

MIB-configuratievoorbeeld voor expressie en gebeurtenis

Inhoud

[Inleiding](#)

[Voorwaarden](#)

[Vereisten](#)

[Gebruikte componenten](#)

[Conventies](#)

[Achtergrondinformatie](#)

[Configureren](#)

[MIB van expressie](#)

[Event MIB](#)

[Verifiëren](#)

[Problemen oplossen](#)

[Opdrachten voor probleemoplossing](#)

[Gerelateerde informatie](#)

[Inleiding](#)

Dit document toont hoe de MIB van de Uitdrukking en MIB van de gebeurtenis moeten worden gecombineerd voor gebruik in foutenbeheersing. Het bijgevoegde voorbeeld is niet realistisch, maar toont veel beschikbare kenmerken.

De router moet twee handelingen uitvoeren:

1. Verzend een val als een loopback interface een bandbreedte hoger dan 100 heeft en administratief laag is
2. De loopback interface sluit af als één van de interfaces zijn bandbreedteverklaring van een bepaalde waarde heeft veranderd

Het voorbeeld wordt getoond met bandbreedte en admin status omdat zij makkelijk te manipuleren zijn van de opdrachtregel en beiden integers en booleaanse waarden tonen.

De opdrachten in dit document gebruiken de object identifier (OID) parameter en niet de objectnamen. Dit maakt het testen mogelijk zonder het MIB te laden.

[Voorwaarden](#)

[Vereisten](#)

Zorg er voordat u de informatie in dit document gebruikt voor dat u aan de volgende voorwaarden

voldoet:

- Het werkstation moet beschikken over Simple Network Management Protocol (SNMP)-tools die zijn geleverd door Hewlett-Packard (HP) OpenView. Andere SNMP-tools werken maar kunnen een andere syntax hebben.
- Het apparaat moet Cisco IOS® software release 12.2(4)T3 of hoger uitvoeren. Eerdere versies ondersteunen de RFC-versie van de Event MIB niet.
- Het platform moet de Event MIB ondersteunen. Raadpleeg het gedeelte "Ondersteunde platform" van de [MIB-ondersteuning](#) voor een lijst met ondersteunde platforms voor Cisco IOS-software release 12.1(3)T.

Gebruikte componenten

De informatie in dit document is gebaseerd op de volgende software- en hardware-versies:



- Cisco IOS-software release 12.3(1)A
- Cisco 3640 modulaire toegangsrouter

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u de potentiële impact van elke opdracht begrijpen.

Conventies

Raadpleeg [Cisco Technical Tips Conventions](#) (Conventies voor technische tips van Cisco) voor meer informatie over documentconventies.

Achtergrondinformatie

- Met de expressie MIB kan de gebruiker zijn eigen MIB object maken op basis van een combinatie van andere objecten. Raadpleeg voor meer informatie [RFC 2982](#) .
- Het gebeurtenis MIB staat de gebruiker toe om het apparaat zijn eigen MIB voorwerpen te hebben gecontroleerd en om acties (bericht of **SNMP SET** opdrachten) te genereren op basis van een bepaalde gebeurtenis. Raadpleeg voor meer informatie [RFC 2981](#) .

Configureren

Opmerking: Sommige regels uitvoercode worden over twee regels weergegeven zodat ze beter op het scherm passen.

In dit voorbeeld is de ifIndex van de loopback interface gelijk aan 16.

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

De namen van de variabelen voor de eerste gebeurtenis beginnen met `e1` en die gerelateerd aan de tweede start met `e2`. De routernaam is "router" en de lees-schrijfstring is "private".

MIB van expressie

De expressie maken 1

Maak eerst een expressie die een waarde van 1 teruggeeft als de conditie, alsSpeed groter is dan 100.000 EN als AdminStatus voor de loopback interface is. Als niet aan de voorwaarde wordt voldaan, geeft het de waarde 0 terug.


1. [expExpressionDeltaInterval](#)-Dit object wordt niet gebruikt. Er is geen reden om een expressie te berekenen als deze niet wordt ondervraagd. Als geen waarde wordt ingesteld, wordt de expressie berekend wanneer het object wordt gevraagd. De naam van de uitdrukking is e1exp, wat in de ASCII-tabel overeenkomt met 101 49 101 120 112.

2. [expNameStatus](#) - Dit vernietigt een uiteindelijke oude expressie die wordt gemaakt.
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](#) - Maken en wachten.
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](#)-Dit creëert de index die later moet worden gebruikt om het resultaat van de expressie terug te winnen.
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](#)-Here .1 (de gekozen expExpressionIndex) is de beschrijving van de expressie.
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](#)-Dit is de expressie zelf, de variabelen \$1 en \$2 worden bij de volgende stap gedefinieerd. De enige toegestane exploitanten zijn (zie voor nadere informatie [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. [expObjectID](#)
.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](#) - De twee waarden worden genomen in absolute waarden (voor Delta, neem 2 als waarde).

```
# 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](#)—De object-IDs zijn niet beveiligd. Dit is de standaardwaarde, dus stel expObjectIDWildcard niet in.

10. [expObjectStatus](#)—Stel de rijen in expObjectTable in op actief.

