表达式MIB和事件MIB配置示例

目录

简介

本文档说明如何将表达式MIB和事件MIB组合在故障管理中使用。包含的示例不现实，但显示了许多可用功能。

路由器必须执行两项操作：

1. 如果环回接口的带宽大于100且管理性关闭，则发送陷阱
2. 如果其中一个接口的带宽语句从定义的值更改，环回接口将关闭

该示例以带宽和管理状态显示，因为它们易于从命令行操作，同时显示整数和布尔值。

本文档中的命令使用对象标识符(OID)参数，而不是对象名称。这允许在不加载MIB的情况下进行测试。

先决条件

要求

使用本文档中的信息之前，请确保满足以下必备条件：

- 工作站应具有Hewlett-Packard(HP)Openview提供的简单网络管理协议(SNMP)工具。其他SNMP工具可以工作，但可能有不同的语法。
- 设备必须运行Cisco IOS®软件版本12.2(4)T3或更高版本。早期版本不支持事件MIB的RFC版本。

- 平台必须支持事件MIB。有关Cisco IOS软件版本12.1(3)T支持的平台列表，请参阅事件MIB支持的“支持的平台”部分。

使用的组件

本文档中的信息基于以下软件和硬件版本：

- 思科IOS软件版本12.3(1a)
- 思科3640模块化接入路由器

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您使用的是真实网络，请确保您已经了解所有命令的潜在影响。

规则

有关文件规则的更多信息请参见“ Cisco技术提示规则”。

背景信息

- 表达式MIB允许用户根据其他对象的组合创建自己的MIB对象。有关详细信息，请参阅 RFC 2982 。
- 事件MIB允许用户让设备监控其自己的MIB对象，并根据定义的事件生成操作(通**知或SNMP** SET命令)。有关详细信息，请参阅 RFC 2981 。

配置

注意：输出代码的某些行显示在两行上，以便更好地适合您的屏幕。

在本例中，环回接口的ifIndex等于16。

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

与第一个事件相关的变量名称以`e1`开头，与第二个事件相关的变量名称以`e2`开头。路由器名称为“router”，读/写社区字符串为“private”。

表达式MIB

创建表达式1

首先，如果条件大于100,000，且环回接口`ifAdminStatusdown1`表达式。如果条件不满足，则返回值0。

1. expExpressionDeltaInterval — 不使用此对象。没有轮询表达式时，没有理由计算表达式。如果未设置值，则在查询对象时计算表达式。表达式名称`e1exp`，在ASCII表中对应于`101 49 101 120 112`。
2. expNameStatus — 这会销毁最终创建的旧表达式。
   ```
   # 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 — 创建并等待。
```
# 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 — 这将创建索引，以便稍后用于检索表达式的结果。
```
# 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 — 此处。1（选择的expExpressionIndex）是表达式的说明。
```
# 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 — 这是表达式本身，变量$1和$2在下一步中定义。唯一允许的运算符是(有关详细信息，请参阅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 — 这两个值以绝对值取值（对于Delta，以2作为值）。
```
# 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 — 对象ID不是通配符。这是默认值，因此请勿snmpset expObjectIDWildcard。
10. expObjectStatus — 将expObjectTable中的行设置为活动。
```
# 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. 激活表达式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
```

测试表达式1

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

1. 如果满足条件，则expValueCounter32Val的值为1(因为expExpressionValueType的值保持不

变，所以结果为counter32)。**注：**类型不能是浮点值。

```
# 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. 如果条件不满足，则值为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. 如果条件不满足，则值为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
```

创建和测试表达式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 — 这表1.3.6.1.2.1.2.2.1.5而不是对象。

```
# 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. 测试：

```
# 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
```

事件MIB

创建事件1

现在创建一个事件，每60秒检查一个表达式的输出值，并将其与引用进行比较。当引用与表达式值匹配时，使用所选的VARBIND触发陷阱。

1. 在触发器表中创建触发器。触发器的名称为trigger1，在ASCII代码中为116 114 105 103 103 101 114 49。老板是汤姆：116 111 109.mteTriggerEntry的索引由触发器所有者和触发器名称组成。索引的第一个值为mteOwner提供字符数。在本例中，tom有三个字符，因此索引是：3.116.111.109.116.114.105.103.103.101.114.49
2. 销毁旧条目（如果存在）。
3. 将触发器状态设置为**创建并等待**。
4. 最后一步激活它：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 — 第一个表达式的值e1exp。MIB对象的对象标识符是要采样的标识符，以查看触发器是否应触发。

```
# 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 — 不为值ID使用通配符。

