間のプライオリティ設定。</p> <ul style="list-style-type: none"> • nice 0 : ポリシーがデフォルトの実行時間優先度レベルで実行されるよう指定します。 • nice 1 : ポリシーがデフォルト優先度レベルよりも低い実行時間優先度で実行されるよう指定します。 <p>この引数が指定されない場合、デフォルトの実行時間プライオリティが使用されます。</p>

結果文字列

なし

_cerno を設定

なし

Event_reqinfo

```
"event_id %u event_type %u event_type_string {%s} %u card %u sub_card %u"
"event_severity {%s} event_pub_sec %u event_pub_msec %u overall_result %u"
"new_failure {%s} action_notify {%s} tt %u tc %u bl %u ci %u pc %u cn {%s}"
"sn {%s} tn# {%s} ta# %s ec# {%s} rc# %u lf# {%s} tf# %u cf# %u tr# {%s}"
"tr#p# {%s} tr#d# {%s}"
```

イベントタイプ	説明
action_notify	TRUE または FALSE の、GOLD イベントでのアクション通知情報。
bl	起動診断レベル、次のいずれかの値である。 <ul style="list-style-type: none"> • 0 : 完全診断 • 1 : 最小診断 • 2 : バイパス診断
card	GOLD イベントのカード情報。
cf testnum	連続的な障害。 <i>testnum</i> はテスト番号。たとえば、 cf3 は、テスト 3 の連続的な障害の EEM 組み込み環境変数です。
ci	カードインデックス。
cn	カードの名前。
ec testnum	テストエラーコード。 <i>testnum</i> はテスト番号。たとえば、 ec3 は、テスト 3 のエラーコードの EEM 組み込み環境変数です。
event_id	パブリッシュされた該当イベントの ID を示す一意の番号。同一のイベントで複数のポリシーを実行可能であり、その場合、各ポリシーは同一の event_id を保持します。
event_pub_msec event_pub_sec	イベントが EEM にパブリッシュされたときの、ミリ秒単位および秒単位の時間。
event_severity	GOLD イベントの重大度で、次のいずれかの値を指定できます。 <ul style="list-style-type: none"> • normal • minor • major
event_type	イベントのタイプ。
event_type_string	このイベントタイプのイベントの名前を表す ASCII 文字列。
lf testnum	最終障害時間。 <i>testnum</i> はテスト番号。たとえば、 lf3 は、テスト 3 の最終障害時間の EEM 組み込み環境変数です。 タイムスタンプの形式は <i>mmm dd yyyy hh:mm:ss</i> です。例 : Mar 11 1960 08:47:00。
new_failure	GOLD イベントフラグの新しいテスト障害情報 (False または True) 。

イベントタイプ	説明
overall_result	総合診断結果、次のいずれかの値である。 <ul style="list-style-type: none"> • 0 : OK • 3 : マイナー エラー • 4 : メジャー エラー • 14 : 結果不明
pc	ポート数。
rc testnum	テスト総実行回数。 <i>testnum</i> はテスト番号。たとえば、 rc3 は、テスト 3 の総実行回数の EEM 組み込み変数です。
sn	カードシリアル番号。
sub_card	GOLD 障害イベントが検出されたサブカード。
ta testnum	テスト属性名。 <i>testnum</i> はテスト番号。たとえば、 ta3 は、テスト 3 の属性の EEM 組み込み環境変数です。
tc	テスト数。
tf testnum	合計障害回数。 <i>testnum</i> はテスト番号。たとえば、 tf3 は、テスト 3 の合計障害回数の EEM 組み込み変数です。
tn testnum	テストの名前。 <i>testnum</i> はテスト番号。たとえば、 tn3 は、テスト 3 の名前の EEM 組み込み環境変数です。
tr testnum	テストの結果。 <i>testnum</i> はテスト番号。たとえば、 tr6 はテスト 6 用の EEM 組み込み変数です。テスト 6 はポート単位のテストでも、デバイス単位のテストでもありません。 テスト結果は、次の値のうちのいずれかです。 <ul style="list-style-type: none"> • P : 診断結果 Pass • F : 診断結果 Fail • U : 診断結果 Unknown

イベントタイプ	説明
tr <i>testnum</i> d <i>devnum</i>	<p>デバイスごとのテスト結果。<i>testnum</i> はテスト番号で、<i>devnum</i> はデバイス番号です。たとえば、tr3d20 は、テスト 3、デバイス 20 のテスト結果の EEM 組み込み環境変数です。</p> <p>テスト結果は、次の値のうちのいずれかです。</p> <ul style="list-style-type: none"> • P : 診断結果 Pass • F : 診断結果 Fail • U : 診断結果 Unknown
tr <i>testnum</i> p <i>portnum</i>	<p>ポートごとのテスト結果。<i>testnum</i> はテスト番号で、<i>portnum</i> はデバイス番号です。たとえば、tr5p20 は、テスト 5、ポート 20 のテスト結果の EEM 組み込み環境変数です。</p> <p>テスト結果は、次の値のうちのいずれかです。</p> <ul style="list-style-type: none"> • P : 診断結果 Pass • F : 診断結果 Fail • U : 診断結果 Unknown
tt	<p>テストのタイプ。次のうちのいずれかです。</p> <ul style="list-style-type: none"> • 1 : 起動診断 • 2 : オンデマンド診断 • 3 : スケジュール診断 • 4 : モニターリング診断

event_register_identity

ID イベントの登録を行います。この Tcl コマンド拡張を使用して、AAA 認証または許可が成功または失敗したときや、ポート上での通常のユーザートラフィックのフローが許可された後にイベントを生成します。

構文

```
event_register_identity [tag ?] interface ?
[aaa-attribute ?]
[authc {all | fail | success}]
[authz {all | fail | success}]
[authz-complete]
[mac-address ?]
```

```
[queue_priority {normal | low | high | last}]
[maxrun ?] [nice {0 | 1}]
```

引数

tag	(任意) Tcl スクリプト内で複数のイベント文をサポートするため、Tcl コマンド拡張のトリガーとともに使用できるタグを指定する文字列。
interface	インターフェイス名と照合する正規表現パターン。
aaa-attribute	(任意) 特定の AAA 属性によってイベントをフィルタリングするために使用可能な正規表現。
authc	(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。
authz	(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。
authz-complete	(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。
mac-address	(任意) リモートデバイスの MAC アドレスによってイベントをフィルタリングするために使用可能な正規表現パターン。
maxrun	(任意) スクリプトの最大実行時間 (SSSSSSSSSS[MMM]形式で指定します。SSSSSSSSSS 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)。この引数が指定されない場合、デフォルトの 20 秒ランタイム制限が使用されます。

queue_priority	<p>(任意) 次のような、スクリプトがキューに入れられる優先度レベル。</p> <ul style="list-style-type: none"> • queue_priority low : 3つの優先度レベルの最も低いレベルでスクリプトがキューに入れられるよう指定します。 • queue_priority normal : low プライオリティよりも高く、high プライオリティよりも低い優先度レベルでスクリプトがキューに入れられるよう指定します。 • queue_priority high : 3つの優先度レベルの最も高いレベルで、スクリプトがキューに入れられるよう指定します。 • queue_priority last : 最も低い優先度レベルでスクリプトがキューに入れられるよう指定します。 <p>「queue_priority_last」引数が設定された状態で複数のスクリプトが登録されている場合、これらのスクリプトは、イベントのパブリッシュ順に実行されます。</p> <p>queue_priority 引数によって、登録されているスクリプトの実行優先度ではなく、キューイングの優先度が指定されます。</p> <p>この引数が指定されない場合、デフォルトのキューイング プライオリティは normal です。</p>
nice	<p>(任意) ポリシーの実行時間優先度の設定。nice 引数が 1 に設定されている場合、ポリシーは、デフォルトの優先度よりも低い実行時間優先度で実行されず。デフォルト値は 0 です</p>

結果文字列

なし

_cerno を設定

なし

EEM_EVENT_IDENTITY の Event_reqinfo

```
"event_id %u event_type %u event_type_string {%s} event_pub_sec %u event_pub_msec %u
event_severity %u identity_stage %u identity_status %u interface %u identity_mac %u
identity_<attribute> {%s}"
```

イベントタイプ	説明
event_id	パブリッシュされた該当イベントの ID を示す一意の番号。同一のイベントで複数のポリシーを実行可能であり、その場合、各ポリシーは同一の event_id を保持します。
event_type	イベントのタイプ。

イベントタイプ	説明
event_type_string	このイベントタイプのイベントの名前を表す ASCII 文字列。
event_pub_sec event_pub_msec	EEM に対してイベントがパブリッシュされた、秒単位およびミリ秒単位の時間。
event_severity	イベントの重大度。
identity_stage	authentication、authentication、または authorization-complete のステージのうちのいずれか。
identity_status	Success または fail_authc、fail_aaa_server、fail_no_response、fail_timeout、fail_authz のいずれかの障害タイプ。 authorization-complete は常に success になります。
interface	イベントのインターフェイス。
identity_mac	イベントのリモートデバイスの MAC アドレス。
identity_<attribute>	属性リストまたは値リスト内のその AAA 属性に対応する値に対する AAA 属性ごとの一連のダイナミック変数。

event_register_interface

インターフェイスカウンタイベントの登録を行います。この Tcl コマンド拡張を使用すると、指定されたインターフェイスカウンタが指定されたしきい値を超えたときに、イベントが生成されます。

構文

```
event_register_interface [tag ?] name ?
parameter ? entry_op gt|ge|eq|ne|lt|le
entry_val ? entry_val_is_increment TRUE|FALSE
entry_type value|increment|rate
[exit_comb or|and]
[exit_op gt|ge|eq|ne|lt|le]
[exit_val ?] [exit_val_is_increment TRUE|FALSE]
[exit_type value|increment|rate]
[exit_time ?] [poll_interval ?]
[average_factor ?] [queue_priority low|normal|high|last]
[maxrun ?] [nice 0|1]
```

引数

tag	(任意) Tcl スクリプト内で複数のイベント文をサポートするため、Tcl コマンド拡張のトリガーとともに使用できるタグを指定する文字列。
-----	---

name	(必須) イーサネット 0/0 など、モニターされるインターフェイスの名前。省略形と空白は使用できません。
parameter	<p>(必須) 比較されるカウンタの名前は、次のとおりです。</p> <ul style="list-style-type: none"> • input_errors : ラント、ジャイアント、バッファなし、CRC、フレーム、オーバーラン、および無視されたカウントが含まれます。他の入力関連のエラーも、入力エラー カウントが大きくなる場合があります。一部のデータグラムには、複数のエラーがあります。したがって、この合計は、列挙型入力エラー カウントとのバランスが取れない場合があります。 • input_errors_crc : 発信元 LAN ステーションまたは遠隔エンドデバイスによって生成される巡回冗長検査が、受信したデータから計算されるチェックサムに一致しません。 • input_errors_frame : 受信した不正確なパケット数。CRC エラーが発生し、8 ビットの非整数の数です。 • input_errors_overrun : 入力レートが、レシーバのデータ処理能力を超えたために、レシーバハードウェアによって、受信データをハードウェア バッファに渡せなかった回数。 • input_packets_dropped : 入力キューがいっぱいのため、廃棄されたパケット数。 • interface_resets : インターフェイスが完全にリセットされた回数。 • output_buffer_failures : 障害が発生したバッファ数およびスワップされたバッファ数。 • output_buffer_swappedout : DRAM にスワップされたパケット数。

parameter (続き)	<ul style="list-style-type: none"> • output_errors : 調べられているインターフェイスからのデータグラムの最終的な送信が妨害されたすべてのエラーの合計。一部のデータグラムには、複数のエラーがある場合があり、また、他のデータグラムには、特に表形式のカテゴリに当てはまらないエラーがある場合があるため、これは、列挙型出力エラーの合計とのバランスが取れないことがあります。 • output_errors_underrun : トランスミッタが、デバイスが処理可能な速度よりも高速だった回数。 • output_packets_dropped : 出力キューがいっぱいのため、廃棄されたパケット数。 • receive_broadcasts : インターフェイスによって受信されたブロードキャストパケットまたはマルチキャストパケットの数。 • receive_giants : メディアの最大パケットサイズを超過したために廃棄されたパケット数。 • receive_rate_bps : 1秒あたりのバイト単位でのインターフェイス受信レート。 • receive_rate_pps : 1秒あたりのパケット単位でのインターフェイス受信レート。 • receive_runts : メディアの最小パケットサイズよりも小さいために廃棄されたパケット数。 • receive_throttle : バッファまたはプロセッサが過負荷などの理由で、ポート上のレシーバが無効にされた回数。 • reliability : 5分間の幾何平均で計算される、255の分数でのインターフェイスの信頼性 (255/255 が 100% の信頼性)。 • rxload : 255の分数でのインターフェイスの受信レート (255/255 が 100%)。 • transmit_rate_bps : 1秒あたりのバイト単位でのインターフェイス送信レート。 • transmit_rate_pps : 1秒あたりのパケット単位でのインターフェイス送信レート。 • txload : 255の分数でのインターフェイスの送信レート (255/255 が 100%)。
entry_op	(必須) 現在のインターフェイスの値を開始値と比較するために使用される比較演算子。真の場合、イベントが発生し、終了基準を満たすまでイベント モニターリングがディセーブルにされます。
entry_val	(必須) イベントがトリガーされる値。

entry_val_is_increment	<p>(必須) TRUE の場合、entry_val フィールドは増分差異として処理され、現在のカウンタの値とイベントが最後に真であったとき（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）の値との差異と、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。FALSE の場合、entry_val フィールドが現在のカウンタの値に対して比較されます。</p> <p>(注) このキーワードは廃止されました。これを指定した場合、その構文は同等な entry-type キーワード構文に変換されます。</p>
entry-type	<p>entry-val 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。</p> <p>値は、entry-val 引数の実際の値として定義されます。</p> <p>増分では、entry-val フィールドは増分差異として使用され、entry-val は、現在のカウンタの値と、イベントが最後に真であったとき（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。</p> <p>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、average-factor の値に、poll-interval の値を乗じたものです。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 average-factor 値サンプルの平均は、変更のレートとして取得されます。</p>
exit_comb	<p>(任意) イベント トリガーの再準備に必要な終了条件テストの組み合わせを示すために使用されます。and 演算子が指定される場合、再準備のためには、終了値と終了時間テストの両方が真である必要があります。or 演算子が指定される場合、イベント モニターリングの再準備のためには、終了値または終了時間テストのいずれかが真である可能性があります。</p>
exit_op	<p>(任意) 現在のインターフェイスの値を終了値と比較するために使用される比較演算子。真の場合、このイベントのイベント モニターリングが再度イネーブルにされます。</p>
exit_val	<p>(任意) イベントが再度監視されるように再準備される値。</p>
exit_val_is_increment	<p>(任意) TRUE の場合、exit_val フィールドは増分差異として処理され、現在のカウンタの値と、イベントが最後に真であったときの値との差異と、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。FALSE の場合、exit_val フィールドが現在のカウンタの値に対して比較されます。</p> <p>(注) Cisco IOS Release 12.4(20)T では、このキーワードは廃止予定で、指定された場合、構文は同等の exit-type キーワード構文に変換されます。</p>

exit-type	<p>(任意) exit-val 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。</p> <p>値は、exit-val 引数の実際の値として定義されます。</p> <p>増分では、exit-val フィールドは増分差異として使用され、exit-val は、現在のカウンタの値と、イベントが最後に真であったとき（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。</p> <p>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、average-factor の値に、poll-interval の値を乗じたものです。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 average-factor 値サンプルの平均は、変更のレートとして取得されます。</p>
exit_time	<p>(任意) イベントが再度監視されるように再準備される時間 (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)。</p>
poll_interval	<p>(任意) サンプルが収集される頻度 (SSSSSSSSSS[.MMM] 形式で指定します。SSSSSSSSSS は、60 ~ 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)。ポーリング間隔の値には、1 秒よりも小さい値は使用できません。デフォルト値は 1 秒です。</p>
average-factor	<p>(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 1 から 64 の範囲の数。average-factor の値は、poll-interval の値を乗じた値で、ミリ秒単位で導き出される期間です。最少平均係数値は 1 です。</p>

queue_priority	<p>(任意) 次のような、スクリプトがキューに入れられる優先度レベル。</p> <ul style="list-style-type: none"> • queue_priority low : 3つの優先度レベルの最も低いレベルでスクリプトがキューに入れられるよう指定します。 • queue_priority normal : low プライオリティよりも高く、high プライオリティよりも低い優先度レベルでスクリプトがキューに入れられるよう指定します。 • queue_priority high : 3つの優先度レベルの最も高いレベルで、スクリプトがキューに入れられるよう指定します。 • queue_priority last : 最も低い優先度レベルでスクリプトがキューに入れられるよう指定します。 <p>「queue_priority_last」引数が設定された状態で複数のスクリプトが登録されている場合、これらのスクリプトは、イベントのパブリッシュ順に実行されます。</p> <p>(注) queue_priority 引数によって、登録されているスクリプトの実行優先度ではなく、キューイングの優先度が指定されます。</p> <p>この引数が指定されない場合、デフォルトのキューイング プライオリティは normal です。</p>
maxrun	<p>(任意) スクリプトの最大ランタイム (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)。この引数が指定されない場合、デフォルトの20秒ランタイム制限が使用されます。</p>
nice	<p>(任意) ポリシーの実行時間優先度の設定。nice 引数が1に設定されている場合、ポリシーは、デフォルトの優先度よりも低い実行時間優先度で実行されます。デフォルト値は0です</p>

