

使用SNMP，如何添加，修改和去除在Catalyst的VLANs

目录

[简介](#)

[先决条件](#)

[要求](#)

[组件](#)

[规则](#)

[背景](#)

[MIB变量的详细信息 — 包括对象标识符\(OID\)](#)

[使用SNMP将VLAN添加到Cisco Catalyst交换机](#)

[逐步指导](#)

[使用SNMP将VLAN添加到Cisco Catalyst交换机](#)

[一步说明](#)

[使用SNMP从Cisco Catalyst交换机中删除VLAN](#)

[逐步指导](#)

[使用SNMP将端口添加到Cisco Catalyst交换机上的VLAN](#)

[如何将端口从一个VLAN更改为另一个VLAN](#)

[相关信息](#)

简介

本文档介绍如何在使用简单网络管理协议 (SNMP) 的 Cisco Catalyst 交换机上创建和删除 VLAN。它还介绍如何使用 SNMP 将端口添加到 VLAN。

先决条件

要求

在使用本文档中的信息之前，请确保您了解：

- ifTable和ifIndexes如何工作
- VLAN在Cisco Catalyst交换机上的工作方式
- 如何查看Cisco Catalyst交换机上的VLAN信息
- SNMP get、set和walk命令的一般使用

组件

本文档适用于运行常规Catalyst OS或Catalyst IOS的Catalyst交换机，这些交换机支持IF-MIB、

CISCO-VTP-MIB和CISCO-VLAN-MEMBERSHIP-MIB。本文档中的信息基于以下软件和硬件版本：

- 运行CatIOS 12.0(5)WC5a的Catalyst 3524XL
- 在 <http://www.net-snmp.org/> 可上获取 NET-SNMP 版本 5.0.6

本文档中的信息都是基于特定实验室环境中的设备创建的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您在实时网络中工作，则在使用任何命令之前，请确保您了解任何命令的潜在影响。

规则

有关文档规则的详细信息，请参阅 [Cisco 技术提示规则](#)。

背景

MIB变量的详细信息 — 包括对象标识符(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 instanciate 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


```

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 }

```

[使用SNMP将VLAN添加到Cisco Catalyst交换机](#)

[逐步指导](#)

在以下示例中，VLAN 11已添加到交换机：

1. 要检查交换机上当前配置了哪些VLAN，请在vtpVlanState OID上发出snmpwalk:注意：OID中的最后一个编号是VLAN编号。

```

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. 验证版本是否正由其他NMS工作站或设备使用。如果您看到以下消息，则此版本未使用：MIB:

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

3. 此版本未使用，因此开始编辑是安全的。将vtpVlanEditOperation设置为复制状态（整数2）。这样您就可以创建VLAN。

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

4. 要使编辑权限的当前所有者可见，可以在发出命令vtpVlanEditBufferOwner时设置所有者。

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

5. 此示例显示如何验证表是否存在：

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

6. 本示例是VLAN 11，它显示了如何创建行并设置类型和名称：

```
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
```

7. 设置vtpVlanEditDot10Said。这是VLAN编号+ 100000转换为十六进制。本示例创建VLAN 11，因此vtpVlanEditDot10Said应为： $11 + 100000 = 100011$ ->十六进制：000186AB

```
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 -- -- -- -- -- -- -- -- -- -- .....
```

8. 创建VLAN 11后，必须应用修改。再次使用vtpVlanEditOperation OID。此时，请使用Apply确认设置：

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

9. 检验VLAN是否已成功创建。使用OID vtpVlanApplyStatus。检查该过程，直到状态为：：

```
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
```

10. 最后一项操作是提交修改并释放权限，以便其他用户可以从其NMS添加、修改或删除VLAN。

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

11. 验证缓冲区为空：

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

12. 使用CLI命令show vlan或snmpwalk验证是否在交换机上创建了VLAN 11。：

```
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
...
```

[使用SNMP将VLAN添加到Cisco Catalyst交换机](#)

[一步说明](#)

一步过程使用OID编号，而不是像之前的分步过程那样使用OID名称。请参阅MIB[详细信息](#)以进行转换。本示例创建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
```

注意：某些SNMP版本要求您在SNMP SET命令中的OID之前使用(.)。

如何将端口从一个VLAN更改为另一个VLAN

本示例演示了端口Fast Eth 0/3如何属于VLAN 48以及如何将其移至VLAN 1 (默认VLAN) :

1. 要验证ifIndex Fast Eth 0/3具有哪个，请发出ifDescr的snmpwalk:

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

2. 由于您知道端口Fast Eth 0/3的ifIndex为4，因此您可以验证端口当前属于哪个VLAN:

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

3. 端口属于VLAN 48。

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

4. 要将端口从VLAN 48移到VLAN 1，请发出vmVlan snmpset。

5. 要验证端口是否更改为其他VLAN，请再次查询vmVlan:

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

您也可以在交换机本身上验证这一点：更改前：

```
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
```

更改后：

```
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
```

注意：您可以进行其他更改，如VLAN名称、所有者等。有关OID的详细信息，请参阅整个MIB。

相关信息

- [技术支持 - Cisco Systems](#)