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

Contenu

Introduction

Ce document montre comment combiner la MIB Expression et la MIB Event pour une utilisation dans la gestion des pannes. L'exemple inclus n'est pas réaliste mais montre de nombreuses fonctionnalités disponibles.

Le routeur doit effectuer deux actions :

1. Envoyer un déroutement si une interface de bouclage a une bande passante supérieure à 100 et est désactivée administrativement
2. L'interface de bouclage s'arrête si l'instruction de bande passante de l'une des interfaces est modifiée à partir d'une valeur définie

L'exemple est illustré avec l'état de bande passante et d'administration, car ils sont faciles à manipuler à partir de la ligne de commande et affichent des valeurs entières et booléennes.

Les commandes de ce document utilisent le paramètre d'identificateur d'objet (OID) et non les noms d'objet. Cela permet de tester sans charger la MIB.

Conditions préalables

Conditions requises

Avant d'utiliser les informations de ce document, assurez-vous de respecter les conditions

préalables suivantes :

- La station de travail doit disposer d'outils SNMP (Simple Network Management Protocol) fournis par Hewlett-Packard (HP) Openview. D'autres outils SNMP fonctionnent mais peuvent avoir une syntaxe différente.
- Le périphérique doit exécuter le logiciel Cisco IOS® Version 12.2(4)T3 ou ultérieure. Les versions antérieures ne prennent pas en charge la version RFC de la MIB d'événements.
- La plate-forme doit prendre en charge la MIB d'événements. Pour obtenir la liste des plates-formes prises en charge par le logiciel Cisco IOS Version 12.1(3)T, reportez-vous à la section « Plate-forme prise en charge » de la prise en charge de la base MIB d'événements.

Components Used

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

- Logiciel Cisco IOS Version 12.3(1a)
- Routeur d'accès modulaire Cisco 3640

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Conventions

For more information on document conventions, refer to the Cisco Technical Tips Conventions.

Informations générales

- La MIB Expression permet à l'utilisateur de créer son propre objet MIB en fonction d'une combinaison d'autres objets. Pour plus d'informations, reportez-vous à la RFC 2982 .
- La base MIB d'événements permet à l'utilisateur de demander au périphérique de surveiller ses propres objets MIB et de générer des actions (commandes de notification ou **SNMP SET**) en fonction d'un événement défini. Pour plus d'informations, reportez-vous à la RFC 2981 .

Configuration

Remarque : Certaines lignes du code de sortie sont affichées sur deux lignes pour mieux s'insérer dans votre écran.

Dans cet exemple, 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 variable liés au premier événement commencent par `e1` et ceux liés au deuxième début par `e2`. Le nom du routeur est « router » et la chaîne de communauté en lecture/écriture est « private ».

MIB Expression

Création de l'expression 1

Créez d'abord une expression qui renvoie une valeur de 1 si la condition, `ifSpeed est supérieure à 100 000` AND `ifAdminStatus est désactivée` 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 lorsqu'elle n'est pas interrogée. Si aucune valeur n'est définie, l'expression est calculée lorsque l'objet est interrogé.Le nom de l'expression est `e1exp`, qui dans le tableau ASCII correspond à `101 49 101 120 112`.
2. expNameStatus : cette option détruit une ancienne expression 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 : crée l'index à utiliser ultérieurement 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 (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 autorisés sont (pour plus de détails, reportez-vous à la 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 : les deux valeurs sont prises en valeurs absolues (pour 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 ID d'objet ne sont pas cardés en caractères génériques. Il s'agit de la valeur par défaut, donc ne pas snmpset expObjectIDWildcard.
10. expObjectStatus : définissez les lignes de expObjectTable sur 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. Activez 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 (comme la valeur d'expExpressionValueType reste inchangée, le résultat est un compteur32).**Remarque :** le type ne peut pas être une valeur en 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
   ```

Création et test d'Expression 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 : indique que 1.3.6.1.2.1.2.2.1.5 est une table et non 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
```

MIB d'événements

Création de l'événement 1

Maintenant, créez un événement qui vérifie la valeur de sortie de la première expression toutes les 60 secondes et la compare à une référence. Lorsque la référence correspond à la valeur de l'expression, un déroutement est déclenché avec le VARBIND sélectionné.

