

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

## Contenido

[Introducción](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Convenciones](#)

[Background](#)

[El CISCO-VTP-MIB](#)

[El MIB puente](#)

[Detalles de las variables MIB: incluye identificadores de objetos \(OID\)](#)

[Recuperar la información CAM dinámica con SNMP](#)

[Step-by-Step Instructions](#)

[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).

## [Prerequisites](#)

## [Requirements](#)


Antes de utilizar la información de este documento, asegúrese de cumplir con los siguientes requisitos previos:

- Entienda cómo obtener VLAN de un switch Catalyst a través de SNMP.
- Comprender cómo utilizar la [indexación de cadenas de comunidad SNMP](#).
- Uso general de los comandos get y walk de SNMP.

## [Componentes Utilizados](#)

Este documento se aplica a los switches Catalyst que ejecutan Catalyst OS normal o Catalyst IOS® que soportan [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. All of the devices used in this document started with a cleared (default) configuration. 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](#).


## [Background](#)

### [El CISCO-VTP-MIB](#)

Para acceder a los MIBs que tienen una instancia separada para cada VLAN, primero debe entender cómo utilizar la [indexación de cadena de comunidad](#). A continuación, debe conocer las VLAN específicas que están activas en un switch determinado. Desde el [CISCO-VTP-MIB](#), puede obtener las VLAN activas en un switch usando el objeto [vtpVlanState](#). La razón para utilizar el objeto vtpVlanState, y no vtpVlanName u otro objeto, es que puede determinar en una operación el número de índice y que una VLAN está operativa.

En el ejemplo siguiente se proporciona más información.

### [El MIB puente](#)

Desde BRIDGE-MIB, que se extrae de [RFC 1493](#) , puede utilizar la [dirección](#) de [dot1dTpFdb](#) de la [dot1dTpFdbTable](#), donde el valor es igual a 3 o [aprendido](#), para determinar qué direcciones de Control de Acceso a Medios (MAC) son switches en la tabla de reenvío ... Este valor se almacena como una dirección MAC de unidifusión para la cual el puente tiene información de reenvío y/o filtrado. Estos valores de dirección MAC por sí solos no significan mucho y pueden producir muchos datos. Por lo tanto, debe contar el número de entradas y almacenar ese valor de recuento, basándose en un `dot1dTpFdbStatus` (.1.3.6.1.2.1.17.4.3.1.3) igual a *aprendido* (valor de 3).

**Nota:** BRIDGE-MIB utiliza la indexación de cadenas de comunidad para acceder a una instancia determinada de MIB como se describe en la [indexación de cadenas de comunidad SNMP](#).

Los datos de dirección MAC de tendencias son valiosos para realizar un seguimiento del número total de entradas CAM (direcciones MAC) aprendidas dinámicamente por el switch. Esta supervisión ayuda a realizar un seguimiento de la flexibilidad de la red, especialmente cuando se relaciona con el número total de LAN virtuales (VLAN) por switch. Por ejemplo, si tiene una VLAN definida en el switch y ve 8000 direcciones MAC, sabe que tiene 8000 direcciones MAC para una VLAN, que es extensa para una subred.

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

Las definiciones de valor son:

- **otros (1):** Ninguna de las siguientes. Esto incluye casos en los que se está utilizando algún otro objeto MIB (no la instancia correspondiente de `dot1dTpFdbPort`, ni una entrada en la `dot1dStaticTable`) para determinar si se están reenviando las direcciones de trama al valor de la instancia correspondiente de `dot1dTpFdbAddress` y cómo hacerlo.
- **inválido (2):** Esta entrada ya no es válida (por ejemplo, se aprendió, pero ha caducado desde entonces), pero todavía no se ha vaciado de la tabla.
- **aprendido (3):** Se aprendió el valor de la instancia correspondiente de `dot1dTpFdbPort` y se está utilizando.
- **yo (4):** El valor de la instancia correspondiente de `dot1dTpFdbAddress` representa una de las direcciones del puente. La instancia correspondiente de `dot1dTpFdbPort` indica cuál de los puertos del puente tiene esta dirección.
- **mgmt (5):** El valor de la instancia correspondiente de `dot1dTpFdbAddress` también es el valor de una instancia existente de `dot1dStaticAddress`.

## Detalles de las variables MIB: incluye identificadores de objetos (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 }

## [Recuperar la información CAM dinámica con SNMP](#)

### [Step-by-Step Instructions](#)

Siga estos pasos para obtener información CAM dinámica con SNMP.

