Ejemplo de Configuración de MIB de Expresión y MIB de Evento

Contenido

Introducción

Este documento muestra cómo combinar la MIB de Expresión y la MIB de Evento para su uso en la administración de fallas. El ejemplo incluido no es realista, pero muestra muchas funciones disponibles.

El router debe realizar dos acciones:

1. Enviar una trampa si una interfaz de loopback tiene un ancho de banda superior a 100 y está administrativamente inactiva
2. La interfaz de loopback se apaga si una de las interfaces tiene su sentencia de ancho de banda cambiada de un valor definido

El ejemplo se muestra con ancho de banda y estado de administrador porque son fáciles de manipular desde la línea de comandos y muestran valores enteros y booleanos.

Los comandos de este documento utilizan el parámetro identificador de objeto (OID) y no los nombres de objeto. Esto permite la prueba sin cargar la MIB.

Prerequisites

Requirements

Antes de utilizar la información de este documento, asegúrese de cumplir con los siguientes

requisitos previos:

- La estación de trabajo debe tener herramientas SNMP (Simple Network Management Protocol, protocolo simple de administración de red) proporcionadas por Hewlett-Packard (HP) Openview. Otras herramientas SNMP funcionan pero pueden tener una sintaxis diferente.
- El dispositivo debe ejecutar Cisco IOS® Software Release 12.2(4)T3 o posterior. Las versiones anteriores no soportan la versión RFC de la MIB de Evento.
- La plataforma debe soportar la MIB de Evento. Para obtener una lista de las plataformas soportadas para la versión 12.1(3)T del software del IOS de Cisco, refiérase a la sección "Plataforma Soportada" del Soporte de MIB de Evento.

Componentes Utilizados

La información que contiene este documento se basa en las siguientes versiones de software y hardware.

- Versión 12.3(1a) del software del IOS de Cisco
- Router de acceso modular 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.

Convenciones

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

Antecedentes

- La MIB de expresión permite al usuario crear su propio objeto MIB basado en una combinación de otros objetos. Para obtener más información, consulte RFC 2982 .
- La MIB de Evento permite al usuario tener el dispositivo monitoreando sus propios objetos MIB y generar acciones (**comandos** de notificación o **SNMP SET**) basadas en un evento definido. Para obtener más información, consulte RFC 2981 .

Configurar

Nota: Algunas líneas del código de salida se muestran en dos líneas para que encajen mejor en la pantalla.

En este ejemplo, el ifIndex de la interfaz de loopback es 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
```

Los nombres de variables relacionados con el primer evento comienzan con `e1` y los relacionados con el segundo comienzan con `e2`. El nombre del router es "router" y la cadena de comunidad de

lectura/escritura es "private".

MIB de expresión

Creación de la expresión 1

Primero cree una expresión que devuelva un valor de 1 si la condición, `ifSpeed es mayor que 100,000` Y `ifAdminStatus está desactivado` para la interfaz de loopback. Si no se cumple la condición, devuelve el valor 0.

1. expExpressionDeltaInterval: este objeto no se utiliza.No hay razón para calcular una expresión cuando no se sondea. Si no se establece ningún valor, la expresión se calcula cuando se consulta el objeto.El nombre de la expresión es `e1exp`, que en la tabla ASCII corresponde a `101 49 101 120 112`.
2. expNameStatus: esto destruye una expresión antigua que se crea.
   ```
   # 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: Crear y esperar.
   ```
   # 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: se crea el índice que se utilizará más adelante para recuperar el resultado de la expresión.
   ```
   # 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: Aquí .1 (el expExpressionIndex seleccionado) es la descripción de la expresión.
   ```
   # 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: es la expresión misma, las variables $1 y $2 se definen en el siguiente paso.Los únicos operadores permitidos son (para obtener detalles, consulte 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: los dos valores se toman en valores absolutos (para Delta, tome 2 como 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: los ID de objeto no están comodín. Este es el valor predeterminado, por lo que no snmpset expObjectIDWildcard.
10. expObjectStatus: establezca las filas de expObjectTable en 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. Active la expresión 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
    ```

Prueba de la expresión 1

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

1. Si se cumple la condición, el valor de expValueCounter32Val es 1 (ya que el valor de expExpressionValueType permanece inalterado, el resultado es counter32).**Nota:** El tipo no puede ser un valor de coma flotante.
   ```
   # 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 no se cumple la condición, el valor es 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 no se cumple la condición, el valor es 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
   ```

Creación y prueba de la expresión 2

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

1. expObjectIDWildcard: indica que 1.3.6.1.2.1.2.2.1.5 es una tabla y no un 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. Prueba:

```
# 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
```

Evento MIB

Creación del evento 1

Ahora, cree un evento que verifique el valor de salida de la primera expresión cada 60 segundos y lo compare con una referencia. Cuando la referencia coincide con el valor de expresión, se activa una trampa con el VARBIND seleccionado.

1. Cree el disparador en la tabla del disparador.El nombre del disparador es trigger1, que en código ASCII es `116 114 105 103 103 101 114 49`.El propietario es tom: `116 111 109`.El índice de mteTriggerEntry se compone del propietario del disparador y del nombre del disparador. El primer valor del índice indica el número de caracteres del mteOwner.En este caso, hay tres caracteres para tom, por lo que el índice es:
   `3.116.111.109.116.114.105.103.103.101.114.49.`
2. Destruya la entrada antigua si existe.
3. Establezca el estado del disparador para **crear y esperar**.
4. El último paso lo activa: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: el valor de la primera expresión es `e1exp`.El identificador de objeto del objeto MIB es el que se muestra para ver si el disparador debe activarse.

