Esempio di configurazione MIB dell'espressione e MIB dell'evento

Sommario

Introduzione

In questo documento viene illustrato come combinare il MIB Expression e il MIB Event per l'utilizzo nella gestione degli errori. L'esempio riportato non è realistico, ma mostra molte delle funzioni disponibili.

Il router deve eseguire due azioni:

1. Invia una trap se un'interfaccia di loopback ha una larghezza di banda superiore a 100 ed è disattivata a livello amministrativo
2. L'interfaccia di loopback viene chiusa se l'istruzione della larghezza di banda di una delle interfacce è stata modificata rispetto a un valore definito

Nell'esempio viene mostrato lo stato admin e bandwidth, in quanto sono facili da modificare dalla riga di comando e mostrano entrambi i valori integer e booleani.

I comandi di questo documento utilizzano il parametro identificatore oggetto (OID) e non i nomi degli oggetti. Ciò consente di eseguire il test senza caricare il MIB.

Prerequisiti

Requisiti

Prima di utilizzare le informazioni contenute in questo documento, verificare che siano soddisfatti i

seguenti prerequisiti:

- La workstation deve disporre degli strumenti SNMP (Simple Network Management Protocol) forniti da Hewlett-Packard (HP) Openview. Altri strumenti SNMP funzionano ma possono avere sintassi diversa.
- Sul dispositivo deve essere eseguito il software Cisco IOS® versione 12.2(4)T3 o successive. Le versioni precedenti non supportano la versione RFC del MIB di eventi.
- La piattaforma deve supportare l'evento MIB. Per un elenco delle piattaforme supportate per il software Cisco IOS versione 12.1(3)T, fare riferimento alla sezione "Piattaforma supportata" del supporto MIB evento.

Componenti usati

Le informazioni fornite in questo documento si basano sulle seguenti versioni software e hardware:

- Software Cisco IOS release 12.3(1a)
- Cisco 3640 Modular Access Router

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Convenzioni

Per ulteriori informazioni sulle convenzioni usate, consultare il documento Cisco sulle convenzioni nei suggerimenti tecnici.

Premesse

- L'espressione MIB consente all'utente di creare il proprio oggetto MIB in base a una combinazione di altri oggetti. Per ulteriori informazioni, consultare la RFC 2982 .
- L'evento MIB consente all'utente di avere il dispositivo che monitora i propri oggetti MIB e di generare azioni (notifiche o comandi **SNMP SET**) in base a un evento definito. Per ulteriori informazioni, consultare la RFC 2981 .

Configurazione

Nota: alcune righe del codice di output vengono visualizzate su due righe per adattarsi meglio allo schermo.

In questo esempio, ifIndex dell'interfaccia di loopback è uguale a 16.

```
# snmpget -v 2c -c private router .1.3.6.1.2.1.2.2.1.2.16
IF-MIB::ifDescr.16 = STRING: Loopback0
```

I nomi delle variabili relativi al primo evento iniziano con `e1` e quelli relativi al secondo inizio con `e2`.

Il nome del router è "router" e la stringa della community di lettura/scrittura è "private".

MIB espressione

Creazione dell'espressione 1

Creare innanzitutto un'espressione che restituisca il valore 1 se la condizione `ifSpeed è maggiore di 100.000` AND `ifAdminStatus è inattiva` per l'interfaccia di loopback. Se la condizione non viene soddisfatta, viene restituito il valore 0.