```
# 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. Activeert expressie 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
```

[De expressie testen 1](#)

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

1. Als aan de voorwaarde wordt voldaan, is de waarde van [expValueCounter32Val 1](#) (aangezien de waarde van [expExpressionValueType](#) onveranderd blijft, is het resultaat een teller32). **Opmerking:** het type kan geen drijvende-kommawaarde zijn.

```
# 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. Als niet aan de voorwaarde is voldaan, is de waarde 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. Als niet aan de voorwaarde is voldaan, is de waarde 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
```

[Expressie maken en testen 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](#)—Dit geeft aan dat 1.3.6.1.2.1.2.2.1.5 een tabel is en geen object.

```
# 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
```

Event MIB

Gebeurtenis maken 1

Maak nu een gebeurtenis die de uitvoerwaarde van de eerste expressie elke 60 seconden controleert en vergelijkt deze met een referentie. Wanneer de referentie de expressiewaarde aanpast, wordt er een val geactiveerd met de gekozen VARBIND.

1. Maak de trigger in de trigger tabel. De naam van de trekker is trigger1, dat in ASCII-code ¹¹⁶_{114 105 103 103 101 114 49} is. De eigenaar is tom: ^{116 111 109}. De index van de datum TriggerEntry bestaat uit de activapigenaar en de activapandaam. De eerste waarde van de index geeft het aantal tekens voor de lokale eigenaar. In dit geval zijn er drie tekens voor tom, zodat de index is: 3.116.111.109.116.114.105.103.103.101.114.49.
2. Vernietig de oude ingang als deze bestaat.
3. Stel de trigger status in om **te maken en te wachten**.
4. Met de laatste stap wordt het geactiveerd: [OpstartenStatus](#)

```
# 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
```

[mementeTriggerValueID](#) - De waarde van de eerste expressie is e_{1exp} . De objectherkenner van het MIB - object is degene om te controleren of de trekker moet ontbranden.

```
# 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
```

[mementeTriggerValueIDWildcard](#)—zonder een jokerteken voor de waarde-ID te gebruiken.

```
# 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
```

[MetteTriggerTest](#)—Bestaan (0), boolean (1), en drempelwaarde (2). De methode om een van de bovenstaande waarden te selecteren is een complex. Om een bestaan te selecteren, geef een waarde in acht cijfers waarin de eerste 1 is, zoals 1000000 of 100xxxxxx. Voor een boolean moet het tweede cijfer een 1 zijn: 01000000 of 010xxxxxx. Voor een drempelwaarde moet het

derde cijfer een getal van 1 zijn: 00100000 of 001xxxxxx. Het is het gemakkelijkst om op deze manier te werken: Voor een bestaan is de waarde octetvezelhex-80. Voor boolean is de waarde octettrighex-40. Voor drempelwaarde is de waarde octettrighex-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](#) - Dit bepaalt het aantal seconden dat moet worden gewacht tussen de activiteitsmonsters. De minimumwaarde wordt ingesteld met het object ResourceSampleMinimale (de standaardinstelling is 60 seconden), waardoor deze waarde het CPU-gebruik verhoogt, zodat dit voorzichtig moet gebeuren.

```
# 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
```

[memteTriggerSampleType](#) - deze zijn absolute waarde (1) en deltaValue (2). In dit geval is de waarde absoluut:

```
# 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
```

[MateTrigger](#) - Dit is een controle die een trigger om te worden ingesteld maar niet gebruikt. Stel dit in op ware (standaard is onjuist).

```
# 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
```

Nu de trigger is aangemaakt, definieer dan de gebeurtenis die de trigger zal gebruiken. De naam van de gebeurtenis is event1. [EventEntryStatus](#)

```
# 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
```

[StteEventActions](#)—deze zijn kennisgeving (0) en reeks (1). Het proces is hetzelfde als voor MetteTriggerTest. De melding is 10xxxxxxx en de optie is 01xxxxx.

```
# 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
```

Deze volgende stap definieert de test die moet worden uitgevoerd op het object dat is geselecteerd voor trigger1. [Multiservice-TriggerBooleanVergelijking](#)—deze zijn ongelijk (1), gelijk (2), minder (3), minder (4), groter (5) en groter of gelijk (6). In dit geval—gelijk.

