

# Comment configurer les paramètres d'alarme et d'événement RMON à l'aide de commandes SNMP

## Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Conventions](#)

[Informations générales](#)

[Procédure pas à pas](#)

[Création d'un événement](#)

[Création d'une alarme](#)

[Exemple](#)

[Vérifier](#)

[Dépanner](#)

[Informations connexes](#)

## [Introduction](#)

Ce document fournit une configuration d'échantillon pour l'alarme de Surveillance à distance (RMON) et des configurations d'événement utilisant des commandes SNMP.

## [Conditions préalables](#)

### [Conditions requises](#)

Aucune spécification déterminée n'est requise pour ce document.

### [Composants utilisés](#)

Pour suivre les procédures dans ce document, votre périphérique doit prendre en charge le MIB RMON. Vous pouvez vérifier ceci aux [outils MIB de Cisco IOS](#) (clients [enregistrés](#) seulement).

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est opérationnel, assurez-vous que vous comprenez l'effet potentiel de toute commande.

## Conventions

Pour plus d'informations sur les conventions utilisées dans ce document, reportez-vous aux [Conventions relatives aux conseils techniques Cisco](#).

## Informations générales

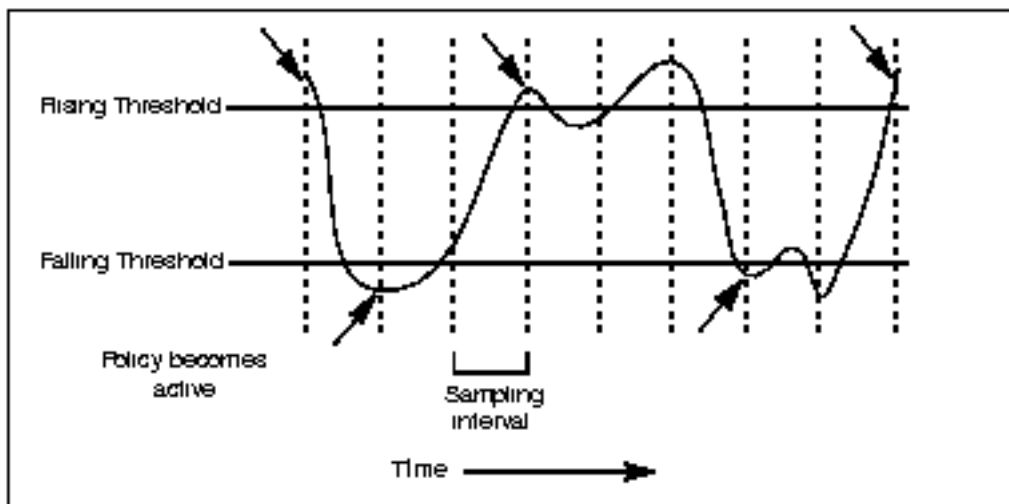
Le but de l'alarme et de l'événement de Surveillance à distance (RMON) est de surveiller un certain objet MIB sur le périphérique, et avertit l'administrateur système si une de ces valeurs est extinction de la plage définie.

L'alarme surveille un objet spécifique dans le MIB et déclenche un événement quand la condition (chute ou seuil montant) est atteinte.

L'événement est le déroutement ou le log a généré quand l'alarme le déclenche. Un exemple d'une montée et d'un seuil de chute est :

n=value monitored by the alarm. The falling threshold is "5" and the rising threshold is "15"

L'alarme déclenche un événement quand une des deux valeurs est atteinte.



Valeur	Déroutement	commentaire
n1=16	augmentation	a atteint la valeur montante : 15
n2=4	chute	a atteint la valeur de chute : 5
n3=6	aucun	entre 5 et 15
n4=6	augmentation	a atteint la valeur montante : 15
n5=13	aucun	bonne condition
n6=20	aucun	au-dessus de 15 mais n'est pas allé au-dessous de 5 depuis le dernier événement
n7=4	chute	a atteint la valeur de chute : 5

n8=2 0	aucun	au-dessous de 5 mais n'est pas allé au-dessus de 15 depuis le dernier événement
n9=1 6	augmentation	a atteint la valeur montante : 15

Vous pouvez configurer l'alarme et l'événement de RMON sur des Routeurs utilisant l'interface de ligne de commande (CLI) (référez-vous à [configurer l'alarme et les configurations d'événement de RMON de l'interface de ligne de commande](#)), et sur des Routeurs et des Commutateurs utilisant des commandes de Protocole SNMP (Simple Network Management Protocol). Les paramètres à modifier font partie du [MIB RMON](#).

## Procédure pas à pas

### Création d'un événement

Utilisez cette commande de créer un événement :

```
# snmpset -c <read_write_community> <device_name> .1.3.6.1.2.1.16.9.1.1.x.y <variable type>
<value>
```

Choisissez d'abord l'ID d'événement (variable *y*).