1. expExpressionDeltaInterval: questo oggetto non viene utilizzato.Non c'è motivo di calcolare un'espressione quando non viene sottoposta a polling. Se non viene impostato alcun valore, l'espressione viene calcolata quando si esegue una query sull'oggetto.Il nome dell'espressione è `e1exp`, che nella tabella ASCII corrisponde a `101 49 101 120 112`.
2. expNameStatus: in questo modo viene eliminata un'eventuale espressione precedente creata.
   ```
   # snmpset -v 2c -c private router 1.3.6.1.4.1.9.10.22.1.2.3.1.3.101.49.101.120.112 integer 6
   ```
3. expNameStatus: creazione e attesa.
   ```
   # snmpset -v 2c -c private router 1.3.6.1.4.1.9.10.22.1.2.3.1.3.101.49.101.120.112 integer 5
   ```
4. expExpressionIndex: crea l'indice da utilizzare successivamente per recuperare il risultato dell'espressione.
   ```
   # snmpset -v 2c -c private router 1.3.6.1.4.1.9.10.22.1.2.3.1.2.101.49.101.120.112 gauge 1
   ```
5. expExpressionComment: .1 (l'expExpressionIndex scelto) è la descrizione dell'espressione.
   ```
   # snmpset -v 2c -c private router 1.3.6.1.4.1.9.10.22.1.3.1.1.4.1 octetstring "e1 expression"
   ```
6. expExpression: si tratta dell'espressione stessa. Le variabili $1 e $2 vengono definite al passaggio successivo.Gli unici operatori consentiti sono (per ulteriori informazioni, fare riferimento alla RFC 2982):

   *( ) - (unary) + - * / % & | ^ << >> ~ ! && || == != > >= < <=*

   ```
   # snmpset -v 2c -c private router 1.3.6.1.4.1.9.10.22.1.3.1.1.2.1 octetstring '$1 < 100000 && $2 == 2'
   ```
7. IDSggetto
   ```
   .1 is for the variable $1 => ifSpeed
   .2 for $2 => ifAdminStatus
   ```

   ```
   # snmpset -v 2c -c private router 1.3.6.1.4.1.9.10.22.1.3.2.1.2.1.1 objectidentifier 1.3.6.1.2.1.2.2.1.5.16
   # snmpset -v 2c -c private router 1.3.6.1.4.1.9.10.22.1.3.2.1.2.1.2 objectidentifier 1.3.6.1.2.1.2.2.1.7.16
   ```
8. expObjectSampleType: i due valori vengono presi in valori assoluti (per Delta, prendere 2 come valore).

```
# snmpset -v 2c -c private router 1.3.6.1.4.1.9.10.22.1.3.2.1.4.1.1 integer 1
# snmpset -v 2c -c private router 1.3.6.1.4.1.9.10.22.1.3.2.1.4.1.2 integer 1
```

9. expObjectIDWildcard: gli ID oggetto non sono caratteri jolly. Si tratta del valore predefinito, pertanto non eseguire snmpset expObjectIDWildcard.
10. expObjectStatus: imposta le righe della tabella expObject su active.
```
# snmpset -v 2c -c private router 1.3.6.1.4.1.9.10.22.1.3.2.1.10.1.1 integer 1
# snmpset -v 2c -c private router 1.3.6.1.4.1.9.10.22.1.3.2.1.10.1.2 integer 1
```

11. Attivate l'espressione 1.
```
# snmpset -v 2c -c private router 1.3.6.1.4.1.9.10.22.1.2.3.1.3.101.49.101.120.112 integer
1
```

Verifica dell'espressione 1

```
router(config)#interface loopback 0
router(config-if)#shutdown
router(config-if)#bandwidth 150
```

1. Se la condizione viene soddisfatta, il valore di expValueCounter32Val è 1 (poiché il valore di expExpressionValueType rimane invariato, il risultato è un counter32).**Nota:** il tipo non può essere un valore a virgola mobile.
```
# snmpwalk -v 2c -c private router 1.3.6.1.4.1.9.10.22.1.4.1.1.2
cisco.ciscoExperiment.22.1.4.1.1.2.1.0.0.0 : Counter: 1


router(config-if)#bandwidth 150000
```

2. Se la condizione non viene soddisfatta, il valore è 0.
```
# snmpwalk -v 2c -c private router 1.3.6.1.4.1.9.10.22.1.4.1.1.2
cisco.ciscoExperiment.22.1.4.1.1.2.1.0.0.0 : Counter: 0

router(config-if)#bandwidth 1
router(config-if)#no shutdown
```

3. Se la condizione non viene soddisfatta, il valore è 0.
```
# snmpwalk -v 2c -c private router 1.3.6.1.4.1.9.10.22.1.4.1.1.2
cisco.ciscoExperiment.22.1.4.1.1.2.1.0.0.0 : Counter: 0
```

