

Compreendendo Valores de Índices de Tabela no SNMP

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Conventions](#)

[Iniciando com IfIndex](#)

[Objetos de eleição](#)

[Agrupando objetos com base em ifIndex](#)

[Reunindo objetos caso a tabela não esteja indexada por ifIndex ou indexada de modo cruzado](#)

[Correlacionando BRIDGE-MIB a IF-MIB](#)

[Informações Relacionadas](#)

[Introduction](#)

Quando realizar o pooling de objetos de Simple Network Management Protocol (SNMP), você deve, às vezes, saber o que exatamente está sendo agrupado em pools. A fim compreender inteiramente isso, é necessário saber como correlacionar o objeto está sendo agrupado em pool com o que você deseja agrupar. Este documento aborda os princípios de como usar os índices no SNMP para agrupar objetos em tabelas.

[Prerequisites](#)

[Requirements](#)

Os leitores deste documento devem estar cientes destes tópicos:

- Conhecimento geral do SNMP
- Software usado para consultar dispositivos Cisco via SNMP

[Componentes Utilizados](#)

As informações neste documento são baseadas nestas versões de software e hardware:

- UCD SNMP versão 4.2
- Cisco Catalyst 5509 com Cisco IOS® Software Release 5.5(7)

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is

live, make sure that you understand the potential impact of any command.

Conventions

For more information on document conventions, refer to the [Cisco Technical Tips Conventions](#).

Iniciando com ifIndex

Uma das primeiras coisas a aprender, quando você está lidando com SNMP, é [ifIndex](#). Esta é uma chave primária de todos os objetos. Considere que todas as interfaces (físicas e lógicas) sejam divididas e atribuíam um valor. Esse valor é atribuído durante a inicialização de um dispositivo e não pode ser alterado. Se alguma informação precisa ser pesquisada para essa interface específica, ela deve usar esse valor atribuído.

ifIndex é definido no IF-MIB ([RFC 1213](#) ) desta forma:

```
InterfaceIndex ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS current
    DESCRIPTION
        "A unique value, greater than zero, for each interface
        or interface sub-layer in the managed system. It is
        recommended that values are assigned contiguously
        starting from 1. The value for each interface sub-
        layer must remain constant at least from one re-
        initialization of the entity's network management
        system to the next re-initialization."
    SYNTAX Integer32 (1..2147483647)
```

Para qualquer MIB, uma maneira rápida de dizer que índice organiza uma tabela é observar a entrada da tabela:

```
ifEntry OBJECT-TYPE
    SYNTAX IfEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry containing management information applicable
        to a particular interface."
    INDEX { ifIndex }
    ::= { ifTable 1 }
```

Dado um MIB e uma entrada de tabela, você pode determinar como a tabela é indexada. A próxima seção fornece exemplos de ifIndex.

Objetos de eleição

Agrupando objetos com base em ifIndex

Quando você emite o comando **snmpwalk** para pesquisar um objeto baseado em ifIndex ([ifName](#)) para a porta 7/4 no switch, você obtém esta saída:

```
sj-cse-568: snmpwalk 172.16.99.60 public ifname
```

```

ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.1 = sc0
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.2 = s10
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.3 = VLAN-1
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.4 = VLAN-1002
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.5 = VLAN-1004
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.6 = VLAN-1005
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.7 = VLAN-1003
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.8 = 7/1
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.9 = 7/2
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.10 = 7/3
!--- This is the relevant line: ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.11 = 7/4
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.12 = 7/5
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.13 = 7/6
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.14 = 7/7
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.15 = 7/8
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.16 = 7/9
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.17 = 7/10
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.18 = 7/11
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.19 = 7/12
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.20 = ATM8/0
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.22 = /A
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.23 = /B
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.24 = Nu0
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.25 = LEC/ATM8/0.10
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.532 = 3/1
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.533 = 3/2
!--- Output suppressed.

