Exemple de configuration d'expression MIB et d'événement MIB

Contenu

Introduction

Ce document affiche comment combiner le MIB d'expression et l'événement MIB pour l'usage en Gestion de défaut. L'exemple inclus n'est pas réaliste mais affiche beaucoup de caractéristiques disponibles.

Le routeur doit exécuter deux actions :

1. Envoyez un déroutement si une interface de bouclage a un supérieur à 100 de bande passante et est administrativement en baisse
2. L'interface de bouclage s'est arrêtée si une des interfaces a son instruction de bande passante changée d'une valeur définie

L'exemple est affiché avec l'état de bande passante et d'admin parce qu'il est facile les manipuler de la ligne de commande et afficher l'entier et les valeurs booléennes.

Les commandes dans ce document utilisent le paramètre de l'identifiant d'objet (OID) et pas les noms d'objet. Ceci permet le test sans charger le MIB.

Conditions préalables

Conditions requises

Avant d'utiliser les informations dans ce document, assurez-vous que vous rencontrez les

conditions préalables suivantes :

- Le poste de travail devrait avoir des outils de Protocole SNMP (Simple Network Management Protocol) fournis par Hewlett-Packard (HP) Openview. D'autres outils SNMP fonctionnent mais peuvent avoir la syntaxe différente.
- Le périphérique doit exécuter la version de logiciel 12.2(4)T3 ou ultérieures de Cisco IOS®. Les versions antérieures ne prennent en charge pas la version RFC de l'événement MIB.
- La plate-forme doit prendre en charge l'événement MIB. Pour une liste de Plateformes prises en charge pour le Logiciel Cisco IOS version 12.1(3)T, référez-vous à la section « de plate-forme prise en charge » de support d'événement MIB.

Composants utilisés

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- Version du logiciel Cisco IOS 12.3(1a)
- Routeur d'accès modulaire de Cisco 3640

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est opérationnel, assurez-vous que vous comprenez l'effet potentiel de toute commande.

Conventions

Pour plus d'informations sur les conventions de documents, reportez-vous à Conventions relatives aux conseils techniques Cisco.

Informations générales

- Le MIB d'expression permet à l'utilisateur pour créer son propre objet MIB basé sur une combinaison d'autres objets. Le pour en savoir plus, se rapportent à RFC 2982 .
- L'événement MIB permet à l'utilisateur pour avoir le périphérique surveillant ses propres objets MIB et pour générer des actions (notification ou commandes de **SNMP SET**) basées sur un événement défini. Le pour en savoir plus, se rapportent à RFC 2981 .

Configurez

Note: Certaines des lignes du code de sortie sont affichées plus de deux lignes pour s'adapter mieux sur votre écran.

Dans cet exemple, l'ifIndex de l'interface de bouclage est égal à 16.

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

Les noms de la variable ont associé au premier début d'événement avec l'E1 et ceux liés au

deuxième début avec `e2`. Le nom de routeur est « routeur » et la chaîne lecture/écriture de la communauté est « privée. »

Le MIB d'expression

Création de l'expression 1

Créez d'abord une expression qui renvoie une valeur de 1 si la condition, `ifSpeed est plus grande que 100,000` ET l'`ifAdminStatus est vers le bas` pour l'interface de bouclage. Si la condition n'est pas remplie, elle renvoie la valeur 0.

1. expExpressionDeltaInterval — Cet objet n'est pas utilisé.Il n'y a aucune raison de calculer une expression quand elle n'est pas votée. Si aucune valeur n'est placée, l'expression est calculée quand l'objet est questionné.Le nom d'expression est `e1exp`, qui dans la table ASCII correspond à `101 49 101 120 112`.
2. expNameStatus — Ceci détruit une vieille expression certaine qui est créée.
   ```
   # 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 — Créez et attendez.
   ```
   # 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 — Ceci crée l'index pour utiliser plus tard pour récupérer le résultat de l'expression.
   ```
   # 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 — Ici .1 (l'expExpressionIndex choisi) est la description de l'expression.
   ```
   # 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 — C'est l'expression elle-même, les variables $1 et $2 sont définies à l'étape suivante.Les seuls opérateurs permis sont (pour des détails, référez-vous à 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 — Les deux valeurs sont des valeurs absolues rentrées (pour le

delta, prenez 2 comme valeur).

```
# 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 — Les object id pas wildcarded. C'est la valeur par défaut, ne font pas ainsi expObjectIDWildcard de snmpset.
10. expObjectStatus — Placez les lignes dans l'expObjectTable à l'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. Lancez l'expression 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
```

Test de l'expression 1

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

1. Si la condition est remplie, la valeur d'expValueCounter32Val est 1 (car la valeur de l'expExpressionValueType demeure sans changement, le résultat est un counter32).**Note**: Le type ne peut pas être un nombre à virgule flottante.

```
# 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. Si la condition n'est pas remplie, la valeur est 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. Si la condition n'est pas remplie, la valeur est 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
```

Expression de création et de test 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 — Ceci indique que `1.3.6.1.2.1.2.2.1.5` est une table et pas un objet.
```
# 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
```

Événement MIB

Création de l'événement 1

Créez maintenant un événement qui vérifie la valeur de sortie de la première expression toutes les 60 secondes et la compare à une référence. Quand la référence apparie la valeur d'expression, un déroutement est déclenché avec le VARBIND choisi.