結果文字列

なし

_cernno を設定

なし

Event_reqinfo

```
"event_id %u event_type %u event_type_string {%s} %u event_pub_sec %u event_pub_msec %u"
"event_severity {%s} name {%s} parameter {%s} value %d"
```

イベントタイプ	説明
event_id	パブリッシュされた該当イベントの ID を示す一意の番号。同一のイベントで複数のポリシーを実行可能であり、その場合、各ポリシーは同一の event_id を保持します。
event_type	イベントのタイプ。
event_type_string	このイベントタイプのイベントの名前を表す ASCII 文字列。
event_pub_sec event_pub_msec	EEM に対してイベントがパブリッシュされた、秒単位およびミリ秒単位の時間。
event_severity	インターフェイスイベントの重大度で、次のいずれかの値を指定できます。 <ul style="list-style-type: none"> • normal • minor • major
name	インターフェイスの名前。
parameter	パラメータの名前。
value	指定された entry_val_is_increment によって、トリガーされた最後のイベントに対する増加または減少の差異、または、監視されているパラメータの絶対値。

event_register_ioswdsysmon

IOSWDSysMon イベントの登録を行います。この Tcl コマンド拡張を使用すると、Cisco IOS タスクが指定された CPU 使用率またはメモリしきい値を超えたときに、イベントが生成されます。Cisco IOS タスクは、ネイティブ Cisco IOS の Cisco IOS プロセスと呼ばれます。

構文

```
event_register_ioswdsysmon [tag ?] [timewin ?] [sub12op and|or] [sub1 ?] [sub2 ?]
[queue_priority low|normal|high|last] [maxrun ?] [nice 0|1]
```

引数

tag	(任意) Tcl スクリプト内で複数のイベント文をサポートするため、Tcl コマンド拡張のトリガーとともに使用できるタグを指定する文字列。
-----	---

timewin	(任意) イベントが生成されるようにするために、すべてのサブイベントが発生する必要がある時間ウィンドウを定義します (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)。
sub12_op	(任意) サブイベント 1 とサブイベント 2 とを比較する組み合わせ演算子。
sub1	(任意) サブイベント 1 の指定。
sub2	(任意) サブイベント 2 の指定。
queue_priority	<p>(任意) 次のような、スクリプトがキューに入れられる優先度レベル。</p> <ul style="list-style-type: none"> • queue_priority low : 3 つの優先度レベルの最も低いレベルでスクリプトがキューに入れられるよう指定します。 • queue_priority normal : low プライオリティよりも高く、high プライオリティよりも低い優先度レベルでスクリプトがキューに入れられるよう指定します。 • queue_priority high : 3 つの優先度レベルの最も高いレベルで、スクリプトがキューに入れられるよう指定します。 • queue_priority last : 最も低い優先度レベルでスクリプトがキューに入れられるよう指定します。 <p>「queue_priority_last」引数が設定された状態で複数のスクリプトが登録されている場合、これらのスクリプトは、イベントのパブリッシュ順に実行されます。</p> <p>(注) queue_priority 引数によって、登録されているスクリプトの実行優先度ではなく、キューイングの優先度が指定されます。</p> <p>この引数が指定されない場合、デフォルトのキューイング プライオリティは normal です。</p>
maxrun	(任意) スクリプトの最大ランタイム (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)。この引数が指定されない場合、デフォルトの 20 秒ランタイム制限が使用されます。
nice	(任意) ポリシーの実行時間優先度の設定。nice 引数が 1 に設定されている場合、ポリシーは、デフォルトの優先度よりも低い実行時間優先度で実行されます。デフォルト値は 0 です

サブイベントの構文

```
cpu_proc path ? taskname ? op gt|ge|eq|ne|lt|le val ? [period ?]
mem_proc path ? taskname ? op gt|ge|eq|ne|lt|le val ? [is_percent TRUE|FALSE] [period ?]
```

サブイベントの引数

cpu_proc	(必須) CPU 統計情報のサンプル収集の使用を指定します。
path	(必須) ソフトウェア モジュール方式イメージのみ。監視される Cisco IOS スケジューラが含まれる POSIX プロセスのパス名。たとえば、/sbin/cdp2.iosproc など。
taskname	(必須) 監視される Cisco IOS タスクの名前。
op	(必須) 収集される使用サンプルを指定値と比較するために使用される比較演算子。真の場合、このイベントが発生します。
val	(必須) 比較される値。
period	(任意) 収集されるサンプルの平均が計算される経過時間 (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)。この引数が指定されない場合は、最新のサンプルが使用されます。
mem_proc	(必須) メモリ統計情報のサンプル収集の使用を指定します。
is_percent	(任意) 指定値がパーセンテージかどうか。

結果文字列

なし

_cerno を設定

なし

Event_reqinfo

```
"event_id %u event_type %u event_type_string {%s} %u event_pub_sec %u event_pub_msec %u"
"num_subs %u"
```

イベントタイプ	説明
event_id	パブリッシュされた該当イベントの ID を示す一意の番号。同一のイベントで複数のポリシーを実行可能であり、その場合、各ポリシーは同一の event_id を保持します。
event_type	イベントのタイプ。
event_type_string	このイベントタイプのイベントの名前を表す ASCII 文字列。
event_pub_sec event_pub_msec	EEM に対してイベントがパブリッシュされた、秒単位およびミリ秒単位の時間。
num_subs	サブイベントの番号。

サブイベント情報文字列は、次のような、CPU_UTIL サブイベント用です。

```
"{type %s procname {%s} pid %u taskname {%s} taskid %u value %u sec %ld msec %ld}"
```

サブイベントタイプ	説明
type	サブイベントのタイプ。
procname	このサブイベントの POSIX プロセス名。
pid	このサブイベントの POSIX プロセス ID。
taskname	このサブイベントの Cisco IOS タスク名。
taskid	このサブイベントの Cisco IOS タスク ID。
value	測定された間隔での、実際の平均 CPU 使用率。
sec , msec	この測定間隔の経過時間。

サブイベント情報文字列は、次のような、MEM_UTIL サブイベント用です。

```
"{type %s procname {%s} pid %u taskname {%s} taskid %u is_percent %s value %u diff %d"
"sec %ld msec %ld}"
```

サブイベントタイプ	説明
type	サブイベントのタイプ。
procname	このサブイベントの POSIX プロセス名。
pid	このサブイベントの POSIX プロセス ID。
taskname	このサブイベントの Cisco IOS タスク名。
taskid	このサブイベントの Cisco IOS タスク ID。
is_percent	値がパーセント値かどうかによって、TRUE または FALSE。
value	この測定された間隔の KB 単位でのメモリ使用量の合計、または実際のメモリ使用率の平均。
diff	測定された間隔で最も古いサンプルと、最新のサンプルとの、パーセンテージでの違い。負の値は、減少を表します。
sec , msec	この測定間隔の経過時間。

event_register_ipsla

event ipsla コマンドによってトリガーされるイベントの登録を行います。この Tcl コマンド拡張を使用すると、IPSLA の応答がトリガーされるときに、イベントがパブリッシュされます。イベントの登録には、グループ ID または動作 ID が必要です。

構文

```
event_register_ipsla [tag ?] group_name ? operation_id ? [reaction_type ?]
[dest_ip_addr ?][queue_priority low|normal|high|last] [maxrun ?] [nice 0|1]
```

引数

tag	(任意) Tcl スクリプト内で複数のイベント文をサポートするため、Tcl コマンド拡張のトリガーとともに使用できるタグを指定する文字列。
group_name	(必須) IP SLA グループ名を指定します。
operation_id	(必須) IP SLA 動作 ID を指定します。番号は 1 から 2147483647 の範囲の整数です。

reaction_type	<p>(任意) 指定した IP SLA 動作に対する応答を指定します。</p> <p>IP SLA 反応のタイプ : 次のキーワードのいずれかを指定できます :</p> <p>connectionLoss、icpif、jitterAvg、jitterDSAvg、jitterSDAvg、maxOfNegativeDS、maxOfNegativeSD、maxOfPositiveDS、maxOfPositiveSD、mos、packetLateArrival、packetLossDS、packetLossSD、packetMIA、packetOutOfSequence、rtt、timeout または verifyError を指定できます。</p> <p>IP SLA の応答。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 • icpif • jitterAvg • jitterDSAvg • jitterSDAvg • maxOfNegativeDS • maxOfNegativeSD • maxOfPositiveDS • maxOfPositiveSD • mos • packetLateArrival • packetLossDS • packetLossSD • packetMIA • packetOutOfSequence • rtt • timeout • verifyError
dest_ip_address	<p>(任意) IP SLA イベントが監視される宛先ポートの宛先 IP アドレスを指定します。</p>

queue_priority	<p>(任意) 次のような、スクリプトがキューに入れられる優先度レベル。</p> <ul style="list-style-type: none"> • queue_priority low : 3つの優先度レベルの最も低いレベルでスクリプトがキューに入れられるよう指定します。 • queue_priority normal : low プライオリティよりも高く、high プライオリティよりも低い優先度レベルでスクリプトがキューに入れられるよう指定します。 • queue_priority high : 3つの優先度レベルの最も高いレベルで、スクリプトがキューに入れられるよう指定します。 • queue_priority last : 最も低い優先度レベルでスクリプトがキューに入れられるよう指定します。 <p>「queue_priority_last」引数が設定された状態で複数のスクリプトが登録されている場合、これらのスクリプトは、イベントのパブリッシュ順に実行されます。</p> <p>(注) queue_priority 引数によって、登録されているスクリプトの実行優先度ではなく、キューイングの優先度が指定されます。</p> <p>この引数が指定されない場合、デフォルトのキューイング プライオリティは normal です。</p>
maxrun	<p>(任意) スクリプトの最大実行時間 (SSSSSSSSSS[MMM]形式で指定します。SSSSSSSSSS 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)。この引数が指定されない場合、デフォルトの 20 秒ランタイム制限が使用されます。</p>
nice	<p>(任意) ポリシーの実行時間優先度の設定。nice 引数が 1 に設定されている場合、ポリシーは、デフォルトの優先度よりも低い実行時間優先度で実行されます。デフォルト値は 0 です</p>

結果文字列

なし

_cerno を設定

なし

Event_reqinfo

```
"event_ID %u event_type %u event_pub_sec %u event_pub_msec %u event_severity %u" "group_name
%u operation_id %u condition %u reaction_type %u dest_ip_addr %u" "threshold_rising %u
threshold_falling%u measured_threshold_value %u" "threshold_count1 %u threshold count2 %u"
```

Event Type	Description
event_id	パブリッシュされた該当イベントの ID を示す一意の番号。同一のイベントで複数のポリシーを実行可能であり、その場合、各ポリシーは同一の event_id を保持します。
event_type	フローの作成、アップデート、および削除を監視するイベントのタイプ。
event_type_string	このイベントタイプのイベントの名前を表す ASCII 文字列。
event_pub_sec event_pub_msec	EEM に対してイベントがパブリッシュされた、秒単位およびミリ秒単位の時間。
event_severity	イベントの重大度。
group_name	IPSLA グループの名前。
operation_id	IPSLA 動作 ID。
condition	IPSLA の条件で、次の 1 つを使用できます。 <ul style="list-style-type: none"> • cleared • occurred
reaction_type	IPSLA 応答タイプ。
dest_ip_address	IPSLA 宛先 IP アドレス。
threshold rising	IPSLA で設定されている上昇しきい値。
threshold falling	IPSLA で設定されている下限しきい値。
measured_threshold_value	IPSLA 動作の測定されたしきい値。
threshold_count1	しきい値 type1 の引数に対応します。
threshold_count2	しきい値 type2 の引数に対応します。

event_register_mat

MAT イベントの登録を行います。この Tcl コマンド拡張を使用して、mac-address-table で MAC アドレスが学習されたときにイベントを生成します。

構文

```
event_register_identity [tag ?] interface ?
```

```
[mac-address ?]
[type {add | delete}]
[hold-down ?]
[maxrun ?]
```

引数

tag	(任意) Tcl スクリプト内で複数のイベント文をサポートするため、Tcl コマンド拡張のトリガーとともに使用できるタグを指定する文字列。
interface	インターフェイス名と照合する正規表現パターン。
mac-address	インターフェイス パラメータを指定していない場合には必須です。リモートデバイスの MAC アドレスによってイベントをフィルタリングするために使用可能な正規表現パターン。
type	(任意) 追加または削除の mac-address-table イベントタイプに基づいてフィルタリングします。指定しなかった場合、イベントをトリガーするかどうかの判断にそのイベントタイプが使用されません。
hold-down	(任意) mac-address-table イベントが着信した場合、ポリシーを処理する前にそのイベントを 1 ~ 4294967295 秒間待機させるようにホールドダウン タイマーを設定できます。このタイマーを設定しなかった場合は、ポリシーの処理は遅延しません。
maxrun	(任意) スクリプトの最大実行時間 (SSSSSSSSSS[.MMM] 形式で指定します。SSSSSSSSSS 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)。この引数が指定されない場合、デフォルトの 20 秒ランタイム制限が使用されます。

結果文字列

なし

_cerno を設定

なし

EEM_EVENT_MAT の Event_reqinfo

```
"event_id %u event_type %u event_type_string {%s} event_pub_sec %u event_pub_msec %u
event_severity %u notification %u intf_name %u mac_address {%s}"
```

イベントタイプ	説明
event_id	パブリッシュされた該当イベントの ID を示す一意の番号。同一のイベントで複数のポリシーを実行可能であり、その場合、各ポリシーは同一の event_id を保持します。
event_type	イベントのタイプ。

イベントタイプ	説明
event_type_string	このイベントタイプのイベントの名前を表す ASCII 文字列。
event_pub_sec event_pub_msec	EEM に対してイベントがパブリッシュされた、秒単位およびミリ秒単位の時間。
event_severity	イベントの重大度。
notification	通知のタイプ：追加または削除。
intf_name	アドレス テーブル エントリのインターフェイス名。
mac_address	アドレス テーブルのエントリの MAC アドレス。

event_register_neighbor_discovery

ネイバー探索イベントの登録この Tcl コマンド拡張を使用して、Cisco Discovery Protocol (CDP) または Link Layer Discovery Protocol (LLDP) のキャッシュ エントリまたはインターフェイス リンク ステータスが変った場合にイベントを生成します。

構文

```
event_register_neighbor_discovery [tag ?] interface ?
[cdp {add | update | delete | all}]
[lldp {add | update | delete | all}]
[link-event]
[line-event]
[queue_priority {normal | low | high | last}]
[maxrun ?] [nice {0 | 1}]
```

引数

tag	(任意) Tcl スクリプト内で複数のイベント文をサポートするため、Tcl コマンド拡張のトリガーとともに使用できるタグを指定する文字列。
interface	インターフェイス名と照合する正規表現パターン。

cdp	<p>CDP のマッチング 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を指定する必要があります。</p> <ul style="list-style-type: none"> • 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 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	<p>LLDP のマッチング 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を指定する必要があります。</p> <ul style="list-style-type: none"> • 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 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	<p>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。</p>
link-event	<p>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。</p>

queue_priority	<p>(任意) 次のような、スクリプトがキューに入れられる優先度レベル。</p> <ul style="list-style-type: none"> • queue_priority low : 3つの優先度レベルの最も低いレベルでスクリプトがキューに入れられるよう指定します。 • queue_priority normal : low プライオリティよりも高く、high プライオリティよりも低い優先度レベルでスクリプトがキューに入れられるよう指定します。 • queue_priority high : 3つの優先度レベルの最も高いレベルで、スクリプトがキューに入れられるよう指定します。 • queue_priority last : 最も低い優先度レベルでスクリプトがキューに入れられるよう指定します。 <p>「queue_priority_last」引数が設定された状態で複数のスクリプトが登録されている場合、これらのスクリプトは、イベントのパブリッシュ順に実行されます。</p> <p>queue_priority引数によって、登録されているスクリプトの実行優先度ではなく、キューイングの優先度が指定されます。</p> <p>この引数が指定されない場合、デフォルトのキューイングプライオリティは normal です。</p>
maxrun	<p>(任意) スクリプトの最大実行時間 (SSSSSSSSSS[.MMM]形式で指定します。SSSSSSSSSS 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)。この引数が指定されない場合、デフォルトの 20 秒ランタイム制限が使用されます。</p>
nice	<p>(任意) ポリシーの実行時間優先度の設定。nice 引数が 1 に設定されている場合、ポリシーは、デフォルトの優先度よりも低い実行時間優先度で実行されます。デフォルト値は 0 です</p>