```

Nessa saída de uma pesquisa de ifName ([ifDescr](#) em roteadores), observe que há um número anexado a cada linha, após ifName. Esse é o ifIndex atribuído à interface real na mesma linha. Isso significa que a segunda linha da pesquisa, porta 7/4, recebe um ifIndex de 11. Se desejar informações sobre a porta 7/4 de um objeto ifIndexado, use um índice de 11. Isso significa adicionar um .11 ao final de um identificador de objeto MIB (OID), para recuperar a instância desse objeto que corresponde aos mesmos valores ifIndex.

[Reunindo objetos caso a tabela não esteja indexada por ifIndex ou indexada de modo cruzado](#)

Às vezes, as tabelas não são indexadas por ifIndex, como com BRIDGE-MIB. Esta saída examina como é indexada:

```

dot1dBasePortEntry OBJECT-TYPE
    SYNTAX Dot1dBasePortEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "A list of information for each port of the
        bridge."
    REFERENCE
        "IEEE 802.1D-1990: Section 6.4.2, 6.6.1"
    INDEX { dot1dBasePort }
    ::= { dot1dBasePortTable 1 }

```

Essa saída mostra que [dot1dBasePortEntry](#) é indexado por dot1dBasePort. Como isso se traduz de volta ao ifIndex? BRIDGE-MIB acessa um objeto chamado dot1dBasePortIfIndex. O objeto é definido desta maneira:

```

dot1dBasePortIfIndex OBJECT-TYPE

```

```
SYNTAX    INTEGER
ACCESS    read-only
STATUS    mandatory
DESCRIPTION
    "The value of the instance of the ifIndex object,
    defined in MIB-II, for the interface corresponding
    to this port."
 ::= { dot1dBasePortEntry 2 }
```

Essa saída mostra como correlacionar de BRIDGE-MIB a IF-MIB. O próximo exemplo mostra como tudo se encaixa.

Observação: BRIDGE-MIB é construído por vlan, portanto a comunidade "public@vlan-id" deve ser usada para ambientes que não sejam de vlan1.

Correlacionando BRIDGE-MIB a IF-MIB

Quando você emite um **snmpwalk** no BRIDGE-MIB, você obtém a próxima saída de exemplo para um Índice. Use dot1dBasePortIfIndex (.1.3.6.1.2.1.17.1.4.1.2) para mapeá-lo de volta para ifIndex. Depois de ter o ifIndex, use-o para pesquisar outros objetos com base em ifIndex.

```
sj-cse-568: snmpwalk 172.16.99.60 public .1.3.6.1.2.1.17.1.4.1.2
```

```
17.1.4.1.2.203 = 671
17.1.4.1.2.204 = 672
17.1.4.1.2.205 = 673
17.1.4.1.2.206 = 674
17.1.4.1.2.207 = 675
17.1.4.1.2.208 = 676
17.1.4.1.2.209 = 677
17.1.4.1.2.210 = 678
17.1.4.1.2.211 = 679
17.1.4.1.2.212 = 680
17.1.4.1.2.213 = 681
17.1.4.1.2.214 = 682
17.1.4.1.2.215 = 683
17.1.4.1.2.216 = 684
17.1.4.1.2.257 = 581
17.1.4.1.2.385 = 8
17.1.4.1.2.386 = 9
17.1.4.1.2.387 = 10
17.1.4.1.2.388 = 11
17.1.4.1.2.389 = 12
17.1.4.1.2.390 = 13
17.1.4.1.2.391 = 14
17.1.4.1.2.392 = 15
17.1.4.1.2.393 = 16
17.1.4.1.2.394 = 17
17.1.4.1.2.395 = 18
17.1.4.1.2.396 = 19
17.1.4.1.2.449 = 22
```

A linha de texto em negrito (17.1.4.1.2.388 = 11) mostra que .388 é um índice. Como você pesquisou o objeto dot1dBasePortIfIndex de BRIDGE-MIB, .388 é dot1dBasePortIfIndex. O 11 na linha de saída é na verdade o ifIndex. Se você coletar as informações desta pesquisa e da pesquisa anterior, poderá determinar que a porta 7/4 tem um ifIndex de 11 e um dot1dBasePortIfIndex (Index para BRIDGE-MIB) de .388.

Informações Relacionadas

- [Suporte Técnico - Cisco Systems](#)