Creazione e test dell'espressione 2

```
# snmpset -v 2c -c private router 1.3.6.1.4.1.9.10.22.1.2.3.1.3.101.50.101.120.112 integer 6
# snmpset -v 2c -c private router 1.3.6.1.4.1.9.10.22.1.2.3.1.3.101.50.101.120.112 integer 5
# snmpset -v 2c -c private router 1.3.6.1.4.1.9.10.22.1.2.3.1.2.101.50.101.120.112 gauge 2
# snmpset -v 2c -c private router 1.3.6.1.4.1.9.10.22.1.3.1.1.4.2 octetstring "e2 expression"
# snmpset -v 2c -c private router 1.3.6.1.4.1.9.10.22.1.3.1.1.2.2 octetstring ''($1 * 18) / 23'
# snmpset -v 2c -c private router 1.3.6.1.4.1.9.10.22.1.3.2.1.2.2.1 objectidentifier
1.3.6.1.2.1.2.2.1.5
```

1. expObjectIDWildcard: indica che la versione `1.3.6.1.2.1.2.2.1.5` è una tabella e non un

oggetto.

```
# snmpset -v 2c -c private router 1.3.6.1.4.1.9.10.22.1.3.2.1.3.2.1 integer 1
# snmpset -v 2c -c private router 1.3.6.1.4.1.9.10.22.1.3.2.1.4.2.1 integer 1
# snmpset -v 2c -c private router 1.3.6.1.4.1.9.10.22.1.3.2.1.10.2.1 integer 1
# snmpset -v 2c -c private router 1.3.6.1.4.1.9.10.22.1.2.3.1.3.101.50.101.120.112 integer
1
```

2. Test:

```
# snmpwalk router 1.3.6.1.4.1.9.10.22.1.4.1.1
[...]
cisco.ciscoExperiment.22.1.4.1.1.2.2.0.0.10 : Counter: 0
cisco.ciscoExperiment.22.1.4.1.1.2.2.0.0.11 : Counter: 23250000
cisco.ciscoExperiment.22.1.4.1.1.2.2.0.0.12 : Counter: 42949672
cisco.ciscoExperiment.22.1.4.1.1.2.2.0.0.13 : Counter: 18450
cisco.ciscoExperiment.22.1.4.1.1.2.2.0.0.14 : Counter: 150
cisco.ciscoExperiment.22.1.4.1.1.2.2.0.0.15 : Counter: 1350
cisco.ciscoExperiment.22.1.4.1.1.2.2.0.0.16 : Counter: 9600
```

MIB evento

Creazione evento 1

Creare ora un evento che controlli il valore di output della prima espressione ogni 60 secondi e lo confronti con un riferimento. Quando il riferimento corrisponde al valore dell'espressione, viene attivata una trap con la variabile VARBIND scelta.

1. Creare il trigger nella tabella dei trigger.Il nome del trigger è trigger1, che nel codice ASCII è `116 114 105 103 103 101 114 49`.Il proprietario è Tom: `116 111 109`.L'indice di mteTriggerEntry è composto dal proprietario e dal nome del trigger. Il primo valore dell'indice fornisce il numero di caratteri per il mteOwner.In questo caso, tom contiene tre caratteri, quindi l'indice è: `3.116.111.109.116.114.105.103.103.101.114.49 .`
2. Eliminare la voce precedente, se esistente.
3. Impostare lo stato del trigger su **create e wait**.
4. L'ultimo passaggio consente di attivarlo:.StatoVoceTrigger

```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.2.2.1.15.3.116.111.109.116.114.105.103.103.101.114.49
integer 6
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.2.2.1.15.3.116.111.109.116.114.105.103.103.101.114.49
integer 5
```

mteTriggerValueID: il valore della prima espressione è `e1exp`.L'identificatore di oggetto dell'oggetto MIB è quello da campionare per verificare se il trigger deve essere attivato.

```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.2.2.1.6.3.116.111.109.116.114.105.103.103.101.114.49
objectidentifier
1.3.6.1.4.1.9.10.22.1.4.1.1.2.1.0.0.0
```

mteTriggerValueIDWildcard: senza utilizzare un carattere jolly per l'ID valore.

```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.2.2.1.7.3.116.111.109.116.114.105.103.103.101.114.49
integer 2
```

