

# Expressão MIB e exemplo de configuração do MIB de evento

## Índice

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Convenções](#)

[Informações de Apoio](#)

[Configurar](#)

[A expressão MIB](#)

[MIB de evento](#)

[Verificar](#)

[Troubleshooting](#)

[Comandos para Troubleshooting](#)

[Informações Relacionadas](#)

## [Introdução](#)

Este documento mostra como combinar a expressão MIB e o MIB de evento para o uso no gerenciamento de defeito. O exemplo incluído não é realístico mas mostra muitas características disponíveis.

O roteador deve executar duas ações:

1. Envie uma armadilha se uma interface de loopback tem uma largura de banda mais altamente de 100 e esteja administrativamente para baixo
2. A interface de loopback fechou se uma das relações tem sua instrução de largura de banda mudada de um valor definido

O exemplo é mostrado com largura de banda e status administrativo porque são fáceis de manipular da linha de comando e para mostrar o inteiro e os valores de booleano.

Os comandos neste documento usam o parâmetro do identificador de objeto (OID) e não os nomes de objeto. Isto reserva testar sem carregar o MIB.

## [Pré-requisitos](#)

### [Requisitos](#)

Antes de usar a informação neste documento, assegure-se de que você encontre as seguintes

condições prévias:

- A estação de trabalho deve ter as ferramentas do Simple Network Management Protocol (SNMP) fornecidas por Hewlett-Packard (HP) Openview. Outras ferramentas SNMP funcionam mas podem ter a sintaxe diferente.
- O dispositivo deve executar o Software Release 12.2(4)T3 ou Mais Recente de Cisco IOS®. As versões anterior não apoiam a versão de RFC do MIB de evento.
- A plataforma deve apoiar o MIB de evento. Para uma lista de plataformas suportadas para o Cisco IOS Software Release 12.1(3)T, refira a seção da “plataforma suportada” do [apoio do MIB de evento](#).

## Componentes Utilizados

As informações neste documento são baseadas nestas versões de software e hardware:

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

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a sua rede estiver ativa, certifique-se de que entende o impacto potencial de qualquer comando.

## Convenções

Para obter mais informações sobre convenções de documento, consulte as [Convenções de dicas técnicas Cisco](#).

## Informações de Apoio

- A expressão MIB permite que o usuário crie seu próprio objeto MIB baseado em uma combinação de outros objetos. Para mais informação, refira o [RFC 2982](#).
- O MIB de evento permite que o usuário tenha o dispositivo que monitora seus próprios objetos MIB e gerencia as ações (notificação ou **comandos SNMP SET**) baseadas em um evento definido. Para mais informação, refira o [RFC 2981](#).

## Configurar

**Note:** Algumas das linhas de código de saída são indicadas sobre duas linhas para caber melhor em sua tela.

Neste exemplo, o ifIndex da interface de loopback é igual 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
```

Os nomes variável relativos ao primeiro começo do evento com e1 e aqueles relativos ao segundo começo com o e2. O nome de roteador é “roteador” e o string de comunidade de leitura/gravação

é “privado.”

## A expressão MIB

### Criando a expressão 1

Crie primeiramente uma expressão que retorne um valor de 1 se a circunstância, `ifSpeed` é maior de 100,000 E o `ifAdminStatus` esteja para baixo para a interface de loopback. Se a circunstância não é estada conforme, retorna o valor 0.

1. [expExpressionDeltaInterval](#) — Este objeto não é usado. Não há nenhuma razão calcular uma expressão quando não é votada. Se nenhum valor é ajustado, a expressão está calculada quando o objeto é perguntado. O nome da expressão é `e1exp`, que na tabela ASCII corresponde a `101 49 101 120 112`.

2. [expNameStatus](#) — Isto destrói uma expressão velha eventual que seja criada.  

```
# 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](#) — Crie e espere.  

```
# 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](#) — Isto cria o deslocamento predeterminado para usar-se mais tarde para recuperar o resultado da expressão.  

