식 MIB 및 이벤트 MIB 컨피그레이션 예

목차

소개

이 문서에서는 장애 관리에 사용할 식 MIB와 이벤트 MIB를 결합하는 방법을 보여 줍니다.포함된 예제는 현실적이지 않지만 사용 가능한 많은 기능을 보여 줍니다.

라우터는 두 가지 작업을 수행해야 합니다.

1. 루프백 인터페이스의 대역폭이 100보다 크고 관리적으로 다운된 경우 트랩 전송
2. 인터페이스 중 하나에 정의된 값에서 대역폭 문이 변경된 경우 루프백 인터페이스가 종료됩니다.

이 예제는 명령줄에서 쉽게 조작할 수 있고 show integer 및 boolean 값을 모두 제공하므로 bandwidth 및 admin 상태와 함께 표시됩니다.

이 문서의 명령은 개체 이름이 아니라 OID(개체 식별자) 매개 변수를 사용합니다.이렇게 하면 MIB를 로드하지 않고 테스트할 수 있습니다.

사전 요구 사항

요구 사항

이 문서의 정보를 사용하기 전에 다음 사전 요구 사항을 충족해야 합니다.

- 워크스테이션에는 HP(Hewlett-Packard) Openview에서 제공하는 SNMP(Simple Network Management Protocol) 도구가 있어야 합니다.다른 SNMP 툴은 작동하지만 구문은 다를 수 있습니다.

- 디바이스는 Cisco IOS® 소프트웨어 릴리스 12.2(4)T3 이상을 실행해야 합니다.이전 버전에서는 이벤트 MIB의 RFC 버전을 지원하지 않습니다.
- 플랫폼은 이벤트 MIB를 지원해야 합니다.Cisco IOS Software Release 12.1(3)T에 대해 지원되는 플랫폼 목록은 이벤트 MIB 지원의 "지원되는 플랫폼" 섹션을 참조하십시오.

사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

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

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다.이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다.현재 네트워크가 작동 중인 경우, 모든 명령어의 잠재적인 영향을 미리 숙지하시기 바랍니다.

표기 규칙

문서 규칙에 대한 자세한 내용은 Cisco 기술 팁 표기 규칙을 참조하십시오.

배경 정보

- 표현식 MIB를 사용하면 다른 객체의 조합을 기반으로 자신의 MIB 객체를 생성할 수 있습니다.자세한 내용은 RFC 2982를 참조하십시오.
- 이벤트 MIB를 사용하면 디바이스에서 자체 MIB 객체를 모니터링하고 정의된 이벤트를 기반으로 작업(알림 또는 **SNMP SET** 명령)을 생성할 수 있습니다.자세한 내용은 RFC 2981을 참조하십시오.

구성

참고: 일부 출력 코드는 화면에 더 잘 맞도록 두 행에 표시됩니다.

이 예에서는 루프백 인터페이스의 ifIndex가 16입니다.

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

첫 번째 이벤트와 관련된 변수 이름은 `e1`으로 시작하고 두 번째 시작과 관련된 변수 이름은 `e2`로 시작합니다. 라우터 이름은 "router"이고 읽기/쓰기 커뮤니티 문자열은 "private"입니다.

식 MIB

표현식 1 생성

먼저 조건이 100,000보다 , 루프백 인터페이스에 대해 `Speed 100,000`보다 `ifAdminStatus down` 경우 값 1을 반환하는 표현식을 만듭니다.조건이 충족되지 않으면 값 0을 반환합니다.

1. expExpressionDeltaInterval - 이 객체는 사용되지 않습니다.폴링되지 않을 때 식을 계산할 이

유가 없습니다.값을 설정하지 않으면 개체를 쿼리할 때 식이 계산됩니다.표현식 이름은 e1exp이며, ASCII 테이블의 101 49 101 120 112 해당합니다.

