

# Cómo obtener entradas CAM dinámicas (tabla CAM) para switches Catalyst con SNMP

## Contenido

[Introducción](#)

[prerrequisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Convenciones](#)

[Antecedente](#)

[El CISCO-VTP-MIB](#)

[El MIB puente](#)

[Detalles de las variables MIB--Incluye los identificadores de objeto \(los OID\)](#)

[Extraiga la información del CAM dinámico con el SNMP](#)

[Instrucciones Paso a Paso](#)

[Verificación](#)

[Información Relacionada](#)

## [Introducción](#)

Este documento describe cómo recopilar las entradas de CAM (Dynamic Content-Addressable Memory) de los switches Catalyst mediante SNMP (Simple Network Management Protocol).

## [prerrequisitos](#)

### [Requisitos](#)

Antes de usar la información en este documento, asegúrese de que usted resuelve los requisitos previos siguientes:

- Entienda cómo conseguir los VLA N de un switch de Catalyst vía el SNMP.
- Entienda cómo utilizar la [indización de la cadena de comunidad SNMP](#).
- Uso general de los comandos **get** y **walk** SNMP.

### [Componentes Utilizados](#)

Este documento se aplica a los switches de Catalyst que ejecutan el Catalyst regular OS o IOS de Catalyst el <sup>®</sup> que soportan el [BRIDGE-MIB](#). La información que contiene este documento se basa en las versiones de software y hardware indicadas a continuación.

- Catalyst 3524XL que ejecuta CatIOS 12.0(5)WC5a

- Catalyst 2948G que ejecuta CatOS 6.3(3)
- NET-SNMP disponible en <http://www.net-snmp.org/>

La información que se presenta en este documento se originó a partir de dispositivos dentro de un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si la red está funcionando, asegúrese de haber comprendido el impacto que puede tener un comando antes de ejecutarlo.

## Convenciones

Para obtener más información sobre las convenciones del documento, consulte [Convenciones de Consejos Técnicos de Cisco](#).

## Antecedente

### El CISCO-VTP-MIB

Para acceder el MIB que tiene un caso distinto para cada VLA N, usted debe primero entender cómo utilizar la [indexación de cadenas de comunidad](#). Después, usted necesita conocer los VLA N específicos que son activos en un Switch dado. [Del CISCO-VTP-MIB](#), usted puede obtener los VLAN activos en un Switch usando el objeto del [vtpVlanState](#). La razón para utilizar el objeto del vtpVlanState, y no el vtpVlanName u otro objeto, es que usted puede determinar en una operación el número del índice y que un VLA N es operativo.

Más información se da en el ejemplo abajo.

### El MIB puente

[Del BRIDGE-MIB](#), que se extrae del [RFC 1493](#), usted puede utilizar el [dot1dTpFdbAddress](#) del `dot1dTpFdbTable`, donde está igual a 3 o *aprendido el* valor, para determinar qué Media Access Control (MAC) Address están en la tabla de reenvío en el Switch. Se salva este valor mientras que un Unicast MAC Address para el cual el Bridge tenga la expedición y/o filtrado de información. Estos valores de la dirección MAC solamente no significan mucho y pueden presentar muchos datos. Por lo tanto, usted necesita contar el número de entradas y salvar ese valor de la cuenta, sobre la base de un .1.3.6.1.2.1.17.4.3.1.3) igual `dot1dTpFdbStatus` (a *aprendido* (valor de 3).

**Note:** El BRIDGE-MIB utiliza la indexación de cadenas de comunidad para acceder un caso particular del MIB según lo descrito en la [indización de la cadena de comunidad SNMP](#).

Tener los datos de la dirección MAC tiene valor para no perder de vista el número total de entradas CAM (direcciones MAC) aprendidas dinámicamente por el Switch. Esta supervisión ayuda a no perder de vista la llanura en su red, especialmente al correlacionar al número total de LAN virtuales (VLA N) por el Switch. Por ejemplo, si usted tiene un VLA N definido en el Switch y usted para ver 8,000 direcciones MAC, usted sabe que usted tiene 8,000 direcciones MAC para un VLA N, que es extenso para una subred.

Un objeto de MIB relacionado del [BRIDGE-MIB](#) (RFC 1493) es `dot1dTpFdbStatus`. Este MIB proporciona el estatus de la entrada de MAC Address.