```
# 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](#) — Aqui `.1` (o `expExpressionIndex` escolhido) são a descrição da expressão.  

```
# 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](#) — Esta é a expressão própria, as variáveis `$1` e `$2` é definida na próxima etapa. Os únicos operadores permitidos são (para detalhes, refira o [RFC 2982](#)):  

```
# snmpset -v 2c -c private router 1.3.6.1.4.1.9.10.22.1.3.1.1.4.1 octetstring "e1 expression"
```

```
# 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](#)  

```
# 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'
```

```
# 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](#) — Os dois valores são valores absolutos recolhidos (para o delta, tome 2 como o valor).

```
# 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](#) — As identificações de objeto não wildcarded. Este é o valor padrão, não faz assim expObjectIDWildcard do snmpset.

10. [expObjectStatus](#) — Ajuste as fileiras no expObjectTable ao 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. Ative a expressão 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
```

## [Testando a expressão 1](#)

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

1. Se a circunstância é estada conforme, o valor de [expValueCounter32Val](#) é 1 (como o valor do [expExpressionValueType](#) permanece inalterado, o resultado é um counter32).**Note:** O tipo não pode ser um valor do ponto flutuante.

```
# 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 a circunstância não é estada conforme, o valor é 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 a circunstância não é estada conforme, o valor é 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
```

## [Expressão criadora e de teste 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](#) — Isto indica que 1.3.6.1.2.1.2.2.1.5 são uma tabela e não um objeto.

```
# 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. Teste:

```
# 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 de evento

### Criando o evento 1

Crie agora um evento que verifique o valor de saída da primeira expressão cada 60 segundos e o compare com uma referência. Quando a referência combina o valor da expressão, uma armadilha está provocada com o VARBIND escolhido.

1. Crie o disparador na tabela do disparador. O nome do disparador é trigger1, que no código ASCII é 116 114 105 103 103 101 114 49. O proprietário é tom: 116 111 109. O deslocamento predeterminado do mteTriggerEntry é composto do proprietário do disparador e do nome do disparador. O primeiro valor do deslocamento predeterminado dá o número de caracteres para o mteOwner. Neste caso, há três caracteres para tom, assim que o deslocamento predeterminado é: 3.116.111.109.116.114.105.103.103.101.114.49.
2. Destrua a entrada velha se existe.
3. Ajuste o estado do disparador para criar e esperar.
4. A última etapa ativa-o: [mteTriggerEntryStatus](#)

```
# 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](#) — O valor da primeira expressão é  $e_{1exp}$ . O identificador de objeto do objeto MIB é esse a provar para ver se o disparador atear fogo.

```
# 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](#) — Sem usar um convite para o valor ID.

```
# 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
```

[mais mteTriggerTest](#) — Existência (0), (1) booleano, e ponto inicial (2). O método para selecionar um dos valores acima é um complexo. Para selecionar uma existência, dê um valor em oito dígitos em que o primeiro é um 1, tal como 10000000 ou 100xxxxx. Para um booleano, o segundo dígito deve ser um 1: 01000000 ou 010xxxxx. Para um ponto inicial, o terceiro dígito deve ser um 1: 00100000 ou 001xxxxx. É mais fácil trabalhar esta maneira: Para a existência, o valor é octetstringhex — 80. Para booleano, o valor é octetstringhex — 40. Para o ponto inicial, o valor é octetstringhex — 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](#) — Isto determina o número de segundos esperar entre amostras do disparador. O valor mínimo é ajustado com o mteResourceSampleMinimum do objeto (o padrão é 60 segundos), abaixando este valor aumenta o USO de CPU, assim que deve ser feito com cuidado.

```
# 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](#) — Estes são o absoluteValue (1) e o deltaValue (2). Neste caso, o valor é absoluto:

```
# 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](#) — Este é um controle que permita que um disparador seja configurado mas não usado. Ajuste-o para retificar (o padrão é falso).

```
# 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
```

Agora que o disparador como criado, define o evento o disparador usar-se-á. O nome do evento é event1.[mteEventEntryStatus](#)

```
# 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](#) — Estes são a notificação (0) e o grupo (1). O processo é o mesmo que para mais mteTriggerTest. A notificação é 10xxxxxxx e grupo é 01xxxxxxx.

```
# 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
```

Esta próxima etapa define o teste a ser feito no objeto selecionado para trigger1.[mteTriggerBooleanComparison](#) — Estes são (1) desigual, igual (2), menos (3), (6) lessOrEqual (4), maior (5), e greaterOrEqual. Neste caso — igual.

```
# 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](#) — Este é o valor a usar-se para o teste. Se o valor de 1.3.6.1.4.1.9.10.22.1.4.1.1.2.1.0.0.0 é igual a 1, a seguir a circunstância está estada conforme.