1. Recupere las VLAN. Utilice **snmpwalk** en el objeto [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 VLAN, obtenga la tabla de direcciones MAC (usando [indización de cadena de comunidad](#)) [dot1dTpFdbAddress](#) (.1.3.6.1.2.1.17.4.3.1.1). En el ejemplo, VLAN 2 no contiene entradas 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 VLAN, obtenga el número de puerto de puente, [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 que el puerto puente [ifIndex](#) (1.3.6.1.2.1.2.2.1.1) mapee, [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. Camine por el [ifName](#) (.1.3.6.1.2.1.31.1.1.1.1) para que el valor ifIndex obtenido en el paso 4 pueda relacionarse con un nombre de puerto adecuado:

```
nms-server2:/home/ccarring> snmpwalk -On -c public 14.32.6.17 ifName
.1.3.6.1.2.1.31.1.1.1.1.1 = STRING: sc0
.1.3.6.1.2.1.31.1.1.1.1.2 = STRING: sl0
.1.3.6.1.2.1.31.1.1.1.1.3 = STRING: me1
.1.3.6.1.2.1.31.1.1.1.1.4 = STRING: VLAN-1
.1.3.6.1.2.1.31.1.1.1.1.5 = STRING: VLAN-1002
.1.3.6.1.2.1.31.1.1.1.1.6 = STRING: VLAN-1004
.1.3.6.1.2.1.31.1.1.1.1.7 = STRING: VLAN-1005
.1.3.6.1.2.1.31.1.1.1.1.8 = STRING: VLAN-1003
.1.3.6.1.2.1.31.1.1.1.1.9 = STRING: 2/1
.1.3.6.1.2.1.31.1.1.1.1.10 = STRING: 2/2
.1.3.6.1.2.1.31.1.1.1.1.11 = STRING: 2/3
.1.3.6.1.2.1.31.1.1.1.1.12 = STRING: 2/4
.1.3.6.1.2.1.31.1.1.1.1.13 = STRING: 2/5
.1.3.6.1.2.1.31.1.1.1.1.14 = STRING: 2/6
.1.3.6.1.2.1.31.1.1.1.1.15 = STRING: 2/7
.1.3.6.1.2.1.31.1.1.1.1.16 = STRING: 2/8
.1.3.6.1.2.1.31.1.1.1.1.17 = STRING: 2/9
.1.3.6.1.2.1.31.1.1.1.1.18 = STRING: 2/10
.1.3.6.1.2.1.31.1.1.1.1.19 = STRING: 2/11
.1.3.6.1.2.1.31.1.1.1.1.20 = STRING: 2/12
.1.3.6.1.2.1.31.1.1.1.1.21 = STRING: 2/13
.1.3.6.1.2.1.31.1.1.1.1.22 = STRING: 2/14
.1.3.6.1.2.1.31.1.1.1.1.23 = STRING: 2/15
.1.3.6.1.2.1.31.1.1.1.1.24 = STRING: 2/16
.1.3.6.1.2.1.31.1.1.1.1.25 = STRING: 2/17
.1.3.6.1.2.1.31.1.1.1.1.26 = STRING: 2/18
.1.3.6.1.2.1.31.1.1.1.1.27 = STRING: 2/19
.1.3.6.1.2.1.31.1.1.1.1.28 = STRING: 2/20
.1.3.6.1.2.1.31.1.1.1.1.29 = STRING: 2/21
.1.3.6.1.2.1.31.1.1.1.1.30 = STRING: 2/22
```

```

.1.3.6.1.2.1.31.1.1.1.1.31 = STRING: 2/23
.1.3.6.1.2.1.31.1.1.1.1.32 = STRING: 2/24
.1.3.6.1.2.1.31.1.1.1.1.33 = STRING: 2/25
.1.3.6.1.2.1.31.1.1.1.1.34 = STRING: 2/26
.1.3.6.1.2.1.31.1.1.1.1.35 = STRING: 2/27
.1.3.6.1.2.1.31.1.1.1.1.36 = STRING: 2/28
.1.3.6.1.2.1.31.1.1.1.1.37 = STRING: 2/29
.1.3.6.1.2.1.31.1.1.1.1.38 = STRING: 2/30
.1.3.6.1.2.1.31.1.1.1.1.39 = STRING: 2/31
.1.3.6.1.2.1.31.1.1.1.1.40 = STRING: 2/32
.1.3.6.1.2.1.31.1.1.1.1.41 = STRING: 2/33
.1.3.6.1.2.1.31.1.1.1.1.42 = STRING: 2/34
.1.3.6.1.2.1.31.1.1.1.1.43 = STRING: 2/35