```
# 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 : sin utilizar un comodín para el 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
```

   mteTriggerTest: existencia (0), booleana (1) y umbral (2).El método para seleccionar uno de los valores anteriores es complejo. Para seleccionar una existencia, asigne un valor en ocho

dígitos en los que el primero sea un 1, como `1000000` o `100xxxxx`.Para un booleano, el segundo dígito debe ser un 1: `0100000` o `010xxxxx`.Para un umbral, el tercer dígito debe ser un 1: `0010000` o `001xxxxx`.Es más fácil trabajar de esta manera:Para la existencia, el valor es octetstringhex—80.Para booleano, el valor es octetstringhex—40.Para el umbral, el valor es 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: determina el número de segundos de espera entre las muestras de disparador.El valor mínimo se establece con el objeto mteResourceSampleMinimum (el valor predeterminado es 60 segundos), al reducir este valor se aumenta el uso de la CPU, por lo que se debe hacer con 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: valores absolutos (1) y valor delta (2). En este caso, el valor es 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: control que permite configurar un disparador pero no utilizarlo. Configúrelo en true (el valor predeterminado es 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
```

Ahora que se ha creado el disparador, defina el evento que utilizará el disparador. El nombre del evento es 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: son la notificación (0) y el conjunto (1).El proceso es el mismo que para mteTriggerTest. La notificación es `10xxxxxx` y el conjunto es `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
```

Este paso siguiente define la prueba que se realizará en el objeto seleccionado para trigger1.mteTriggerBooleanComparison: son desiguales (1), iguales (2), menores (3), menores o iguales (4), mayores (5) y mayoresOiguales (6). En este 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: valor que se utiliza para la prueba. Si el valor de `1.3.6.1.4.1.9.10.22.1.4.1.1.2.1.0.0.0` es igual a 1, se cumple la condición.

```
# 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
```

Ahora defina el objeto que se enviará con el 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"
```

Cree la tabla de objetos.Enviar el valor de `1.3.6.1.2.1.2.2.1.5.16` como VARBIND con la trampa.Object Table 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: no se utiliza ningún comodín.

```
# 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
```

Active la tabla de objetos.

```
# 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
```

Asocie el objeto al evento1.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"
```

Activa el 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
```

Active el 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
```

Trampa recibida

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

Nota: El objeto 6 es el VARBIND que se agregó.

Creación del evento 2

Siga estos pasos:

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: valor de la primera expresión y de mteTriggerValueIDWildcard.Esta vez, el proceso descarta el ID de valor, el identificador de objeto del objeto MIB que se muestra para determinar si se activa el disparador.

```
# 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: Umbral.
```
# 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 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. Cree un evento en la tabla de eventos // 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: el valor 40 es para Set, lo que significa que cuando se cumple la condición, el router emite un **comando snmp set**. En este caso, hace el Set por sí mismo, pero también podría realizar la operación en un dispositivo remoto.
```
# snmpset -v 2c -c private router
1.3.6.1.2.1.88.1.4.2.1.3.3.116.111.109.101.118.101.110.116.50
octetstringhex "40"
```

9. Habilite el 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. Establezca El Umbral del Disparador en la Tabla del Disparador // índice = mteTriggerName—Trigger2.Dado que se trata de un umbral, proporcione valores para las condiciones que fallan y que aumentan. Tomemos sólo la condición de ascenso esta vez.
11. mteTriggerThresholdDeltaRising: valor de umbral para comprobar.
```
# 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: identificador de objeto del objeto MIB que se va a establecer. Aquí, ifAdminStatus para la interfaz 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: valor que se debe establecer (2 para 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. Activa el 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. Active el 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
```

Nota: Aquí, `10.48.71.71` es la dirección del propio router.

Verificación

Esta sección proporciona información que se debe utilizar para confirmar que la configuración funciona correctamente.

La herramienta Output Interpreter (sólo para clientes registrados) permite utilizar algunos comandos “show” y ver un análisis del resultado de estos comandos.

```
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
```

Troubleshoot

Esta sección proporciona información para solucionar problemas de configuración.

Comandos para resolución de problemas

Estos son los comandos para habilitar la depuración:

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

Nota: Antes de ejecutar **comandos debug**, consulte Información Importante sobre Comandos Debug.

Información Relacionada

- MIB de Expresión: RFC 2982
- Evento MIB: RFC 2981
- EXPRESSION-MIB.my / EVENT-MIB.my
- Guía de funciones de IOS: Soporte de evento MIB
- Soporte Técnico - Cisco Systems