```
# 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
```

Defina agora o objeto a ser enviado com o evento. [mteTriggerBooleanObjectsOwner](#)

```
# 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"
```

[mteTriggerBooleanObjects](#)

```
# 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"
```

[mteTriggerBooleanEventOwner](#)

```
# 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"
```

[mteTriggerBooleanEvent](#)

```
# 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"
```

Crie a tabela do objeto. Envie o valor de 1.3.6.1.2.1.2.2.1.5.16 como o VARBIND com a armadilha. Objete o [mteObjectsName da](#) tabela — Objects1. [mteObjectsEntryStatus](#)

```
# 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
```

[mteObjectsID](#)

```
# 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](#) — Não há nenhum convite usado.

```
# 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
```

Ative a tabela do objeto.

```
# 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
```

Anexe o objeto ao event1. [Notifique o mteEventName](#) — Event1. [mteEventNotificationObjectsOwner](#)

```
# 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"
```

### [mteEventNotificationObjects](#)

```
# 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"
```

Ative o disparador.

```
# 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
```

Ative o 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
```

## [Armadilha recebida](#)

```
# 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
```

**Note:** O objeto 6 é o VARBIND que foi adicionado.

## [Criando o evento 2](#)

Siga estes passos:

### 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](#) — Este é o valor da primeira expressão e [mteTriggerValueIDWildcard](#). Esta hora, os convites do processo o valor ID, o identificador de objeto do objeto MIB de provar para determinar se o disparador ateia fogo.

```
# 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. [mais mteTriggerTest](#) — Ponto inicial.

```
# 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. [mteTriggerFrequency](#)

```
# 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](#) — Valor do 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. Crie um evento no [mteEventName de](#) //da tabela de evento — 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](#) — O valor 40 é para o grupo, significando isso quando a circunstância é estada conforme, os problemas do roteador um **comando snmp set**. Neste caso, faz o grupo para se, mas poderia igualmente fazer a operação em um 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. Permita o 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. Ajuste o ponto inicial do disparador no deslocamento predeterminado = no [mteTriggerName de](#) //da tabela do disparador — Trigger2. Porque é um ponto inicial, dê valores para circunstâncias de falha e de aumentação. Tome somente à circunstância de aumentação este tempo.

#### 11. [mteTriggerThresholdDeltaRising](#) — Este é o valor de limiar a verificar contra.

```
# 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. [mteTriggerThresholdDeltaRisingEventOwner](#)

```
# 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. [mteTriggerThresholdDeltaRisingEvent](#)

```
# 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](#) — Este é o identificador de objeto do objeto MIB a ajustar-se. Aqui,

ifAdminStatus para a interface de 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](#) — Este é o valor a ajustar-se (2 para para baixo).

```
# 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. Ative o disparador.

```
# 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. Ative o 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
```

## Resultado

```
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
```

**Note:** Aqui, 10.48.71.71 é o endereço do roteador próprio.

## Verificar

Esta seção fornece a informação para usar-se para confirmar a configuração está trabalhando corretamente.

A [Output Interpreter Tool](#) ([somente clientes registrados](#)) oferece suporte a determinados comandos show, o que permite exibir uma análise da saída do 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

## Troubleshooting

Esta seção fornece a informação para usar-se para pesquisar defeitos a configuração.

### Comandos para Troubleshooting

Estes são os comandos permitir a eliminação de erros:

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

**Note:** [Antes de emitir comandos de depuração, consulte Informações Importantes sobre Comandos de Depuração.](#)

### Informações Relacionadas

- [Expressão MIB: RFC 2982](#)
- [MIB de evento: RFC 2981](#)
- [EXPRESSION-MIB.my/EVENT-MIB.my](#)
- [Guia de função IO: Apoio do MIB de evento](#)
- [Suporte Técnico - Cisco Systems](#)