

表達式MIB和事件MIB配置示例

目錄

[簡介](#)

[必要條件](#)

[需求](#)

[採用元件](#)

[慣例](#)

[背景資訊](#)

[設定](#)

[表達式MIB](#)

[事件MIB](#)

[驗證](#)

[疑難排解](#)

[疑難排解指令](#)

[相關資訊](#)

簡介

本文檔介紹如何將表達式MIB和事件MIB組合起來用於故障管理。包含的示例不現實，但顯示了許多可用功能。

路由器必須執行兩項操作：

1. 如果回送介面的頻寬高於100且處於管理性關閉狀態，則傳送陷阱
2. 如果其中一個介面的bandwidth語句從定義的值更改，則環回介面關閉

該示例以頻寬和管理狀態顯示，因為它們易於從命令列操作，並且同時顯示integer和boolean值。

本文檔中的命令使用對象識別符號(OID)引數，而不是對象名稱。這允許在不載入MIB的情況下進行測試。

必要條件

需求

使用本檔案中的資訊之前，請確保符合以下必要條件：

- 工作站應具有Hewlett-Packard(HP)Openview提供的簡單網路管理協定(SNMP)工具。其他SNMP工具可以使用，但語法可能不同。
- 裝置必須運行Cisco IOS®軟體版本12.2(4)T3或更高版本。早期版本不支援事件MIB的RFC版本。
-

- 平台必須支援事件MIB。有關Cisco IOS軟體版本12.1(3)T支援的平台的清單，請參閱[事件MIB支援](#)的「支援的平台」部分。

採用元件

本文中的資訊係根據以下軟體和硬體版本：

- Cisco IOS軟體版本12.3(1a)
- 思科3640模組化存取路由器

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路正在作用，請確保您已瞭解任何指令可能造成的影響。

慣例

如需文件慣例的詳細資訊，請參閱[思科技術提示慣例](#)。

背景資訊

- Expression MIB允許使用者基於其他對象的組合建立自己的MIB對象。如需詳細資訊，請參閱[RFC 2982](#)。
- 事件MIB允許使用者讓裝置監控自己的MIB對象，並根據定義的事件生成操作(通知或SNMP SET命令)。如需詳細資訊，請參閱[RFC 2981](#)。

設定

注意：輸出代碼的某些行會顯示在兩行上，以便更好地適合您的螢幕。

在本示例中，環回介面的ifIndex等於16。

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

與第一個事件相關的變數名稱以e₁開頭，與第二個事件相關的變數名稱以e₂開頭。路由器名稱為「router」，讀/寫社群字串為「private」。

表達式MIB

建立表達式1

首先建立一個表達式，如果環回介面的條件(ifSpeed100,000)和ifAdminStatusdown，則返回值1。如果不滿足條件，則返回值0。

1. [expExpressionDeltaInterval](#) — 未使用此對象。未輪詢表達式時，沒有理由計算該表達式。如果未設定值，則在查詢對象時計算表達式。表達式名稱為e_{1exp}，在ASCII表中對應於101 49 101 120 112。
2. [expNameStatus](#) — 這將銷毀最終建立的舊表達式。
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

3. [expNameStatus](#) — 建立和等待。

```
# 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](#) — 這將建立稍後用於檢索表達式結果的索引。

```
# 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](#) — 此處。1 (所選expExpressionIndex) 是表達式的說明。

```
# 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](#) — 這是表達式本身，將在下一步定義變數\$1和\$2。僅允許使用的運算子有(有關詳細資訊，請參閱[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 < 10000 && $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](#) — 這兩個值是以絕對值取的 (對於「增量」，取值為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](#) — 對象ID不是萬用字元。這是預設值，因此不要使用snmpset expObjectIDWildcard。10. [expObjectStatus](#) — 將expObjectTable中的行設定為活動狀態。

```
# 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. 啟用表達式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
```

測試表達式1

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

1. 如果滿足條件，[expValueCounter32Val](#)的值是1(由於[expExpressionValueType](#)的值保持不變

, 因此結果為計數器32)。註：型別不能是浮點值。

```
# 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. 如果不滿足條件，則值為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. 如果不滿足條件，則值為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
```

建立和測試表達式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](#) — 這表示1.3.6.1.2.1.2.2.1.5是表而不是對象。