Suivez cette procédure pour créer un événement. Pour chaque étape, il y a une description de l'étape, du nom de l'objet MIB à modifier, de l'object id (OID), du *type* <variable, et du <value > de la commande générique.

1. Effacez un vieil événement certain qui aurait utilisé ID= " y » (veuillez-vous d'abord pour vérifier que vous n'avez plus besoin de lui. Autrement, utilisez un autre ID).

```
* eventStatus / .1.3.6.1.2.1.16.9.1.1.7.y
* variable type=integer
* value=4
```

**Remarque:** Utilisez la même commande d'effacer l'événement une fois nécessaire.

2. Entrez le mode de création d'événement :

```
eventStatus / .1.3.6.1.2.1.16.9.1.1.7.y
* variable type=integer
* value v=2
```

3. Spécifiez la description d'événement :

```
* eventDescription / .1.3.6.1.2.1.16.9.1.1.2.y
* variable type=string (for Net-snmp) or octetsting (for Openview)
* value = a description of the event
```

4. Spécifiez le type d'événement que vous voulez :

```
* eventType / .1.3.6.1.2.1.16.9.1.1.3.y
* variable type=integer
* value =
"1" => none
"2" => log
"3" => snmp-trap
```

```
"4" => log-and-trap
```

#### 5. Spécifiez la chaîne de la communauté pour le déROUTement :

```
* eventCommunity / .1.3.6.1.2.1.16.9.1.1.4.y  
* variable type=string (for Net-snmp) or octetsting (for Openview)  
* value="<trap_community_string>"
```

#### 6. Spécifiez le propriétaire de l'événement :

```
* eventOwner / .1.3.6.1.2.1.16.9.1.1.6.y  
* variable type=string (for Net-snmp) or octetsting (for Openview)  
* value="<event_owner>"
```

#### 7. Lancez l'événement :

```
* eventStatus / .1.3.6.1.2.1.16.9.1.1.7.y  
* variable type=integer  
* value=1
```

## Création d'une alarme

Utilisez cette commande de créer une alarme :

```
# snmpset -c .1.3.6.1.2.1.16.3.1.1.x.y <read_write_community> <device_name> <variable type>  
<value>
```

#### 1. Effacez une vieille alarme certaine qui aurait utilisé ID=y (premier contrôle si vous n'avez plus besoin de lui. Autrement, utilisez un autre ID) :

```
* alarmStatus / .1.3.6.1.2.1.16.3.1.1.12.y  
* variable type=integer  
* value=4
```

#### 2. Entrez le mode de création d'alarme :

```
* alarmStatus / .1.3.6.1.2.1.16.3.1.1.12.y  
* variable type=integer  
* value=2
```

#### 3. Placez l'intervalle (dont en quelques secondes) au-dessus les données sont échantillonnées et comparées à la montée et aux seuils de chute :

```
* alarmInterval / .1.3.6.1.2.1.16.3.1.1.2.y  
* variable type=integer  
* value=<n_seconds>
```

#### 4. Spécifiez l'OID que vous voulez surveiller :

```
* alarmVariable / .1.3.6.1.2.1.16.3.1.1.3.y  
* variable type=objid (for Net-snmp) or objectidentifier (for Openview)  
* value=<oid_to_check>
```

#### 5. Définissez le type d'échantillon que vous voulez :

```
* alarmSampleType / .1.3.6.1.2.1.16.3.1.1.4.y  
* variable type=integer  
* value=<rising_threshold> "1" => absoluteValue "2" => deltaValue
```

#### 6. Spécifiez quels déclencheurs une alarme :

```
* alarmStartupAlarm / .1.3.6.1.2.1.16.3.1.1.6.y  
* variable type=integer  
* value=  
"1" => risingAlarm  
"2" => fallingAlarm  
"3" => risingOrFallingAlarm
```

## 7. Définissez le seuil montant :

```
* alarmRisingThreshold / .1.3.6.1.2.1.16.3.1.1.7.y
* variable type=integer
* value=<rising_threshold>
```

## 8. Définissez le seuil de chute :

```
* alarmFallingThreshold / .1.3.6.1.2.1.16.3.1.1.8.y
* variable type=integer
* value=<falling_threshold>
```

## 9. Spécifiez l'ID d'événement que vous voulez déclencher quand le seuil montant est franchi :

```
* alarmRisingEventIndex / .1.3.6.1.2.1.16.3.1.1.9.y
* variable type=integer
* value=<event_ID>
```

## 10. Spécifiez l'ID d'événement quand le seuil de chute est franchi :

```
* alarmFallingEventIndex / .1.3.6.1.2.1.16.3.1.1.9.y
* variable type=integer
* value=<event_ID>
```

## 11. Spécifiez le propriétaire de l'alarme :

```
* alarmOwner / .1.3.6.1.2.1.16.3.1.1.11.y
* variable type=string (for Net-snmp) or octetsting (for Openview)
* value=<owner>
```

## 12. Lancez l'alarme :

```
* alarmStatus / .1.3.6.1.2.1.16.3.1.1.12.y
* variable type=integer
* value=1
```

## Exemple

Dans cet exemple, le **safari** est utilisé pour envoyer un déroutement quand le nombre d'octets allant dans l'interface 12 pendant les deux dernières minutes est au-dessus de 140000000 ou en-dessous de 10.