Las definiciones del valor son:

- **el otro (1):** Ninguno del siguiente. Esto incluye los casos donde un cierto otro objeto de MIB (no la instancia correspondiente de `dot1dTpFdbPort`, ni una entrada en el `dot1dStaticTable`) se está utilizando para determinar si y cómo los direccionamientos de la trama al valor de la instancia correspondiente del `dot1dTpFdbAddress` se están remitiendo.
- **(2) inválido:** Esta entrada es no más válida (por ejemplo, era docta pero ha envejecido desde entonces hacia fuera), pero todavía no se ha vaciado de la tabla.
- **(3) docto:** El valor de la instancia correspondiente del `dot1dTpFdbPort` era docto, y se está utilizando.
- **uno mismo (4):** El valor de la instancia correspondiente del `dot1dTpFdbAddress` representa uno de los direccionamientos del Bridge. La instancia correspondiente del `dot1dTpFdbPort` indica cuáles de los puertos del Bridge tienen este direccionamiento.
- **mgmt (5):** El valor de la instancia correspondiente del `dot1dTpFdbAddress` es también el valor de un caso existente de `dot1dStaticAddress`.

## Detalles de las variables MIB--Incluye los identificadores de objeto (los OID)

```
vtpVlanState OBJECT-TYPE
SYNTAX INTEGER { operational(1),
suspended(2),
mtuTooBigForDevice(3),
mtuTooBigForTrunk(4) }
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The state of this VLAN.
```

The state 'mtuTooBigForDevice' indicates that this device cannot participate in this VLAN because the VLAN's MTU is larger than the device can support.

The state 'mtuTooBigForTrunk' indicates that while this VLAN's MTU is supported by this device, it is too large for one or more of the device's trunk ports."

```
::= { vtpVlanEntry 2 }
```

```
.1.3.6.1.2.1.17.4.3.1.1
```

```
dot1dTpFdbAddress OBJECT-TYPE
-- FROM BRIDGE-MIB
-- TEXTUAL CONVENTION MacAddress
SYNTAX OCTET STRING (6)
MAX-ACCESS read-only
STATUS Mandatory
DESCRIPTION "A unicast MAC address for which the
bridge has forwarding and/or filtering information."
::= { iso(1) org(3) dod(6) internet(1) mgmt(2) mib-2(1)
dot1dBridge(17) dot1dTp(4) dot1dTpFdbTable(3) dot1dTpFdbEntry(1) 1 }
```

```
.1.3.6.1.2.1.17.4.3
```

```
dot1dTpFdbTable OBJECT-TYPE
-- FROM BRIDGE-MIB
DESCRIPTION "A table that contains information about unicast
entries for which the bridge has forwarding and/or filtering information.
This information is used by the transparent bridging function in
determining how to propagate a received frame."
::= { iso(1) org(3) dod(6) internet(1) mgmt(2) mib-2(1) dot1dBridge(17)
```

```
dot1dTp(4) 3 }
```

```
.1.3.6.1.2.1.17.5.1
```

```
dot1dStaticTable OBJECT-TYPE
```

```
-- FROM BRIDGE-MIB
```

```
DESCRIPTION      "A table containing filtering information configured
into the bridge by (local or network) management specifying the set of ports
to which frames received from specific ports and containing specific destination
addresses are allowed to be forwarded. The value of zero in this table as the
port number from which frames with a specific destination address are received,
is used to specify all ports for which there is no specific entry in this table
for that particular destination address. Entries are valid for unicast and for
group/broadcast addresses."
```

```
::= { iso(1) org(3) dod(6) internet(1) mgmt(2) mib-2(1) dot1dBridge(17)
dot1dStatic(5) 1 }
```

```
.1.3.6.1.2.1.17.4.3.1.2
```

```
dot1dTpFdbPort OBJECT-TYPE
```

```
-- FROM BRIDGE-MIB
```

```
SYNTAX            Integer
```

```
MAX-ACCESS        read-only
```

```
STATUS            Mandatory
```

```
DESCRIPTION      "Either the value "0", or the port number of the port
on which a frame having a source address equal to the value of the corresponding
instance of dot1dTpFdbAddress has been seen. A value of "0" indicates that the
port number has not been learned, but that the bridge does have some
forwarding/filtering information about this address (that is, in the StaticTable).
```

```
Implementors are encouraged to assign the port value to
this object whenever it is learned, even for addresses for which the corresponding
value of dot1dTpFdbStatus is not learned(3)."
```

```
::= { iso(1) org(3) dod(6) internet(1) mgmt(2) mib-2(1) dot1dBridge(17) dot1dTp(4)
dot1dTpFdbTable(3) dot1dTpFdbEntry(1) 2 }
```

## [Extraiga la información del CAM dinámico con el SNMP](#)

### [Instrucciones Paso a Paso](#)

Siga los siguientes pasos para obtener la información del CAM dinámico con el SNMP.

