

Ping MIB の実装

内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[表記法](#)

[スクリプトの例](#)

[スクリプト](#)

[関連情報](#)

概要

Ping Group は、Cisco Management ブランチ (.1.3.6.1.4.1.9.9.16.) にある Cisco 独自の MIB の一部です。Ping Group を使用して、管理ステーションからリモート デバイス間のインターネット制御メッセージ プロトコル (ICMP) アクティビティをセットアップ、実行、検索できます。

前提条件

要件

このドキュメントに特有の要件はありません。

使用するコンポーネント

このドキュメントの内容は、特定のソフトウェアやハードウェアのバージョンに限定されるものではありません。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期 (デフォルト) 設定の状態から起動しています。対象のネットワークが実稼働中である場合には、どのようなコマンドについても、その潜在的な影響について確実に理解しておく必要があります。

表記法

ドキュメント表記の詳細は、「[シスコ テクニカル ティップスの表記法](#)」を参照してください。

スクリプトの例

サンプルスクリプトを使用して、HP OpenViewまたはNetViewでこのシーケンスを開始できます

。ネットワーク管理プラットフォームのGUIを使用して、**snmpsets**および**snmpget**として次のコマンドを入力することもできます。GUIにアクセスするには、次のいずれかの方法を使用します。

- メニューバーから、[Tools]、[MIB Browser]の順に選択します。**SNMP**を選択します。
- コマンドラインで**xnmbrowser**と入力します。

スクリプト

```
Management_Station-----Router_Source-----Router_Dest
echo "##### Create the instance #####"
```

```
##### We've chosen 333 at random. 333 will be the row instance to use for this particular
##### ping experiment. After the ping, the row will be deleted.
##### This keeps the table clean. Router_Source is the dns name of the device we are
##### working with, and public is its RW community string. The values for
##### ciscoPingEntryStatus status are as follows (see Ping MIB): ##### 1 - active ##### 2 -
notInService ##### 3 - notReady ##### 4 - createAndGo ##### 5 - createAndWait ##### 6 -
destroy ##### We will clear out any previous entries by setting ciscoPingEntryStatus = 6
(destroy) snmpset -c public Router_Source .1.3.6.1.4.1.9.9.16.1.1.1.16.333 integer 6 ##### We
start building the row by setting ciscoPingEntryStatus = 5 (createAndWait) echo snmpset -c
public Router_Source .1.3.6.1.4.1.9.9.16.1.1.1.16.333 integer 5 echo echo "##### Now let's set
the characteristics of the ping #####" ##### Only the first three sets below are REQUIRED.
The rest have default ##### values. #Set ciscoPingEntryOwner = any_name snmpset -c public
Router_Source .1.3.6.1.4.1.9.9.16.1.1.1.15.333 octetstring any_name #Set ciscoPingProtocol = 1 =
ip (see CISCO-TC-V1SMI.my CiscoNetworkProtocol) snmpset -c public Router_Source
.1.3.6.1.4.1.9.9.16.1.1.1.2.333 integer 1 #Set ciscoPingAddress = #.#.#--take Remote_Dest's ip
& convert each octet to hex snmpset -c public Router_Source .1.3.6.1.4.1.9.9.16.1.1.1.3.333
octetstringhex "AB 44 76 67" #Set the packet count to 20 (ciscoPingPacketCount) snmpset -c
public Router_Source .1.3.6.1.4.1.9.9.16.1.1.1.4.333 integer 20 #Set the packetsize to 100
(ciscoPingPacketSize) snmpset -c public Router_Source .1.3.6.1.4.1.9.9.16.1.1.1.5.333 integer
100 echo echo "##### Now let's verify that the ping is ready to go and launch it #####" #Get
ciscoPingEntryStatus and make sure it is now equal to 2. This means # notInService which
indicates that we're ready to go. snmpget -c public Router_Source
.1.3.6.1.4.1.9.9.16.1.1.1.16.333 # Set ciscoPingEntryStatus = 1 to tell it to activate. snmpset
-c public Router_Source .1.3.6.1.4.1.9.9.16.1.1.1.16.333 integer 1 echo echo "##### Let's look
at the results. #####" snmpwalk -c public Router_Source .1.3.6.1.4.1.9.9.16.1.1.1 echo echo
"##### Now that we've gotten the results, let's destroy the row #####" snmpset -c public
Router_Source .1.3.6.1.4.1.9.9.16.1.1.1.16.333 integer 6
```

関連情報

- [テクニカルサポート - Cisco Systems](#)