結果文字列

なし

_cerno を設定

なし

EEM_EVENT_NEIGHBOR_DISCOVERY の Event_reqinfo

```
"event_id %u event_type %u event_type_string {%s} event_pub_sec %u event_pub_msec %u
event_severity %u nd_notification {%s}"
```

イベントタイプ	説明
event_id	パブリッシュされた該当イベントのIDを示す一意の番号。同一のイベントで複数のポリシーを実行可能であり、その場合、各ポリシーは同一の event_id を保持します。
event_type	イベントのタイプ。
event_type_string	このイベントタイプのイベントの名前を表す ASCII 文字列。
event_pub_sec event_pub_msec	EEM に対してイベントがパブリッシュされた、秒単位およびミリ秒単位の時間。
event_severity	イベントの重大度。
共通の Event_Reqinfo	
nd_notification	通知のタイプ : cdp-add、cdp-update、cdp-delete、lldp-add、lldp-update、lldp-delete、link、line。
nd_intf_linkstatus	現在のインターフェイスリンクのステータス。up または down。
nd_intf_linestatus	現在のインターフェイス回線のステータス。down、goingdown、init、testing、up、reset、admindown、deleted。
nd_local_intf_name	イベントのローカルインターフェイスの名前。
nd_short_local_intf_name	イベントのローカルインターフェイスの短い名前。
nd_port_id	CDP プロトコルまたは LLDP プロトコルのいずれかで識別されたポート ID。これは、リンクまたは回線プロトコルのイベントには設定されません。
CDP-specific Event_reqinfo	
nd_protocol	イベントをトリガーしたプロトコルを識別します。CDP の場合は常に cdp に設定されます。
nd_proto_notif	イベント、追加、更新、または削除をトリガーしたプロトコルイベントのタイプを特定します。
nd_proto_new_entry	1 に設定されている場合、キャッシュエントリは新規であるため、イベントはトリガーされており、それ以外の場合は 0 に設定されます。
nd_cdp_entry_name	CDP テーブル内の CDP キャッシュエントリの名前。

イベントタイプ	説明
nd_cdp_hold_time	CDP キャッシュ エントリが期限切れになり、CDP テーブルから削除されるまでの残り時間。この時間は、CDP ネイバーからの更新によって最大値にリセットされます。新しいエントリの場合は通常、0 に設定されます。
nd_cdp_mgmt_domain	CDP VTP 管理ドメイン。
nd_cdp_platform	リモート デバイスによって報告されるプラットフォームの名前。
nd_cdp_version	リモートデバイスで実行されているコードのバージョン。
nd_cdp_capabilities_string	文字列形式の CDP capabilities フィールドのコンテンツ：ルータ、トランスブリッジ、ソースルートブリッジ、スイッチ、ホスト、IGMP、リピータ、電話、リモートで管理されているデバイス、CVTA 電話ポート、2ポート MAC リレー、または、カンマで区切ったこれらの組み合わせ。
nd_cdp_capabilities_bits	先頭に 0x が付加された 16 進数内の CDP 機能ビット。
nd_cdp_capabilities_bit_[0-31]	capabilities フィールドのそのビットが設定されている場合は YES に、設定されていない場合は NOT に設定される一連の値。
LLDP-specific Event_reqinfo	
nd_protocol	イベントをトリガーしたプロトコルを識別します。LLDP の場合は常に lldp に設定されます。
nd_proto_notif	イベント、追加、更新、または削除をトリガーしたプロトコル イベントのタイプを特定します。
nd_proto_new_entry	1 に設定されている場合、キャッシュエントリは新規であるため、イベントはトリガーされており、それ以外の場合は 0 に設定されます。
nd_lldp_chassis_id	LLDP キャッシュ エントリからの chassis id フィールド。
nd_lldp_system_name	LLDP キャッシュ エントリからのシステム名。
nd_lldp_system_description	LLDP キャッシュ エントリからの system description フィールド。
nd_lldp_ttl	LLDP キャッシュ エントリからの LLDP time to live フィールド。
nd_lldp_port_description	LLDP キャッシュ エントリからの port description フィールド。

イベントタイプ	説明
nd_lldp_system_capabilities_string	LLDP キャッシュ エントリからの LLDP system capabilities フィールド。この文字列には、O、P、B、W、R、T、C、S、またはこれらの任意の組み合わせをカンマで区切って含めることができます。
nd_lldp_enabled_capabilities_string	LLDP キャッシュ エントリからの LLDP enabled system capabilities フィールド。この文字列には、O、P、B、W、R、T、C、S、またはこれらの任意の組み合わせをカンマで区切って含めることができます。
nd_lldp_system_capabilities_bits	LLDP キャッシュ エントリからの LLDP system capabilities bits フィールド。先頭に 0x が付加された 16 進数です。
nd_lldp_enabled_capabilities_bits	LLDP キャッシュ エントリからの LLDP enabled capabilities bits フィールド。先頭に 0x が付加された 16 進数です。
nd_lldp_capabilities_bits	LLDP キャッシュ エントリからの LLDP capabilities bits フィールド。先頭に 0x が付加された 16 進数です。
nd_lldp_capabilities_bit_[0-31]	capabilities フィールドのそのビットが設定されている場合は YES に、設定されていない場合は NOT に設定される一連の値。

event_register_nf

NetFlow イベントが **event nf** コマンドによってトリガーされるときイベントの登録を行います。この Tcl コマンド拡張を使用すると、NetFlow の応答がトリガーされるときに、イベントがパブリッシュされます。

構文

```
event_register_nf [tag ?] monitor_name ? event_type create|update|delete
exit_event_type create|update|delete event1-event4 ? [maxrun ?] [nice 0|1]
```

引数

tag	(任意) Tcl スクリプト内で複数のイベント文をサポートするため、Tcl コマンド拡張のトリガーとともに使用できるタグを指定する文字列。
monitor_name	(必須) NetFlow モニターの名前。
event_type	(必須) フローの作成、アップデート、および削除を監視するイベントのタイプ。

exit_event_type	(必須) 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)。
event1- event4	(必須) 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 、および event4 です。 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。
maxrun	(任意) スクリプトの最大ランタイム (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)。この引数が指定されない場合、デフォルトの 20 秒ランタイム制限が使用されます。
nice	(任意) ポリシーの実行時間優先度の設定。nice 引数が 1 に設定されている場合、ポリシーは、デフォルトの優先度よりも低い実行時間優先度で実行されます。デフォルト値は 0 です

サブイベントの構文

```
field ? rate_interval ? event1 only entry_value ? entry_op eq|ge|gt|le|lt|wc
[exit_value ?] [exit_op eq|ge|gt|le|lt|wc] [exit_rate_interval ? event1 only]
```

サブイベントの引数

field	(必須) 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。 <ul style="list-style-type: none"> • counter {bytes packets} : カウンタフィールドを指定します。 • datalink {dot1q mac} : データリンク (レイヤ 2) フィールドを指定します。 • flow {direction sampler} : フロー識別フィールドを指定します。 • interface {input output} : インターフェイスフィールドを指定します。 • ipv4 field-type : IPv4 フィールドを指定します。 • ipv6 field-type : IPv6 フィールド • routing routing-attribute -- : ルーティング属性を指定します。 • timestamp sysuptime {first last}-- : タイムスタンプフィールドを指定します。 • transport field-type : トランスポートレイヤフィールドを指定します。
rate_interval	(必須) レートの計算に使用されるレート間隔値を秒単位で指定します。このフィールドは、event1 でのみ有効です。

entry_value	(必須) フィールドまたはレート値を指定します。
entry_op	(必須) フィールド演算子を指定します。 次の比較演算子の値が有効です。 <ul style="list-style-type: none"> • eq : 次の値と等しい • ge : 次の値以上 • gt : 次の値より大きい • le : 次の値以下 • lt : 次の値より小さい • wc : ワイルドカード
exit_value	(任意) イベントが再度監視されるように再準備される値。
exit_op	(任意) 現在のイベントフィールドまたはレート値を終了値と比較するために使用される比較演算子。真の場合、このイベントのイベント監視が再度イネーブルにされます。 次の比較演算子の値が有効です。 <ul style="list-style-type: none"> • eq : 次の値と等しい • ge : 次の値以上 • gt : 次の値より大きい • le : 次の値以下 • lt : 次の値より小さい • wc : ワイルドカード
exit_rate_interval	(任意) 終了レート値の計算に使用される終了レート間隔値を秒単位で指定します。このフィールドは、event1 でのみ有効です。

結果文字列

なし

_cerrno を設定

なし

Event_reqinfo

```
"event_ID %u event_type %u event_type_string {%s} event_pub_sec %u event_pub_msec %u
event_severity %u monitor_name %u event1-event4_field %u event1-event4_value
```

イベント タイプ	説明
event_id	パブリッシュされた該当イベントのIDを示す一意の番号。同一のイベントで複数のポリシーを実行可能であり、その場合、各ポリシーは同一の event_id を保持します。
event_type	フローの作成、アップデート、および削除を監視するイベントのタイプ。
event_type_string	このイベントタイプのイベントの名前を表す ASCII 文字列。
event_pub_sec event_pub_msec	EEM に対してイベントがパブリッシュされた、秒単位およびミリ秒単位の時間。
event_severity	NetFlow イベントの重大度。
montior_name	NetFlow モニターの名前。
event1-event4_field	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 、および event4 です。
event1-event4_value	監視するイベント値とその属性を指定します。有効な値は event1 、 event2 、 event3 、および event4 です。

event_register_none

event manager run コマンドによってトリガーされるイベントの登録を行います。これらのイベントは、このイベントをスクリーニングする None イベントディテクタによって処理されます。

構文

```
event_register_none [tag ?] [sync {yes|no}] [default ?] [queue_priority
low|normal|high|last] [maxrun ?] [nice 0|1]
```

引数

tag	(任意) Tcl スクリプト内で複数のイベント文をサポートするため、Tcl コマンド拡張のトリガーとともに使用できるタグを指定する文字列。
sync	(任意) このキーワードを完了するには、「yes」または「no」が必要です。 <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。

デフォルト	<p>(任意) CLI イベント デテクタがポリシーの終了を待つ時間 (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)。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。</p>
queue_priority	<p>(任意) 次のような、スクリプトがキューに入れられる優先度レベル。</p> <ul style="list-style-type: none"> • queue_priority low : 3 つの優先度レベルの最も低いレベルでスクリプトがキューに入れられるよう指定します。 • queue_priority normal : low プライオリティよりも高く、high プライオリティよりも低い優先度レベルでスクリプトがキューに入れられるよう指定します。 • queue_priority high : 3 つの優先度レベルの最も高いレベルで、スクリプトがキューに入れられるよう指定します。 • queue_priority last : 最も低い優先度レベルでスクリプトがキューに入れられるよう指定します。 <p>「queue_priority_last」引数が設定された状態で複数のスクリプトが登録されている場合、これらのスクリプトは、イベントのパブリッシュ順に実行されます。</p> <p>(注) queue_priority 引数によって、登録されているスクリプトの実行優先度ではなく、キューイングの優先度が指定されます。</p> <p>この引数が指定されない場合、デフォルトのキューイング プライオリティは normal です。</p>
maxrun	<p>(任意) スクリプトの最大ランタイム (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)。この引数が指定されない場合、デフォルトの 20 秒ランタイム制限が使用されます。</p>
nice	<p>(任意) ポリシーの実行時間優先度の設定。nice 引数が 1 に設定されている場合、ポリシーは、デフォルトの優先度よりも低い実行時間優先度で実行されず。デフォルト値は 0 です</p>

結果文字列

なし

_cerno を設定

なし

Event_reqinfo

```
"event_id %u event_type %u event_type_string {%s} event_pub_sec %u event_pub_msec %u
event_severity %u arg %u"
```

イベント タイプ	説明
event_id	パブリッシュされた該当イベントの ID を示す一意の番号。同一のイベントで複数のポリシーを実行可能であり、その場合、各ポリシーは同一の event_id を保持します。
event_type	イベントのタイプ。
event_type_string	このイベント タイプのイベントの名前を表す ASCII 文字列。
event_pub_sec event_pub_msec	EEM に対してイベントがパブリッシュされた、秒単位およびミリ秒単位の時間。
event_severity	イベントの重大度。
argc arg1 arg2 arg3 arg4 arg6 arg7 arg8 arg9 arg10 arg11 arg12 arg13 arg14 arg15	Extensible Markup Language (XML) Simple Object Access Protocol (SOAP) コマンドからスクリプトに渡されるパラメータ。

event_register_oir

活性挿抜 (OIR) イベントの登録を行います。この Tcl コマンド拡張を使用すると、ハードウェアカード OIR イベントの発生時に発生するイベントに基づいて、ポリシーが実行されます。

これらのイベントは、このイベントをスクリーニングする OIR イベント デテクタによって処理されます。

構文

```
event_register_oir [tag ?] [queue_priority low|normal|high|last] [maxrun ?] [nice 0|1]
```

引数

tag	(任意) Tcl スクリプト内で複数のイベント文をサポートするため、Tcl コマンド拡張のトリガーとともに使用できるタグを指定する文字列。
queue_priority	<p>(任意) 次のような、スクリプトがキューに入れられる優先度レベル。</p> <ul style="list-style-type: none"> • queue_priority low : 3つの優先度レベルの最も低いレベルでスクリプトがキューに入れられるよう指定します。 • queue_priority normal : low プライオリティよりも高く、high プライオリティよりも低い優先度レベルでスクリプトがキューに入れられるよう指定します。 • queue_priority high : 3つの優先度レベルの最も高いレベルで、スクリプトがキューに入れられるよう指定します。 • queue_priority last : 最も低い優先度レベルでスクリプトがキューに入れられるよう指定します。 <p>「queue_priority_last」引数が設定された状態で複数のスクリプトが登録されている場合、これらのスクリプトは、イベントのバブリッシュ順に実行されます。</p> <p>(注) queue_priority 引数によって、登録されているスクリプトの実行優先度ではなく、キューイングの優先度が指定されます。</p> <p>この引数が指定されない場合、デフォルトのキューイング プライオリティは normal です。</p>
maxrun	(任意) スクリプトの最大ランタイム (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)。この引数が指定されない場合、デフォルトの 20 秒ランタイム制限が使用されます。
nice	(任意) ポリシーの実行時間優先度の設定。nice 引数が 1 に設定されている場合、ポリシーは、デフォルトの優先度よりも低い実行時間優先度で実行されます。デフォルト値は 0 です

結果文字列

なし

_cerno を設定

なし

Event_reqinfo

```
"event_id %u event_type %u event_type_string {%s} event_pub_sec %u event_pub_msec %u"
"slot %u event %s"
```

イベントタイプ	説明
event_id	パブリッシュされた該当イベントの ID を示す一意の番号。同一のイベントで複数のポリシーを実行可能であり、その場合、各ポリシーは同一のイベント ID を保持します。
event_type	イベントのタイプ。
event_type_string	このイベントタイプのイベントの名前を表す ASCII 文字列。
event_pub_sec event_pub_msec	EEM に対してイベントがパブリッシュされた、秒単位およびミリ秒単位の時間。
slot	影響が及ぼされるカードのスロット番号。
event	OIR の削除イベントまたは OIR の挿入イベントを表す、removed または online の文字列を示します。

event_register_process

プロセス イベントの登録を行います。この Tcl コマンド拡張を使用すると、Cisco IOS ソフトウェア モジュール方式プロセスの開始時と停止時に発生するイベントに基づいて、ポリシーが実行されます。これらのイベントは、このイベントをスクリーニングする System Manager イベント ディテクタによって処理されます。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

```
event_register_process [tag ?] abort|term|start|user_restart|user_shutdown
[sub_system ?] [version ?] [instance ?] [path ?] [node ?]
[queue_priority low|normal|high|last] [maxrun ?] [nice 0|1]
```

