

Comment obtenir des entrées CAM dynamiques (table CAM) pour des commutateurs Catalyst à l'aide de SNMP

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Components Used](#)

[Conventions](#)

[Fond](#)

[CISCO-VTP-MIB](#)

[BRIDGE-MIB](#)

[Détails des variables MIB - Inclut les OID \(Object Identifier\)](#)

[Récupérer les informations CAM dynamiques avec SNMP](#)

[Step-by-Step Instructions](#)

[Vérification](#)

[Informations connexes](#)

Introduction

Ce document décrit comment collecter les entrées dynamiques de mémoire de contenu adressable (CAM) pour des commutateurs Catalyst utilisant le protocole de gestion de réseau simple (SNMP).

Conditions préalables

Conditions requises

Avant d'utiliser les informations de ce document, assurez-vous de respecter les conditions préalables suivantes :

- Comprendre comment obtenir des VLAN à partir d'un commutateur Catalyst via SNMP.
- Comprendre comment utiliser l'[indexation des chaînes de communauté SNMP](#).
- Utilisation générale des commandes **get** et **walk** SNMP.

Components Used

Ce document s'applique aux commutateurs Catalyst qui exécutent Catalyst OS ou Catalyst IOS® standard qui prennent en charge [BRIDGE-MIB](#). Les informations dans ce document sont basées

sur les versions de logiciel et de matériel ci-dessous.

- Catalyst 3524XL exécutant CatIOS 12.0(5)WC5a
- Catalyst 2948G exécutant CatOS 6.3(3)
- NET-SNMP disponible à l'adresse <http://www.net-snmp.org/> 

Les informations présentées dans ce document ont été créées à partir de périphériques dans un environnement de laboratoire spécifique. All of the devices used in this document started with a cleared (default) configuration. Si vous travaillez dans un réseau opérationnel, assurez-vous de bien comprendre l'impact potentiel de toute commande avant de l'utiliser.

[Conventions](#)

Pour plus d'informations sur les conventions des documents, référez-vous aux [Conventions utilisées pour les conseils techniques de Cisco](#).

[Fond](#)

[CISCO-VTP-MIB](#)

Afin d'accéder aux MIB qui ont une instance distincte pour chaque VLAN, vous devez d'abord comprendre comment utiliser l'[indexation des chaînes de communauté](#). Ensuite, vous devez connaître les VLAN spécifiques actifs sur un commutateur donné. À partir de [CISCO-VTP-MIB](#), vous pouvez obtenir les VLAN actifs sur un commutateur à l'aide de l'objet [vtpVlanState](#). La raison d'utiliser l'objet vtpVlanState, et non vtpVlanName ou un autre objet, est que vous pouvez déterminer dans une opération le numéro d'index et qu'un VLAN est opérationnel.

Plus d'informations sont fournies dans l'exemple ci-dessous.

[BRIDGE-MIB](#)

À partir du BRIDGE-MIB, qui est extrait de [RFC 1493](#) , vous pouvez utiliser la [dot1dTpFdbAddress](#) à partir du `dot1dTpFdbTable`, où la valeur est égale à 3 ou *apprise*, pour déterminer les adresses MAC (Media Access Control) dans le sur le commutateur. Cette valeur est stockée en tant qu'adresse MAC de monodiffusion pour laquelle le pont dispose d'informations de transfert et/ou de filtrage. Ces seules valeurs d'adresse MAC ne signifient pas grand chose et peuvent produire beaucoup de données. Par conséquent, vous devez compter le nombre d'entrées et stocker cette valeur de comptage, basée sur un `dot1dTpFdbStatus` (.1.3.6.1.2.1.17.4.3.1.3) égal à *appris* (valeur 3).

Remarque : BRIDGE-MIB utilise l'indexation des chaînes de communauté pour accéder à une instance particulière de la base de données MIB, comme décrit dans l'[indexation des chaînes de communauté SNMP](#).

Les données d'adresse MAC de tendance sont utiles pour suivre le nombre total d'entrées CAM (adresses MAC) apprises dynamiquement par le commutateur. Cette surveillance permet de suivre l'état d'intégrité de votre réseau, en particulier lors de la corrélation avec le nombre total de VLAN par commutateur. Par exemple, si vous avez un VLAN défini sur le commutateur et que vous voyez 8 000 adresses MAC, vous savez que vous avez 8 000 adresses MAC pour un VLAN, qui est étendu pour un sous-réseau.

