

# Пример конфигурации MIB выражения и MIB события

## Содержание

[Введение](#)

[Предварительные условия](#)

[Требования](#)

[Используемые компоненты](#)

[Условные обозначения](#)

[Общие сведения](#)

[Настройка](#)

[MIB выражения](#)

[База MIB \(управляющей информации для событий\)](#)

[Проверка](#)

[Устранение неполадок](#)

[Команды для устранения неполадок](#)

[Дополнительные сведения](#)

## Введение

Этот документ показывает, как объединить MIB Выражения и Базу MIB (управляющей информации для событий) для использования в защите от ошибок и неисправностей. Включенный пример не реалистичен, но показывает много доступных функций.

Маршрутизатор должен выполнить два действия:

1. Передайте trap-сообщение, если интерфейс обратной связи имеет пропускную способность выше, чем 100 и административно выключен
2. Если одному из интерфейсов изменили его инструкцию для пропускной способности от определенного значения, интерфейс обратной связи завершает работу

Пример показывают с пропускной способностью и административным статусом, потому что ими легко манипулировать из командной строки, и оба показывают целые и булевы значения.

Команды в этом документе используют идентификатор объекта (OID) параметр а не имена объекта. Это позволяет тестировать, не загружая MIB.

## Предварительные условия

### Требования

Прежде чем использовать сведения из этого документа, убедитесь, что следующие условия выполнены:

- Рабочей станции должен предоставить программные средства Протокола SNMP Hewlett-Packard (HP) Openview. Другие программные средства SNMP работают, но могут иметь другой синтаксис.
- Устройство должно выполнить релиз 12.2 программного обеспечения Cisco IOS (4) T3 или позже. Более ранние версии не поддерживают версию RFC Базы MIB (управляющей информации для событий).
- Платформа должна поддерживать Базу MIB (управляющей информации для событий). Для списка поддерживаемых платформ для программного обеспечения Cisco IOS версии 12.1(3)T обратитесь к разделу "Поддерживаемой платформы" [Поддержки Базы MIB \(управляющей информации для событий\)](#).

## Используемые компоненты

Сведения, содержащиеся в данном документе, касаются следующих версий программного обеспечения и оборудования:

- Программное обеспечение Cisco IOS версии 12.3 (1a)
- Модульный маршрутизатор доступа Cisco 3640

Сведения, представленные в этом документе, были получены от устройств, работающих в специальной лабораторной среде. Все устройства, описанные в этом документе, были запущены с чистой (стандартной) конфигурацией. В рабочей сети необходимо изучить потенциальное воздействие всех команд до их использования.

## Условные обозначения

[Дополнительные сведения об условных обозначениях см. в документе Технические рекомендации Cisco. Условные обозначения.](#)

## Общие сведения

- MIB Выражения позволяет пользователю создавать свой собственный объект MIB на основе комбинации других объектов. Для получения дополнительной информации обратитесь к [RFC 2982](#).
- База MIB (управляющей информации для событий) позволяет пользователю иметь устройство, контролирующее его собственные объекты MIB и генерировать действия (уведомление или команды **SNMP SET**) на основе определенного события. Для получения дополнительной информации обратитесь к [RFC 2981](#).

## Настройка

**Примечание:** Некоторые линии кода выхода отображены более чем две линии для лучшей адаптации на экран.

В данном примере ifIndex интерфейса обратной связи равен 16.

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

Имена переменной, отнесенные к первому событию, запускаются с e1 и отнесенных к второму запуску с e2. Имя маршрутизатора является "маршрутизатором", и строка имени и пароля чтения-записи является "частной".

## MIB выражения

### Создание выражения 1

Сначала создайте выражение, которое возвращает значение 1 если условие, `ifSpeed is greater than 100,000 AND ifAdminStatus is down` для интерфейса обратной связи. Если условие не соблюдают, оно возвращает значение 0.