```
# 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](#) - Dit is de waarde die voor de test moet worden gebruikt. Indien de waarde van 1.3.6.1.4.1.9.10.22.1.4.1.1.2.1.0.0.0 gelijk is aan 1, is aan de voorwaarde

voldaan.

```
# 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
```

Bepaal nu het object dat met de gebeurtenis moet worden

verstuurd. [meteTriggerBooleanObjectsOwner](#)

```
# 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"
```

[GemteTriggerBooleanObjects](#)

```
# 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"
```

[metoTriggerBooleanEventOwner](#)

```
# 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"
```

[mete TriggerBooleanEvent](#)

```
# 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"
```

Maak de doeltabel. Verzend de waarde van 1.3.6.1.2.1.2.2.1.5.16 als VARBIND met de val.Object Tabel [ObjectName](#)—Objects1. [OpdrachtStatus](#)

```
# 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
```

[meteObjectsID](#)

```
# 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
```

[MatteObjectsIDWildcard](#)—Er is geen wildkaart gebruikt.

```
# 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
```

Activeert de objecttabel.

```
# 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
```

Bevestig het object aan de

gebeurtenis1. [MateEventName](#)—Event1. [EventKennisgevingExemplaarEigenaar](#)

```
# 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"
```

[EventKennisgevingExemplaren](#)

```
# 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"
```

Activeert de 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
```

Activeert de gebeurtenis.

```
# 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
```

[Vak ontvangen](#)

```
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
```

Opmerking: object 6 is de VARBIND die is toegevoegd.

[Event 2 creëren](#)

Ga als volgt te werk:

1. [regTriggerName](#)—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. [memteTriggerValueID](#) - Dit is de waarde van de eerste expressie en [mteTriggerValueIDWildcard](#). In dit geval, de procesbosbranden de waarde-ID, de object-identificatie van het MIB object om te kijken of de trigger branden.

```
# 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. [Starttest](#) — Drempel.

```
# 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. [MetaTriggerFrequentie](#)

```
# 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. [veelTriggerSampleType](#)—Delta waarde.

```
# 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. [mogelijk](#)

```
# 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. Maak een gebeurtenis in de eventtabel // [veelEventName](#)—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. [MetteEventActions](#)-De waarde 40 is voor Stel, dat betekent dat wanneer de voorwaarde wordt voldaan, de router een **snmp set** opdracht uitvoert. In dit geval maakt het de Instellen voor zichzelf, maar het kan ook de bediening op een extern apparaat maken.

```
# 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. Schakel de gebeurtenis in.

```
# 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. Stel de drempelwaarde voor trigger in in de Trigger Tabel // index =

[mteTriggerName](#)—Trigger2 in.Aangezien het een drempel is, geef dan waarden voor dalende en stijgende omstandigheden. Neem deze keer alleen de stijgende conditie.

11. [memteTriggerDrempelDeltaRising](#) - Dit is de drempelwaarde om te controleren.

```
# 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. [DemoTriggerDrempelDeltaRisingEventEigenaar](#)

```
# 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. [DemoTriggerDrempelDeltaRisingEvent](#)

```
# 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. [GemteEventSetObject](#)-Dit is de object identifier van het MIB-object dat moet worden ingesteld. Hier, als AdminStatus voor de loopback interface.

```
# 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. [StteEventSetValue](#) - Dit is de waarde die u wilt instellen (2 voor beneden).

```
# 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. Activeert de 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. Activeert de gebeurtenis.

```
# 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
```

[Resultaat](#)

```
router(config)#int lo1
router(config-if)#bandwidth 500000
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
```

Opmerking: Hier is 10.48.71.71 het adres van de router zelf.

[Verifiëren](#)

Deze sectie verschaft informatie om te bevestigen dat de configuratie correct werkt.

Bepaalde opdrachten met **show** worden ondersteund door de tool [Output Interpreter \(alleen voor geregistreerde klanten\)](#). Hiermee kunt u een analyse van de output van opdrachten met **show** genereren.

```
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

Problemen oplossen

Deze sectie verschaft informatie om te gebruiken om de configuratie problemen op te lossen.



Opdrachten voor probleemoplossing

Dit zijn de opdrachten om het fouilleren in te schakelen:

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

Opmerking: Voordat u **debug**-opdrachten afgeeft, raadpleegt u [Belangrijke informatie over debug-opdrachten](#).

Gerelateerde informatie

- [Onderdruk MIB : RFC 2982](#) 
- [Voorval MIB : RFC 2981](#) 
- [EXPRESSIE-MIB.my / EVENT-MIB.my](#)
- [IOS-functiehandleiding: MIB-ondersteuning voor gebeurtenissen](#)
- [Technische ondersteuning - Cisco-systemen](#)