

SNMP を使用する Catalyst での VLAN の追加、修正、削除の方法

目次

[概要](#)

[前提条件](#)

[要件](#)

[コンポーネント](#)

[表記法](#)

[背景説明](#)

[オブジェクト識別子 \(OID\) を含む、MIB 変数の詳細](#)

[SNMP を使用する Cisco Catalyst スイッチへの VLAN の追加](#)

[手順説明](#)

[SNMP を使用する Cisco Catalyst スイッチへの VLAN の追加](#)

[ワン ステップの説明](#)

[SNMP を使用する Cisco Catalyst スイッチからの VLAN の削除](#)

[手順説明](#)

[SNMP を使用する Cisco Catalyst スイッチの VLAN へのポートの追加](#)

[ポートをある VLAN から他の VLAN に変更する方法](#)

[関連情報](#)

概要

このドキュメントでは、Simple Network Management Protocol (SNMP; 簡易ネットワーク管理プロトコル) を使用している Cisco Catalyst スイッチで、VLAN を作成および削除する方法について説明しています。また、SNMP を使用する VLAN にポートを追加する方法についても説明しています。

前提条件

要件

このドキュメントの情報を使用する前に、次のことを理解している必要があります。

- ifTable および ifIndexes が動作する仕組み
- Cisco Catalyst スイッチで VLAN が動作する仕組み
- Cisco Catalyst スイッチで VLAN 情報を表示する方法
- SNMP の `get`、`set`、および `walk` コマンドの一般的な使用方法

コンポーネント

このドキュメントの対象となるのは、標準の Catalyst OS または IF-MIB、CISCO-VTP-MIB、および CISCO-VLAN-MEMBERSHIP-MIB をサポートする Catalyst IOS が稼働している Catalyst スイッチです。このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づくものです。

- CatIOS 12.0(5)WC5a が稼働する Catalyst 3524XL
- NET-SNMP バージョン 5.0.6 は、<http://www.net-snmp.org/> で入手できます。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されたものです。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。対象のネットワークが実稼働中である場合には、どのような作業についても、その潜在的な影響について確実に理解しておく必要があります。

表記法

ドキュメント表記の詳細は、『[シスコテクニカルティップスの表記法](#)』を参照してください。

背景説明

オブジェクト識別子 (OID) を含む、MIB 変数の詳細

```
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 'apply'ing
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 instantiate 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 }

[SNMP を使用する Cisco Catalyst スイッチへの VLAN の追加](#)

[手順説明](#)

次に示す例では、VLAN 11 をスイッチに追加しています。

1. 現在スイッチに設定されている VLAN を確認するには、vtpVlanState の OID で snmpwalk コマンドを発行します。注: OID 内の最後の番号は VLAN 番号です。

```
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. そのエディションが他の NMS ステーションまたはデバイスで使用されているかどうかを確認します。次のメッセージが表示された場合は、そのエディションは使用されていません。

no MIB objects contained under subtree:

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

3. そのエディションが使用されていなければ、編集を始めても大丈夫です。

vtpVlanEditOperation を設定して、コピー状態 (integer 2) にします。これで、VLAN を作成できるようになります。

```
snmpset -c private crumpy vtpVlanEditOperation.1 integer 2
```

```
cisco.ciscoMgmt.ciscoVtpMIB.vtpMIBObjects.vlanEdit.vtpEditControlTable.vtpEditControlEntry.vtpVlanEditOperation.1 : INTEGER: copy
```

- 現在のオーナーの編集権限を参照可能にするために、`vtpVlanEditBufferOwner` コマンドを発行する際にオーナーを設定できます。

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

- 次の例では、テーブルの存在を確認する方法を示しています。

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

- 次の例は VLAN 11 で、行を作成し、タイプと名前を設定する方法を示しています。

```
snmpset -c private crumpy vtpVlanEditRowStatus.1.11 integer 4
cisco.ciscoMgmt.ciscoVtpMIB.vtpMIBObjects.vlanEdit.vtpVlanEditTable.vtpVlanEditEntry.vtpVlanEditRowStatus.1.11 : INTEGER: createAndGo snmpset -c private crumpy vtpVlanEditType.1.11 integer 1
cisco.ciscoMgmt.ciscoVtpMIB.vtpMIBObjects.vlanEdit.vtpVlanEditTable.vtpVlanEditEntry.vtpVlanEditType.1.11 : INTEGER: ethernet snmpset -c private crumpy vtpVlanEditName.1.11 octetstring "test_11_gerald"
cisco.ciscoMgmt.ciscoVtpMIB.vtpMIBObjects.vlanEdit.vtpVlanEditTable.vtpVlanEditEntry.vtpVlanEditName.1.11 : DISPLAY STRING- (ascii): test_11_gerald
```

- `vtpVlanEditDot10Said` を設定します。これは VLAN 番号 + 100000 が 16 進数に変換されたものです。この例では VLAN 11 を作成しているため、`vtpVlanEditDot10Said` は「11 + 100000 = 100011」、16 進数で「000186AB」になります。

```
snmpset -c private crumpy vtpVlanEditDot10Said.1.11 octetstringhex 000186AB
cisco.ciscoMgmt.ciscoVtpMIB.vtpMIBObjects.vlanEdit.vtpVlanEditTable.vtpVlanEditEntry.vtpVlanEditDot10Said.1.11 : OCTET STRING- (hex): length = 4 0: 00 01 86 ab -- -- -- --
-----
```

