

Comment ajouter, modifier et supprimer les VLAN sur Catalyst à l'aide de SNMP

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants](#)

[Conventions](#)

[Fond](#)

[Détails des variables MIB — Y compris les identifiants d'objet \(OID\)](#)

[Ajoutez un VLAN à un commutateur Cisco Catalyst avec le SNMP](#)

[Instructions pas à pas](#)

[Ajoutez un VLAN à un commutateur Cisco Catalyst avec le SNMP](#)

[Instructions détaillée une](#)

[Supprimez un VLAN d'un commutateur Cisco Catalyst avec le SNMP](#)

[Instructions pas à pas](#)

[Ajoutez un port à un VLAN sur un commutateur Cisco Catalyst avec le SNMP](#)

[Comment changer un port d'un VLAN à un autre VLAN](#)

[Informations connexes](#)

[Introduction](#)

Ce document explique comment créer et supprimer des VLAN sur un commutateur Cisco Catalyst qui utilise le protocole SNMP. Il décrit également comment ajouter des ports à un VLAN avec le protocole SNMP.

[Conditions préalables](#)

[Conditions requises](#)

Avant que vous utilisiez les informations dans ce document, assurez-vous que vous comprenez :

- Comment les ifTable et les ifIndexes fonctionnent
- Comment les VLAN travaillent aux commutateurs Cisco Catalyst
- Comment visualiser les informations VLAN sur des Catalyst de Cisco commute
- L'utilisation générale du snmp get, du **positionnement**, et des commandes d'**inspection**

[Composants](#)

Ce document est pour les Commutateurs de Catalyst qui exécutent le SYSTÈME D'EXPLOITATION régulier de Catalyst ou l'IOS de Catalyst qui prennent en charge l'IF-MIB, le CISCO-VTP-MIB et le CISCO-VLAN-MEMBERSHIP-MIB. Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- Catalyst 3524XL exécutant CatIOS 12.0(5)WC5a
- Version 5.0.6 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 fonctionnez dans un réseau vivant, avant que vous utilisiez n'importe quelle commande assurez-vous que vous comprenez l'impact potentiel de n'importe quelle commande.

Conventions

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

Fond

Détails des variables MIB — Y compris les identifiants d'objet (OID)

1.3.6.1.4.1.9.9.46.1.3.1.1.2 (CISCO-VTP-MIB)

```
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 }
```

1.3.6.1.4.1.9.9.46.1.4.1.1.1 (CISCO-VTP-MIB)

```
vtpVlanEditOperation OBJECT-TYPE
    SYNTAX      INTEGER { none(1),
                        copy(2),
                        apply(3),
                        release(4),
                        restartTimer(5)
                    }
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION "This object always has the value 'none' when read. When
```

written, each value causes the appropriate action:

'copy' - causes the creation of rows in the vtpVlanEditTable exactly corresponding to the current global VLAN information for this management domain. If the Edit Buffer (for this management domain) is not currently empty, a copy operation fails. A successful copy operation starts the deadman-timer.

'apply' - first performs a consistent check on the the modified information contained in the Edit Buffer, and if consistent, then tries to instantiate the modified information as the new global VLAN information. Note that an empty Edit Buffer (for the management domain) would always result in an inconsistency since the default VLANs are required to be present.

'release' - flushes the Edit Buffer (for this management domain), clears the Owner information, and aborts the deadman-timer. A release is generated automatically if the deadman-timer ever expires.

'restartTimer' - restarts the deadman-timer.

'none' - no operation is performed."

::= { vtpEditControlEntry 1 }

1.3.6.1.4.1.9.9.46.1.4.1.1.3 (CISCO-VTP-MIB)

vtpVlanEditBufferOwner OBJECT-TYPE

SYNTAX OwnerString

MAX-ACCESS read-create

STATUS current

DESCRIPTION "The management station which is currently using the Edit Buffer for this management domain. When the Edit Buffer for a management domain is not currently in use, the value of this object is the zero-length string. Note that it is also the zero-length string if a manager fails to set this object when invoking a copy operation."

::= { vtpEditControlEntry 3 }

1.3.6.1.4.1.9.9.46.1.4.2.1.11 (CISCO-VTP-MIB)

vtpVlanEditRowStatus OBJECT-TYPE

SYNTAX RowStatus

1:active

2:notInService

3:notReady

4:createAndGo

5:createAndWait

6:destroy

MAX-ACCESS read-create

STATUS current

DESCRIPTION "The status of this row. Any and all columnar objects in an existing row can be modified irrespective of the status of the row.