1. Créez le déclencheur dans la table de déclencheurs.Le nom du déclencheur est trigger1, qui dans le code ASCII est `116 114 105 103 103 101 114 49`.Le propriétaire est tom : `116 111 109`.L'index de mteTriggerEntry est composé du propriétaire du déclencheur et du nom du déclencheur. La première valeur de l'index donne le nombre de caractères du mteOwner.Dans ce cas, il y a trois caractères pour tom, l'index est donc : `3.116.111.109.116.114.105.103.103.101.114.49 .`
2. Détruisez l'ancienne entrée si elle existe.
3. Définissez l'état du déclencheur pour **créer et attendre**.
4. La dernière étape l'active :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'identificateur d'objet de l'objet MIB est celui à échantillonner pour voir si le déclencheur doit s'activer.

```
# 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 de caractère générique 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), booléen (1) et seuil (2).La méthode de sélection de l'une des valeurs ci-dessus est complexe. Pour sélectionner une existence, donnez une valeur en

huit chiffres dans lesquels le premier est un 1, tel que `10000000` ou `100xxx`.Pour un booléen, le deuxième chiffre doit être un 1 : `0100000` ou `010xxxxx`.Pour un seuil, le troisième chiffre doit être un 1 : `0010000` ou `001xxxxx`.Il est plus facile de fonctionner de cette manière :Pour exister, la valeur est octetstringhex—80.Pour le 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 : détermine le nombre de secondes à attendre entre les échantillons de déclencheurs.La valeur minimale est définie avec l'objet mteResourceSampleMinimum (la valeur par défaut est de 60 secondes), la diminution de cette valeur augmente l'utilisation du CPU, donc il faut le faire avec précaution.

```
# 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 : il s'agit de absoluteValue (1) et 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 : contrôle qui permet de configurer un déclencheur mais qui n'est pas utilisé. Définissez-la sur true (la valeur par défaut est 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
```

Maintenant que le déclencheur a été créé, définissez l'événement que le déclencheur utilisera. Le nom de l'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 : il s'agit de notifications (0) et de set (1).Le processus est le même que pour mteTriggerTest. La notification est `10xxxxxx` et définie 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 effectuer sur l'objet sélectionné pour trigger1.mteTriggerBooleanComparison : ces valeurs sont inégales (1), égales (2), inférieures (3), inférieuresOrEqual (4), supérieures (5) et supérieuresOrEqual (6). 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 : valeur à utiliser pour le test. Si la valeur du `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'objets.Envoyez la valeur `1.3.6.1.2.1.2.2.1.5.16` comme VARBIND avec le déroutement.Table d'objets mteObjectsName —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 : aucun caractère générique n'est 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
```

Activez la table d'objets.

```
# 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
```

Associez l'objet à l'événement1.Notify 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"
```

Activez 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
```

Activez 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
```

Interruption reçue

```
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
```

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 : valeur de la première expression et de mteTriggerValueIDWildcard.Cette fois-ci, le processus masque l'ID de valeur, l'identificateur d'objet de l'objet MIB à 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 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 événement dans la table d'événements // mteEventName —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 définie pour Set, ce qui signifie que lorsque la condition est remplie, le routeur émet une commande **snmp set**. Dans ce cas, il crée l'ensemble pour lui-même, mais il peut également effectuer l'opération 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. Définissez le seuil de déclenchement dans la table de déclenchement // index = mteTriggerName—Trigger2.Comme il s'agit d'un seuil, donnez des valeurs pour les conditions défaillantes et montantes. Ne prenez que l'état en hausse cette fois.
11. mteTriggerThresholdDeltaRising : valeur de seuil à vérifier.
```
# 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 : identificateur d'objet de l'objet MIB à définir. 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 : valeur à définir (2 pour 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. Activez 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. Activez 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
```

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

Vérification

Cette section fournit des informations à utiliser pour confirmer que 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
```

Dépannage

Cette section fournit des informations à utiliser pour dépanner la configuration.

Dépannage des commandes

Voici les commandes permettant d'activer le débogage :

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

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

Informations connexes

- MIB d'expression : RFC 2982
- MIB d'événements : RFC 2981
- EXPRESSION-MIB.my / EVENT-MIB.my
- Guide des fonctionnalités IOS : Prise en charge MIB d'événements
- Support technique - Cisco Systems