How To Get Dynamic CAM Entries (CAM Table) for Catalyst Switches Using SNMP

Document ID: 13492

Introduction

This document describes how to collect Dynamic Content–Addressable Memory (CAM) entries for Catalyst switches using Simple Network Management Protocol (SNMP).

Prerequisites

Requirements

Before using the information in this document, ensure that you meet the following prerequisites:

- Understand how to get VLANs from a Catalyst switch via SNMP.
- Understand how to use Community String Indexing with SNMP.
- General use of SNMP **get** and **walk** commands

Components Used

This document applies to Catalyst switches that run regular Catalyst OS or Catalyst IOS® that support the BRIDGE–MIB ⬈. The information in this document is based on the software and hardware versions below.

- Catalyst 3524XL running CatIOS 12.0(5)WC5a
- Catalyst 2948G running CatOS 6.3(3)
- NET–SNMP available at http://www.net–snmp.org/

The information presented in this document was created from devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If you are working in a live network, ensure that you understand the potential impact of any command before using it.

Conventions

For more information on document conventions, see the Cisco Technical Tips Conventions.

Background

The CISCO–VTP–MIB

In order to access MIBs that have a separate instance for each VLAN, you must first understand how to use community string indexing. Next, you need to know the specific VLANs that are active on a given switch. From the CISCO–VTP–MIB, you can obtain the active VLANs on a switch using the vtpVlanState object. The reason to use the vtpVlanState object, and not vtpVlanName or another object, is that you can determine in one operation the index number and that a VLAN is operational.

More information is given in the example below.

The BRIDGE–MIB

From the BRIDGE–MIB, which is extracted from RFC 1493 , you can use the dot1dTpFdbAddress from the `dot1dTpFdbTable`, where the value is equal to 3 or *learned*, to determine what Media Access Control (MAC) addresses are in the forwarding table on the switch. This value is stored as a unicast MAC address for which the bridge has forwarding and/or filtering information. These MAC address values alone do not mean much and can produce a lot of data. Therefore, you need to count the number of entries and store that count value, based on a `dot1dTpFdbStatus` (.1.3.6.1.2.1.17.4.3.1.3) equal to *learned* (value of 3).

Note: The BRIDGE–MIB uses community string indexing to access a particular instance of the MIB as described in SNMP Community String Indexing.

Trending MAC address data is valuable for keeping track of the total number of CAM entries (MAC addresses) learned dynamically by the switch. This monitoring helps keep track of the flatness in your network, especially when correlating to the total number of Virtual LANs (VLANs) per switch. For example, if you have one VLAN defined on the switch and you see 8,000 MAC addresses, you know you have 8,000 MAC addresses for one VLAN, which is extensive for one subnet.

A related MIB object from the BRIDGE–MIB (RFC 1493) is `dot1dTpFdbStatus`. This MIB provides the status of the MAC address entry.

The value definitions are:

- **other (1):** None of the following. This includes cases where some other MIB object (not the corresponding instance of `dot1fTpFdbPort`, nor an entry in the `dot1dStaticTable`) is being used to determine if and how frame addresses to the value of the corresponding instance of `dot1dTpFdbAddress` are being forwarded.
- **invalid (2):** This entry is no longer valid (for example, it was learned but has since aged out), but has not yet been flushed from the table.
- **learned (3):** The value of the corresponding instance of `dot1dTpFdbPort` was learned, and is being used.
- **self (4):** The value of the corresponding instance of `dot1dTpFdbAddress` represents one of the bridge's addresses. The corresponding instance of `dot1dTpFdbPort` indicates which of the bridge's ports has this address.

- **mgmt (5):** The value of the corresponding instance of `dot1dTpFdbAddress` is also the value of an existing instance of `dot1dStaticAddress`.

Details of the MIB Variables––Includes Object Identifiers (OIDs)

```
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 }
```

Retrieve the Dynamic CAM Information with SNMP

Step–by–Step Instructions

Follow these steps to obtain dynamic CAM information with SNMP.

1. Retrieve the VLANs. Use **snmpwalk** on the vtpVlanState object (.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. For each VLAN, get the MAC address table (using community string indexing) dot1dTpFdbAddress (.1.3.6.1.2.1.17.4.3.1.1). In the example, VLAN 2 contains no entries in the table:

```
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. For each VLAN, get the bridge port number, 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. Get the bridge port to ifIndex (1.3.6.1.2.1.2.2.1.1) mapping, dot1dBasePortIfIndex
   (.1.3.6.1.2.1.17.1.4.1.2):

```
            nms-server2:/home/ccarring> snmpwalk -c public 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. Walk the ifName (.1.3.6.1.2.1.31.1.1.1.1) so that the ifIndex value obtained in step 4 can be correllated with a proper port name:

```
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
```

Now the port information obtained can be used, for example:

♦ From Step 2 , there is a MAC address: .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
♦ From Step 3: .1.3.6.1.2.1.17.4.3.1.2.0.208.211.106.71.251 = INTEGER: 113

This tells you that this MAC address (00 D0 D3 6A 47 FB) is from bridge port number 113.
♦ From Step 4, the bridge port number 113 has an ifIndex number 57 .1.3.6.1.2.1.17.1.4.1.2.113 = INTEGER: 57
♦ From Step 5, the ifIndex 57 corresponds to port 2/49 .1.3.6.1.2.1.31.1.1.1.1.57 = STRING: 2/49

Compare that with the output from the **show cam dynamic** command output for CatOS switches, or **show mac** command output for CatIOS switches. You see a match for 1 00−d0−d3−6a−47−fb 2/49 [ALL].

Verify

This section provides information you can use to confirm your configuration is working properly.

1. Telnet to your switch.
2. From the command line, issue the appropriate command:

♦ CatOS devices: **show cam dynamic**
♦ CatIOS devices: **show mac**
3. Compare the output with the results obtained by the procedure specified here.

```
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)
```

NetPro Discussion Forums – Featured Conversations

Networking Professionals Connection is a forum for networking professionals to share questions, suggestions, and information about networking solutions, products, and technologies. The featured links are some of the most recent conversations available in this technology.

NetPro Discussion Forums – Featured Conversations for Network Management
Network Infrastructure: Network Management
Virtual Private Networks: Network and Policy Management

Related Information

- **SNMP Community String Indexing**
- **Technical Support – Cisco Systems**

Updated: Jul 09, 2007   Document ID: 13492

Cisco – How To Get Dynamic CAM Entries (CAM Table) for Catalyst Switches Using SNMP