mteTriggerTest: esistenza (0), valore booleano (1) e soglia (2).Il metodo di selezione di uno dei valori precedenti è complesso. Per selezionare un'esistenza, specificate un valore in otto cifre in cui la prima è 1, ad esempio `10000000` o `100xxxxx`.Per un valore booleano, la seconda cifra deve essere 1: `0100000` o `010xxxxx`.Per una soglia, la terza cifra deve essere 1: `00100000` o `001xxxxx`.Èpiù facile lavorare in questo modo:Per l'esistenza, il valore è ottetstringhex—80.Per booleano, il valore è ottetstringhex—40.Per threshold, il valore è ottetstringhex—20.

```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.2.2.1.4.3.116.111.109.116.114.105.103.103.101.114.49
octetstringhex "40"
```

mteTriggerFrequency: determina il numero di secondi di attesa tra gli esempi di trigger.Il valore minimo viene impostato con l'oggetto mteResourceSampleMinimum (il valore predefinito è 60 secondi). Se si riduce questo valore, l'utilizzo della CPU aumenta, pertanto è necessario eseguire l'operazione con attenzione.

```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.2.2.1.11.3.116.111.109.116.114.105.103.103.101.114.49
gauge 60
```

mteTriggerSampleType: valore assoluto (1) e valore delta (2). In questo caso, il valore è assoluto:

```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.2.2.1.5.3.116.111.109.116.114.105.103.103.101.114.49
integer 1
```

mteTriggerEnabled: controllo che consente di configurare ma non utilizzare un trigger. Impostarla su true (il valore predefinito è false).

```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.2.2.1.14.3.116.111.109.116.114.105.103.103.101.114.49
integer 1
```

Dopo aver creato il trigger, definire l'evento che verrà utilizzato dal trigger. Il nome dell'evento è event1.MetaStatoVoceEvento

```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.4.2.1.5.3.116.111.109.101.118.101.110.116.49
integer 6
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.4.2.1.5.3.116.111.109.101.118.101.110.116.49
integer 5
```

mteEventActions: notifiche (0) e impostate (1).Il processo è lo stesso utilizzato per mteTriggerTest. La notifica è `10xxxxxx` e impostata su `01xxxxxx`.

```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.4.2.1.3.3.116.111.109.101.118.101.110.116.49
octetstringhex "80"
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.4.2.1.4.3.116.111.109.101.118.101.110.116.49
integer 1
```

Il passaggio successivo definisce il test da eseguire sull'oggetto selezionato per trigger1.mteTriggerBooleanComparison: sono diversi (1), uguali (2), minori (3), minoriOrEqual (4), maggiori (5) e maggioriOrEqual (6). In questo caso, è uguale a.

```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.2.5.1.1.3.116.111.109.116.114.105.103.103.101.114.49
```

```
integer 2
```

mteTriggerBooleanValue: valore da utilizzare per il test. Se il valore di `1.3.6.1.4.1.9.10.22.1.4.1.1.2.1.0.0.0` è uguale a 1, la condizione è soddisfatta.

```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.2.5.1.2.3.116.111.109.116.114.105.103.103.101.114.49
integer 1
```

Definire ora l'oggetto da inviare con l'evento.ProprietarioOggettiBooleaniTrigger

```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.2.5.1.4.3.116.111.109.116.114.105.103.103.101.114.49
octetstring "tom"
```

OggettoBooleanoTrigger

```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.2.5.1.5.3.116.111.109.116.114.105.103.103.101.114.49
octetstring "objects1"
```

ProprietarioEventoBooleanoTrigger

```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.2.5.1.6.3.116.111.109.116.114.105.103.103.101.114.49
octetstring "tom"
```

EventoBooleanoTrigger

```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.2.5.1.7.3.116.111.109.116.114.105.103.103.101.114.49
octetstring "event1"
```

Create la tabella degli oggetti.Inviate il valore di `1.3.6.1.2.1.2.1.5.16` come VARBIND con la trappola.Object Table mteObjectsName—Oggetti1.ImpostaStatoVoceOggetti

```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.3.1.1.5.3.116.111.109.8.111.98.106.101.99.116.115.49.1
integer 6
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.3.1.1.5.3.116.111.109.8.111.98.106.101.99.116.115.49.1
integer 5
```