Le safari est le logiciel du Cisco IOS 2500 (C2500-JS-L), version 12.1(9), le LOGICIEL de VERSION (fc1).

Cet exemple a été également essayé avec succès sur le logiciel WS-C6506, version NmpSW : 6.1(1b).

**Remarque:** Dans le Catalyst, il n'y a aucune commande CLI de vérifier la configuration, mais elle peut être faite avec la commande de **snmpwalk** sur le serveur.

Sur le routeur et le commutateur, cette configuration survit à une recharge.

```
safari# show rmon events
Event table is empty
```

```
# snmpset -c private safari .1.3.6.1.2.1.16.9.1.1.7.123 integer 4
16.9.1.1.7.123 = 4
# snmpset -c private safari .1.3.6.1.2.1.16.9.1.1.7.123 integer 2
16.9.1.1.7.123 = 2
```

```
safari#show rmon events
Event 123 is under creation, owned by
Description is
```

Event firing causes nothing, last fired 00:00:00

```
# snmpset -c private safari .1.3.6.1.2.1.16.9.1.1.2.123 string "test_event"
16.9.1.1.2.123 = "test_event"
# snmpset -c private safari .1.3.6.1.2.1.16.9.1.1.3.123 integer 4
16.9.1.1.3.123 = 4
# snmpset -c private safari .1.3.6.1.2.1.16.9.1.1.4.123 string "public"
16.9.1.1.4.123 = "public"
# snmpset -c private safari .1.3.6.1.2.1.16.9.1.1.6.123 string "event_owner"
16.9.1.1.6.123 = "event_owner"
# snmpset -c private safari .1.3.6.1.2.1.16.9.1.1.7.123 integer 1
16.9.1.1.7.123 = 1
```

safari# **show rmon events**

Event 123 is active, owned by event\_owner  
Description is test\_event  
Event firing causes log and trap to community public, last fired 00:00:00

safari# **show rmon alarm**

Alarm table is empty

```
# snmpset -c private safari .1.3.6.1.2.1.16.3.1.1.12.321 integer 2
16.3.1.1.12.321 = 2
```

safari# **show rmon alarm**

Alarm 321 is under creation, owned by  
Monitors ccitt.0 every 10 second(s)  
Taking absolute samples, last value was 0  
Rising threshold is 0, assigned to event 0  
Falling threshold is 0, assigned to event 0  
On startup enable rising or falling alarm

```
# snmpset -c private safari .1.3.6.1.2.1.16.3.1.1.2.321 integer 120
16.3.1.1.2.321 = 120
# snmpset -c private safari .1.3.6.1.2.1.16.3.1.1.3.321 objid .1.3.6.1.2.1.2.2.1.10.12
16.3.1.1.3.321 = OID: interfaces.ifTable.ifEntry.ifInOctets.12
# snmpset -c private safari .1.3.6.1.2.1.16.3.1.1.4.321 integer 2
16.3.1.1.4.321 = 2
# snmpset -c private safari .1.3.6.1.2.1.16.3.1.1.6.321 integer 3
16.3.1.1.6.321 = 3
# snmpset -c private safari .1.3.6.1.2.1.16.3.1.1.7.321 integer 140000000
16.3.1.1.7.321 = 140000000
# snmpset -c private safari .1.3.6.1.2.1.16.3.1.1.8.321 integer 10
16.3.1.1.8.321 = 10
# snmpset -c private safari .1.3.6.1.2.1.16.3.1.1.9.321 integer 123
16.3.1.1.9.321 = 123
# snmpset -c private safari .1.3.6.1.2.1.16.3.1.1.10.321 integer 123
16.3.1.1.10.321 = 123
# snmpset -c private safari .1.3.6.1.2.1.16.3.1.1.11.321 string "alarm_owner"
16.3.1.1.11.321 = "alarm_owner"
# snmpset -c private safari .1.3.6.1.2.1.16.3.1.1.12.321 integer 1
16.3.1.1.12.321 = 1
```

safari# **show rmon alarm**

Alarm 321 is active, owned by alarm\_owner  
Monitors ifEntry.10.1 every 120 second(s)  
Taking delta samples, last value was 130244  
Rising threshold is 140000000, assigned to event 123  
Falling threshold is 10, assigned to event 123  
On startup enable rising or falling alarm

[Vérifier](#)

Aucune procédure de vérification n'est disponible pour cette configuration.

## Dépanner

Il n'existe actuellement aucune information de dépannage spécifique pour cette configuration.

## Informations connexes

- [Configurer l'alarme et les configurations d'événement de RMON de l'interface de ligne de commande](#)
- [Support d'événement MIB](#)
- [RFC 1757](#)
- [Support technique - Cisco Systems](#)