```
# 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. 測試:

```
# 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

建立事件1

現在建立一個事件，該事件每60秒檢查一次表達式的輸出值，並將其與引用進行比較。當引用與表達式值匹配時，將使用所選VARBIND觸發陷阱。

1. 在觸發器表中建立觸發器。觸發器的名稱為trigger1，其ASCII代碼為116 114 105 103 103 101 114 49。所有者是湯姆：116 111 109.mteTriggerEntry的索引由觸發器所有者和觸發器名稱組成。索引的第一個值提供mteOwner的字元數。在這個例子中，湯姆有三個角色，因此索引是：3.116.111.109.116.114.105.103.103.101.114.49。
2. 銷毀舊條目（如果存在）。
3. 將觸發器狀態設定為建立和等待。
4. 最後一步會啟用它：[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](#) — 第一個表達式的值是`elexp`。MIB對象的對象識別符號是進行取樣以檢視觸發器是否應該觸發的識別符號。

```
# 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](#) — 不使用萬用字元作為值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](#) — 存在(0)、布林值(1)和閾值(2)。選擇上述值之一的方法非常複雜。要選擇存在，請提供一個八位數字的值，其中第一個數字是1，例如10000000或100xxxxxx。對於布林值，第二個數字必須是1:0100000或010xxxxxx。對於閾值，第三個數字必須是1:0010000或001xxxxxx。這樣操作最容易：對於存在，值為octetstringhex - 80。對於boolean，值為octetstringhex - 40。對於閾值，值為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](#) — 它確定在觸發器取樣之間等待的秒數。最小值是使用object `mteResourceSampleMinimum`設定的（預設值為60秒），降低此值會增加CPU使用率，因此必須小心操作。

```
# 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](#) — 這些是`absoluteValue(1)`和`deltaValue(2)`。在這種情況下，該值為絕對值：

```
# 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](#) — 這是一個允許配置但不使用觸發器的控制元件。將其設定為`true`（預設值為`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
```

既然已經建立觸發器，請定義觸發器將使用的事件。事件名稱為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](#) — 這些是通知(0)和設定(1)。此過程與mteTriggerTest的過程相同。通知為10xxxxxxx，並設定為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
```

以下步驟定義要對為trigger1選擇的對象執行的測試。[mteTriggerBooleanComparison](#) — 這些值是不等的(1)、等於(2)、小於(3)、小於或等於(4)、大於(5)和大於或等於(6)。在這種情況下，等於。

```
# 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](#) — 這是用於測試的值。如果1.3.6.1.4.1.9.10.22.1.4.1.1.2.1.0.0.01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
```

現在定義要隨事件傳送的對象。[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"
```

建立對象表。將1.3.6.1.2.1.2.2.1.5.16VARBIND與陷阱一起傳送。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](#) — 未使用萬用字元。

```
# 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
```

啟用對象表。

```
# 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
```

將對象附加到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"
```

啟用觸發器。

```
# 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
```

啟用事件。

```
# 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
```

接收的陷阱

```
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：對象6是新增的VARBIND。

建立事件2

請遵循以下步驟：

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](#) — 這是第一個表達式和[mteTriggerValueIDWildcard](#)的[值](#)。這一次，該進程將值ID（要取樣的MIB對象的對象識別符號）作為萬用字元，以確定是否觸發觸發器。

```
# 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](#) — 閾值。

```
# 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](#)中建立事件。

```
# 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用於Set，這意味著當條件滿足時，路由器會發出[snmp set](#)命令。在這種情況下，它會為自身生成Set，但也可能在遠端裝置上執行操作。

```
# 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. 啟用事件。

```
# 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. 在觸發器表//index = mteTriggerName - Trigger2中設置觸發器閾值。由於這是一個閾值，請給出失敗和上升條件的值。這次只看上升的情況。

11. [mteTriggerThresholdDeltaRising](#) — 這是要檢查的閾值。

```
# 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](#) — 這是要設定的MIB對象的對象識別符號。這裡，是環回介面的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](#) — 這是要設定的值 (2表示關閉) 。

```
# 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. 啟用觸發器。

```
# 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. 啟用事件。

```
# 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
```

結果

```
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
```

註：其中10.48.71.71是路由器本身的地址。

驗證

本節提供的資訊用於確認組態是否正常運作。

[輸出直譯器工具](#)(僅供註冊客戶使用)支援某些show命令，此工具可讓您檢視show命令輸出的分析。

```
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.22.1.4.1.1.2.2.0.0.0, 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
```

疑難排解

本節提供的資訊用於對組態進行疑難排解。

疑難排解指令

以下是啟用調試的命令：

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

注意：發出debug指令之前，請參閱[有關Debug指令的重要資訊](#)。

相關資訊

- [表達式MIB :RFC 2982](#)
- [事件MIB:RFC 2981](#)
- [EXPRESSION-MIB.my / EVENT-MIB.my](#)
- [IOS功能指南：事件MIB支援](#)
- [技術支援 - Cisco Systems](#)