A row is not qualified for activation until instances of at least its vtpVlanEditType, vtpVlanEditName and vtpVlanEditDot10Said columns have appropriate values.

The management station should endeavor to make all rows consistent in the table before 'applying' the buffer. An inconsistent entry in the table will cause the entire

buffer to be rejected with the vtpVlanApplyStatus object set to the appropriate error value."
 ::= { vtpVlanEditEntry 11 }

1.3.6.1.4.1.9.9.46.1.4.2.1.3.1.48 (CISCO-VTP-MIB)

vtpVlanEditType OBJECT-TYPE

SYNTAX VlanType
MAX-ACCESS read-create
STATUS current
DESCRIPTION "The type which this VLAN would have.
An implementation may restrict access to this object."
DEFVAL { ethernet }
 ::= { vtpVlanEditEntry 3 }

1.3.6.1.4.1.9.9.46.1.4.2.1.4.1.48 (CISCO-VTP-MIB)

vtpVlanEditName OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..32))
MAX-ACCESS read-create
STATUS current
DESCRIPTION "The name which this VLAN would have. This name would be
used as the ELAN-name for an ATM LAN-Emulation segment of
this VLAN.

An implementation may restrict access to this object."
 ::= { vtpVlanEditEntry 4 }

1.3.6.1.4.1.9.9.46.1.4.2.1.6.1.48 (CISCO-VTP-MIB)

vtpVlanEditDot10Said OBJECT-TYPE

SYNTAX OCTET STRING (SIZE (4))
MAX-ACCESS read-create
STATUS current
DESCRIPTION "The value of the 802.10 SAID field which would be used for
this VLAN.

An implementation may restrict access to this object."
 ::= { vtpVlanEditEntry 6 }

1.3.6.1.4.1.9.9.46.1.4.1.1.2.1 (CISCO-VTP-MIB)

vtpVlanApplyStatus OBJECT-TYPE

SYNTAX INTEGER { inProgress(1),
succeeded(2),
configNumberError(3),
inconsistentEdit(4),
tooBig(5),
localNVStoreFail(6),
remoteNVStoreFail(7),
editBufferEmpty(8),
someOtherError(9)
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The current status of an 'apply' operation to instanciate
the Edit Buffer as the new global VLAN information (for this
management domain). If no apply is currently active, the
status represented is that of the most recently completed
apply. The possible values are:

inProgress - 'apply' operation in progress;

succeeded - the 'apply' was successful (this value is

```

also used when no apply has been invoked since the
last time the local system restarted);

configNumberError - the apply failed because the value of
vtpVlanEditConfigRevNumber was less or equal to
the value of current value of
managementDomainConfigRevNumber;

inconsistentEdit - the apply failed because the modified
information was not self-consistent;

tooBig - the apply failed because the modified
information was too large to fit in this VTP
Server's non-volatile storage location;

localNVStoreFail - the apply failed in trying to store
the new information in a local non-volatile
storage location;

remoteNVStoreFail - the apply failed in trying to store
the new information in a remote non-volatile
storage location;

editBufferEmpty - the apply failed because the Edit
Buffer was empty (for this management domain).

someOtherError - the apply failed for some other reason
(e.g., insufficient memory)."
 ::= { vtpEditControlEntry 2 }

```

1.3.6.1.4.1.9.9.68.1.2.2.1.2 (CISCO-VLAN-MEMBERSHIP-MIB)

```

vmVlan OBJECT-TYPE
    SYNTAX      INTEGER(0..4095)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The VLAN id of the VLAN the port is assigned to
        when vmVlanType is set to static or dynamic.
        This object is not instantiated if not applicable.

        The value may be 0 if the port is not assigned
        to a VLAN.

        If vmVlanType is static, the port is always
        assigned to a VLAN and the object may not be
        set to 0.

        If vmVlanType is dynamic the object's value is
        0 if the port is currently not assigned to a VLAN.
        In addition, the object may be set to 0 only."
 ::= { vmMembershipEntry 2 }

```

[Ajoutez un VLAN à un commutateur Cisco Catalyst avec le SNMP](#)

[Instructions pas à pas](#)

Dans l'exemple présenté ci-dessous, le VLAN 11 est ajouté au commutateur :