2. expNameStatus - 생성된 최종 이전 표현식을 삭제합니다.
```
# 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—생성 후 대기합니다.
```
# 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 - 나중에 표현식 결과를 검색하는 데 사용할 인덱스를 생성합니다.
```
# 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 - 여기서 .1(선택한 expExpressionIndex)은 표현식에 대한 설명입니다.
```
# 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 - 표현식 자체이며, 다음 단계에서 $1 및 $2 변수가 정의됩니다.허용되는 연산자는 다음과 같습니다(자세한 내용은 RFC 2982 참조).
```
( ) - (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. exp객체 ID
```
.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 - 두 값은 절대 값으로 설정됩니다(Delta의 경우 값으로 2).
```
# 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 - 객체 ID가 와일드카드가 아닙니다.기본값이므로 snmpObjectIDWildcard를 설정하지 마십시오.
10. expObjectStatus - expObjectTable의 행을 활성으로 설정합니다.
```
# 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. 식 1을 활성화합니다.
```
# 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
```

표현식 1 테스트

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

1. 조건이 충족되면 expValueCounter32Val의 값은 1입니다(expExpressionValueType의 값은 변경되지 않으므로 결과는 counter32).**참고:** 유형은 부동 소수점 값이 될 수 없습니다.
   ```
   # 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. 조건이 충족되지 않으면 값은 0입니다.
   ```
   # 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. 조건이 충족되지 않으면 값은 0입니다.
   ```
   # 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
   ```

표현식 생성 및 테스트 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—1.3.6.1.1.2.2.1.5가 객체가 아닌 테이블임을 나타냅니다.
   ```
   # 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. 테스트:
   ```
   # 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
   ```

이벤트 MIB

이벤트 생성 1

이제 60초마다 첫 번째 식의 출력 값을 확인하고 이를 참조와 비교하는 이벤트를 만듭니다.참조가 표현식 값과 일치하면 선택한 VARBIND로 트랩이 트리거됩니다.

1. 트리거 테이블에서 트리거를 생성합니다.트리거 이름은 trigger1이며, ASCII 코드에서 `116 114 105 103 103 101 114 49`.주인은 톰입니다.`116 111 109` .mteTriggerEntry의 인덱스는 트리거 소유자 및 트리거 이름으로 구성됩니다.인덱스의 첫 번째 값은 mteOwner의 문자 수를 제공합니다.이 경우 tom에는 세 개의 문자가 있으므로 색인은 다음과 같습니다 .3.116.111.109.116.114.105.103.103.101.114.49.
2. 기존 항목이 있으면 제거합니다.
3. 트리거 상태를 **만들고 대기하도록 설정합니다**.
4. 마지막 단계에서는 다음을 활성화합니다.mte트리거 항목 상태

```
# 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 - 첫 번째 표현식 값은 `e1exp`입니다.MIB 개체의 개체 식별자는 트리거가 실행되어야 하는지 여부를 샘플링하는 것입니다.

```
# 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 - 값 ID에 와일드카드를 사용하지 않습니다.

```
# 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—존재 여부(0), 부울(1) 및 임계값(2).위의 값 중 하나를 선택하는 방법은 복잡합니다.존재를 선택하려면 첫 번째 숫자가 1인 8자리 숫자(예: `10000000` 또 `100xxxxx`) .부울의 경우 두 번째 숫자는 1이어야 합니다.`01000000` 또는 `010xxxxx`.임계값의 경우 세 번째 숫자는 1이어야 합니다.`00100000` 또는 `001xxxxx`.이렇게 하는 것이 가장 쉽습니다.존재의 경우 값은 octetestringhex—80입니다.boolean의 경우 값은 octestringhex—40입니다.임계값의 경우 값은 octestringhex—20입니다.

```
# 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 - 트리거 샘플 간에 대기할 시간(초)을 결정합니다.최소값은 object mteResourceSampleMinimum(기본값은 60초)으로 설정되므로 이 값을 낮추면 CPU 사용량이 증가하므로 신중하게 수행해야 합니다.

```
# 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 - absoluteValue(1) 및 deltaValue(2)입니다. 이 경우 값은 절대입니다.

```
# 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 - 트리거를 구성하되 사용하지 않도록 하는 컨트롤입니다.true(기본값은 false)로 설정합니다.

```
# 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
```

트리거가 생성되었으므로 트리거에서 사용할 이벤트를 정의합니다.이벤트 이름은 event1입니다.mte이벤트 항목 상태

```
# 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 - 알림(0)과 설정(1)입니다.프로세스는 mteTriggerTest와 동일합니다.알림은 10xxxxxx, 설정된 알림은 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
```

다음 단계에서는 trigger1에 대해 선택된 객체에 대해 수행할 테스트를 정의합니다.mteTriggerBooleanComparison - 같지 않음(1), 같음(2), less(3), lessOrEqual(4), greater(5) 및 greaterOrEqual(6)입니다. 이 경우 같음.