1. Extraiga los VLA N. Utilice el **snmpwalk** en el objeto del [vtpVlanState](#) (.1.3.6.1.4.1.9.9.46.1.3.1.1.2):

```
nms-server2:/home/ccarring> snmpwalk -c public 14.32.6.17 vtpVlanState
CISCO-VTP-MIB::vtpVlanState.1.1 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.2 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.6 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.7 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.8 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.11 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.12 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.14 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.18 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.19 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.20 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.21 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.41 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.42 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.43 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.44 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.100 = INTEGER: operational(1)
```

```
CISCO-VTP-MIB::vtpVlanState.1.101 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.123 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.401 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.1002 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.1003 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.1004 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.1005 = INTEGER: operational(1)
```

2. Para cada VLA N, consiga el [dot1dTpFdbAddress de la](#) tabla de la dirección MAC (usando la [indexación de cadenas de comunidad](#)) (.1.3.6.1.2.1.17.4.3.1.1). En el ejemplo, el VLAN2 no contiene ninguna entrada en la tabla:

```
nms-server2:/home/ccarring> snmpwalk -c public@1 14.32.6.17 dot1dTpFdbAddress
.1.3.6.1.2.1.17.4.3.1.1.0.208.211.106.71.251 = Hex-STRING: 00 D0 D3 6A 47 FB
```

```
nms-server2:/home/ccarring> snmpwalk -c public@2 14.32.6.17 dot1dTpFdbAddress
```

```
nms-server2:/home/ccarring> snmpwalk -c public@6 14.32.6.17 dot1dTpFdbAddress
```

```
.1.3.6.1.2.1.17.4.3.1.1.0.2.185.144.76.102 = Hex-STRING: 00 02 B9 90 4C 66
```

```
.1.3.6.1.2.1.17.4.3.1.1.0.2.253.106.170.243 = Hex-STRING: 00 02 FD 6A AA F3
```

```
.1.3.6.1.2.1.17.4.3.1.1.0.16.13.56.16.0 = Hex-STRING: 00 10 0D 38 10 00
```

```
.1.3.6.1.2.1.17.4.3.1.1.0.96.84.144.248.0 = Hex-STRING: 00 60 54 90 F8 00
```

```
.1.3.6.1.2.1.17.4.3.1.1.0.208.2.214.120.10 = Hex-STRING: 00 D0 02 D6 78 0A
```

```
.1.3.6.1.2.1.17.4.3.1.1.0.208.211.54.162.60 = Hex-STRING: 00 D0 D3 36 A2 3C
```

```
.1.3.6.1.2.1.17.4.3.1.1.0.224.30.159.10.210 = Hex-STRING: 00 E0 1E 9F 0A D2
```

```
nms-server2:/home/ccarring> snmpwalk -c public@7 14.32.6.17 dot1dTpFdbAddress
```

```
.1.3.6.1.2.1.17.4.3.1.1.0.16.13.161.24.32 = Hex-STRING: 00 10 0D A1 18 20
```

... and so forth for each VLAN discovered in the first step.

3. Para cada VLA N, consiga el número de puerto del Bridge, el [dot1dTpFdbPort](#) (.1.3.6.1.2.1.17.4.3.1.2):

```
nms-server2:/home/ccarring> snmpwalk -c public@1 14.32.6.17 dot1dTpFdbPort
```

```
.1.3.6.1.2.1.17.4.3.1.2.0.208.211.106.71.251 = INTEGER: 113
```

```
nms-server2:/home/ccarring> snmpwalk -c public@2 14.32.6.17 dot1dTpFdbPort
```

```
nms-server2:/home/ccarring> snmpwalk -c public@6 14.32.6.17 dot1dTpFdbPort
```

```
.1.3.6.1.2.1.17.4.3.1.2.0.2.185.144.76.102 = INTEGER: 113
```

```
.1.3.6.1.2.1.17.4.3.1.2.0.2.253.106.170.243 = INTEGER: 113
```

```
.1.3.6.1.2.1.17.4.3.1.2.0.6.83.198.64.173 = INTEGER: 113
```

```
.1.3.6.1.2.1.17.4.3.1.2.0.16.13.56.16.0 = INTEGER: 113
```

```
.1.3.6.1.2.1.17.4.3.1.2.0.96.84.144.248.0 = INTEGER: 113
```

```
.1.3.6.1.2.1.17.4.3.1.2.0.208.2.214.120.10 = INTEGER: 113
```

```
.1.3.6.1.2.1.17.4.3.1.2.0.208.211.54.162.60 = INTEGER: 113
```

```
.1.3.6.1.2.1.17.4.3.1.2.0.224.30.159.10.210 = INTEGER: 65
```

```
nms-server2:/home/ccarring> snmpwalk -c public@7 14.32.6.17 dot1dTpFdbPort
```

```
.1.3.6.1.2.1.17.4.3.1.2.0.16.13.161.24.32 = INTEGER: 113
```