1. [expExpressionDeltaInterval](#) — Этот объект не используется. Нет никакой причины вычислить выражение, когда оно не опрошено. Если никакое значение не установлено, выражение вычислено, когда объект делают запрос. Названием выражения является `e1exp`, который в таблице ASCII соответствует 101 49 101 120 112.
2. [expNameStatus](#) — Это уничтожает возможное старое выражение, которое создано.

```
# 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](#) — Создайте и ждите.

```
# 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](#) — Это создает индекс для использования позже для получения результата выражения.

```
# 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](#) — Здесь.1 (выбранный `expExpressionIndex`) описание выражения.

```
# 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](#) — Это - само выражение, переменные, 1\$ и 2\$ определены в следующем шаге. Единственные разрешенные операторы (для подробных данных, обратитесь к [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](#) — Два значения приняты в абсолютных значениях (для Дельты, возьмите 2 в качестве значения).

```
# 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](#) — Идентификаторы объекта не являются `wildcarded`. Это - значение по умолчанию, не `snmpset expObjectIDWildcard` - также.
10. [expObjectStatus](#) — Установите строки в `expObjectTable` к активному.

```
# 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. Активируйте выражение 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
```

### Тестирование выражения 1

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

1. Если условие соблюдают, значение [expValueCounter32Val](#) равняется 1 (поскольку значение [expExpressionValueType](#) остается неизменным, результатом является counter32). **Примечание:** Тип не может быть значением с плавающей точкой.
 

```
# 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. Если условие не соблюдают, значение 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. Если условие не соблюдают, значение 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
```

## Создание и тестирование выражения 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](#) — Это указывает, что 1.3.6.1.2.1.2.2.1.5 является таблицей и не ОБЪЕКТОМ.
 

```
# 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. ТЕСТ:
 

```
# 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
```

## База MIB (управляющей информации для событий)

### Создание события 1

Теперь создайте событие, которое проверяет выходное значение первого выражения каждые 60 секунд и сравнивает его со ссылкой. Когда ссылка совпадает со значением выражения, trap-сообщение инициировано с выбранным VARBIND.

1. Создайте триггер в триггерной таблице. Название триггера является trigger1, который в коде ASCII является 116 114 105 103 103 101 114 49. Владелец является tom: 116 111 109. Индекс mteTriggerEntry составлен из триггерного владельца и триггерного названия. Первое значение индекса дает количество символов для mteOwner. В этом случае существует три символа для tom, таким образом, индекс:
 

```
3.116.111.109.116.114.105.103.103.101.114.49.
```
2. Уничтожьте старую запись, если она существует.
3. Заставьте триггерный статус **создавать и ждать**.
4. Последний шаг активирует его:
 

```
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](#) — Значением первого выражения является `elexp`. Идентификатор объекта объекта MIB является тем для выборки, чтобы видеть, должен ли сработать триггер. # `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](#) — Не используя

подстановочный знак для ID значения. # `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](#) — Существование (0), булевская переменная (1), и порог (2). Метод для выбора одного из вышеупомянутых значений является комплексом. Для выбора существования дайте значение в восьми цифрах, в которых первым является 1, такой как `10000000` или `100xxxxxx`. Для булевской переменной вторая цифра должна быть 1:

`01000000` или `010xxxxxx`. Для порога третий разряд должен быть 1: `00100000` или

`001xxxxxx`. Является самым легким проложить себе путь: Для существования значение

является `octetstringhex` — `80`. Для булевской переменной значение является

`octetstringhex` — `40`. Для порога значение является `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](#) — Это определяет кол-во секунд для ожидания

между триггерными выборками. Минимальное значение установлено с объектом

`mteResourceSampleMinimum` (по умолчанию составляет 60 секунд), понижать это

значение увеличивает использование ЦПУ, поэтому это должно быть сделанный

тщательно. # `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](#) — Это `absoluteValue` (1) и `deltaValue` (2). В этом случае значение

является абсолютным. # `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](#) — Это - контроль, который позволяет триггеру настраиваться, но не

использоваться. Установите его в `True` (по умолчанию является ложью). # `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` Теперь, когда триггер, как создано, определите событие, которое будет

использовать триггер. Название события является `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 —
```

Они - уведомление (0) и устанавливают (1). Процесс совпадает с для `mteTriggerTest`.