IDommentoOggetti

```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.3.1.1.3.3.116.111.109.8.111.98.106.101.99.116.115.49.1
objectidentifier 1.3.6.1.2.1.2.2.1.5.16
```

mteObjectsIDWildcard: non viene utilizzato alcun carattere jolly.

```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.3.1.1.4.3.116.111.109.8.111.98.106.101.99.116.115.49.1
integer 1
```

Attivate la tabella degli oggetti.

```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.3.1.1.5.3.116.111.109.8.111.98.106.101.99.116.115.49.1
integer 1
```

Allegare l'oggetto all'evento 1.Notify nomeEventoMte—Evento1.ProprietarioOggettiNotificaEvento

```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.4.3.1.2.3.116.111.109.101.118.101.110.116.49
```

```
octetstring "tom"
```

OggettiNotificaEventi

```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.4.3.1.3.3.116.111.109.101.118.101.110.116.49
octetstring "objects1"
```

Attivare il trigger.

```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.2.2.1.15.3.116.111.109.116.114.105.103.103.101.114.49
integer 1
```

Attivare l'evento.

```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.4.2.1.5.3.116.111.109.101.118.101.110.116.49
integer 1
```

Trap ricevuta

```
Enterprise : 1.3.6.1.2.1.88.2
Trap type : ENTERPRISE SPECIFIC (6)
Specific trap type: 1
object 1 : mteHotTrigger
value : STRING: "trigger1"
object 2 : mteHotTargetName
value: ""
object 3 : mteHotContextName
value: ""
object 4: mteHotOID
value: OID: 1.3.6.1.4.1.9.10.22.1.4.1.1.2.1.0.0.0
object 5: mteHotValue
value: INTEGER: 1
object 6:  1.3.6.1.2.1.2.2.1.5.16
value: Gauge32: 1000
```

Nota: l'oggetto 6 è la variabile VARBIND aggiunta.

Creazione evento 2

Attenersi alla procedura seguente:

1. mteTriggerName: Trigger2.

   ```
   # snmpset -v 2c -c private router
   1.3.6.1.2.1.88.1.2.2.1.15.3.116.111.109.116.114.105.103.103.101.114.50
   integer 6
   # snmpset -v 2c -c private router
   1.3.6.1.2.1.88.1.2.2.1.15.3.116.111.109.116.114.105.103.103.101.114.50
   integer 5
   ```

2. mteTriggerValueID: valore della prima espressione e di mteTriggerValueIDWildcard.In questo caso, il processo utilizza caratteri jolly per l'ID valore, ovvero l'identificatore dell'oggetto MIB da campionare per determinare se il trigger viene attivato.

   ```
   # snmpset -v 2c -c private router
   1.3.6.1.2.1.88.1.2.2.1.6.3.116.111.109.116.114.105.103.103.101.114.50
   objectidentifier
   ```

```
1.3.6.1.4.1.9.10.22.1.4.1.1.2.2.0.0
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.2.2.1.7.3.116.111.109.116.114.105.103.103.101.114.50
integer 1
```

3. MteTriggerTest: soglia.
```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.2.2.1.4.3.116.111.109.116.114.105.103.103.101.114.50
octetstringhex "20"
```

4. FrequenzaTrigger
```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.2.2.1.11.3.116.111.109.116.114.105.103.103.101.114.50
gauge 60
```

5. mteTriggerSampleType: valore delta.
```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.2.2.1.5.3.116.111.109.116.114.105.103.103.101.114.50
integer 2
```

6. .MteTriggerEnabled
```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.2.2.1.14.3.116.111.109.116.114.105.103.103.101.114.50
integer 1
```

7. Creare un evento nella tabella eventi // nomeEventoMte—event2.
```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.4.2.1.5.3.116.111.109.101.118.101.110.116.50
integer 6
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.4.2.1.5.3.116.111.109.101.118.101.110.116.50
integer 5
```