```
# 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 — 存在(0)、布尔值(1)和阈值(2)。选择上述值之一的方法是复杂的。要选择存在，请以八位数字表示一个值，其中第一个数字为1，例如1000000或100xxxxx。对于布尔值，第二位必须是1:01000000或010xxxx。对于阈值，第三位必须是1:00100000或001xxxx。这种工作方式最简单：对于存在，值为octeinghex - 80。对于布尔值，该值为octetstringex - 40。对于阈值，值为octetstringex - 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 — 这确定触发器样本之间等待的秒数。最小值是使用对象mteResourceSampleMinimum设置的（默认值为60秒），降低此值会增加CPU使用率，因此必须小心操作。

```
# 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 — 这些是absoluteValue(1)和deltaValue(2)。 在本例中，值为绝对值：

```
# 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 — 这是允许配置但不使用触发器的控件。将其设置为true（默认为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
```

创建触发器后，定义触发器将使用的事件。事件名称为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 — 这些是通知(0)和设置(1)。该过程与mteTriggerTest的过程相同。通知为10xxxxxx设置为01xxxxx。

```
# 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
```

此下一步定义要对为trigger1选择的对象执行的测试。mteTriggerBooleanComparison — 这些是不等价(1)、等价(2)、小值(3)、小值(4)、大值(5)和大值(6)。 在这种情况下，等于。

```
# 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 — 这是用于测试的值。如果值1.3.6.1.4.1.9.10.22.1.4.1.2.1.0.0.01，则满足条件。

```
# 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
```

现在定义要随事件一起发送的对象。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"
```

创建对象表。将值1.3.6.1.2.1.2.2.1.5.16VARBIND一起发送。对象表mteObjectsName — 对象1。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 — 未使用通配符。

```
# 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
```

激活对象表。

```
# 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
```

将对象附加到event1。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"
```

激活触发器。

```
# 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
```

激活事件。

```
# 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
```

已接收陷阱

```
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
```

注意：对象6是添加的VARBIND。

创建事件2

执行下列步骤：

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 — 这是第一个表达式和mteTriggerValueIDWildcard的值。此时，进程会通配符值ID，即要采样的MIB对象的对象标识符，以确定触发器是否触发。
```
# 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 — 阈值。
```
# 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 — 增量值。
```
# 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. 在事件表// mteEventName - event2中创建事件。
```
# 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 — 值40用于Set，这意味着当满足条件时，路由器会发出**snmp set**命令。在这种情况下，它为自身设置，但也可以在远程设备上操作。
```
# 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. 启用事件。

```
# 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. 在触发器表//索引= mteTriggerName - Trigger2中设置触发器阈值。由于这是一个阈值，因此请给出故障和上升情况的值。这次只看上升的情况。
11. mteTriggerThresholdDeltaRising — 这是要检查的阈值。

```
# 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 — 这是要设置的MIB对象的对象标识符。此处为环回接口的ifAdminStatus。

```
# 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 — 这是要设置的值（2表示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. 激活触发器。

```
# 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. 激活事件。

```
# 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
```

结果

```
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
```

注意：此处10.48.71.71是路由器本身的地址。

验证

本部分提供用于确认配置工作正常的信息。

命令输出解释程序工具（仅限注册用户）支持某些 **show 命令，使用此工具可以查看对 show 命令输出的分析。**

```
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
```

故障排除

本节提供用于排除配置故障的信息。

故障排除命令

以下是启用调试的命令：

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

注意：在发出debug命**令**之前，请参阅有关debug命令的重要信息。

相关信息

- 表达式MIB:RFC 2982
- 事件MIB:RFC 2981
- EXPRESSION-MIB.my / EVENT-MIB.my
- IOS功能指南：事件MIB支持
- 技术支持 - Cisco Systems