```
# 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 - 테스트에 사용할 값입니다.1.3.6.1.4.1.9.10.22.1.4.1.1.2.1.0.0.0 값이 1이면 조건이 충족됩니다.

```
# 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
```

이제 이벤트와 함께 전송할 객체를 정의합니다.mteTriggerBoolean객체소유자

```
# 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"
```

mteTrigger부울 객체

```
# 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"
```

mteTriggerBoolean이벤트 소유자

```
# 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"
```

mteTriggerBoolean이벤트

```
# 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"
```

객체 테이블을 생성합니다.1.3.6.1.2.1.2.2.1.5.16 값을 트랩과 함께 VARBIND로 전송합니다 .Object Table mteObjectsName - Objects1.mte개체항목상태

```
# 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
```

mte개체ID

```
# 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 - 와일드카드가 사용되지 않습니다.

```
# 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
```

개체 테이블을 활성화합니다.

```
# 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
```

객체를 event1에 연결합니다.Notify mteEventName—Event1.mte이벤트알림객체소유자

```
# 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"
```

mte이벤트 알림 개체

```
# 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"
```

트리거를 활성화합니다.

```
# 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
```

이벤트를 활성화합니다.

```
# 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
```

트랩 수신

```
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
```
참고: 객체 6은 추가된 VARBIND입니다.

이벤트 생성 2

다음 단계를 수행합니다.

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 - 첫 번째 식 및 mteTriggerValueIDWildcard의 값입니다.이번에는 프로세스가 트리거가 발생하는지 확인하기 위해 샘플링할 MIB 객체의 객체 식별자, 값 ID를 와일드카드로 사용합니다.
   ```
   # 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 - 임계값.
   ```
   # 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. mte트리거 빈도
   ```
   # 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 - 델타 값입니다.
   ```
   # 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. mte트리거 사용
   ```
   # 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. 이벤트 테이블 // mteEventName—event2에서 이벤트를 만듭니다.
   ```
   # 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—값 40은 Set에 대한 값으로, 조건이 충족되면 라우터가 snmp set 명령을 **실행합니다**.이 경우 Set은 자신을 위해 설정되지만 원격 디바이스에서 작업을 수행할 수도 있습니다.
```
# 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. 이벤트를 활성화합니다.
```
# 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. 트리거 테이블 // 인덱스에서 트리거 임계값을 설정합니다. mteTriggerName—Trigger2.임계값이므로 실패와 상승 상태의 값을 지정합니다.이번에는 상승조건만 가져가세요.
11. mteTriggerThresholdDeltaRising - 확인할 임계값 값입니다.
```
# 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 - 설정할 MIB 개체의 개체 식별자입니다.여기에 루프백 인터페이스에 대한 ifAdminStatus가 있습니다.
```
# 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 - 설정할 값(2의 경우 down)입니다.
```
# 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. 트리거를 활성화합니다.
```
# 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. 이벤트를 활성화합니다.
```
# 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
```

결과

```
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
```

참고: 여기서 `10.48.71.71`은 라우터 자체의 주소입니다.

다음을 확인합니다.

이 섹션에서는 컨피그레이션이 제대로 작동하는지 확인하는 데 사용할 정보를 제공합니다.

일부 **show 명령은** 출력 인터프리터 툴 에서 지원되는데(등록된 고객만), 이 툴을 사용하면 **show 명령 출력의 분석 결과를 볼 수 있습니다.**

```
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
```

문제 해결

이 섹션에서는 컨피그레이션 문제를 해결하는 데 사용할 정보를 제공합니다.

문제 해결 명령

디버깅을 활성화하는 명령은 다음과 같습니다.

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

참고: debug 명령을 실행하기 전에 디버그 명령에 대한 중요 정보를 참조하십시오.

관련 정보

- 식 MIB:RFC 2982
- 이벤트 MIB:RFC 2981
- EXPRESSION-MIB.my / EVENT-MIB.my

- IOS 기능 가이드:이벤트 MIB 지원
- Technical Support - Cisco Systems