引数

tag	(任意) Tcl スクリプト内で複数のイベント文をサポートするため、Tcl コマンド拡張のトリガーとともに使用できるタグを指定する文字列。
-----	---

abort	(必須) プロセスの異常な終了。ゼロではない終了ステータスでの終了、カーネル生成信号の受信、またはユーザー要求のために送信されない SIGTERM 信号または SIGKILL 信号の受信のため、プロセスが強制終了されることがあります。
term	(必須) プロセスの正常な終了。
start	(必須) プロセスの開始。
user_restart	(必須) CLI コマンドからのプロセスの再起動要求が原因でプロセスを終了。
user_shutdown	(必須) CLI コマンドからのプロセスの終了要求が原因でプロセスを終了。
sub_system	(任意) プロセス イベントをパブリッシュした EEM ポリシーに割り当てられる番号。他のすべての番号は Cisco での使用のために予約されているため、番号は 798 に設定されます。
version	(任意) バージョンマネージャによって割り当てられるプロセスのバージョン番号。major_number.minor_number.level の形式である必要があります。指定される場合、バージョン番号の各コンポーネントは、1～4294967295 の範囲の整数である必要があります。
instance	(任意) プロセス インスタンス ID。指定される場合、この引数は、1～4294967295 の範囲の整数である必要があります。
path	(任意) プロセスパス名 (正規表現文字列)。process-name 引数の値に空白文字が含まれている場合、二重引用符で囲む必要があります。すべてのプロセスを照合するには、パス「*」を使用します。
ノード	(任意) ノード名は、「node」という語句と、それに続く、次の形式を使用してスラッシュ文字で区切られた2つのフィールドで構成される、文字列です。 node<slot-number>/<cpu-number> slot-number は、ハードウェア スロット番号です。cpu-number は、ハードウェア CPU 番号です。たとえば、スロット 0 にある Cisco Catalyst 6500 シリーズスイッチのスーパーバイザカードの SP CPU は、node0/0 と指定されます。たとえば、スロット 0 にある Cisco Catalyst 6500 シリーズスイッチのスーパーバイザカードの RP CPU は、node0/1 と指定されます。node 引数が指定されない場合、デフォルトのノード指定は、常に、すべての該当するノードを表す正規表現パターンマッチ * です。

queue_priority	<p>(任意) 次のような、スクリプトがキューに入れられる優先度レベル。</p> <ul style="list-style-type: none"> • queue_priority low : 3つの優先度レベルの最も低いレベルでスクリプトがキューに入れられるよう指定します。 • queue_priority normal : low プライオリティよりも高く、high プライオリティよりも低い優先度レベルでスクリプトがキューに入れられるよう指定します。 • queue_priority high : 3つの優先度レベルの最も高いレベルで、スクリプトがキューに入れられるよう指定します。 • queue_priority last : 最も低い優先度レベルでスクリプトがキューに入れられるよう指定します。 <p>「queue_priority_last」引数が設定された状態で複数のスクリプトが登録されている場合、これらのスクリプトは、イベントのパブリッシュ順に実行されます。</p> <p>(注) queue_priority 引数によって、登録されているスクリプトの実行優先度ではなく、キューイングの優先度が指定されます。</p> <p>この引数が指定されない場合、デフォルトのキューイング プライオリティは normal です。</p>
maxrun	<p>(任意) スクリプトの最大ランタイム (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)。この引数が指定されない場合、デフォルトの 20 秒ランタイム制限が使用されます。</p>
nice	<p>(任意) ポリシーの実行時間優先度の設定。nice 引数が 1 に設定されている場合、ポリシーは、デフォルトの優先度よりも低い実行時間優先度で実行されます。デフォルト値は 0 です</p>

任意の引数が指定されない場合、イベントは、引数のすべての可能な値に対して照会されます。複数の引数が存在する場合、すべての条件が一致したときに、プロセスイベントが発生します。

結果文字列

なし

_cerno を設定

なし

Event_reqinfo

```
"event_id %u event_type %u event_type_string {%s} event_pub_sec %u event_pub_msec %u"
"sub_system 0x%x instance %u process_name {%s} path {%s} exit_status 0x%x"
```

```
"respawn_count %u last_respawn_sec %ld last_respawn_msec %ld fail_count %u"
"dump_count %u node_name {%s}"
```

イベントタイプ	説明
event_id	パブリッシュされた該当イベントの ID を示す一意の番号。同一のイベントで複数のポリシーを実行可能であり、その場合、各ポリシーは同一の event_id を保持します。
event_type	イベントのタイプ。
event_type_string	このイベントタイプのイベントの名前を表す ASCII 文字列。
event_pub_sec event_pub_msec	EEM に対してイベントがパブリッシュされた、秒単位およびミリ秒単位の時間。
sub_system	アプリケーション固有のイベントをパブリッシュした EEM ポリシーに割り当てられる番号。他のすべての番号は Cisco での使用のために予約されているため、番号は 798 に設定されます。
instance	プロセス インスタンス ID。
process_name	プロセス名。
path	パスを含むプロセスの絶対名。
exit_status	プロセスの最後の終了ステータス。
respawn_count	プロセスが再起動された回数。
last_respawn_sec last_respawn_msec	最後の再起動が発生したカレンダー時間。
fail_count	失敗した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、0 にリセットされます。
dump_count	プロセスで取られたコア ダンプの数。
node_name	プロセスが存在するノードの名前。ノード名は、「node」という語句と、それに続く、次の形式を使用してスラッシュ文字で区切られた 2 つのフィールドで構成される、文字列です。 node slot-number / cpu-number slot-number は、ハードウェア スロット番号です。cpu-number は、ハードウェア CPU 番号です。

event_register_resource

Embedded Resource Manager (ERM) イベントの登録を行います。この Tcl コマンド拡張を使用すると、指定されたポリシーの ERM イベント レポートに基づいて、ポリシーが実行されます。ERM イベントは、EEM リソース イベントによってスクリーニングされ、これによって、指定された ERM ポリシーへの一致が発生したときに、EEM ポリシーを実行できます。

構文

```
event_register_resource policy policy-name [queue_priority low|normal|high|last]
[maxrun ?] [nice 0|1]
```

引数

ポリシー	(必須) ポリシーの使用を指定します。
policy-name	(必須) ERM ポリシーの名前。
queue_priority	<p>(任意) 次のような、スクリプトがキューに入れられる優先度レベル。</p> <ul style="list-style-type: none"> • <code>queue_priority low</code> : 3つの優先度レベルの最も低いレベルでスクリプトがキューに入れられるよう指定します。 • <code>queue_priority normal</code> : <code>low</code> プライオリティよりも高く、<code>high</code> プライオリティよりも低い優先度レベルでスクリプトがキューに入れられるよう指定します。 • <code>queue_priority high</code> : 3つの優先度レベルの最も高いレベルで、スクリプトがキューに入れられるよう指定します。 • <code>queue_priority last</code> : 最も低い優先度レベルでスクリプトがキューに入れられるよう指定します。 <p>「<code>queue_priority_last</code>」引数が設定された状態で複数のスクリプトが登録されている場合、これらのスクリプトは、イベントのパブリッシュ順に実行されます。</p> <p>(注) <code>queue_priority</code> 引数によって、登録されているスクリプトの実行優先度ではなく、キューイングの優先度が指定されます。</p> <p>この引数が指定されない場合、デフォルトのキューイング プライオリティは <code>normal</code> です。</p>
maxrun	<p>(任意) スクリプトの最大ランタイム (<code>SSSSSSSSSS[.MMM]</code> 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)。この引数が指定されない場合、デフォルトの 20 秒ランタイム制限が使用されます。</p>

nice	(任意) ポリシーの実行時間優先度の設定。nice 引数が 1 に設定されている場合、ポリシーは、デフォルトの優先度よりも低い実行時間優先度で実行されません。デフォルト値は 0 です
------	---

結果文字列

なし

_cerrno を設定

なし

Event_reqinfo

```
"event_id %u event_type %u event_type_string {%s} %u event_pub_sec %u event_pub_msec %u"
"owner_id %lld user_id %lld" time_sent %llu dampen_time %d notify_data_flags %u"
"level {%s} direction {%s} configured_threshold %u current_value %u"
"policy_violation_flag {%s} policy_id %d"
```

イベントタイプ	説明
event_id	パブリッシュされた該当イベントの ID を示す一意の番号。同一のイベントで複数のポリシーを実行可能であり、その場合、各ポリシーは同一の event_id を保持します。
event_type	イベントのタイプ。
event_type_string	このイベントタイプのイベントの名前を表す ASCII 文字列。
event_pub_sec event_pub_msec	EEM に対してイベントがパブリッシュされた、秒単位およびミリ秒単位の時間。
owner_id	Embedded Resource Manager (ERM) オーナー ID。
user_id	ERM ユーザー ID。
time_sent	ERM イベント時間、ナノ秒単位。
dampen_time	ERM 減衰時間、ナノ秒単位。
notify_data_flags	ERM 通知データ フラグ。
level	ERM イベントレベル。イベントレベルは、Normal、Minor、Major、および Critical の 4 つです。
direction	ERM イベント方向。イベント方向は、アップ、ダウン、または、変更なしのうちのいずれかです。
configured_threshold	設定されている ERM しきい値。
current_value	ERM によって報告された、現在の値。

イベントタイプ	説明
policy_violation_flag	ERM ポリシー違反フラグ (False または True)。
policy_id	ERM ポリシー ID。

event_register_rf

冗長ファシリティ (RF) イベントの登録を行います。この Tcl コマンド拡張を使用すると、RF の進行またはステータス イベントの通知が発生したときに、ポリシーが実行されます。

構文

```
event_register_rf [tag ?] event ?  
[queue_priority low|normal|high|last]  
[maxrun ?] [nice 0|1]
```

引数

tag	(任意) Tcl スクリプト内で複数のイベント文をサポートするため、Tcl コマンド拡張のトリガーとともに使用できるタグを指定する文字列。
-----	---

event	<p>(必須) RF の進行またはステータスイベントの名前。有効な値は次のとおりです。</p> <ul style="list-style-type: none">• RF_PROG_ACTIVE• RF_PROG_ACTIVE_DRAIN• RF_PROG_ACTIVE_FAST = 200• RF_PROG_ACTIVE_PRECONFIG• RF_PROG_ACTIVE_POSTCONFIG• RF_PROG_EXTRALOAD• RF_PROG_HANDBACK• RF_PROG_INITIALIZATION• RF_PROG_PLATFORM_SYNC• RF_PROG_STANDBY_BULK• RF_PROG_STANDBY_COLD• RF_PROG_STANDBY_CONFIG• RF_PROG_STANDBY_FILESYS• RF_PROG_STANDBY_HOT• RF_PROG_STANDBY_OIR_SYNC_DONE• RF_REGISTRATION_STATUS• RF_STATUS_MAINTENANCE_ENABLE• RF_STATUS_MANUAL_SWACT• RF_STATUS_OPER_REDUNDANCY_MODE_CHANGE• RF_STATUS_PEER_COMM• RF_STATUS_PEER_PRESENCE• RF_STATUS_REDUNDANCY_MODE_CHANGE• RF_STATUS_SWACT_INHIBIT
-------	---

queue_priority	<p>(任意) 次のような、スクリプトがキューに入れられる優先度レベル。</p> <ul style="list-style-type: none"> • queue_priority low : 3つの優先度レベルの最も低いレベルでスクリプトがキューに入れられるよう指定します。 • queue_priority normal : low プライオリティよりも高く、high プライオリティよりも低い優先度レベルでスクリプトがキューに入れられるよう指定します。 • queue_priority high : 3つの優先度レベルの最も高いレベルで、スクリプトがキューに入れられるよう指定します。 • queue_priority last : 最も低い優先度レベルでスクリプトがキューに入れられるよう指定します。 <p>「queue_priority_last」引数が設定された状態で複数のスクリプトが登録されている場合、これらのスクリプトは、イベントのパブリッシュ順に実行されます。</p> <p>(注) queue_priority 引数によって、登録されているスクリプトの実行優先度ではなく、キューイングの優先度が指定されます。</p> <p>この引数が指定されない場合、デフォルトのキューイング プライオリティは normal です。</p>
maxrun	<p>(任意) スクリプトの最大ランタイム (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)。この引数が指定されない場合、デフォルトの 20 秒ランタイム制限が使用されます。</p>
nice	<p>(任意) ポリシーの実行時間優先度の設定。nice 引数が 1 に設定されている場合、ポリシーは、デフォルトの優先度よりも低い実行時間優先度で実行されます。デフォルト値は 0 です</p>

結果文字列

なし

_cerno を設定

なし

Event_reqinfo

```
"event_id %u event_type %u event_type_string {%s} %u event_pub_sec %u event_pub_msec %u"
"event {%s}"
```

イベントタイプ	説明
event_id	パブリッシュされた該当イベントの ID を示す一意の番号。同一のイベントで複数のポリシーを実行可能であり、その場合、各ポリシーは同一の event_id を保持します。
event_type	イベントのタイプ。
event_type_string	このイベントタイプのイベントの名前を表す ASCII 文字列。
event_pub_sec event_pub_msec	EEM に対してイベントがパブリッシュされた、秒単位およびミリ秒単位の時間。
event	このイベントが発生する原因となる RF の進行またはステータス イベント通知。

event_register_routing

event routing コマンドによってトリガーされるイベントの登録を行います。これらのイベントは、ルートエントリが **Routing Information Base (RIB)** インフラストラクチャで変更されるときに、ルーティング イベント デテクタによって処理され、イベントがパブリッシュされます。この Tcl コマンド拡張を使用すると、このスクリプトのルーティングポリシーが実行されます。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。

構文

```
event_register_routing [tag ?] network ? length [ge|le|ne] [type add|remove|modify|all]
[protocol ?] [queue_priority normal|low|high|last] [maxrun ?] [nice {0 | 1}]
```

引数

tag	(任意) Tcl スクリプト内で複数のイベント文をサポートするため、Tcl コマンド拡張のトリガーとともに使用できるタグを指定する文字列。
network	ネットワーク IP アドレスを指定します。ネットワーク番号には、任意の有効な IP アドレスまたはプレフィックスを指定できます。

length	<p>ネットワークマスクの長さをビット単位で指定します。ビットマスクは0から32までの番号を使用できます。</p> <ul style="list-style-type: none"> • ge (任意) 照合されるプレフィックスの最小の長さを指定します。ge キーワードは、演算子の「以上」を表します。 • le (任意) 照合されるプレフィックスの最大の長さを指定します。le キーワードは、演算子の「以下」を表します。 • ne (任意) プレフィックスの長さを照合しない指定をします。ne キーワードは、演算子の「等しくない」を表します。 <p>ge キーワード、le キーワード、および ne キーワードが設定されない場合、ネットワーク長の完全一致が処理されます。</p>
type	<p>(任意) 必要なポリシーのトリガーを指定します。タイプオプションは、add、remove、modify、および all です。デフォルトは all です。</p>
プロトコル	<p>(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。</p> <p>次のプロトコルのいずれかを使用できます：all、bgp、connected、eigrp、isis、iso-igrp、mobile、odr、ospf、rip、static デフォルトは all です。</p>
queue_priority	<p>(任意) 次のような、スクリプトがキューに入れられる優先度レベル。</p> <ul style="list-style-type: none"> • queue_priority low : 3つの優先度レベルの最も低いレベルでスクリプトがキューに入れられるよう指定します。 • queue_priority normal : low プライオリティよりも高く、high プライオリティよりも低い優先度レベルでスクリプトがキューに入れられるよう指定します。 • queue_priority high : 3つの優先度レベルの最も高いレベルで、スクリプトがキューに入れられるよう指定します。 • queue_priority last : 最も低い優先度レベルでスクリプトがキューに入れられるよう指定します。 <p>「queue_priority_last」引数が設定された状態で複数のスクリプトが登録されている場合、これらのスクリプトは、イベントのパブリッシュ順に実行されます。</p> <p>(注) queue_priority 引数によって、登録されているスクリプトの実行優先度ではなく、キューイングの優先度が指定されます。</p> <p>この引数が指定されない場合、デフォルトのキューイングプライオリティは normal です。</p>
maxrun	<p>(任意) スクリプトの最大ランタイム (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)。この引数が指定されない場合、デフォルトの20秒ランタイム制限が使用されます。</p>

nice	(任意) ポリシーの実行時間優先度の設定。nice 引数が 1 に設定されている場合、ポリシーは、デフォルトの優先度よりも低い実行時間優先度で実行されず。デフォルト値は 0 です
------	---

結果文字列

なし

_cerrno を設定

なし

Event_reqinfo

```
"event_id %u event_type %u event_type_string {%s} %u event_pub_sec %u event_pub_msec %u"
"event_severity {%s} %u network %u mask %u protocol %u lastgateway %u distance %u" "time_sec %u
time_msec %u metric %u lastinterface %u"
```

イベントタイプ	説明
event_id	パブリッシュされた該当イベントの ID を示す一意の番号。同一のイベントで複数のポリシーを実行可能であり、その場合、各ポリシーは同一の event_id を保持します。
event_type	イベントのタイプ。
event_type_string	このイベントタイプのイベントの名前を表す ASCII 文字列。
event_pub_sec event_pub_msec	EEM に対してイベントがパブリッシュされた、秒単位およびミリ秒単位の時間。
event_severity	イベントの重大度。
network	IP アドレス形式のネットワーク プレフィックス。
mask	IP アドレス形式のネットワーク マスク。
protocol	ネットワーク プロトコルのタイプ。
type	追加、削除、または変更するイベントのタイプ。
lastgateway	最後に認識されたゲートウェイ。
distance	アドミニストレーティブ ディスタンス。
time_sec time_msec	イベントが EEM にパブリッシュされたときの、秒単位およびミリ秒単位でのイベントの時間。
metric	パス メトリック。
lastinterface	最後に認識されたインターフェイス。

event_register_rpc

EEM SSH リモートプロシージャコール (RPC) コマンドによってトリガーされるイベントの登録を行います。これらのイベントは、このイベントをスクリーニングする RPC イベントディテクタによって処理されます。この Tcl コマンド拡張を使用すると、このスクリプトの RPC ポリシーが実行されます。