- VLAN 11 を作成したら、修正を加える必要があります。OID の `vtpVlanEditOperation` を再度使用します。今回は「Apply」を使用して設定を確認します。

```
snmpset -c private crumpy vtpVlanEditOperation.1 integer 3
cisco.ciscoMgmt.ciscoVtpMIB.vtpMIBObjects.vlanEdit.vtpEditControlTable.vtpEditControlEntry.vtpVlanEditOperation.1 : INTEGER: apply
```

- VLAN が正しく作成されたことを確認します。OID の `vtpVlanApplyStatus` を使用します。ステータスが「succeeded」になるまでプロセスを確認します。

```
snmpget -c public crumpy vtpVlanApplyStatus.1 vtpVlanApplyStatus.1 : INTEGER: inProgress
snmpget -c public crumpy vtpVlanApplyStatus.1 vtpVlanApplyStatus.1 : INTEGER: inProgress
snmpget -c public crumpy vtpVlanApplyStatus.1 vtpVlanApplyStatus.1 : INTEGER: succeeded
```

- 最後の処理として、修正を確定し、権限を解除して、他のユーザがそれぞれの NMS から VLAN を追加、修正、削除できるようにします。

```
snmpset -c private crumpy vtpVlanEditOperation.1 integer 4 vtpVlanEditOperation.1 :
INTEGER: release
```

- バッファが空であることを確認します。

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

- CLI コマンドの `show vlan` か、`snmpwalk` を使用して、スイッチに VLAN 11 が作成されたことを確認します。

```
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.1.11 : 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 ...
```

[SNMP を使用する Cisco Catalyst スイッチへの VLAN の追加](#)

ワンステップの説明

ワンステッププロセスでは、上記の手順ごとの説明での OID 名ではなく、OID 番号を使用します。変換については、「[MIB 変数の詳細](#)」を参照してください。次の例では VLAN 6 を作成しています。

```
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
```

注: 一部の SNMP バージョンでは、SNMP SET コマンドで OID の前に (.) を使用する必要があります。

SNMP を使用する Cisco Catalyst スイッチからの VLAN の削除

手順説明

この例では、スイッチから VLAN 48 を削除しています。詳細は、「[SNMP による Cisco Catalyst スイッチへの VLAN の追加](#)」を参照してください。VLAN を削除するこのセクションと、VLAN を追加するセクションとの違いは、vtpVlanEditRowStatus に対して CreateAndGo コマンドではなく destroy を使用することです。

1. VLAN 48 を削除するコマンドを発行します。

```
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. VLAN 48 が削除されたことを確認するには、CLI で vtpVlanState または show vlan を使
します。

```
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 ...
```

SNMP を使用する Cisco Catalyst スイッチの VLAN へのポート の追加

次の例では、ファストイーサネットのポート 0/5 を VLAN 48 に追加する方法について説明して
います。

1. Fast Eth 0/5 の ifIndex を確認するために、ifDescr を指定した snmpwalk を発行します。


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

2. ポート Fast Eth 0/5 が持つ ifIndex が 6 であることがわかったため、ポートを VLAN 48 に追加します。

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

3. 同じ OID についてクエリーを再度発行して、ポートが正しく追加されたことを確認します。

```
snmpget -c public crumpy vmVlan.6
cisco.ciscoMgmt.ciscoVlanMembershipMIB.ciscoVlanMembershipMIBObjects.vmMembership.vmMembers
hipTable.vmMembershipEntry.vmVlan.6 : INTEGER: 48 これはスイッチでも確認できます。
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
```

ポートをある VLAN から他の VLAN に変更する方法

この例では、ポート Fast Eth 0/3 が VLAN 48 に属する仕組みと、それを VLAN 1 (デフォルトの VLAN) に移動する方法について説明しています。

1. Fast Eth 0/3 が持つ ifIndex を確認するために、ifDescr を指定した snmpwalk を発行します。

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

2. ポート Fast Eth 0/3 が持つ ifIndex が 4 であることがわかったため、ポートが現在属している VLAN を確認できます。

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

3. このポートは VLAN 48 に属しています。

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

4. このポートを VLAN 48 から VLAN 1 に移動するには、vmVlan を指定した snmpset を発行します。

5. ポートが他の VLAN に変更されたことを確認するために、vmVlan のクエリーを再度発行します。

```
snmpget -c public crumpy vmVlan.4
cisco.ciscoMgmt.ciscoVlanMembershipMIB.ciscoVlanMembershipMIBObjects.vmMembership.vmMembers
hipTable.vmMembershipEntry.vmVlan.4 : INTEGER: 1 これはスイッチでも確認できます。変更前
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 変更
後
```

```
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 注
```

: VLAN 名、オーナーなど、他の変更も行うことができます。OID の詳細については、MIB 全体を参照してください。

関連情報

- [テクニカルサポート - Cisco Systems](#)