Expresión MIB y ejemplo de configuración del evento MIB

Contenido

Introducción
prerrequisitos
Requisitos
Componentes Utilizados
Convenciones
Antecedentes
Configurar
La expresión MIB
Evento MIB
Verificación
Troubleshooting
Comandos para resolución de problemas
Información Relacionada

Introducción

Este documento muestra cómo combinar la expresión MIB y el evento MIB para el uso en administración de fallas. El ejemplo incluido no es realista sino muestra muchas características disponibles.

El router debe realizar dos acciones:

1. Envíe un desvío si un Loopback Interface tiene un ancho de banda más arriba de 100 y está administrativo abajo
2. El Loopback Interface apaga si una de las interfaces tiene su sentencia de ancho de banda cambiada de un valor definido

El ejemplo se muestra con el ancho de banda y el estado del administrador porque son fáciles de manipular de la línea de comando y mostrar el número entero y los valores booleanos.

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

prerrequisitos

Requisitos

Antes de usar la información en este documento, asegúrese de que usted resuelve los requisitos

previos siguientes:

- El puesto de trabajo debe tener herramientas del Simple Network Management Protocol (SNMP) proporcionadas por Hewlett-Packard (HP) Openview. Otras herramientas SNMP funcionan pero pueden tener diverso sintaxis.
- El dispositivo debe funcionar con el Software Release 12.2(4)T3 o Posterior de Cisco IOS®. Las versiones anteriores no soportan la versión RFC del evento MIB.
- La plataforma debe soportar el evento MIB. Para una lista de plataformas admitidas para el Cisco IOS Software Release 12.1(3)T, refiera a la sección de la “plataforma admitida” del soporte del evento MIB.

Componentes Utilizados

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

- Cisco IOS Software Release 12.3(1a)
- Router de acceso modular del Cisco 3640

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si la red está funcionando, asegúrese de haber comprendido el impacto que puede tener cualquier comando.

Convenciones

Para obtener más información sobre las convenciones del documento, consulte las Convenciones de Consejos Técnicos de Cisco.

Antecedentes

- La expresión MIB permite que el usuario cree su propio objeto de MIB basado en una combinación de otros objetos. Para más información, refiera al RFC 2982 .
- El evento MIB permite que el usuario tenga el dispositivo que monitorea sus propios objetos de MIB y genere las acciones (notificación o los **comandos SNMP SET**) basadas en un evento definido. Para más información, refiera al RFC 2981 .

Configurar

Nota: Algunas de las líneas de código de salida se visualizan sobre dos líneas para caber mejor sobre su pantalla.

En este ejemplo, el ifIndex del Loopback Interface 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 variable relacionados con el primer comienzo del evento con el `e1` y ésos relacionados con el segundo comienzo con `e2`. El nombre del router es “router” y la cadena de comunidad de lectura/grabación es “soldado.”

La expresión MIB

Crear la expresión 1

Primero cree una expresión que vuelva un valor de 1 si la condición, `ifSpeed es mayor de 100,000` Y el `ifAdminStatus está abajo` para el Loopback Interface. Si la condición no se cumple, vuelve el valor 0.

1. expExpressionDeltaInterval — Este objeto no se utiliza.No hay razón para calcular una expresión cuando no se sondea. Si no se fija ningún valor, se calcula la expresión cuando se pregunta 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 vieja expresión eventual se cree que.`# 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 — Cree y 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 — Esto crea el índice para utilizar más adelante para extraer 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 elegido) 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 — Ésta es la expresión sí mismo, las variables $1 y $2 se define en el siguiente paso.Los únicos operadores permitidos son (para los detalles, refiera al 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 son valores absolutos admitidos (para el delta, tome 2 como el 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 del objeto no wildcarded. Éste es el valor predeterminado, no hace tan expObjectIDWildcard del snmpset.
10. expObjectStatus — Fije las filas en el expObjectTable al 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 (como sigue habiendo el valor del expExpressionValueType sin cambiar, el resultado es un counter32).**Nota:** El tipo no puede ser un valor del punto 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 la condición no se cumple, 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 la condición no se cumple, 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

Expresión que crea y de prueba 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 — Esto 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

Crear el evento 1

Ahora cree un evento que marque el valor del resultado de la primera expresión cada 60 segundos y lo compare con una referencia. Cuando la referencia hace juego el valor de la expresión, un desvío se acciona con el VARBIND elegido.