Уведомлением является `10xxxxxxx`, и набором является `01xxxxxxx`. # `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
```

Этот следующий шаг определяет тест, который будет сделан на объекте, выбранном для

`trigger1`. [mteTriggerBooleanComparison](#) — Они не равны (1), равняются (2), меньше (3),

`lessOrEqual` (4), больше (5), и `greaterOrEqual` (6). В этом случае — равный. # `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 — Это - значение для использования для теста. Если
```

значение `1.3.6.1.4.1.9.10.22.1.4.1.1.2.1.0.0.0` равно 1, то условие соблюдают. # `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
```

Теперь определите объект, который будет передаваться с

событием. [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"
Составьте объектную таблицу.Передайте значение 1.3.6.1.2.1.2.2.1.5.16 как VARBIND
с trap-сообщением.Объектная Таблица 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 — Нет никакого используемого
ПОДСТАНОВОЧНОГО ЗНАКА.# 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
Активируйте объектную таблицу.# 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
Подключите объект к event1.Уведомьте 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"
Активируйте триггер.# 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
Активизируйте событие.# 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

```

## [Полученное trap-сообщение](#)

```

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

```

**Примечание:** Объект 6 является VARBIND, который был добавлен.

## [Создание события 2](#)

Выполните следующие действия:

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](#) — Это - значение первого выражения и

[mteTriggerValueIDWildcard](#). На этот раз, подстановочные знаки процесса ID значения, идентификатор объекта объекта MIB для выборки, чтобы определить, срабатывает ли

```
триггер.# 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](#) — Порог.# 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](#) — Значение дельты.# 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. Создайте событие в конечном счете таблица//[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](#) — Когда условие соблюдают, проблемы маршрутизатора команда **snmp set**, значение 40 для Набора, означая это. В этом случае это делает Набор для себя, но это могло также сделать операцию на удаленном устройстве.# 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. Включите событие.# 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. Установите Триггерный Порог в Триггерной Таблице//индекс = [mteTriggerName](#) — Trigger2. Поскольку это - порог, дайте значения для сбоя и повышения условий. Возьмите только возрастающее условие на этот раз.
11. [mteTriggerThresholdDeltaRising](#) — Это - пороговое значение для проверки против.# 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](#) — Это - идентификатор объекта от объекта MIB для установки. Здесь, ifAdminStatus для интерфейса обратной СВЯЗИ.# 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](#) — Это - значение для установки (2 для вниз).# 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. Активируйте триггер.# 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. Активизируйте событие.# 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

## Результат

```
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
```

Примечание: Здесь, 10.48.71.71 адрес самого маршрутизатора.

## Проверка

Этот раздел предоставляет сведения для использования, чтобы подтвердить, что конфигурация работает должным образом.

**Некоторые команды show поддерживаются Средством интерпретации выходных данных(только зарегистрированные клиенты), которое позволяет просматривать аналитику выходных данных команды 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
```

## Устранение неполадок

Этот раздел предоставляет сведения для использования для устранения проблем конфигурации.

### Команды для устранения неполадок

Это команды для включения отладку:

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

**Примечание:** Прежде чем применять команды отладки, ознакомьтесь с разделом "Важные сведения о командах отладки".



## Дополнительные сведения

- [MIB выражения: RFC 2982](#)
- [База MIB \(управляющей информации для событий\): RFC 2981](#)
- [ВЫРАЖЕНИЕ-MIB.my / СОБЫТИЕ-MIB.my](#)
- [Руководство по функциям и свойствам IOS: Поддержка события MIB](#)
- [Техническая поддержка - Cisco Systems](#)