... and so forth for each VLAN discovered in the first step.

4. Consiga el puerto de Bridge al [ifIndex](#) (.1.3.6.1.2.1.2.2.1.1) que asocia, el [dot1dBasePortIfIndex](#) (.1.3.6.1.2.1.17.1.4.1.2):

```
nms-server2:/home/ccarring> snmpwalk -c public@1 14.32.6.17 dot1dBasePortIfIndex
```

```
.1.3.6.1.2.1.17.1.4.1.2.68 = INTEGER: 12
```

```
.1.3.6.1.2.1.17.1.4.1.2.69 = INTEGER: 13
```

```
.1.3.6.1.2.1.17.1.4.1.2.70 = INTEGER: 14
```

```
.1.3.6.1.2.1.17.1.4.1.2.71 = INTEGER: 15
```

```
.1.3.6.1.2.1.17.1.4.1.2.72 = INTEGER: 16
```

```
.1.3.6.1.2.1.17.1.4.1.2.74 = INTEGER: 18
```

```
.1.3.6.1.2.1.17.1.4.1.2.76 = INTEGER: 20
```

```
.1.3.6.1.2.1.17.1.4.1.2.77 = INTEGER: 21
```

```
.1.3.6.1.2.1.17.1.4.1.2.78 = INTEGER: 22
```

```
.1.3.6.1.2.1.17.1.4.1.2.79 = INTEGER: 23
```

```
.1.3.6.1.2.1.17.1.4.1.2.80 = INTEGER: 24
```

```
.1.3.6.1.2.1.17.1.4.1.2.81 = INTEGER: 25
```

```
.1.3.6.1.2.1.17.1.4.1.2.82 = INTEGER: 26
```

```
.1.3.6.1.2.1.17.1.4.1.2.83 = INTEGER: 27
```

```
.1.3.6.1.2.1.17.1.4.1.2.84 = INTEGER: 28
.1.3.6.1.2.1.17.1.4.1.2.85 = INTEGER: 29
.1.3.6.1.2.1.17.1.4.1.2.86 = INTEGER: 30
.1.3.6.1.2.1.17.1.4.1.2.87 = INTEGER: 31
.1.3.6.1.2.1.17.1.4.1.2.88 = INTEGER: 32
.1.3.6.1.2.1.17.1.4.1.2.89 = INTEGER: 33
.1.3.6.1.2.1.17.1.4.1.2.90 = INTEGER: 34
.1.3.6.1.2.1.17.1.4.1.2.91 = INTEGER: 35
.1.3.6.1.2.1.17.1.4.1.2.92 = INTEGER: 36
.1.3.6.1.2.1.17.1.4.1.2.93 = INTEGER: 37
.1.3.6.1.2.1.17.1.4.1.2.94 = INTEGER: 38
.1.3.6.1.2.1.17.1.4.1.2.95 = INTEGER: 39
.1.3.6.1.2.1.17.1.4.1.2.96 = INTEGER: 40
.1.3.6.1.2.1.17.1.4.1.2.98 = INTEGER: 42
.1.3.6.1.2.1.17.1.4.1.2.99 = INTEGER: 43
.1.3.6.1.2.1.17.1.4.1.2.100 = INTEGER: 44
.1.3.6.1.2.1.17.1.4.1.2.101 = INTEGER: 45
.1.3.6.1.2.1.17.1.4.1.2.102 = INTEGER: 46
.1.3.6.1.2.1.17.1.4.1.2.103 = INTEGER: 47
.1.3.6.1.2.1.17.1.4.1.2.104 = INTEGER: 48
.1.3.6.1.2.1.17.1.4.1.2.105 = INTEGER: 49
.1.3.6.1.2.1.17.1.4.1.2.106 = INTEGER: 50
.1.3.6.1.2.1.17.1.4.1.2.107 = INTEGER: 51
.1.3.6.1.2.1.17.1.4.1.2.108 = INTEGER: 52
.1.3.6.1.2.1.17.1.4.1.2.109 = INTEGER: 53
.1.3.6.1.2.1.17.1.4.1.2.110 = INTEGER: 54
.1.3.6.1.2.1.17.1.4.1.2.111 = INTEGER: 55
.1.3.6.1.2.1.17.1.4.1.2.112 = INTEGER: 56
.1.3.6.1.2.1.17.1.4.1.2.113 = INTEGER: 57
.1.3.6.1.2.1.17.1.4.1.2.114 = INTEGER: 58
```