1. Afin de vérifier quels VLAN sont actuellement configurés sur le commutateur, émettez un **snmpwalk** sur le **vtpVlanState** OID :**Note**: Le dernier nombre dans l'OID est le nombre VLAN.

```
snmpwalk -c public crumpy vtpVlanState
cisco.ciscoMgmt.ciscoVtpMIB.vtpMIBObjects.vlanInfo.vtpVlanTable.vtpVlanEntry.vtpVlanState.1
.1 : INTEGER: operational
cisco.ciscoMgmt.ciscoVtpMIB.vtpMIBObjects.vlanInfo.vtpVlanTable.vtpVlanEntry.vtpVlanState.1
.48 : INTEGER: operational
cisco.ciscoMgmt.ciscoVtpMIB.vtpMIBObjects.vlanInfo.vtpVlanTable.vtpVlanEntry.vtpVlanState.1
.1002 : INTEGER: operational
```

2. Vérifiez si l'édition est en service par une autre station ou un périphérique NMS. L'édition est non utilisable si vous voyez ce message : aucun objets MIB contenus sous le sous-arbre :

```
snmpwalk -c public crumpy vtpVlanEditTable
no MIB objects contained under subtree.
```

3. L'édition est non utilisable, ainsi il est sûr de commencer à éditer. Placez le **vtpVlanEditOperation** à l'état de copie (entier 2). Ceci te permet pour créer le VLAN.

```
snmpset -c private crumpy vtpVlanEditOperation.1 integer 2
cisco.ciscoMgmt.ciscoVtpMIB.vtpMIBObjects.vlanEdit.vtpEditControlTable.vtpEditControlEntry.
vtpVlanEditOperation.1 : INTEGER: copy
```

4. Afin de rendre le propriétaire en cours de l'autorisation d'éditer visible, vous pouvez placer le propriétaire quand vous émettez la commande, **vtpVlanEditBufferOwner**.

```
snmpset -c private crumpy vtpVlanEditBufferOwner.1 octetstring "Gerald"
cisco.ciscoMgmt.ciscoVtpMIB.vtpMIBObjects.vlanEdit.vtpEditControlTable.vtpEditControlEntry.
vtpVlanEditBufferOwner.1 : OCTET STRING- (ascii): Gerald
```

5. Cet exemple affiche comment vérifier que la table existe :

```
snmpwalk -c public crumpy vtpVlanEditTable
vtpVlanEditState.1.1 : INTEGER: operational
vtpVlanEditState.1.2 : INTEGER: operational
vtpVlanEditState.1.3 : INTEGER: operational
..
```

6. Cet exemple est VLAN 11 et t'affiche comment créer une ligne et placer le type et le nom :

```
snmpset -c private crumpy vtpVlanEditRowStatus.1.11 integer 4
cisco.ciscoMgmt.ciscoVtpMIB.vtpMIBObjects.vlanEdit.vtpVlanEditTable.vtpVlanEditEntry.vtpVla
nEditRowStatus.1.11 : INTEGER: createAndGo
```

```
snmpset -c private crumpy vtpVlanEditType.1.11 integer 1
cisco.ciscoMgmt.ciscoVtpMIB.vtpMIBObjects.vlanEdit.vtpVlanEditTable.vtpVlanEditEntry.vtpVla
nEditType.1.11 : INTEGER: ethernet
```

```
snmpset -c private crumpy vtpVlanEditName.1.11 octetstring "test_11_gerald"
cisco.ciscoMgmt.ciscoVtpMIB.vtpMIBObjects.vlanEdit.vtpVlanEditTable.vtpVlanEditEntry.vtpVla
nEditName.1.11 : DISPLAY STRING- (ascii): test_11_gerald
```

7. Placez le **vtpVlanEditDot10Said**. C'est le nombre VLAN + 100000 traduits à l'hexadécimal. Cet exemple crée le VLAN 11, ainsi le **vtpVlanEditDot10Said** devrait être : 11 + 100000 =