構文

```
event_register_rpc [queue_priority {normal | low | high | last}] [maxrun <sec.msec>]
[nice {0 | 1}] [default <sec.msec>]
```

引数

queue_priority	<p>(任意) 次のような、スクリプトがキューに入れられる優先度レベル。</p> <ul style="list-style-type: none"> • queue_priority low : 3 つの優先度レベルの最も低いレベルでスクリプトがキューに入れられるよう指定します。 • queue_priority normal : low プライオリティよりも高く、high プライオリティよりも低い優先度レベルでスクリプトがキューに入れられるよう指定します。 • queue_priority high : 3 つの優先度レベルの最も高いレベルで、スクリプトがキューに入れられるよう指定します。 • queue_priority last : 最も低い優先度レベルでスクリプトがキューに入れられるよう指定します。 <p>「queue_priority_last」引数が設定された状態で複数のスクリプトが登録されている場合、これらのスクリプトは、イベントのパブリッシュ順に実行されます。</p> <p>(注) queue_priority 引数によって、登録されているスクリプトの実行優先度ではなく、キューイングの優先度が指定されます。</p> <p>この引数が指定されない場合、デフォルトのキューイングプライオリティは normal です。</p>
maxrun	<p>(任意) スクリプトの最大ランタイム (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)。この引数が指定されない場合、デフォルトの 20 秒ランタイム制限が使用されます。</p>
nice	<p>(任意) ポリシーの実行時間優先度の設定。nice 引数が 1 に設定されている場合、ポリシーは、デフォルトの優先度よりも低い実行時間優先度で実行されます。デフォルト値は 0 です</p>

デフォルト	(任意) CLI イベント デテクタがポリシーの終了を待つ時間 (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)。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。
-------	---

結果文字列

なし

_cerrno を設定

なし

Event_reqinfo

```
"event_id %u event_type %u event_type_string {%s} event_pub_sec %u event_pub_msec %u arg %u"
```

イベントタイプ	説明
event_id	パブリッシュされた該当イベントの ID を示す一意の番号。同一のイベントで複数のポリシーを実行可能であり、その場合、各ポリシーは同一の event_id を保持します。
event_type	イベントのタイプ。
event_type_string	このイベントタイプのイベントの名前を表す ASCII 文字列。
event_pub_sec event_pub_msec	EEM に対してイベントがパブリッシュされた、秒単位およびミリ秒単位の時間。

argc arg0 arg1 arg2 arg3 arg4 arg6 arg7 arg8 arg9 arg10 arg11 arg12 arg13 arg14	Extensible Markup Language (XML) Simple Object Access Protocol (SOAP) コマンドからスクリプトに渡されるパラメータ。
--	--

event_register_snmp

簡易ネットワーク管理プロトコル (SNMP) 統計イベントの登録を行います。この Tcl コマンド拡張を使用すると、SNMP オブジェクト ID (OID) によって指定されたカウンタが、定義されたしきい値に近くなったときに、ポリシーが実行されます。

構文

```
event_register_snmp [tag ?] oid ? get_type exact|next
entry_op gt|ge|eq|ne|lt|le entry_val ?
entry_type value|increment|rate
[exit_comb or|and]
[exit_op gt|ge|eq|ne|lt|le] [exit_val ?]
[exit_type value|increment|rate]
[exit_time ?] poll_interval ? [average_factor ?]
[queue_priority low|normal|high|last]
[maxrun ?] [nice 0|1]
```

引数

tag	(任意) Tcl スクリプト内で複数のイベント文をサポートするため、Tcl コマンド拡張のトリガーとともに使用できるタグを指定する文字列。
-----	---

oid	<p>(必須) SNMP ドット付き表記でのデータエレメントの OID 番号 (たとえば、1.3.6.1.2.1.2.1.0)。次のタイプの OID を使用できます。</p> <ul style="list-style-type: none"> • COUNTER_TYPE • COUNTER_64_TYPE • GAUGE_TYPE • INTEGER_TYPE • OCTET_PRIM_TYPE • OPAQUE_PRIM_TYPE • TIME_TICKS_TYPE
entry_op	<p>(必須) 現在の OID データの値を開始値と比較するために使用される開始比較演算子。真の場合、イベントが発生し、終了基準を満たすまでイベントモニタリングがディセーブルにされます。</p>
get_type	<p>(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 引数が「exact」の場合、指定された OID の値が取得されます。get_type 引数が「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。</p>
entry_val	<p>(必須) SNMP イベントが発生させる必要があるかどうかを判断するために、現在の OID データの値を比較する必要がある値。</p>
entry-type	<p>entry-val 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。</p> <p>値は、entry-val 引数の実際の値として定義されます。</p> <p>増分では、entry-val フィールドは増分差異として使用され、entry-val は、現在のカウンタの値と、イベントが最後に真であったとき (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。</p> <p>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、average-factor の値に、poll-interval の値を乗じたものです。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 average-factor 値サンプルの平均は、変更のレートとして取得されます。</p>
exit_comb	<p>(任意) イベントモニタリングが再度イネーブルにされるよう、終了基準が満たされているかどうかを判断するために必要な、終了条件テストの組み合わせを示す、終了組み合わせ演算子を使用します。「and」の場合は、終了基準を満たすために、終了値と終了時間テストの両方を渡す必要があります。「or」の場合は、終了基準を満たすために、終了値または終了時間テストのいずれかを渡します。exit_comb が「and」の場合、exit_op と exit_val (exit_time) が存在する必要があります。exit_comb が「or」の場合、(exit_op と exit_val) または (exit_time) が存在する必要があります。</p>

exit_op	(任意) 現在の OID データの値を終了値と比較するために使用される終了比較演算子。真の場合、このイベントのイベント モニターリングが再度イネーブルにされます。
exit_val	(任意) 終了基準を満たすかどうかを判断するために、現在の OID データの値を比較する必要がある値。
exit-type	(任意) exit-val 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。 値は、exit-val 引数の実際の値として定義されます。 増分では、exit-val フィールドは増分差異として使用され、exit-val は、現在のカウンタの値と、イベントが最後に真であったとき（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。 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、average-factor の値に、poll-interval の値を乗じたものです。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 average-factor 値サンプルの平均は、変更のレートとして取得されます。
exit_time	(任意) イベント モニターリングが再度イネーブルにされる時に発生するイベントの後の、POSIX タイマーユニットの数。SSSSSSSSSS[.MMM] 形式で指定します。SSSSSSSSSS は、0 ~ 4294967295 の秒数を表す整数である必要があります。MMM はミリ秒を表し、0 ~ 999 の整数である必要があります。
poll_interval	(必須) POSIX タイマーユニットの連続的なポーリング間隔。間隔は、現在、最小で 1 秒に設定されます (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)。
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 1 から 64 の範囲の数。average-factor の値は、poll-interval の値を乗じた値で、ミリ秒単位で導き出される期間です。最少平均係数値は 1 です。

queue_priority	<p>(任意) 次のような、スクリプトがキューに入れられる優先度レベル。</p> <ul style="list-style-type: none"> • queue_priority low : 3つの優先度レベルの最も低いレベルでスクリプトがキューに入れられるよう指定します。 • queue_priority normal : low プライオリティよりも高く、high プライオリティよりも低い優先度レベルでスクリプトがキューに入れられるよう指定します。 • queue_priority high : 3つの優先度レベルの最も高いレベルで、スクリプトがキューに入れられるよう指定します。 • queue_priority last : 最も低い優先度レベルでスクリプトがキューに入れられるよう指定します。 <p>「queue_priority_last」引数が設定された状態で複数のスクリプトが登録されている場合、これらのスクリプトは、イベントのパブリッシュ順に実行されます。</p> <p>(注) queue_priority 引数によって、登録されているスクリプトの実行優先度ではなく、キューイングの優先度が指定されます。</p> <p>この引数が指定されない場合、デフォルトのキューイング プライオリティは normal です。</p>
maxrun	<p>(任意) スクリプトの最大ランタイム (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)。この引数が指定されない場合、デフォルトの 20 秒ランタイム制限が使用されます。</p>
nice	<p>(任意) ポリシーの実行時間優先度の設定。nice 引数が 1 に設定されている場合、ポリシーは、デフォルトの優先度よりも低い実行時間優先度で実行されます。デフォルト値は 0 です</p>

結果文字列

なし

_cerno を設定

なし

Event_reqinfo

```
"event_id %u event_type %u event_type_string {%s} %u event_pub_sec %u event_pub_msec %u"
"event_severity {%s} oid {%s} val {%s} delta_val {%s}"
```

イベントタイプ	説明
event_id	パブリッシュされた該当イベントの ID を示す一意の番号。同一のイベントで複数のポリシーを実行可能であり、その場合、各ポリシーは同一の event_id を保持します。
event_type	イベントのタイプ。
event_type_string	このイベントタイプのイベントの名前を表す ASCII 文字列。
event_pub_sec event_pub_msec	EEM に対してイベントがパブリッシュされた、秒単位およびミリ秒単位の時間。
event_severity	SNMP イベントの重大度で、次のいずれかの値を指定できます。 <ul style="list-style-type: none"> • normal • minor • major
oid	SNMP ドット付き表記での、データ エLEMENT のオブジェクト ID。
val	データ エLEMENT の値。
delta_val	ポリシーの値間のデルタ値。

event_register_snmp_notification

簡易ネットワーク管理プロトコル (SNMP) 通知トラップ イベントの登録を行います。この Tcl コマンド拡張を使用すると、特定のインターフェイスまたはアドレスで、指定された SNMP オブジェクト ID (OID) で SNMP トラップが検出されるたびに、ポリシーが実行されます。SNMP 通知が Tcl ポリシーを使用して動作するようにするには、**snmp-server manager** CLI コマンドを有効にする必要があります。

構文

```
event_register_snmp_notification [tag ?] oid ? oid_val ?
op {gt|ge|eq|ne|lt|le}
[maxrun ?]
[src_ip_address ?]
[dest_ip_address ?]
[queue_priority {normal|low|high|last}]
[maxrun ?]
[nice {0|1}]
[default ?]
[direction {incoming|outgoing}]
[msg_op {drop|send}]
```

引数

tag	(任意) Tcl スクリプト内で複数のイベント文をサポートするため、Tcl コマンド拡張のトリガーとともに使用できるタグを指定する文字列。
oid	(必須) SNMP ドット付き表記でのデータ要素の OID 番号 (たとえば、1.3.6.1.2.1.2.1.0)。指定された OID がドット (.) で終わっている場合、ドットの前の OID 番号で始まっているすべての OID が、照会されます。次のタイプの OID を使用できます。 <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	(必須) SNMP イベントを発生させる必要があるかどうかを判断するために、現在の OID データの値を比較する必要がある OID 値。
op	(必須) 現在の OID データの値を、SNMP プロトコルデータユニット (PDU) の OID データ値と比較するために使用される、比較演算子。真の場合、イベントが発生します。
maxrun	(任意) スクリプトの最大実行時間 (sssssss[.mmm] 形式で指定します。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)。この引数が指定されない場合、デフォルトの 20 秒ランタイム制限が使用されます。
src_ip_address	(任意) SNMP 通知トラップが発信される発信元 IP アドレス。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。
dest_ip_address	(任意) SNMP 通知トラップが送信される宛先 IP アドレス。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。

queue_priority	<p>(任意) 次のような、スクリプトがキューに入れられる優先度レベル。</p> <ul style="list-style-type: none"> • queue_priority low : 3つの優先度レベルの最も低いレベルでスクリプトがキューに入れられるよう指定します。 • queue_priority normal : low プライオリティよりも高く、high プライオリティよりも低い優先度レベルでスクリプトがキューに入れられるよう指定します。 • queue_priority high : 3つの優先度レベルの最も高いレベルで、スクリプトがキューに入れられるよう指定します。 • queue_priority last : 最も低い優先度レベルでスクリプトがキューに入れられるよう指定します。 <p>queue_priority_last 引数が設定された状態で複数のスクリプトが登録されている場合、これらのスクリプトは、イベントのパブリッシュ順に実行されます。</p> <p>(注) queue_priority 引数によって、登録されているスクリプトの実行優先度ではなく、キューイングの優先度が指定されます。</p> <p>この引数が指定されない場合、デフォルトのキューイングプライオリティは normal です。</p>
デフォルト	<p>(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、秒単位での時間を指定します。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。</p>
nice	<p>(任意) ポリシーの実行時間優先度の設定。nice 引数が 1 に設定されている場合、ポリシーは、デフォルトの優先度よりも低い実行時間優先度で実行されます。デフォルト値は 0 です</p>
direction	<p>(任意) 発着信 SNMP トラップまたは通知 PDU がフィルタリングする方向。デフォルトは incoming です。</p>
msg_op	<p>(任意) イベントが一度トリガーされると、SNMP PDU (廃棄または送信) で行われるアクション。デフォルトは send です。</p>

結果文字列

なし

_cernno を設定

なし

Event_reqinfo

```
"event_id %u event_type %u event_type_string {%s} %u event_pub_sec %u event_pub_msec %u
event_severity {%s}" "oid {%s} oid_val {%s} src_ip_addr {%s} dest_ip_addr {%s} x_x_x_x_x_x
(varbinds) {%s} trunc_vb_buf {%s} trap_oid {%s} enterprise_oid {%s} generic_trap %u
specific_trap %u"
```

イベントタイプ	説明
event_id	パブリッシュされた該当イベントの ID を示す一意の番号。同一のイベントで複数のポリシーを実行可能であり、その場合、各ポリシーは同一の event_id を保持します。
event_type	イベントのタイプ。
event_type_string	このイベントタイプのイベントの名前を表す ASCII 文字列。
event_pub_sec event_pub_msec	EEM に対してイベントがパブリッシュされた、秒単位およびミリ秒単位の時間。
oid	ユーザー指定オブジェクト ID。
oid_val	ユーザー指定オブジェクト ID 値。
src_ip_addr	SNMP プロトコルデータユニット (PDU) の発信元 IP アドレス。
dest_ip_addr	SNMP PDU の宛先の IP アドレス。
x_x_x_x_x (varbinds)	SNMP PDU varbind 情報。
trap_oid	トラップ OID 値を示します。
enterprise_oid	エンタープライズ OID 値を示します。
generic_trap	汎用トラップタイプの番号の 1 つを示します。0 から 6 の、7 つの汎用トラップタイプがあります。
specific_trap	指定されたトラップ コードの番号の 1 つを示します。

event_register_snmp_object

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の登録を行います。この Tcl コマンド拡張を使用すると、特定のインターフェイスまたはアドレスで、指定された SNMP オブジェクト ID (OID) で SNMP が検出されるときに、値が置き換えられます。

構文

```
event_register_snmp_object oid ?
```

```

type {int|uint|counter|counter64|gauge|ipv4|oid|string}
sync {yes|no}
skip {yes|no}
[istable {yes|no}]
[default ?]
[queue_priority {normal|low|high|last}]
[maxrun ?]
[nice {0|1}]

```

引数

oid	<p>(必須) SNMP ドット付き表記でのデータ要素の OID 番号 (たとえば、1.3.6.1.2.1.2.1.0)。指定された OID がドット (.) で終わっている場合、ドットの前の OID 番号で始まっているすべての OID が、照会されます。次のタイプの OID を使用できます。</p> <ul style="list-style-type: none"> • COUNTER_TYPE • COUNTER_64_TYPE • GAUGE_TYPE • INTEGER_TYPE • OCTET_PRIM_TYPE • OPAQUE_PRIM_TYPE • TIME_TICKS_TYPE
type	(必須) OID 値のタイプ。
sync	<p>(必須) 「yes」は、EEM ポリシーが通知されることを意味します。アプレット <code>set_exit_status</code> または Tcl 戻り値が 0 の場合、SNMP によって、要求が処理されます。戻り値が 1 の場合、SNMP によって、<code>get</code> 要求のポリシーで指定された値が使用され、<code>set</code> 要求は処理されません。「no」は、EEM は通知されず、SNMP によって要求が処理されることを意味します。</p> <p>1 つの OID のみが、同期ポリシーに関連付けられます。ただし、複数の同期ポリシーが、同じ OID に登録できます。</p>
skip	<p><code>sync</code> 引数が <code>no</code> の場合は必須で、<code>sync</code> 引数が <code>yes</code> の場合は不要です <code>skip</code> 引数が「yes」の場合、SNMP によって要求が処理されることを意味します。 <code>skip</code> 引数が「no」の場合、SNMP は、オブジェクトが存在しないかのように動作することを意味します。</p>
istable	<p>(任意) 値「no」は、OID がスカラーオブジェクトであることを意味し、「yes」は、OID がテーブルオブジェクトであることを意味します。</p>