8. mteEventActions: il valore 40 si riferisce a Set, ossia quando la condizione viene soddisfatta, il router invia un comando **snmp set**. In questo caso, esegue il Set da solo, ma potrebbe anche eseguire l'operazione su un dispositivo remoto.
```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.4.2.1.3.3.116.111.109.101.118.101.110.116.50
octetstringhex "40"
```

9. Attivare l'evento.
```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.4.2.1.4.3.116.111.109.101.118.101.110.116.50
integer 1
```

10. Impostare la soglia del trigger nella tabella Trigger // index = mteTriggerName—Trigger2.Trattandosi di una soglia, è opportuno fornire dei valori per le condizioni di insuccesso e di aumento. Questa volta prendete solo la condizione che sale.
11. mteTriggerThresholdDeltaRising: valore di soglia da confrontare.
```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.2.6.1.4.3.116.111.109.116.114.105.103.103.101.114.50
integer 100
```

12. ProprietarioEventoDeltaTriggerSoglia
```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.2.6.1.12.3.116.111.109.116.114.105.103.103.101.114.50
```

```
octetstring "tom"
```

13. EventoMteTriggerThresholdDeltaRising
```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.2.6.1.13.3.116.111.109.116.114.105.103.103.101.114.50
octetstring "event2"
```

14. mteEventSetObject: identificatore dell'oggetto MIB da impostare. Qui, ifAdminStatus per l'interfaccia di loopback.
```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.4.4.1.1.3.116.111.109.101.118.101.110.116.50
objectidentifier 1.3.6.1.2.1.2.2.1.7.16
```

15. mteEventSetValue: valore da impostare (2 per down).
```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.4.4.1.3.3.116.111.109.101.118.101.110.116.50
integer 2
```

16. Attivare il trigger.
```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.2.2.1.15.3.116.111.109.116.114.105.103.103.101.114.50
integer 1
```

17. Attivare l'evento.
```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.4.2.1.5.3.116.111.109.101.118.101.110.116.50
integer 1
```

Risultato

```
router(config)#int lo1
router(config-if)#bandwidth 5000000
16:24:11: %SYS-5-CONFIG_I: Configured from 10.48.71.71 by snmp
16:24:13: %LINK-5-CHANGED: Interface Loopback1, changed state to administratively down
16:24:14: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1, changed state to down
```

Nota: qui, `10.48.71.71` è l'indirizzo del router stesso.

Verifica

Le informazioni contenute in questa sezione permettono di verificare che la configurazione funzioni correttamente.