1. Créez le déclencheur dans la table de déclencheur.Le nom du déclencheur est trigger1, qui en code ASCII est `116 114 105 103 103 101 114 49`.Le propriétaire est Tom : `116 111 109`.L'index du mteTriggerEntry se compose de propriétaire de déclencheur et de nom de déclencheur. La première valeur de l'index donne le nombre de caractères pour le mteOwner.Dans ce cas, il y a trois caractères pour Tom, ainsi l'index est : `3.116.111.109.116.114.105.103.103.101.114.49.`
2. Détruisez la vieille entrée si elle existe.
3. Placez l'état de déclencheur **pour créer et attendre**.
4. La dernière étape le lance :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 — La valeur de la première expression est `e1exp`.L'identifiant d'objet de l'objet MIB est celui à échantillonner pour voir si le déclencheur se déclenche.
```
# 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 — Sans utiliser un masque pour l'ID de valeur.
```
# 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 — Existence (0), (1) booléen, et seuil (2).La méthode pour sélectionner une des valeurs ci-dessus est un complexe. Pour sélectionner une existence, donnez une valeur dans huit chiffres dans lesquels le premier est un 1, tel que `10000000` ou `100xxxxx`.Pour un booléen, le deuxième chiffre doit être un 1 : `01000000` ou `010xxxxx`.Pour un seuil, le troisième chiffre doit être un 1 : `00100000` ou `001xxxxx`.Il est le plus facile de fonctionner de cette façon :Pour l'existence, la valeur est octetstringhex — 80.Pour booléen, la valeur est octetstringhex — 40.Pour le seuil, la valeur est 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 — Ceci détermine le nombre de secondes pour attendre entre les échantillons de déclencheur.La valeur minimum est placée avec le mteResourceSampleMinimum d'objet (le par défaut est de 60 secondes), diminuant cette valeur augmente l'utilisation du CPU, ainsi elle doit être faite soigneusement.

```
# 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 — Ce sont l'absoluteValue (1) et le deltaValue (2). Dans ce cas, la valeur est absolue :

```
# 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 — C'est un contrôle qui permet un déclencheur à configurer mais ne pas être utilisé. Placez-le pour rectifier (le par défaut est faux).

```
# 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
```

Maintenant que le déclencheur comme créé, définissent l'événement le déclencheur l'utilisera. Le nom d'événement est 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 — Ce sont la notification (0) et le positionnement (1).Le processus est identique que pour mteTriggerTest. La notification est `10xxxxxx` et positionnement est `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
```

Cette étape suivante définit le test à faire sur l'objet sélectionné pour trigger1.mteTriggerBooleanComparison — Ce sont (1) inégal, égal (2), moins (3), (4) lessOrEqual, plus grand (5), et (6) greaterOrEqual. Dans ce cas — égal.

```
# 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 — C'est la valeur à l'utiliser pour le test. Si la valeur de `1.3.6.1.4.1.9.10.22.1.4.1.1.2.1.0.0.0` est égale à 1, alors la condition est remplie.

```
# 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
```

Définissez maintenant l'objet à envoyer avec l'événement.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"
```

Créez la table d'objet.Envoyez la valeur de `1.3.6.1.2.1.2.2.1.5.16` comme VARBIND avec le déroutement.MteObjectsName de Tableau d'objet — 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 — Il n'y a aucun masque utilisé.

```
# 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
```

Lancez la table d'objet.

```
# 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
```

Reliez l'objet à l'event1.Informez le 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"
```

Lancez le déclencheur.

```
# 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
```

Lancez l'événement.

```
# 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
```

Déroutement reçu

```
# 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: L'objet 6 est le VARBIND qui a été ajouté.

Création de l'événement 2

Suivez ces étapes :

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 — C'est la valeur de la premiers expression et mteTriggerValueIDWildcard.Cette fois, les masques de processus l'ID de valeur, l'identifiant d'objet de l'objet MIB d'échantillonner pour déterminer si le déclencheur se déclenche.

```
# 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 — Seuil.

```
# 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 — Valeur de 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. Créez un mteEventName de //de table d'événement en cas — 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 — La valeur 40 est pour le positionnement, signifiant cela quand la condition est remplie, les problèmes de routeur une commande de **snmp set**. Dans ce cas, il fait le positionnement pour lui-même, mais il pourrait également faire l'exécution sur un périphérique distant.
```
# 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. Activez l'événement.
```
# 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. Placez le seuil de déclencheur dans l'index = le mteTriggerName de //de Tableau de déclencheur — Trigger2.Car c'est un seuil, donnez les valeurs pour des conditions manquantes et en hausse. Prise seulement la condition en hausse cette fois.
11. mteTriggerThresholdDeltaRising — C'est la valeur seuil à vérifier contre.
```
# 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 — C'est l'identifiant d'objet de l'objet MIB à placer. Ici, ifAdminStatus

pour l'interface de bouclage.

```
# 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 — C'est la valeur à placer (2 pour vers le bas).

```
# 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. Lancez le déclencheur.

```
# 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. Lancez l'événement.

```
# 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
```

Résultat

```
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: Ici, `10.48.71.71` est l'adresse du routeur elle-même.

Vérifiez

Cette section fournit des informations pour l'utiliser pour confirmer la configuration fonctionne correctement.

Certaines commandes **show** sont prises en charge par l'Output Interpreter Tool (clients enregistrés uniquement), qui vous permet de voir une analyse de la sortie de la commande 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
```

Dépannez

Cette section fournit des informations pour l'utiliser pour dépanner la configuration.

Dépannage des commandes

Ce sont les commandes d'activer l'élimination des imperfections :

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

Note: Avant d'émettre des commandes **debug**, reportez-vous aux Informations importantes sur les commandes de débogage.

Informations connexes

- MIB d'expression : RFC 2982
- Événement MIB : RFC 2981
- EXPRESSION-MIB.my/EVENT-MIB.my
- Guide de fonctionnalité IOS : Support d'événement MIB
- Support technique - Cisco Systems