デフォルト	<p>(任意) SNMP オブジェクト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は、通常、SNMP サブシステムによって set 要求または get 要求を処理することです。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。</p>
maxrun	<p>(任意) スクリプトの最大実行時間 (sssssssss[.mmm] 形式で指定します。sssssssss 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)。この引数が指定されない場合、デフォルトの 20 秒ランタイム制限が使用されます。</p>
queue_priority	<p>(任意) 次のような、スクリプトがキューに入れられる優先度レベル。</p> <ul style="list-style-type: none"> • queue_priority low : 3 つの優先度レベルの最も低いレベルでスクリプトがキューに入れられるよう指定します。 • queue_priority normal : low プライオリティよりも高く、high プライオリティよりも低い優先度レベルでスクリプトがキューに入れられるよう指定します。 • queue_priority high : 3 つの優先度レベルの最も高いレベルで、スクリプトがキューに入れられるよう指定します。 • queue_priority last : 最も低い優先度レベルでスクリプトがキューに入れられるよう指定します。 <p>queue_priority_last 引数が設定された状態で複数のスクリプトが登録されている場合、これらのスクリプトは、イベントのパブリッシュ順に実行されます。</p> <p>(注) queue_priority 引数によって、登録されているスクリプトの実行優先度ではなく、キューイングの優先度が指定されます。</p> <p>この引数が指定されない場合、デフォルトのキューイング プライオリティは normal です。</p>
nice	<p>(任意) ポリシーの実行時間優先度の設定。nice 引数が 1 に設定されている場合、ポリシーは、デフォルトの優先度よりも低い実行時間優先度で実行されず。デフォルト値は 0 です</p>

結果文字列

なし

_cerno を設定

なし

Event_reqinfo

```
"event_id %u event_type %u event_type_string {%s} %u event_pub_sec %u event_pub_msec %u
event_severity {%s}" "oid {%s} request {%s} request_type {%s} value %u"
```

イベントタイプ	説明
event_id	パブリッシュされた該当イベントの ID を示す一意の番号。同一のイベントで複数のポリシーを実行可能であり、その場合、各ポリシーは同一の event_id を保持します。
event_type	イベントのタイプ。
event_type_string	このイベントタイプのイベントの名前を表す ASCII 文字列。
event_pub_sec event_pub_msec	EEM に対してイベントがパブリッシュされた、秒単位およびミリ秒単位の時間。
event_severity	イベントの重大度。
oid	受信した get 要求または set 要求の SNMP オブジェクトの ID。
request	get または set の要求タイプ。
request_type	要求のタイプ（現在または次の）。
value	set 要求のみ。オブジェクトに設定される値。

event_register_syslog

Syslog イベントの登録を行います。この Tcl コマンド拡張を使用すると、一定の時間内に一定回数の発生後、特定パターンの Syslog メッセージが記録されるときに、ポリシーがトリガーされます。

構文

```
event_register_syslog [tag ?] [occurs ?] [period ?] pattern ?
[priority all|emergencies|alerts|critical|errors|warnings|notifications|
informational|debugging|0|1|2|3|4|5|6|7]
[queue_priority low|normal|high|last]
[severity_fatal] [severity_critical] [severity_major]
[severity_minor] [severity_warning] [severity_notification]
[severity_normal] [severity_debugging]
[maxrun ?] [nice 0|1]
```

引数

tag	(任意) Tcl スクリプト内で複数のイベント文をサポートするため、Tcl コマンド拡張のトリガーとともに使用できるタグを指定する文字列。
-----	---

occurs	(任意) 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。指定される場合、0より大きい値を指定する必要があります。
period	(任意) イベントを発生させるために取る必要がある1つまたは複数のイベントの間の、秒単位およびミリ秒単位の時間の間隔 (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)。この引数が指定されない場合、期間チェックは適用されません。
pattern	(必須) Syslog メッセージパターンマッチの実行に使用される正規表現。この引数は、記録された Syslog メッセージを指定するためにポリシーによって使用されます。
priority	(任意) スクリーニングされるメッセージのプライオリティ。この引数が指定される場合、指定されたロギングプライオリティレベルまたはそれ以下メッセージのみがスクリーニングされます。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0です。
queue_priority	<p>(任意) 次のような、スクリプトがキューに入れられる優先度レベル。</p> <ul style="list-style-type: none"> • queue_priority low : 3つの優先度レベルの最も低いレベルでスクリプトがキューに入れられるよう指定します。 • queue_priority normal : low プライオリティよりも高く、high プライオリティよりも低い優先度レベルでスクリプトがキューに入れられるよう指定します。 • queue_priority high : 3つの優先度レベルの最も高いレベルで、スクリプトがキューに入れられるよう指定します。 • queue_priority last : 最も低い優先度レベルでスクリプトがキューに入れられるよう指定します。 <p>「queue_priority_last」引数が設定された状態で複数のスクリプトが登録されている場合、これらのスクリプトは、イベントのパブリッシュ順に実行されます。</p> <p>(注) queue_priority 引数によって、登録されているスクリプトの実行優先度ではなく、キューイングの優先度が指定されます。</p> <p>この引数が指定されない場合、デフォルトのキューイングプライオリティは normal です。</p>
maxrun	(任意) スクリプトの最大ランタイム (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)。この引数が指定されない場合、デフォルトの20秒ランタイム制限が使用されます。

nice	(任意) ポリシーの実行時間優先度の設定。nice 引数が 1 に設定されている場合、ポリシーは、デフォルトの優先度よりも低い実行時間優先度で実行されません。デフォルト値は 0 です
severity_xxx	(任意) スクリーニングされるイベントの重大度。この引数が指定される場合、指定された重大度のメッセージのみがスクリーニングされます。syslog イベントの重大度レベルのマッピングについては、「Syslog イベントの重大度のマッピング」というタイトルの表を参照してください。

複数の条件が存在する場合、すべての条件が一致したときに、Syslog イベントが発生します。

表 1: Syslog イベントの重大度のマッピング

重大度のキーワード	Syslog のプライオリティ	説明
severity_fatal	LOG_EMERG (0)	システムが使用不可能な状態。
severity_critical	LOG_ALERT (1)	クリティカル条件で、即時対応が必要であることを示す
severity_major	LOG_CRIT (2)	重大な状態。
severity_minor	LOG_ERR (3)	軽微な状態。
severity_warning	LOG_WARNING (4)	警告状態。
severity_notification	LOG_NOTICE (5)	基本的な通知、情報メッセージ
severity_normal	LOG_INFO (6)	正常なイベント、正常な状態に戻ったことを伝える
severity_debugging	LOG_DEBUG (7)	デバッグ メッセージ。

結果文字列

なし

_cerno を設定

なし

Event_reqinfo

```
"event_id %u event_type %u event_type_string {%s} event_pub_sec %u event_pub_msec %u"
"msg {%s}"
```

イベントタイプ	説明
event_id	パブリッシュされた該当イベントのIDを示す一意の番号。同一のイベントで複数のポリシーを実行可能であり、その場合、各ポリシーは同一の event_id を保持します。
event_type	イベントのタイプ。
event_type_string	このイベントタイプのイベントの名前を表す ASCII 文字列。
event_pub_sec event_pub_msec	EEM に対してイベントがパブリッシュされた、秒単位およびミリ秒単位の時間。
msg	パターンと一致する最後の Syslog メッセージ。

event_register_timer

パブリッシャとサブスクリバの両方として、タイマーを作成し、タイマーイベントの登録を行います。時間特有または時間に基づいたポリシーをトリガーする必要があるときに、この Tcl コマンド拡張を使用します。このイベントタイマーは、イベントのパブリッシャとサブスクリバの両方です。パブリッシャの部分は、名前付きタイマーがオフになるという条件を示します。サブスクリバの部分は、イベントが登録されているタイマーの名前を示します。



(注) CRON および絶対時間の指定は、現地時間で動作します。

構文

```
event_register_timer [tag ?] watchdog|countdown|absolute|cron
[name ?] [cron_entry ?]
[time ?]
[queue_priority low|normal|high|last] [maxrun ?]
[nice 0|1]
```

引数

tag	(任意) Tcl スクリプト内で複数のイベント文をサポートするため、Tcl コマンド拡張のトリガーとともに使用できるタグを指定する文字列。
watchdog	(必須) ウォッチドッグタイマー。
countdown	(必須) カウントダウンタイマー。
絶対	(必須) 絶対タイマー。
cron	(必須) CRON タイマー。

name	(任意) タイマーの名前。
------	---------------

cron_entry	
------------	--

(任意) CRON タイマー タイプが指定される場合に、指定する必要があります。他のいずれかのタイマー タイプが指定される場合には、指定しないでください。cron_entry は、UNIX CRON デーモンで使用される部分的な UNIX Crontab エントリ (最初の 5 つのフィールド) です。

cron_entry の指定は、5 つのフィールドが使用されるテキスト文字列で構成されます。フィールドは、空白文字で区切られます。フィールドは、CRON タイマー イベントがトリガーされるとき時刻と日付を表します。フィールドの説明については、「CRON イベントがトリガーされるとき時刻と日付」というタイトルの表を参照してください。

番号の範囲を使用できます。範囲は、ハイフンで区切られる 2 つの数字で表示されます。範囲には、2 つの数字自身も含まれます。たとえば、時刻に入力される 8-11 は、8 時、9 時、10 時、および 11 時での実行を示します。

フィールドはアスタリスク記号 (*) も使用でき、これは常に「first-last」を表します。

リストを使用できます。リストは、カンマで区切られた番号のセット (または範囲) です。例: "1,2,5,9" および "0-4,8-12"。

手順の値は、範囲の組み合わせで使用できます。範囲に続く「/<number>」によって、範囲内での省略値を指定します。たとえば、2 時間ごとにイベントのトリガーを指定する場合、「0-23/2」を hour フィールドで使用できます。アスタリスク記号後にも手順を使用でき、「2 時間ごと」と指定する場合は、「*/2」を使用します。

month フィールドと day of week フィールドには、名前も使用できます。特定の日または月の最初の 3 文字を使用します (ケースは問題ではありません)。名前の範囲またはリストは使用できません。

タイマー イベントがトリガーされる日は、day of month と day of week の 2 つのフィールドで指定できます。両方のフィールドが制限される (つまり * ではない) 場合、いずれかのフィールドが現在の時刻と一致すると、イベントがトリガーされます。たとえば、「30 4 1,15 * 5」の場合、各月の 1 日と 15 日に加え、金曜日の午前 4:30 にイベントがトリガーされます。

最初の 5 つのフィールドの代わりに、7 つの特殊文字列の 1 つが表示されることがあります。これらの 7 つの特殊文字列の説明については、「cron_entry の特殊文字列」というタイトルの表を参照してください。

例 1: 「0 0 1,15 * 1」では、各月の 1 日と 15 日、および月曜日ごとに、真夜中の 0 時に、イベントがトリガーされます。1 つのフィールドによってのみ日を指定する場合、他のフィールドは * に設定する必要があります。「0 0 * * 1」では、月曜日にのみ、真夜中の 0 時に、イベントがトリガーされます。

例 2: 「15 16 1 * *」では、各月の 1 日の午後 4:15 にイベントがトリガーされず。

例 3: 「0 12 * * 1-5」では、各週の月曜日から金曜日まで、正午に、イベントがトリガーされます。

	例 4 : 「@weekly」では、1 週間に一度、日曜日の真夜中の 0 時に、イベントがトリガーされます。
time	<p>(任意) CRON 以外のタイマータイプが指定される場合に、指定する必要があります。CRON タイマータイプが指定される場合には、指定しないでください。ウォッチドッグタイマーと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およびミリ秒の単位での数です。絶対タイマーでは、期限切れ時刻のカレンダー時間です。時間は、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。</p>
queue_priority	<p>(任意) 次のような、スクリプトがキューに入れられる優先度レベル。</p> <ul style="list-style-type: none"> • queue_priority low : 3 つの優先度レベルの最も低いレベルでスクリプトがキューに入れられるよう指定します。 • queue_priority normal : low プライオリティよりも高く、high プライオリティよりも低い優先度レベルでスクリプトがキューに入れられるよう指定します。 • queue_priority high : 3 つの優先度レベルの最も高いレベルで、スクリプトがキューに入れられるよう指定します。 • queue_priority last : 最も低い優先度レベルでスクリプトがキューに入れられるよう指定します。 <p>「queue_priority_last」引数が設定された状態で複数のスクリプトが登録されている場合、これらのスクリプトは、イベントのパブリッシュ順に実行されます。</p> <p>(注) queue_priority 引数によって、登録されているスクリプトの実行優先度ではなく、キューイングの優先度が指定されます。</p> <p>この引数が指定されない場合、デフォルトのキューイング プライオリティは normal です。</p>
maxrun	<p>(任意) スクリプトの最大ランタイム (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)。この引数が指定されない場合、デフォルトの 20 秒ランタイム制限が使用されます。</p>
nice	<p>(任意) ポリシーの実行時間優先度の設定。nice 引数が 1 に設定されている場合、ポリシーは、デフォルトの優先度よりも低い実行時間優先度で実行されます。デフォルト値は 0 です</p>

表 2: *CRON* イベントがトリガーされるときの時刻と日付

フィールド	使用可能な値
minute	0 ~ 59
hour	0 ~ 23
day of month	1 ~ 31
month	1 ~ 12 (または名前、下記を参照)
day of week	0 ~ 7 (0 または 7 が日曜日または名前。「cron_entry の特殊文字列」というタイトルの表を参照)

表 3: *cron_entry* の特殊文字列

文字列	意味
@yearly	1 年に 1 回トリガーする、「0 0 1 * *」。
@annually	@yearly と同じ。
@monthly	1 か月に 1 回トリガーする、「0 0 1 * *」。
@weekly	1 週間に 1 回トリガーする、「0 0 * * 0」。
@daily	1 日に 1 回トリガーする、「0 0 * * *」。
@midnight	@daily と同じ。
@hourly	1 時間に 1 回トリガーする、「0 * * * *」。

結果文字列

なし

_cerno を設定

なし

Event_reqinfo

```
"event_id %u event_type %u event_type_string {%s} event_pub_sec %u event_pub_msec %u"
"timer_type %s timer_time_sec %ld timer_time_msec %ld"
"timer_remain_sec %ld timer_remain_msec %ld"
```

イベントタイプ	説明
event_id	パブリッシュされた該当イベントの ID を示す一意の番号。同一のイベントで複数のポリシーを実行可能であり、その場合、各ポリシーは同一の event_id を保持します。
event_type	イベントのタイプ。
event_type_string	このイベントタイプのイベントの名前を表す ASCII 文字列。
event_pub_sec event_pub_msec	EEM に対してイベントがパブリッシュされた、秒単位およびミリ秒単位の時間。
timer_type	タイマーのタイプ。次のいずれかです。 <ul style="list-style-type: none"> • watchdog • countdown • 絶対
timer_time_sec timer_time_msec	タイマーの期限が切れる時間。
timer_remain_sec timer_remain_msec	次の期限切れ前の残りの時間。

関連項目

event_register_timer_subscriber

event_register_timer_subscriber

サブスクリバとしてタイマーイベントの登録を行います。この Tcl コマンド拡張を使用すると、サブスクリバとして、登録するイベントタイマーの名前が指定されます。イベントタイマーは、別のポリシーまたは別のプロセスに依存して、カウンタが実際に操作されます。たとえば、policyB はタイマー加入者ポリシーとして動作しますが、policyA（タイマーポリシーは不要ですが）では、register_counter、timer_arm、または timer_cancel の各 Tcl コマンド拡張を使用して、policyB で参照されているカウンタが操作されます。

構文

```
event_register_timer_subscriber watchdog|countdown|absolute|cron
name ? [queue_priority low|normal|high|last] [maxrun ?] [nice 0|1]
```

引数

watchdog	(必須) ウォッチドッグタイマー。
----------	-------------------

countdown	(必須) カウントダウン タイマー。
絶対	(必須) 絶対タイマー。
cron	(必須) CRON タイマー。
name	(必須) タイマーの名前。
queue_priority	<p>(任意) 次のような、スクリプトがキューに入れられる優先度レベル。</p> <ul style="list-style-type: none"> • <code>queue_priority low</code> : 3つの優先度レベルの最も低いレベルでスクリプトがキューに入れられるよう指定します。 • <code>queue_priority normal</code> : <code>low</code> プライオリティよりも高く、<code>high</code> プライオリティよりも低い優先度レベルでスクリプトがキューに入れられるよう指定します。 • <code>queue_priority high</code> : 3つの優先度レベルの最も高いレベルで、スクリプトがキューに入れられるよう指定します。 • <code>queue_priority last</code> : 最も低い優先度レベルでスクリプトがキューに入れられるよう指定します。 <p>「<code>queue_priority last</code>」引数が設定された状態で複数のスクリプトが登録されている場合、これらのスクリプトは、イベントのパブリッシュ順に実行されます。</p> <p>(注) <code>queue_priority</code> 引数によって、登録されているスクリプトの実行優先度ではなく、キューイングの優先度が指定されます。</p> <p>この引数が指定されない場合、デフォルトのキューイング プライオリティは <code>normal</code> です。</p>
maxrun	<p>(任意) スクリプトの最大ランタイム (<code>SSSSSSSSSS[.MMM]</code> 形式で指定します。<code>SSSSSSSSSS</code> は、0 ~ 4294967295 の秒数を表す整数で、<code>MMM</code> は 0 ~ 999 のミリ秒数を表す整数である必要があります)。この引数が指定されない場合、デフォルトの 20 秒ランタイム制限が使用されます。</p>
nice	<p>(任意) ポリシーの実行時間優先度の設定。<code>nice</code> 引数が 1 に設定されている場合、ポリシーは、デフォルトの優先度よりも低い実行時間優先度で実行されます。デフォルト値は 0 です</p>