Alcuni comandi **show sono supportati dallo** strumento Output Interpreter (solo utenti registrati); lo strumento permette di visualizzare un'analisi dell'output del comando **show.**

```
router #show management event
 Mgmt Triggers:
 (1): Owner: tom
   (1): trigger1, Comment: , Sample: Abs, Freq: 15
         Test:  Boolean
         ObjectOwner: , Object:
         OID: ciscoExperiment.22.1.4.1.1.2.1.0.0.0, Enabled 1, Row Status 1
```

```
      Boolean Entry:
         Value: 1, Cmp: 2, Start: 1
         ObjOwn: tom, Obj: objects1, EveOwn: tom, Eve: event1

      Delta Value Table:
   (0): Thresh: , Exis: 1, Read: 0, OID: ciscoExperiment.22.1.4.1.1.2.1.0.0.0 , val: 0
   (2): trigger2, Comment: , Sample: Del, Freq: 60
         Test:  Threshold
         ObjectOwner: , Object:
         OID: ciscoExperiment.22.1.4.1.1.2.2.0.0, Enabled 1, Row Status 1
      Threshold Entry:
         Rising: 0, Falling: 0, DeltaRising: 100, DeltaFalling: 0
         ObjOwn: , Obj:
         RisEveOwn: , RisEve: , FallEveOwn: , FallEve:
         DelRisEveOwn: tom, DelRisEve: event2, DelFallEveOwn: , DelFallEve:

      Delta Value Table:
   (0): Thresh: , Exis: 1, Read: 0, OID: ciscoExperiment.22.1.4.1.1.2.2.0.0.1 , val: 62000000
   (1): Thresh: , Exis: 1, Read: 0, OID: ciscoExperiment.22.1.4.1.1.2.2.0.0.2 , val: 4000000
   (2): Thresh: , Exis: 1, Read: 0, OID: ciscoExperiment.22.1.4.1.1.2.2.0.0.3 , val: 617600
   (3): Thresh: , Exis: 1, Read: 0, OID: ciscoExperiment.22.1.4.1.1.2.2.0.0.4 , val: 617600
   (4): Thresh: , Exis: 1, Read: 0, OID: ciscoExperiment.22.1.4.1.1.2.2.0.0.5 , val: 617600
   (5): Thresh: , Exis: 1, Read: 0, OID: ciscoExperiment.22.1.4.1.1.2.2.0.0.6 , val: 617600
   (6): Thresh: , Exis: 1, Read: 0, OID: ciscoExperiment.22.1.4.1.1.2.2.0.0.7 , val: 858993458
   (7): Thresh: , Exis: 1, Read: 0, OID: ciscoExperiment.22.1.4.1.1.2.2.0.0.8 , val: 0
   (8): Thresh: , Exis: 1, Read: 0, OID: ciscoExperiment.22.1.4.1.1.2.2.0.0.9 , val: 62000000
   (9): Thresh: , Exis: 1, Read: 0, OID: ciscoExperiment.22.1.4.1.1.2.2.0.0.10 , val: 0
   (10): Thresh: , Exis: 1, Read: 0, OID: ciscoExperiment.22.1.4.1.1.2.2.0.0.11 , val: 62000000
   (11): Thresh: , Exis: 1, Read: 0, OID: ciscoExperiment.22.1.4.1.1.2.2.0.0.12 , val: 858993458
   (12): Thresh: , Exis: 1, Read: 0, OID: ciscoExperiment.22.1.4.1.1.2.2.0.0.13 , val: 858993458
   (13): Thresh: , Exis: 1, Read: 0, OID: ciscoExperiment.22.1.4.1.1.2.2.0.0.14 , val: 400
   (14): Thresh: , Exis: 1, Read: 0, OID: ciscoExperiment.22.1.4.1.1.2.2.0.0.15 , val: 3600
   (15): Thresh: , Exis: 1, Read: 0, OID: ciscoExperiment.22.1.4.1.1.2.2.0.0.16 , val: 25600

 Mgmt Events:
 (1): Owner: tom
   (1)Name: event1, Comment: , Action: Notify, Enabled: 1 Status: 1
      Notification Entry:
         ObjOwn: tom, Obj: objects1, OID: ccitt.0
   (2)Name: event2, Comment: , Action: Set, Enabled: 1 Status: 1
      Set:
         OID: ifEntry.7.13, SetValue: 2, Wildcard: 2
         TAG: , ContextName:

 Object Table:
 (1): Owner: tom
   (1)Name: objects1, Index: 1, OID: ifEntry.5.13, Wild: 2, Status: 1

 Failures: Event = 44716, Trigger = 0


router #show management expression
Expression: e1exp is active
  Expression to be evaluated is $1 < 100000 && $2 == 2 where:
  $1 = ifEntry.5.13
  Object Condition is not set
  Sample Type is absolute
  Both ObjectID and ObjectConditional are not wildcarded
  $2 = ifEntry.7.13
  Object Condition is not set
  Sample Type is absolute
  Both ObjectID and ObjectConditional are not wildcarded

Expression: e2exp is active
```

```
Expression to be evaluated is ($1 * 18) / 23 where:
$1 = ifEntry.5
Object Condition is not set
Sample Type is absolute
ObjectID is wildcarded
```

Risoluzione dei problemi

Le informazioni contenute in questa sezione permettono di risolvere i problemi relativi alla configurazione.

Comandi per la risoluzione dei problemi

Di seguito sono riportati i comandi per abilitare il debug:

```
router#debug management expression mib
router#debug management event mib
```

Nota: prima di usare i comandi di **debug**, consultare le informazioni importanti sui comandi di debug.

Informazioni correlate

- MIB espressione: RFC 2982
- MIB evento: RFC 2981
- EXPRESSION-MIB.my/EVENT-MIB.my
- Guida alle funzionalità di IOS: Supporto MIB evento
- Supporto tecnico – Cisco Systems