

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

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Conventions](#)

[Fond](#)

[Le CISCO-VTP-MIB](#)

[Le MIB DE PONT](#)

[Détails des variables MIB--Inclut les identifiants d'objet \(les OID\)](#)

[Récupérez les informations de CAM dynamique avec le SNMP](#)

[Instructions pas à pas](#)

[Vérifiez](#)

[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 dans ce document, assurez-vous que vous rencontrez les conditions préalables suivantes :

- Comprenez comment obtenir des VLAN d'un commutateur de Catalyst par l'intermédiaire du SNMP.
- Comprenez comment utiliser l'[indexation de chaîne de caractères de la communauté SNMP](#).
- Utilisation générale de snmp get et de commandes d'**inspection**.

Composants utilisés

Ce document applique aux Commutateurs de Catalyst qui exécutent le [®] régulier de SYSTÈME

D'EXPLOITATION de Catalyst ou IOS de Catalyst qui prennent en charge le [MIB DE PONT](#). 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 chez <http://www.net-snmp.org/>

Les informations présentées dans ce document ont été créées à partir de périphériques dans un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). 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](#)

[Le CISCO-VTP-MIB](#)

Afin d'accéder au MIB qui ont un exemple distinct pour chaque VLAN, vous devez d'abord comprendre comment utiliser l'[indexation de chaîne de la communauté](#). Ensuite, vous devez connaître la particularité VLAN qui sont en activité sur un commutateur donné. [Du CISCO-VTP-MIB](#), vous pouvez obtenir les VLAN actifs sur un commutateur utilisant l'objet de [vtpVlanState](#). La raison d'utiliser l'objet de [vtpVlanState](#), et pas le [vtpVlanName](#) ou un autre objet, est que vous pouvez déterminer dans une exécution l'index et qu'un VLAN est opérationnel.

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

[Le MIB DE PONT](#)

[Du MIB DE PONT](#), qui est extrait de [RFC 1493](#), vous pouvez employer le [dot1dTpFdbAddress](#) du `dot1dTpFdbTable`, où la valeur est égale à 3 ou *apprise*, pour déterminer quelles adresses de Contrôle d'accès au support (MAC) sont dans la table d'expédition sur le commutateur. Cette valeur est enregistrée comme adresse MAC d'unicast pour laquelle la passerelle a l'expédition et/ou les informations de filtrage. Ces seules valeurs d'adresse MAC ne signifient pas beaucoup et peuvent produire beaucoup de données. Par conséquent, vous devez compter le nombre d'entrées et enregistrer cette valeur de compte, basé sur un .1.3.6.1.2.1.17.4.3.1.3) égal `dot1dTpFdbStatus` (à *apprendre* (valeur de 3)).

Note: L'indexation de chaîne de la communauté d'utilisations de MIB DE PONT pour accéder à un exemple particulier du MIB comme décrit dans l'[indexation de chaîne de caractères de la communauté SNMP](#).

Tendre des données d'adresse MAC est valeur pour maintenir le nombre total d'entrées de CAM (adresses MAC) apprises dynamiquement par le commutateur. Cette surveillance aide à maintenir la planéité dans votre réseau, particulièrement en corrélant au nombre total de réseaux locaux virtuels (VLAN) par commutateur. Par exemple, si vous avez un VLAN défini sur le commutateur et vous pour voir 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 relatif MIB du [MIB DE PONT](#) (RFC 1493) est `dot1dTpFdbStatus`. Ce MIB fournit le statut de l'entrée d'adresse MAC.

Les définitions de valeur sont :

- **autre (1)** : Aucune du suivant. Ceci inclut des cas où un autre objet MIB (pas l'exemple correspondant de `dot1dTpFdbPort`, ni une entrée dans le `dot1dStaticTable`) est utilisé pour déterminer si et comment des adresses de trame à la valeur de l'exemple correspondant de `dot1dTpFdbAddress` sont expédiées.
- **(2) non valide** : Cette entrée n'est plus valide (par exemple, on l'a appris mais depuis a vieilli), mais n'a pas été encore vidée de la table.
- **(3) instruit** : La valeur de l'exemple correspondant de `dot1dTpFdbPort` a été apprise, et est utilisée.
- **individu (4)** : La valeur de l'exemple correspondant de `dot1dTpFdbAddress` représente une des adresses de la passerelle. L'exemple correspondant de `dot1dTpFdbPort` indique lesquels des ports de la passerelle ont cette adresse.
- **mgmt (5)** : La valeur de l'exemple correspondant de `dot1dTpFdbAddress` est également la valeur d'un exemple existant de `dot1dStaticAddress`.

Détails des variables MIB--Inclut les identifiants d'objet (les 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 }
```

[Récupérez les informations de CAM dynamique avec le SNMP](#)

[Instructions pas à pas](#)

Suivez ces étapes pour obtenir les informations de CAM dynamique avec le SNMP.

1. Récupérez les VLAN. **Snmwalk** d'utilisation sur l'objet de [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'adresse MAC (utilisant l'[indexation de chaîne de la communauté dot1dTpFdbAddress](#) (.1.3.6.1.2.1.17.4.3.1.1)). Dans l'exemple, le 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 passerelle, [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. Obtenez le port de passerelle au mappage de l'[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. Marchent l'[ifName](#) (.1.3.6.1.2.1.31.1.1.1) de sorte que la valeur d'[ifIndex](#) obtenue dans l'étape 4 puisse corréllated avec un nom de port approprié :

```
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.

Maintenant les informations de port obtenues peuvent être utilisées, par exemple : De l'étape 2, il y a une adresse MAC : .1.3.6.1.2.1.17.4.3.1.1.0.208.211.106.71.251 = Hexa-CHAÎNE : 00 D0 D3 6A 47 FB De l'étape 3 : .1.3.6.1.2.1.17.4.3.1.2.0.208.211.106.71.251 = ENTIER : 113 Ceci t'indique que cette adresse MAC (00 D0 D3 6A 47 FB) est du numéro de port 113 de passerelle. De l'étape 4, le numéro de port 113 de passerelle a un ifIndex le numéro 57 .1.3.6.1.2.1.17.1.4.1.2.113 = ENTIER : 57 De l'étape 5, l'ifIndex 57 correspond au port 2/49 .1.3.6.1.2.1.31.1.1.1.1.57 = CHAÎNE : 2/49 Comparez cela à la sortie de la sortie de commande **dynamique de show cam** pour des Commutateurs de CatOS, ou de la sortie de commande de **show mac** pour des Commutateurs de CatIOS. Vous voyez une correspondance pour 1 00-d0-d3-6a-47-fb 2/49 [TOUT].

Vérifiez

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

1. Telnet à votre commutateur.
2. De la ligne de commande, émettez la commande appropriée : Périphériques de CatOS : **show cam dynamique** Périphériques de CatIOS : [show mac](#)
3. Comparez la sortie aux 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](#)