- (注) タイマーイベントまたはカウンタイベントの登録を行う EEM ポリシーは、パブリッシャとサブスクリバの両方として動作できます。

結果文字列

なし

_cerrno を設定

なし

Event_reqinfo

```
"event_id %u event_type %u event_type_string {%s} event_pub_sec %u event_pub_msec %u"
"timer_type %s timer_time_sec %ld timer_time_msec %ld"
"timer_remain_sec %ld timer_remain_msec %ld"
```

イベントタイプ	説明
event_id	パブリッシュされた該当イベントの ID を示す一意の番号。同一のイベントで複数のポリシーを実行可能であり、その場合、各ポリシーは同一の event_id を保持します。
event_type	イベントのタイプ。
event_type_string	このイベントタイプのイベントの名前を表す ASCII 文字列。
event_pub_sec event_pub_msec	EEM に対してイベントがパブリッシュされた、秒単位およびミリ秒単位の時間。
timer_type	タイマーのタイプ。次のいずれかです。 <ul style="list-style-type: none"> • watchdog • countdown • 絶対
timer_time_sec timer_time_msec	タイマーの期限が切れる時間。
timer_remain_sec timer_remain_msec	次の期限切れ前の残りの時間。

関連項目

event_register_timer

event_register_track

Cisco IOS Object Tracking サブシステムからのレポートイベントの登録を行います。この Tcl コマンド拡張を使用すると、指定され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、ポリシーがトリガーされます。

構文

```
event_register_track ? [tag ?] [state up|down|any] [queue_priority low|normal|high|last]
```

```
[maxrun ?]
[nice 0|1]
```

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。
tag	(任意) Tcl スクリプト内で複数のイベント文をサポートするため、Tcl コマンド拡張のトリガーとともに使用できるタグを指定する文字列。
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 が指定されている場合、トラックされるオブジェクトが down 状態から up 状態に遷移するときにイベントが発生します。 down が指定されている場合、トラックされるオブジェクトが up 状態から down 状態に遷移するときにイベントが発生します。 any が指定されている場合、トラックされるオブジェクトがある状態から別の状態に遷移するときにイベントが発生します。
queue_priority	<p>(任意) 次のような、スクリプトがキューに入れられる優先度レベル。</p> <ul style="list-style-type: none"> • queue_priority low : 3 つの優先度レベルの最も低いレベルでスクリプトがキューに入れられるよう指定します。 • queue_priority normal : low プライオリティよりも高く、high プライオリティよりも低い優先度レベルでスクリプトがキューに入れられるよう指定します。 • queue_priority high : 3 つの優先度レベルの最も高いレベルで、スクリプトがキューに入れられるよう指定します。 • queue_priority last : 最も低い優先度レベルでスクリプトがキューに入れられるよう指定します。 <p>「queue_priority_last」引数が設定された状態で複数のスクリプトが登録されている場合、これらのスクリプトは、イベントのパブリッシュ順に実行されます。</p> <p>(注) queue_priority 引数によって、登録されているスクリプトの実行優先度ではなく、キューイングの優先度が指定されます。</p> <p>この引数が指定されない場合、デフォルトのキューイング プライオリティは normal です。</p>
maxrun	(任意) スクリプトの最大ランタイム (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)。この引数が指定されない場合、デフォルトの 20 秒ランタイム制限が使用されます。
nice	(任意) ポリシーの実行時間優先度の設定。nice 引数が 1 に設定されている場合、ポリシーは、デフォルトの優先度よりも低い実行時間優先度で実行されます。デフォルト値は 0 です

任意の引数が指定されない場合、イベントは、引数のすべての可能な値に対して照会されます。

結果文字列

なし

`_cerrno` を設定

なし

Event_reqinfo

```
"event_id %u event_type %u event_type_string {%s} %u event_pub_sec %u event_pub_msec %u"
"track_number {%u} track_state {%s}"
```

イベントタイプ	説明
<code>event_id</code>	パブリッシュされた該当イベントの ID を示す一意の番号。同一のイベントで複数のポリシーを実行可能であり、その場合、各ポリシーは同一のイベント ID を保持します。
<code>event_type</code>	イベントのタイプ。
<code>event_type_string</code>	このイベントタイプのイベントの名前を表す ASCII 文字列。
<code>event_pub_sec</code> <code>event_pub_msec</code>	EEM に対してイベントがパブリッシュされた、秒単位およびミリ秒単位の時間。
<code>track_number</code>	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。
<code>track_state</code>	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 <code>up</code> または <code>down</code> です。

event_register_wdsysmon

Watchdog System Monitor イベントの登録を行います。この Tcl コマンド拡張を使用すると、いくつかのサブイベントまたは条件の組み合わせである複合イベントの登録が行われます。たとえば、特定処理の CPU の使用率が 80% を超える場合で、かつ処理に使用されるメモリが初期割り当て容量の 50% よりも大きい場合といった条件を組み合わせで登録できます。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

```
event_register_wdsysmon [tag ?] [timewin ?]
[sub12_op and|or|andnot]
[sub23_op and|or|andnot]
```

```
[sub34_op and|or|andnot]
[sub1 subevent-description]
[sub2 subevent-description]
[sub3 subevent-description]
[sub4 subevent-description] [node ?]
[queue_priority low|normal|high|last]
[maxrun ?] [nice 0|1]
```

各引数は、位置に依存しません。



- (注) 演算子の定義は、and (論理 And 操作) or (論理 Or 操作)、andnot (論理 And Not 操作) です。たとえば、「sub12_op and」では、サブイベント 1 およびサブイベント 2 が真であるときにイベントが発生するよう定義されます。「sub23_op or」では、sub12_op で定義された条件が真で、サブイベント 3 が真であるときに、イベントが発生するよう定義されます。ロジックは、次のようにダイアグラム化できます。(((sub1 sub12_op sub2) sub23_op sub3) sub34_op sub4) が真の場合、イベントが発生

引数

tag	(任意) Tcl スクリプト内で複数のイベント文をサポートするため、Tcl コマンド拡張のトリガーとともに使用できるタグを指定する文字列。
timewin	(任意) イベントが生成されるようにするために、すべてのサブイベントが発生する必要がある時間ウィンドウ (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)。
sub12_op	(任意) サブイベント 1 とサブイベント 2 とを比較する組み合わせ演算子。
sub23_op	(任意) サブイベント 1、2 とサブイベント 3 とを比較する組み合わせ演算子。
sub34_op	(任意) サブイベント 1、2、サブイベント 3、サブイベント 4 とを比較する組み合わせ演算子。
sub1	(任意) サブイベント 1 の指定を意味します。
subevent-description	(任意) サブイベントの構文。
sub2	(任意) サブイベント 2 の指定を意味します。
sub3	(任意) サブイベント 3 の指定を意味します。
sub4	(任意) サブイベント 4 の指定を意味します。

ノード	<p>(任意) デッドロック条件が監視されるノード名は、「node」という語句と、それに続く、次の形式を使用してスラッシュ文字で区切られた2つのフィールドで構成される文字列です。</p> <p>node<slot-number>/<cpu-number></p> <p>slot-number は、ハードウェア スロット番号です。cpu-number は、ハードウェア CPU 番号です。たとえば、スロット 0 にある Cisco Catalyst 6500 シリーズ スイッチのスーパーバイザカードの SP CPU は、node0/0 と指定されます。たとえば、スロット 0 にある Cisco Catalyst 6500 シリーズ スイッチのスーパーバイザカードの RP CPU は、node0/1 と指定されます。node 引数が指定されない場合、デフォルトのノード指定は、登録が行われているローカルノードです。</p>
queue_priority	<p>(任意) 次のような、スクリプトがキューに入れられる優先度レベル。</p> <ul style="list-style-type: none"> • queue_priority low : 3 つの優先度レベルの最も低いレベルでスクリプトがキューに入れられるよう指定します。 • queue_priority normal : low プライオリティよりも高く、high プライオリティよりも低い優先度レベルでスクリプトがキューに入れられるよう指定します。 • queue_priority high : 3 つの優先度レベルの最も高いレベルで、スクリプトがキューに入れられるよう指定します。 • queue_priority last : 最も低い優先度レベルでスクリプトがキューに入れられるよう指定します。 <p>「queue_priority_last」引数が設定された状態で複数のスクリプトが登録されている場合、これらのスクリプトは、イベントのパブリッシュ順に実行されます。</p> <p>(注) queue_priority 引数によって、登録されているスクリプトの実行優先度ではなく、キューイングの優先度が指定されます。</p> <p>この引数が指定されない場合、デフォルトのキューイングプライオリティは normal です。</p>
maxrun	<p>(任意) スクリプトの最大ランタイム (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)。この引数が指定されない場合、デフォルトの 20 秒ランタイム制限が使用されます。</p>
nice	<p>(任意) ポリシーの実行時間優先度の設定。nice 引数が 1 に設定されている場合、ポリシーは、デフォルトの優先度よりも低い実行時間優先度で実行されます。デフォルト値は 0 です</p>

サブイベント

subevent description の構文は、7つのケースのうちの1つを使用できます。

subevent descriptions の引数では、number 引数の値に次の制約事項が適用されます。

- dispatch_mgr では、val は、0 ～ 4294967295 の範囲の整数である必要があります。
- cpu_proc および cpu_tot では、val は、0 ～ 100 の整数である必要があります。
- mem_proc、mem_tot_avail、および mem_tot_used では、is_percent が偽の場合、val は、0 ～ 4294967295 の範囲の整数である必要があります。

1. deadlock procname ?

引数

procname	(必須) デッドロック条件をモニターするプロセス名を指定する正規表現。指定された場合、サブイベントによって、時間ウィンドウは無視されます。
----------	---

2. dispatch_mgr [procname ?] [op gt|ge|eq|ne|lt|le] [val ?] [period ?]

引数

procname	(任意) dispatch_manager ステータスをモニターするプロセス名を指定する正規表現。
op	(任意) 収集されたイベント数を指定された値と比較するために使用される比較演算子。真の場合、このイベントが発生します。
val	(任意) 発生したイベント数の値を比較する必要があります。
period	(任意) 発生したイベント数の時間 (SSSSSSSS[.MMM] 形式で指定します。SSSSSSSS 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)。この引数が指定されない場合は、最新のサンプルが使用されます。

3. cpu_proc [procname ?] [op gt|ge|eq|ne|lt|le] [val ?] [period ?]

引数

procname	(任意) CPU の使用条件をモニターするプロセス名を指定する正規表現。
op	(任意) 収集された CPU 使用率サンプル パーセンテージを、指定されたパーセント値と比較するために使用される、比較演算子。真の場合、このイベントが発生します。
val	(任意) サンプル期間の平均 CPU 使用率のパーセント値を比較する必要があります。

period	(任意) サンプルの収集の平均の時間 (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)。この引数が指定されない場合は、最新のサンプルが使用されます。
--------	--

4. cpu_tot [op gt|ge|eq|ne|lt|le] [val ?] [period ?]

引数

op	(任意) 収集された合計システム CPU 使用率サンプルパーセンテージを、指定されたパーセント値と比較するために使用される、比較演算子。真の場合、このイベントが発生します。
val	(任意) サンプル期間の平均 CPU 使用率のパーセント値を比較する必要があります。
period	(任意) サンプルの収集の平均の時間 (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)。この引数が指定されない場合は、最新のサンプルが使用されます。

5. mem_proc [procname ?] [op gt|ge|eq|ne|lt|le] [val ?] [is_percent TRUE|FALSE] [period ?]

引数

procname	(任意) メモリ使用状況をモニターするプロセス名を指定する正規表現。
op	(任意) 収集された使用メモリを、指定された値と比較するために使用される比較演算子。真の場合、このイベントが発生します。
val	(任意) キロバイト単位で指定される、パーセンテージまたは絶対値。パーセンテージは、指定された時間内で最も古いサンプルと、最新のサンプルとの違いを表します。メモリ使用量が時間内で 150 KB から 300 KB に増えた場合、増加パーセンテージは 100 です。これは、測定された値を比較する必要がある値です。
is_percent	(任意) 真の場合、パーセンテージの値が収集され、比較されます。これ以外の場合、絶対値が収集され、比較されます。
period	(任意) is_percent が真に設定される場合、時間のパーセンテージが計算されます。これ以外の場合、収集されるサンプルの平均が計算される時間 (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)。この引数が指定されない場合は、最新のサンプルが使用されます。

6. mem_tot_avail [op gt|ge|eq|ne|lt|le] [val ?] [is_percent TRUE|FALSE] [period ?]

引数

op	(任意) 使用可能な収集されたメモリを指定された値と比較するために使用される比較演算子。真の場合、このイベントが発生します。
val	(任意) キロバイト単位で指定される、パーセンテージまたは絶対値。パーセンテージは、指定された時間内で最も古いサンプルと、最新のサンプルとの違いを表します。使用可能なメモリ使用量が時間内で 300 KB から 150 KB に減った場合、減少パーセンテージは 50 です。これは、測定された値と比較する必要がある値です。
is_percent	(任意) 真の場合、パーセンテージの値が収集され、比較されます。これ以外の場合、絶対値が収集され、比較されます。
period	(任意) is_percent が真に設定される場合、時間のパーセンテージが計算されます。これ以外の場合、収集されるサンプルの平均が計算される時間 (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)。この引数が指定されない場合は、最新のサンプルが使用されます。

```
7. mem_tot_used [op gt|ge|eq|ne|lt|le] [val ?] [is_percent TRUE|FALSE] [period ?]
```

引数

op	(任意) 収集された使用メモリを、指定された値と比較するために使用される比較演算子。真の場合、このイベントが発生します。
val	(任意) キロバイト単位で指定される、パーセンテージまたは絶対値。パーセンテージは、指定された時間内で最も古いサンプルと、最新のサンプルとの違いを表します。メモリ使用量が時間内で 150 KB から 300 KB に増えた場合、増加パーセンテージは 100 です。これは、測定された値と比較する必要がある値です。
is_percent	(任意) 真の場合、パーセンテージの値が収集され、比較されます。これ以外の場合、絶対値が収集され、比較されます。
period	(任意) is_percent が真に設定される場合、時間のパーセンテージが計算されます。これ以外の場合、収集されるサンプルの平均が計算される時間 (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)。この引数が指定されない場合は、最新のサンプルが使用されます。 (注) is_percent が真に設定されている場合、この引数は必須です。これ以外の場合、この引数は任意です。

結果文字列

なし

_cerrno を設定

なし

Event_reqinfo

```
"event_id %u event_type %u event_type_string {%s} %u event_pub_sec %u event_pub_msec %u"
"num_subs %u"
```

イベントタイプ	説明
event_id	パブリッシュされた該当イベントの ID を示す一意の番号。同一のイベントで複数のポリシーを実行可能であり、その場合、各ポリシーは同一の event_id を保持します。
event_type	イベントのタイプ。
event_type_string	このイベントタイプのイベントの名前を表す ASCII 文字列。
event_pub_sec event_pub_msec	EEM に対してイベントがパブリッシュされた、秒単位およびミリ秒単位の時間。
num_subs	サブイベント番号。

サブイベント情報文字列は、次のような、デッドロック サブイベント用です。

```
"(type %s num_entries %u entries {entry 1, entry 2, ...})"
```

サブイベントタイプ	説明
type	Wdsysmon サブイベントのタイプ。
num_entries	デッドロックのプロセスおよびスレッドの番号。
entries	デッドロックのプロセスおよびスレッドの情報。

各エントリは次のとおりです。

```
"(node {%s} procname {%s} pid %u tid %u state %s b_node %s b_procname %s b_pid %u b_tid %u)"
```

このエントリでは、プロセス A のスレッド m によって、プロセス B のスレッド n でブロックされるシナリオが記述されているとすると、次のようになります。

サブイベントタイプ	説明
node	プロセス A のスレッド m があるノードの名前。
procname	プロセス A の名前。
pid	プロセス A のプロセス ID。