.1.3.6.1.2.1.31.1.1.1.1.44 = STRING: 2/36
.1.3.6.1.2.1.31.1.1.1.1.45 = STRING: 2/37
.1.3.6.1.2.1.31.1.1.1.1.46 = STRING: 2/38
.1.3.6.1.2.1.31.1.1.1.1.47 = STRING: 2/39
.1.3.6.1.2.1.31.1.1.1.1.48 = STRING: 2/40
.1.3.6.1.2.1.31.1.1.1.1.49 = STRING: 2/41
.1.3.6.1.2.1.31.1.1.1.1.50 = STRING: 2/42
.1.3.6.1.2.1.31.1.1.1.1.51 = STRING: 2/43
.1.3.6.1.2.1.31.1.1.1.1.52 = STRING: 2/44
.1.3.6.1.2.1.31.1.1.1.1.53 = STRING: 2/45
.1.3.6.1.2.1.31.1.1.1.1.54 = STRING: 2/46
.1.3.6.1.2.1.31.1.1.1.1.55 = STRING: 2/47
.1.3.6.1.2.1.31.1.1.1.1.56 = STRING: 2/48
.1.3.6.1.2.1.31.1.1.1.1.57 = STRING: 2/49
.1.3.6.1.2.1.31.1.1.1.1.58 = STRING: 2/50
.1.3.6.1.2.1.31.1.1.1.1.59 = STRING: VLAN-2
.1.3.6.1.2.1.31.1.1.1.1.60 = STRING: VLAN-6
.1.3.6.1.2.1.31.1.1.1.1.61 = STRING: VLAN-7
.1.3.6.1.2.1.31.1.1.1.1.62 = STRING: VLAN-8
.1.3.6.1.2.1.31.1.1.1.1.63 = STRING: VLAN-11
.1.3.6.1.2.1.31.1.1.1.1.64 = STRING: VLAN-12
.1.3.6.1.2.1.31.1.1.1.1.65 = STRING: VLAN-18
.1.3.6.1.2.1.31.1.1.1.1.66 = STRING: VLAN-19
.1.3.6.1.2.1.31.1.1.1.1.67 = STRING: VLAN-20
.1.3.6.1.2.1.31.1.1.1.1.68 = STRING: VLAN-21
.1.3.6.1.2.1.31.1.1.1.1.69 = STRING: VLAN-41
.1.3.6.1.2.1.31.1.1.1.1.70 = STRING: VLAN-42
.1.3.6.1.2.1.31.1.1.1.1.71 = STRING: VLAN-43
.1.3.6.1.2.1.31.1.1.1.1.72 = STRING: VLAN-44
.1.3.6.1.2.1.31.1.1.1.1.73 = STRING: VLAN-100
.1.3.6.1.2.1.31.1.1.1.1.74 = STRING: VLAN-101
.1.3.6.1.2.1.31.1.1.1.1.75 = STRING: VLAN-123
.1.3.6.1.2.1.31.1.1.1.1.76 = STRING: VLAN-401
.1.3.6.1.2.1.31.1.1.1.1.77 = STRING: VLAN-14

```

Ahora se puede utilizar la información del puerto obtenida, por ejemplo: Desde el Paso 2 , hay una dirección MAC: .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 Desde el paso 3: .1.3.6.1.2.1.17.4.3.1.2.0.208.211.106.71.251 = ENTERO: 113 Esto le indica que esta dirección MAC (00 D0 D3 6A 47 FB) proviene del puerto bridge número 113. Desde el Paso 4, el puerto bridge número 113 tiene un ifIndex número 57. 1.3.6.1.2.1.17.1.4.1.2.113 = INTEGER: 57 Desde el paso 5, el ifIndex 57 corresponde al puerto 2/49. 1.3.6.1.2.1.31.1.1.1.1.57 = STRING: 2/49 Compare eso con el resultado del comando show cam dynamic para switches CatOS o el resultado del comando show mac para switches CatIOS. Verá una coincidencia para 1 00-d0-d3-6a-47-fb 2/49 [ALL].

## Verificación

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

1. Telnet a su switch.
2. Desde la línea de comandos, ejecute el comando apropiado: Dispositivos CatOS: **show cam dynamic** Dispositivos CatIOS: **show mac**
3. Compare el resultado con los resultados obtenidos mediante 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](#)