1. Cree el activador en la tabla del activador.El nombre del activador es trigger1, que en el código ASCII es 116 114 105 103 103 101 114 49.El propietario es tom: 116 111 109.El índice del mteTriggerEntry se compone del propietario del activador y del nombre del activador. El primer valor del índice da el número de caracteres para el mteOwner.En este caso, hay tres caracteres para tom, así que el índice es: 3.116.111.109.116.114.105.103.103.101.114.49.
2. Destruya la vieja entrada si existe.
3. Fije el estatus del activador **para crear y para esperar**.
4. El paso más reciente 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 de MIB es el que está a muestrear para ver si el activador enciende.# **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 usar a 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** el más mteTriggerTest — Existencia (0), (1) boleano, y umbral (2).El método para seleccionar uno

de los valores antedichos es un complejo. Para seleccionar una existencia, dé un valor en ocho dígitos en los cuales el primer sea a1, tal como `10000000` o `100xxxxx`.Para un boleano, el segundo dígito debe ser a1: `01000000` o `010xxxxx`.Para un umbral, el tercer dígito debe ser a1: `00100000` o `001xxxxx`.Es el más fácil trabajar esta manera:Para la existencia, el valor es octetstringhex — 80.Para boleano, 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 — Esto determina el número de segundos para esperar entre las muestras del activador.El valor mínimo se fija con el mteResourceSampleMinimum del objeto (el valor por defecto es 60 segundos), bajando este valor aumenta el USO de la CPU, así que debe ser hecho cuidadosamente.`# 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 — Éstos son el absoluteValue (1) y el deltaValue (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 — Éste es un control que permite que no utilizan un activador sea configurado pero. Fíjelo para verdad (el valor por defecto es 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` Ahora que el activador según lo creado, define el evento el activador utilizará. 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 — Éstos son la notificación (0) y el conjunto (1).El proceso es lo mismo que para la más mteTriggerTest. La notificación es `10xxxxxx` y 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 siguiente paso define la prueba que se hará en el objeto seleccionado para trigger1.mteTriggerBooleanComparison — Éstos son (1) desigual, igual (2), menos (3), (4) lessOrEqual, mayor (5), y (6) greaterOrEqual. 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 — Éste es el valor a utilizar 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, después 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 del objeto.Envíe el valor de `1.3.6.1.2.1.2.2.1.5.16` como VARBIND con el desvío.Opóngase el mteObjectsName de la tabla — 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 hay comodín 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 Active la tabla del 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** Asocie el objeto al event1.Notifique el 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"** Active el activador.**# 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**

Desvío recibido

```
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 fue agregado.

Crear el 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 — Éste es el valor de la primera expresión y mteTriggerValueIDWildcard.Esta hora, los comodines de proceso el valor ID, el identificador de objeto del objeto de MIB de muestrear para determinar si el activador enciende.**# 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. el más 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 del 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 el mteEventName de //de la tabla de eventos — 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 está para el conjunto, significando eso cuando se cumple la condición, los problemas del router un **comando snmp set**. En este caso, hace el conjunto para sí mismo, pero podría también hacer 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. Fije el umbral del activador en el índice = el mteTriggerName de //de la tabla del activador — Trigger2.Pues es un umbral, dé los valores para las condiciones que fallan y de levantamientos. Tarde solamente a condición de levantamiento este vez.
11. el mteTriggerThresholdDeltaRising — Éste es el valor de umbral a marcar 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 — Éste es el identificador de objeto del objeto de MIB a fijar. Aquí, ifAdminStatus para el Loopback Interface.**# snmpset -v 2c -c private router 1.3.6.1.2.1.88.1.4.4.1.1.3.116.111.109.101.118.101.110.116.50 objectidentifier 1.3.6.1.2.1.2.2.1.7.16**
15. mteEventSetValue — Éste es el valor a fijar (2 para abajo).**# 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. Active el activador.**# 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 el direccionamiento del router sí mismo.

Verificación

Esta sección proporciona la información para utilizar para confirmar la configuración está trabajando 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
```

Troubleshooting

Esta sección proporciona la información para utilizar para resolver problemas la configuración.

Comandos para resolución de problemas

Éstos son los comandos de habilitar el debugging:

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

Nota: Antes de ejecutar un comando de depuración, consulte Información importante sobre comandos de depuración.

Información Relacionada

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