サブイベントタイプ	説明
tid	プロセス A のスレッド m のスレッド ID。
state	プロセス A のスレッド m のスレッド状態。次のいずれかになります。 <ul style="list-style-type: none"> • STATE_CONDVAR • STATE_DEAD • STATE_INTR • STATE_JOIN • STATE_MUTEX • STATE_NANOSLEEP • STATE_READY • STATE_RECEIVE • STATE_REPLY • STATE_RUNNING • STATE_SEM • STATE_SEND • STATE_SIGSUSPEND • STATE_SIGWAITINFO • STATE_STACK • STATE_STOPPED • STATE_WAITPAGE • STATE_WAITTHREAD
b_node	プロセス B のスレッドがあるノードの名前。
b_procname	プロセス B の名前。
b_pid	プロセス B のプロセス ID。
b_tid	プロセス B のスレッド n のスレッド ID。0 は、プロセス A のスレッド m は、プロセス B のすべてのスレッド上でブロックされることを意味します。

dispatch_mgr サブイベントについて

```
"{type %s node {%s} procname {%s} pid %u value %u sec %ld msec %ld}"
```

サブイベントタイプ	説明
type	Wdysmon サブイベントのタイプ。
node	POSIX プロセスが存在するノードの名前。
procname	このサブイベントの POSIX プロセス名。
pid	このサブイベントの POSIX プロセス ID。 (注) 前述の 3 つのフィールドは、このディスパッチ マネージャのオーナー プロセスについて説明します。
value	sec 変数と msec 変数が、0 に指定されている、またはイベント登録 Tcl コマンド拡張で指定されていない場合、ディスパッチ マネージャによって処理されたイベント数は、最新サンプル内にあります。時間ウィンドウが指定され、登録 Tcl コマンド拡張でゼロよりも大きい場合、このディスパッチ マネージャによって処理されるイベントの合計数は、該当する時間ウィンドウにあります。
sec msec	イベント登録 Tcl コマンド拡張で、 sec 変数と msec 変数が 0 に指定されているか、または指定されていない場合、両方とも 0 です。登録 Tcl コマンド拡張で時間ウィンドウが指定され、かつその時間ウィンドウがゼロよりも大きい場合、 sec 変数および msec 変数は、この時間ウィンドウの最も古いサンプルと最新のサンプルとの実際の時間の差分です。

cpu_proc サブイベントについて

```
"(type %s node {%s} procname {%s} pid %u value %u sec %ld msec %ld)"
```

サブイベントタイプ	説明
type	Wdysmon サブイベントのタイプ。
node	POSIX プロセスが存在するノードの名前。
procname	このサブイベントの POSIX プロセス名。
pid	このサブイベントの POSIX プロセス ID。 (注) 前述の 3 つのフィールドは、その CPU 使用率がモニターされているプロセスについて説明します。

サブイベントタイプ	説明
value	sec 変数と msec 変数が 0 に指定されている、またはイベント登録 Tcl コマンド拡張で指定されていない場合、プロセスの CPU 使用率は最新サンプル内にあります。時間ウィンドウが指定され、登録 Tcl コマンド拡張でゼロよりも大きい場合、プロセス CPU 使用率の平均は、該当する時間ウィンドウにあります。
sec msec	イベント登録 Tcl コマンド拡張で、 sec 変数と msec 変数が 0 に指定されているか、または指定されていない場合、両方とも 0 です。登録 Tcl コマンド拡張で時間ウィンドウが指定され、かつその時間ウィンドウがゼロよりも大きい場合、 sec 変数および msec 変数は、この時間ウィンドウの最も古いサンプルと最新のサンプルとの実際の時間の差分です。

cpu_tot サブイベントについて

```
"{type %s node {%s} value %u sec %ld msec %ld}"
```

サブイベントタイプ	説明
type	Wdsysmon サブイベントのタイプ。
node	CPU 使用率の合計がモニターされているノードの名前。
value	sec 変数と msec 変数が 0 に指定されている、またはイベント登録 Tcl コマンド拡張で指定されていない場合、合計 CPU 使用率は最新サンプル内にあります。時間ウィンドウが指定され、登録 Tcl コマンド拡張でゼロよりも大きい場合、合計 CPU 使用率の平均は、該当する時間ウィンドウにあります。
sec msec	イベント登録 Tcl コマンド拡張で、 sec 変数と msec 変数が 0 に指定されているか、または指定されていない場合、両方とも 0 です。登録 Tcl コマンド拡張で時間ウィンドウが指定され、かつその時間ウィンドウがゼロよりも大きい場合、 sec 変数および msec 変数は、この時間ウィンドウの最も古いサンプルと最新のサンプルとの実際の時間の差分です。

mem_proc サブイベントについて

```
"{type %s node {%s} procname {%s} pid %u is_percent %s value %u diff %d sec %ld msec %ld}"
```

サブイベントタイプ	説明
type	Wdsysmon サブイベントのタイプ。

サブイベントタイプ	説明
node	POSIX プロセスが存在するノードの名前。
procname	このサブイベントの POSIX プロセス名。
pid	このサブイベントの POSIX プロセス ID。 (注) 前述の 3 つのフィールドは、そのメモリ使用率がモニターされているプロセスについて説明します。
is_percent	TRUE または FALSE のいずれかです。TRUE は、値がパーセント値であることを示します。FALSE は、値が絶対値であることを示します（平均値の場合もあります）。
value	sec 変数と msec 変数が 0 に指定されている、またはイベント登録 Tcl コマンド拡張で指定されていない場合、プロセスで使用されたメモリは最新サンプル内にあります。時間ウィンドウが指定され、登録 Tcl コマンド拡張でゼロよりも大きい場合、プロセスで使用されたメモリ使用率の平均は、該当する時間ウィンドウにあります。
サブイベントタイプ	説明
diff	sec 変数と msec 変数が 0 に指定されている、またはイベント登録 Tcl コマンド拡張で指定されていない場合、 diff は、今まで収集されたプロセスで使用されたメモリの最初のサンプルと、プロセスで使用されたメモリの最新サンプルのパーセンテージの差分です。時間ウィンドウが指定され、登録 Tcl コマンド拡張でゼロよりも大きい場合、 diff は、プロセスで使用されたメモリの使用状況のうち、指定された時間ウィンドウで最も古い値と最新の値のパーセンテージの差分です。
sec msec	イベント登録 Tcl コマンド拡張で、 sec 変数と msec 変数が 0 に指定されているか、または指定されていない場合、両方とも 0 です。登録 Tcl コマンド拡張で時間ウィンドウが指定され、かつその時間ウィンドウがゼロよりも大きい場合、 sec 変数および msec 変数は、この時間ウィンドウの最も古いサンプルと最新のサンプルとの実際の時間の差分です。

is_percent 引数が FALSE であり、イベント登録 Tcl コマンド拡張で **sec** 引数と **msec** 引数が 0 に指定されているか、または指定されていない場合は、次のようになります。

- **value** は最新のサンプルでプロセスによって使用されたメモリです。
- **diff** は 0 です。
- **sec** と **msec** は両方とも 0 です。

is_percent 引数が FALSE であり、イベント登録 Tcl コマンド拡張で時間ウィンドウがゼロよりも大きい値に指定されている場合は、次のようになります。

- **value** は指定された時間ウィンドウでプロセスによって使用されたメモリ サンプル値の平均です。
- **diff** は 0 です。
- **sec** および **msec** は、両方とも、この時間ウィンドウ内の最も古いサンプルのタイムスタンプと最新のサンプルのタイムスタンプの実際の時間の差分です。

is_percent 引数が TRUE であり、イベント登録 Tcl コマンド拡張で時間ウィンドウがゼロよりも大きい値に指定されている場合は、次のようになります。

- **value** は 0 です。
- **diff** は指定された時間ウィンドウの、最も古いプロセスで使用されたメモリ サンプルと最新のプロセスで使用されたメモリ サンプルとのパーセンテージによる差分です。
- **sec** および **msec** は、プロセスで使用されたメモリ サンプルの、この時間ウィンドウ内の最も古いタイムスタンプと最新のタイムスタンプの実際の時間の差分です。

is_percent 引数が TRUE であり、イベント登録 Tcl コマンド拡張で **sec** 引数と **msec** 引数が 0 に指定されているか、または指定されていない場合は、次のようになります。

- **value** は 0 です。
- **diff** は今まで収集された、最初のプロセスで使用されたメモリ サンプルと、最新のプロセスで使用されたメモリ サンプルとのパーセンテージによる差分です。
- **sec** および **msec** は、今まで収集されたプロセスで使用されたメモリの最初のサンプルのタイムスタンプと、プロセスで使用されたメモリの最新のサンプルのタイムスタンプの実際の時間の差分です。

mem_tot_avail サブイベントについて

```
"{type %s node {%s} is_percent %s used %u avail %u diff %d sec %ld msec %ld}"
```

サブイベントタイプ	説明
type	Wdsysmon サブイベントのタイプ。
node	使用可能なメモリの合計がモニターされているノードの名前。
is_percent	TRUE または FALSE のいずれかです。TRUE は、値がパーセント値であることを示します。FALSE は、値が絶対値であることを示します（平均値の場合もあります）。

サブイベント タイプ	説明
used	sec 変数と msec 変数が、0 に指定されるか、または、イベント登録 Tcl コマンド拡張で指定されない場合、使用されたメモリの合計は、最新のサンプルにあります。時間ウィンドウが指定され、登録 Tcl コマンド拡張でゼロよりも大きい場合、使用された合計メモリ使用率の平均は、該当する時間ウィンドウにあります。
avail	sec 変数と msec 変数が 0 に指定されている、または、イベント登録 Tcl コマンド拡張で指定されていない場合、 avail は使用可能な総メモリの最新サンプル内にあります。時間ウィンドウが指定され、登録 Tcl コマンド拡張でゼロよりも大きい場合、 avail は、指定された時間ウィンドウ内での使用可能な総メモリの使用率です。
diff	sec 変数と msec 変数が 0 に指定されている、または、イベント登録 Tcl コマンド拡張で指定されていない場合、 diff は、今まで収集された使用可能な総メモリの最初のサンプルと、使用可能な総メモリの最新サンプルのパーセンテージの差分です。時間ウィンドウが指定され、登録 Tcl コマンド拡張でゼロよりも大きい場合、 diff は、使用可能な総メモリの使用率のうち、指定された時間ウィンドウで最も古い値と最新の値のパーセンテージの差分です。
sec msec	イベント登録 Tcl コマンド拡張で、 sec 変数と msec 変数が 0 に指定されているか、または指定されていない場合、両方とも 0 です。時間ウィンドウが指定され、登録 Tcl コマンド拡張でゼロよりも大きい場合、これらの変数は、この時間ウィンドウの、最も古いサンプルと最新のサンプルとの実際の時間の差分です。

is_percent 引数が FALSE であり、イベント登録 Tcl コマンド拡張で **sec** 引数と **msec** 引数が 0 に指定されているか、または指定されていない場合は、次のようになります。

- **used** は最新のサンプルで使用されたメモリの合計です。
- **avail** は最新のサンプルで使用可能なメモリの合計です。
- **diff** は 0 です。
- **sec** と **msec** は両方とも 0 です。

is_percent 引数が FALSE であり、イベント登録 Tcl コマンド拡張で時間ウィンドウがゼロよりも大きい値に指定されている場合は、次のようになります。

- **used** は 0 です。
- **avail** は指定された時間ウィンドウで使用可能な合計メモリ サンプル値の平均です。
- **diff** は 0 です。
- **sec** および **msec** は、両方とも、この時間ウィンドウ内の使用可能な総メモリの最も古いサンプルのタイムスタンプと最新サンプルのタイムスタンプの実際の時間の差分です。

is_percent 引数が TRUE であり、イベント登録 Tcl コマンド拡張で時間ウィンドウがゼロよりも大きい値に指定されている場合は、次のようになります。

- **used** は 0 です。
- **avail** は 0 です。
- **diff** 指定された時間ウィンドウの、最も古い使用可能なメモリ サンプルの合計と最新の可能なメモリ サンプルの合計とのパーセンテージによる差分です。
- **sec** および **msec** は、両方とも、この時間ウィンドウ内の使用可能な総メモリの最も古いサンプルのタイムスタンプと最新サンプルのタイムスタンプの実際の時間の差分です。

is_percent 引数が TRUE であり、イベント登録 Tcl コマンド拡張で **sec** 引数と **msec** 引数が 0 に指定されているか、または指定されていない場合は、次のようになります。

- **used** は 0 です。
- **avail** は 0 です。
- **diff** 今まで収集された、最初の使用可能なメモリ サンプルの合計と、最新の使用可能なメモリ サンプルの合計との間の、パーセンテージによる差です。
- **sec** および **msec** は、今まで収集された使用可能な総メモリの最初のサンプルのタイムスタンプと、使用可能な総メモリの最新サンプルのタイムスタンプ間の実際の時間の差です。

mem_tot_used サブイベントについて

```
"{type %s node {%s} is_percent %s used %u avail %u diff %d sec %ld msec %ld}"
```

サブイベントタイプ	説明
type	Wdsysmon サブイベントのタイプ。
node	使用されているメモリの合計がモニターされているノードの名前。
is_percent	TRUE または FALSE のいずれかです。TRUE は、値がパーセント値であることを示します。FALSE は、値が絶対値であることを示します（平均値の場合もあります）。
used	sec 変数と msec 変数が、0 に指定されるか、または、イベント登録 Tcl コマンド拡張で指定されない場合、使用されたメモリの合計は、最新のサンプルにあります。時間ウィンドウが指定され、登録 Tcl コマンド拡張でゼロよりも大きい場合、使用された合計メモリ使用率の平均は、該当する時間ウィンドウにあります。

サブイベント タイプ	説明
avail	sec 変数と msec 変数が、0 に指定されている、または、イベント登録 Tcl コマンド拡張で指定されていない場合、 avail は使用されたメモリ合計の最新サンプル内にあります。時間ウィンドウが指定され、登録 Tcl コマンド拡張でゼロよりも大きい場合、 avail は、指定された時間ウィンドウ内での使用されたメモリ合計の使用状況です。
diff	sec 変数と msec 変数が 0 に指定されている、または、イベント登録 Tcl コマンド拡張で指定されていない場合、 diff は、今まで収集された使用されたメモリ合計の最初のサンプルと、使用されたメモリ合計の最新サンプルのパーセンテージの差分です。時間ウィンドウが指定され、登録 Tcl コマンド拡張でゼロよりも大きい場合、 diff は、使用されたメモリ合計の使用状況のうち、指定された時間ウィンドウで最も古い値と最新の値のパーセンテージの差分です。
sec msec	イベント登録 Tcl コマンド拡張で、 sec 変数と msec 変数が 0 に指定されているか、または指定されていない場合、両方とも 0 です。登録 Tcl コマンド拡張で時間ウィンドウが指定され、かつその時間ウィンドウがゼロよりも大きい場合、 sec 変数および msec 変数は、この時間ウィンドウの最も古いサンプルと最新のサンプルとの実際の時間の差分です。

is_percent 引数が FALSE であり、イベント登録 Tcl コマンド拡張で **sec** 引数と **msec** 引数が 0 に指定されているか、または指定されていない場合は、次のようになります。

- **used** は最新のサンプルで使用されたメモリの合計です。
- **avail** は最新のサンプルで使用可能なメモリの合計です。
- **diff** は 0 です。
- **sec** と **msec** は両方とも 0 です。

is_percent 引数が FALSE であり、イベント登録 Tcl コマンド拡張で時間ウィンドウがゼロよりも大きい値に指定されている場合は、次のようになります。

- **used** は指定された時間ウィンドウで使用された合計メモリ サンプル値の平均です。
- **avail** は 0 です。
- **diff** は 0 です。
- **sec** および **msec** は、両方とも、この時間ウィンドウ内の使用されたメモリ合計の最も古いサンプルのタイムスタンプと最新サンプルのタイムスタンプの実際の時間の差分です。

is_percent 引数が TRUE であり、イベント登録 Tcl コマンド拡張で時間ウィンドウがゼロよりも大きい値に指定されている場合は、次のようになります。

- **used** は 0 です。

- **avail** は 0 です。
- **diff** 指定された時間ウィンドウの、使用された最も古いメモリ サンプルの合計と使用された最新のメモリ サンプルの合計とのパーセンテージによる差分です。
- **sec** および **msec** は、両方とも、この時間ウィンドウ内の使用されたメモリ合計の最も古いサンプルのタイムスタンプと最新サンプルのタイムスタンプの実際の時間の差分です。

is_percent 引数が TRUE であり、イベント登録 Tcl コマンド拡張で **sec** 引数と **msec** 引数が 0 に指定されているか、または指定されていない場合は、次のようになります。

- **used** は 0 です。
- **avail** は 0 です。
- **diff** は今まで収集された、使用された最初のメモリ サンプルの合計と、使用された最新のメモリ サンプルの合計との間のパーセンテージによる差です。
- **sec** および **msec** は、今まで収集された使用されたメモリ合計の最初のサンプルのタイムスタンプと、使用されたメモリ合計の最新サンプルのタイムスタンプ間の実際の時間の差です。



(注) サブイベントの説明内部では、各引数は、位置に依存しません。

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。