... and so forth for each VLAN discovered in the first step.

5. Recorre el [ifName](#) (.1.3.6.1.2.1.31.1.1.1) para poder correlated el valor del ifIndex obtenido en el paso 4 con un nombre de puerto apropiado:

```
nms-server2:/home/ccarring> snmpwalk -c public@1 14.32.6.17 dot1dBasePortIfIndex
.1.3.6.1.2.1.17.1.4.1.2.68 = INTEGER: 12
.1.3.6.1.2.1.17.1.4.1.2.69 = INTEGER: 13
.1.3.6.1.2.1.17.1.4.1.2.70 = INTEGER: 14
.1.3.6.1.2.1.17.1.4.1.2.71 = INTEGER: 15
.1.3.6.1.2.1.17.1.4.1.2.72 = INTEGER: 16
.1.3.6.1.2.1.17.1.4.1.2.74 = INTEGER: 18
.1.3.6.1.2.1.17.1.4.1.2.76 = INTEGER: 20
.1.3.6.1.2.1.17.1.4.1.2.77 = INTEGER: 21
.1.3.6.1.2.1.17.1.4.1.2.78 = INTEGER: 22
.1.3.6.1.2.1.17.1.4.1.2.79 = INTEGER: 23
.1.3.6.1.2.1.17.1.4.1.2.80 = INTEGER: 24
.1.3.6.1.2.1.17.1.4.1.2.81 = INTEGER: 25
.1.3.6.1.2.1.17.1.4.1.2.82 = INTEGER: 26
.1.3.6.1.2.1.17.1.4.1.2.83 = INTEGER: 27
.1.3.6.1.2.1.17.1.4.1.2.84 = INTEGER: 28
.1.3.6.1.2.1.17.1.4.1.2.85 = INTEGER: 29
.1.3.6.1.2.1.17.1.4.1.2.86 = INTEGER: 30
.1.3.6.1.2.1.17.1.4.1.2.87 = INTEGER: 31
.1.3.6.1.2.1.17.1.4.1.2.88 = INTEGER: 32
.1.3.6.1.2.1.17.1.4.1.2.89 = INTEGER: 33
.1.3.6.1.2.1.17.1.4.1.2.90 = INTEGER: 34
.1.3.6.1.2.1.17.1.4.1.2.91 = INTEGER: 35
.1.3.6.1.2.1.17.1.4.1.2.92 = INTEGER: 36
.1.3.6.1.2.1.17.1.4.1.2.93 = INTEGER: 37
.1.3.6.1.2.1.17.1.4.1.2.94 = INTEGER: 38
.1.3.6.1.2.1.17.1.4.1.2.95 = INTEGER: 39
.1.3.6.1.2.1.17.1.4.1.2.96 = INTEGER: 40
.1.3.6.1.2.1.17.1.4.1.2.98 = INTEGER: 42
```

```
.1.3.6.1.2.1.17.1.4.1.2.99 = INTEGER: 43
.1.3.6.1.2.1.17.1.4.1.2.100 = INTEGER: 44
.1.3.6.1.2.1.17.1.4.1.2.101 = INTEGER: 45
.1.3.6.1.2.1.17.1.4.1.2.102 = INTEGER: 46
.1.3.6.1.2.1.17.1.4.1.2.103 = INTEGER: 47
.1.3.6.1.2.1.17.1.4.1.2.104 = INTEGER: 48
.1.3.6.1.2.1.17.1.4.1.2.105 = INTEGER: 49
.1.3.6.1.2.1.17.1.4.1.2.106 = INTEGER: 50
.1.3.6.1.2.1.17.1.4.1.2.107 = INTEGER: 51
.1.3.6.1.2.1.17.1.4.1.2.108 = INTEGER: 52
.1.3.6.1.2.1.17.1.4.1.2.109 = INTEGER: 53
.1.3.6.1.2.1.17.1.4.1.2.110 = INTEGER: 54
.1.3.6.1.2.1.17.1.4.1.2.111 = INTEGER: 55
.1.3.6.1.2.1.17.1.4.1.2.112 = INTEGER: 56
.1.3.6.1.2.1.17.1.4.1.2.113 = INTEGER: 57
.1.3.6.1.2.1.17.1.4.1.2.114 = INTEGER: 58
```

... and so forth for each VLAN discovered in the first step.