Instructions détaillée une

L'une étape de processus utilise les nombres OID au lieu des noms OID comme le processus pas à pas précédent. Voyez les [détails MIB](#) pour la traduction. Cet exemple crée VLAN 6 :

```
snmpset -c private crumpy 1.3.6.1.4.1.9.9.46.1.4.1.1.1.1 integer 2
1.3.6.1.4.1.9.9.46.1.4.1.1.3.1 octetstring "gcober"
```

```
snmpset -c private gooroo 1.3.6.1.4.1.9.9.46.1.4.2.1.11.1.6 integer 4
1.3.6.1.4.1.9.9.46.1.4.2.1.3.1.6 integer 1 1.3.6.1.4.1.9.9.46.1.4.2.1.4.1.6 octetstring "vlan6"
1.3.6.1.4.1.9.9.46.1.4.2.1.6.1.6 octetstringhex 000186A6 1.3.6.1.4.1.9.9.46.1.4.1.1.1.1 integer
3
```

```
snmpset -c private gooroo 1.3.6.1.4.1.9.9.46.1.4.1.1.1.1 integer 4
```

```
snmpwalk -c public crumpy 1.3.6.1.4.1.9.9.46.1.3.1.1.2
cisco.ciscoMgmt.ciscoVtpMIB.vtpMIBObjects.vlanInfo.vtpVlanTable.vtpVlanEntry.vtpVlanState.1.1 :
INTEGER: operational
cisco.ciscoMgmt.ciscoVtpMIB.vtpMIBObjects.vlanInfo.vtpVlanTable.vtpVlanEntry.vtpVlanState.1.6 :
INTEGER: operational
cisco.ciscoMgmt.ciscoVtpMIB.vtpMIBObjects.vlanInfo.vtpVlanTable.vtpVlanEntry.vtpVlanState.1.11 :
INTEGER: operational
```

Note: Certaines versions SNMP exigent de vous d'utiliser a (.) avant l'OID dans les commandes de SNMP SET.

Supprimez un VLAN d'un commutateur Cisco Catalyst avec le SNMP

Instructions pas à pas

Dans cet exemple VLAN 48 est supprimé du commutateur. Référez-vous à l'[ajouter un VLAN à Cisco Catalyst avec le SNMP](#) pour en savoir plus [SNMP](#). La différence entre cette section où vous supprimez un VLAN et celui où vous ajoutez un VLAN est que vous utilisez la **destruction** au lieu de la commande de **CreateAndGo** pour le **vtpVlanEditRowStatus** :

1. Émettez la commande de supprimer VLAN 48 :

```
snmpset -c private crumpy vtpVlanEditOperation.1 integer 2
cisco.ciscoMgmt.ciscoVtpMIB.vtpMIBObjects.vlanEdit.vtpEditControlTable.vtpEditControlEntry.
vtpVlanEditOperation.1 : INTEGER: copy
snmpset -c private crumpy vtpVlanEditRowStatus.1.48 integer 6
cisco.ciscoMgmt.ciscoVtpMIB.vtpMIBObjects.vlanEdit.vtpVlanEditTable.vtpVlanEditEntry.vtpVla
nEditRowStatus.1.48 : INTEGER: destroy
```

2. Pour vérifier que VLAN 48 a été supprimé, utilisez le **vtpVlanState** ou affichez le VLAN sur le CLI :

```
snmpwalk -c public crumpy vtpVlanState
cisco.ciscoMgmt.ciscoVtpMIB.vtpMIBObjects.vlanInfo.vtpVlanTable.vtpVlanEntry.vtpVlanState.1
.1 : INTEGER: operational
cisco.ciscoMgmt.ciscoVtpMIB.vtpMIBObjects.vlanInfo.vtpVlanTable.vtpVlanEntry.vtpVlanState.1
```



```
.1002 : INTEGER: operational
...
```

[Ajoutez un port à un VLAN sur un commutateur Cisco Catalyst avec le SNMP](#)

Cet exemple affiche comment ajouter un port Fast Ethernet 0/5 à VLAN 48.

1. Pour vérifier quel ifIndex Eth rapide 0/5 a, émettez un **snmpwalk d'ifDescr** :

```
snmpwalk -c public crumpy ifDescr
...
interfaces.ifTable.ifEntry.ifDescr.6 : DISPLAY STRING- (ascii): FastEthernet0/5
...
```

2. Puisque vous savez que le port Eth rapide 0/5 a un ifIndex de 6, ajoutez le port à VLAN 48 :

```
snmpset -c private crumpy vmVlan.6 integer 48
cisco.ciscoMgmt.ciscoVlanMembershipMIB.ciscoVlanMembershipMIBObjects.vmMembership.vmMembers
hipTable.vmMembershipEntry.vmVlan.6 : INTEGER: 48
```