Un objet MIB associé de la [BRIDGE-MIB](#) (RFC 1493) est `dot1dTpFdbStatus`. Cette MIB fournit l'état de l'entrée d'adresse MAC.

Les définitions de valeur sont les suivantes :

- **autre (1)** : Aucun des éléments suivants. Cela inclut les cas où un autre objet MIB (pas l'instance correspondante de `dot1dTpFdbPort`, ni une entrée dans la `dot1dStaticTable`) est utilisé pour déterminer si et comment les adresses de trame à la valeur de l'instance correspondante de `dot1dTpFdbAddress` sont transférées.
- **non valide (2)** : Cette entrée n'est plus valide (par exemple, elle a été apprise mais a depuis expiré), mais n'a pas encore été supprimée de la table.
- **acquis (3)** : La valeur de l'instance correspondante de `dot1dTpFdbPort` a été apprise et est utilisée.
- **auto (4)** : La valeur de l'instance correspondante de `dot1dTpFdbAddress` représente une des adresses du pont. L'instance correspondante de `dot1dTpFdbPort` indique lequel des ports du pont a cette adresse.
- **mgmt (5)** : La valeur de l'instance correspondante de `dot1dTpFdbAddress` est également la valeur d'une instance existante de `dot1dStaticAddress`.

Détails des variables MIB - Inclut les OID (Object Identifier)

```
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 }

[Récupérer les informations CAM dynamiques avec SNMP](#)

[Step-by-Step Instructions](#)

Suivez ces étapes pour obtenir des informations CAM dynamiques avec SNMP.

1. Récupérez les VLAN. Utilisez **snmpwalk** sur l'objet [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. Pour chaque VLAN, obtenez la table d'adresses MAC (à l'aide de l'[indexation des chaînes de communauté](#) [dot1dTpFdbAddress](#) (.1.3.6.1.2.1.17.4.3.1.1)). Dans l'exemple, VLAN 2 ne contient aucune entrée dans la 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. Pour chaque VLAN, obtenez le numéro de port de pont, [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. Obtenir le port de pont au [mappage ifIndex](#) (1.3.6.1.2.1.2.2.1.1), [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. Parcourez le [ifName](#) (.1.3.6.1.2.1.31.1.1.1.1) afin que la valeur ifIndex obtenue à l'étape 4 puisse être corrélée avec un nom de port approprié :

```
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

```

Maintenant, les informations de port obtenues peuvent être utilisées, par exemple :Àl'étape 2 , il existe une adresse MAC : .1.3.6.1.2.1.17.4.3.1.1.0.208.211.106.71.251 = CHAÎNE hexadécimale : 00 D0 D3 6A 47 FBÀpartir de l'étape 3 : .1.3.6.1.2.1.17.4.3.1.2.0.208.211.106.71.251 = INTEGER : 113Ceci vous indique que cette adresse MAC (00 D0 D3 6A 47 FB) provient du port de pont numéro 113.Àpartir de l'étape 4, le numéro de port de pont 113 a un numéro ifIndex 57.1.3.6.1.2.1.17.1.4.1.2.113 = INTEGER : 57Àpartir de l'étape 5, ifIndex 57 correspond au port 2/49.1.3.6.1.2.1.31.1.1.1.1.57 =

CHAÎNE : 2/49 Comparez cela avec le résultat de la sortie de la commande **show cam dynamic** pour les commutateurs CatOS ou **show mac** pour les commutateurs CatIOS. Vous voyez une correspondance pour 1 00-d0-d3-6a-47-fb 2/49 [ALL].

Vérification

Cette section présente des informations que vous pouvez utiliser pour vous assurer que votre configuration fonctionne correctement.

1. Établissez une connexion Telnet avec votre commutateur.
2. À partir de la ligne de commande, exécutez la commande appropriée : Périphériques CatOS : **show cam dynamic** Périphériques CatIOS : **show mac**
3. Comparez le résultat avec les résultats obtenus par la procédure spécifiée ici.

```
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)
```

Informations connexes

- [Indexation de la chaîne communautaire SNMP](#)
- [Support technique - Cisco Systems](#)