Ahora la información de puerto obtenida se puede utilizar, por ejemplo: Del paso 2, hay una dirección MAC: .1.3.6.1.2.1.17.4.3.1.1.0.208.211.106.71.251 = Hex.-CADENA: 00 D0 D3 6A 47 FB Del paso 3: .1.3.6.1.2.1.17.4.3.1.2.0.208.211.106.71.251 = NÚMERO ENTERO: 113 Esto le dice que esta dirección MAC (00 D0 D3 6A 47 FB) es del número de puerto del Bridge 113. Del paso 4, el número de puerto del Bridge 113 tiene un ifIndex número 57 .1.3.6.1.2.1.17.1.4.1.2.113 = NÚMERO ENTERO: 57 Del paso 5, el ifIndex 57 corresponde al puerto 2/49 .1.3.6.1.2.1.31.1.1.1.1.57 = CADENA: 2/49 Compare eso con la salida del comando **show cam dynamic** hecho salir para los switches CatOS, o la salida del comando **show mac** para el Switches de CatIOS. Usted ve una coincidencia para 1 [ALL] 00-d0-d3-6a-47-fb 2/49.

## Verificación

En esta sección encontrará información que puede utilizar para confirmar que su configuración esté funcionando correctamente.

1. Telnet a su Switch.
2. De la línea de comando, publique el comando apropiado: Dispositivos CatOS: **show cam dynamic** Dispositivos CatIOS: [show mac](#)
3. Compare la salida con los resultados obtenidos por el procedimiento especificado aquí.

```
nms-2948g> (enable) show cam dynamic
```

```
* = Static Entry. + = Permanent Entry. # = System Entry. R = Router Entry.
```

```
X = Port Security Entry $ = Dot1x Security Entry
```

VLAN	Dest MAC/Route Des	[CoS]	Destination Ports or VCs / [Protocol Type]
1	00-d0-d3-6a-47-fb		2/49 [ALL]
6	00-02-b9-90-4c-66		2/49 [ALL]
6	00-02-fd-6a-aa-f3		2/49 [ALL]
6	00-10-0d-38-10-00		2/49 [ALL]
6	00-60-54-90-f8-00		2/49 [ALL]
6	00-c0-1d-99-00-dc		2/49 [ALL]
6	00-d0-02-d6-78-0a		2/49 [ALL]
6	00-d0-d3-36-a2-3c		2/49 [ALL]
6	00-e0-1e-9f-0a-d2		2/1 [ALL]
7	00-10-0d-a1-18-20		2/49 [ALL]
8	00-10-0d-38-10-00		2/49 [ALL]
8	00-10-0d-a1-18-c0		2/49 [ALL]
14	00-d0-d3-36-a2-3c		2/49 [ALL]

```
18      00-00-0c-07-ac-12          2/49 [ALL]
18      00-10-0d-38-10-00        2/49 [ALL]
18      00-d0-d3-36-a2-3c        2/49 [ALL]
19      00-d0-02-d6-78-0a        2/49 [ALL]
41      00-d0-d3-36-a2-3c        2/49 [ALL]
42      00-d0-d3-36-a2-3c        2/49 [ALL]
100     00-04-de-a9-18-00        2/49 [ALL]
100     00-10-0d-38-10-00        2/49 [ALL]
100     00-10-7b-d9-07-60        2/49 [ALL]
100     00-90-27-86-76-e2        2/49 [ALL]
100     00-d0-d3-36-a2-3c        2/49 [ALL]
100     00-e0-1e-68-33-c7        2/49 [ALL]
101     00-d0-d3-36-a2-3c        2/49 [ALL]
Total Matching CAM Entries Displayed  =26
nms-2948g> (enable)
```

## [Información Relacionada](#)

- [Indexación de cadenas de comunidad SNMP](#)
- [Soporte Técnico - Cisco Systems](#)