3. Vérifiez que le port a été correctement ajouté en questionnant le même OID de nouveau.

```
snmpget -c public crumpy vmVlan.6
cisco.ciscoMgmt.ciscoVlanMembershipMIB.ciscoVlanMembershipMIBObjects.vmMembership.vmMembers
hipTable.vmMembershipEntry.vmVlan.6 : INTEGER: 48
```

Vous pouvez également vérifier ceci sur le commutateur :

```
crumpy#sh vlan
```

VLAN Name	Status	Ports	
1	default	active	Fa0/1, Fa0/2, Fa0/3, Fa0/4, Fa0/6, Fa0/7, Fa0/8, Fa0/9, Fa0/10, Fa0/11, Fa0/12, Fa0/13, Fa0/14, Fa0/15, Fa0/16, Fa0/17, Fa0/18, Fa0/19, Fa0/20, Fa0/21, Fa0/22, Fa0/23, Fa0/24, Gi0/1, Gi0/2
48	VLAN0048	active	Fa0/5

[Comment changer un port d'un VLAN à un autre VLAN](#)

Cet exemple explique comment le port Eth rapide 0/3 appartient à VLAN 48 et comment le déplacer à VLAN 1 (VLAN par défaut) :

1. Pour vérifier quel ifIndex Eth rapide 0/3 a, émettez un **snmpwalk d'ifDescr** :

```
snmpwalk -c public crumpy ifDescr
...
interfaces.ifTable.ifEntry.ifDescr.4 : DISPLAY STRING- (ascii): FastEthernet0/3
...
```

2. Puisque vous savez que le port Eth rapide 0/3 a un ifIndex de 4, vous pouvez vérifier à quel

VLAN le port appartient actuellement :

```
snmpget -c public crumpy vmVlan.4
cisco.ciscoMgmt.ciscoVlanMembershipMIB.ciscoVlanMembershipMIBObjects.vmMembership.vmMembers
hipTable.vmMembershipEntry.vmVlan.4 : INTEGER: 48
```

3. Le port appartient à VLAN 48.

```
snmpset -c private crumpy vmVlan.4 integer 1
cisco.ciscoMgmt.ciscoVlanMembershipMIB.ciscoVlanMembershipMIBObjects.vmMembership.vmMembers
hipTable.vmMembershipEntry.vmVlan.4 : INTEGER: 1
```

4. Pour déplacer le port de VLAN 48 au VLAN 1, émettez un **snmpset de vmVlan**.

5. Pour vérifier si le port était changé à l'autre VLAN, questionnez **vmVlan** de nouveau :

```
snmpget -c public crumpy vmVlan.4
cisco.ciscoMgmt.ciscoVlanMembershipMIB.ciscoVlanMembershipMIBObjects.vmMembership.vmMembers
hipTable.vmMembershipEntry.vmVlan.4 : INTEGER: 1
```

Vous pouvez également vérifier ceci sur le commutateur lui-même : Avant la modification :

```
crumpy#sh vlan
VLAN Name                               Status   Ports
-----
1    default                               active  Fa0/1, Fa0/2, Fa0/4, Fa0/5,
                                           Fa0/6, Fa0/7, Fa0/8, Fa0/9,
                                           Fa0/10, Fa0/11, Fa0/12, Fa0/13,
                                           Fa0/14, Fa0/15, Fa0/16, Fa0/17,
                                           Fa0/18, Fa0/19, Fa0/20, Fa0/21,
                                           Fa0/22, Fa0/23, Fa0/24, Gi0/1,
                                           Gi0/2
48   VLAN0048                               active  Fa0/3
```

Après la modification :

```
crumpy#sh vlan
VLAN Name                               Status   Ports
-----
1    default                               active  Fa0/1, Fa0/2, Fa0/3, Fa0/4,
                                           Fa0/5, Fa0/6, Fa0/7, Fa0/8,
                                           Fa0/9, Fa0/10, Fa0/11, Fa0/12,
                                           Fa0/13, Fa0/14, Fa0/15, Fa0/16,
                                           Fa0/17, Fa0/18, Fa0/19, Fa0/20,
                                           Fa0/21, Fa0/22, Fa0/23, Fa0/24,
                                           Gi0/1, Gi0/2
48   VLAN0048                               active
```

Note: Vous pouvez apporter d'autres modifications, telles que le nom VLAN, le propriétaire, et beaucoup plus. Référez-vous au MIB entier pour plus de détails sur l'OID.

[Informations connexes](#)

- [Support technique - Cisco Systems](#)