

Como adicionar, altere, e remova VLAN em um catalizador usando o SNMP

Índice

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes](#)

[Convenções](#)

[Background](#)

[Detalhes dos variáveis MIB — Incluindo os identificadores de objeto \(OID\)](#)

[Adicionar um VLAN a um interruptor do Cisco catalyst com SNMP](#)

[Instruções passo a passo](#)

[Adicionar um VLAN a um interruptor do Cisco catalyst com SNMP](#)

[Instruções de uma etapa](#)

[Suprima de um VLAN de um interruptor do Cisco catalyst com SNMP](#)

[Instruções passo a passo](#)

[Adicionar uma porta a um VLAN em um interruptor do Cisco catalyst com SNMP](#)

[Como mudar uma porta de um VLAN a um outro VLAN](#)

[Informações Relacionadas](#)

[Introdução](#)

Este documento descreve como criar e excluir VLANs em um switch Cisco Catalyst que usa o Simple Network Management Protocol (SNMP). Ele também descreve como adicionar portas a uma VLAN com SNMP.

[Pré-requisitos](#)

[Requisitos](#)

Antes que você use a informação neste documento, assegure-se de que você compreenda:

- Como o ifTable e os ifIndex trabalham
- Como os VLAN trabalham no Cisco catalyst comuta
- Como ver a informação de VLAN no Switches dos Cisco catalyst
- O uso geral de **comandos get, set, e walk** SNMP

[Componentes](#)

Este documento é para Catalyst Switches que executa o OS regular ou o Catalyst IOS do catalizador que apoiam o IF-MIB, o CISCO-VTP-MIB e o CISCO-VLAN-MEMBERSHIP-MIB. As informações neste documento são baseadas nestas versões de software e hardware:

- Catalyst 3524XL executando CatIOS 12.0(5)WC5a
- [A versão NET-SNMP 5.0.6 está disponível em http://www.net-snmp.org/](http://www.net-snmp.org/)

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se você está trabalhando em uma rede viva, antes que você use o comando any certifique-se de que você compreende o impacto potencial do comando any.

Convenções

Para obter mais informações sobre convenções de documento, consulte as [Convenções de dicas técnicas Cisco](#).

Background

Detalhes dos variáveis MIB — Incluindo os identificadores de objeto (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 '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 }

```

[Adicionar um VLAN a um interruptor do Cisco catalyst com SNMP](#)

[Instruções passo a passo](#)

No exemplo mostrado abaixo, o VLAN 11 é adicionado ao interruptor:

1. A fim verificar que VLAN são configurados atualmente no interruptor, emita um **snmpwalk** no **vtpVlanState** OID:**Note:** O último número no OID é o número de 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. Verifique se a edição está no uso por uma outra estação NMS ou dispositivo. A edição não é dentro uso se você vê esta mensagem: nenhuns objetos MIB contidos sob o subtree:

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

3. A edição não é dentro uso, assim que é seguro começar editar. Ajuste o **vtpVlanEditOperation** ao estado da cópia (inteiro 2). Isto permite que você crie o VLAN.

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

4. A fim fazer o proprietário atual da permissão da edição visível, você pode ajustar o proprietário quando você emite o comando, **vtpVlanEditBufferOwner**.

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

5. Este exemplo mostra como verificar que a tabela existe:

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

6. Este exemplo é VLAN 11 e mostra-lhe como criar uma fileira e ajustar o tipo e o nome:

```
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. Ajuste o **vtpVlanEditDot10Said**. Este é o número de VLAN + 100000 traduzidos ao hexadecimal. Este exemplo cria o VLAN 11, assim que o **vtpVlanEditDot10Said** deve ser: 11 + 100000 = 100011 - > encantar: 000186AB

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

8. Quando você tem o VLAN criado 11, você deve aplicar as alterações. Use o **vtpVlanEditOperation** OID outra vez. Este uso do tempo a **aplicação** confirmar os ajustes:

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

9. Verifique que o VLAN esteve criado com sucesso. Use o OID **vtpVlanApplyStatus**. Verifique o processo até que o estado leia: sucedido:

```
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. A última ação é comprometer as alterações e liberar as permissões de modo que outros usuários possam adicionar, alterar, ou suprimir de VLAN de seu NMS.

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

11. Verifique que o buffer está vazio:

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

12. Verifique que o VLAN 11 esteve criado no interruptor com a **mostra do comando CLI vlan** ou com um **snmpwalk**:

```
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
...
```

[Adicionar um VLAN a um interruptor do Cisco catalyst com SNMP](#)

[Instruções de uma etapa](#)

O um processo da etapa usa os números OID em vez dos nomes OID como o processo passo a passo precedente. Veja os [detalhes MIB de tradução](#). Este exemplo cria VLAN 6:

```
snmpset -c private crumphy 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 crumphy 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: Determinadas versões de SNMP exigem-no usar a (.) antes do OID nos comandos SNMP SET.

[Suprima de um VLAN de um interruptor do Cisco catalyst com SNMP](#)

[Instruções passo a passo](#)

Neste exemplo o VLAN 48 é suprimido do interruptor. Refira [adicionar a um VLAN a um Cisco catalyst com o SNMP](#) para mais informação. A diferença entre esta seção onde você suprime de um VLAN e de esse onde você adiciona um VLAN é que você usa o **destruição** em vez do comando **CreateAndGo** para o **vtpVlanEditRowStatus**:

1. Emita o comando suprimir de VLAN 48:

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

2. Para verificar que o VLAN 48 esteve suprimido, use o **vtpVlanState** ou **mostre vlan** no CLI:

```
snmpwalk -c public crumphy 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
...
```

Adicionar uma porta a um VLAN em um interruptor do Cisco catalyst com SNMP

Este exemplo mostra como adicionar um Fast Ethernet 0/5 da porta a VLAN 48.

1. Para verificar que ifIndex o Eth rápido 0/5 tem, emita um **snmpwalk do ifDescr**:

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

2. Desde que você sabe que o Eth rápido 0/5 da porta tem um ifIndex de 6, adicionar a porta a VLAN 48:

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

3. Verifique que a porta esteve adicionada corretamente perguntando o mesmo OID outra vez.

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

Você pode igualmente verificar este no interruptor:

```
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

Como mudar uma porta de um VLAN a um outro VLAN

Este exemplo demonstra como o Eth rápido 0/3 da porta pertence a VLAN 48 e como o mover para VLAN1 (VLAN padrão):

1. Para verificar que ifIndex o Eth rápido 0/3 tem, emita um **snmpwalk do ifDescr**:

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

2. Desde que você sabe que o Eth rápido 0/3 da porta tem um ifIndex de 4, você pode verificar a que VLAN a porta pertence atualmente:

```
snmpget -c public crumpy vmVlan.4
cisco.ciscoMgmt.ciscoVlanMembershipMIB.ciscoVlanMembershipMIBObjects.vmMembership.vmMembers
```

```
hipTable.vmMembershipEntry.vmVlan.4 : INTEGER: 48
```

3. A porta pertence a VLAN 48.

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

4. Para mover a porta de VLAN 48 para o VLAN1, emita um **snmpset de vmVlan**.

5. Para verificar se a porta foi mudada ao outro VLAN, pergunte **vmVlan** outra vez:

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

Você pode igualmente verificar este no interruptor próprio: Antes da mudança:

```
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
```

Após a mudança:

```
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: Você pode fazer outras mudanças, tais como o nome VLAN, o proprietário, e muito mais. Refira o MIB inteiro para mais detalhes no OID.

[Informações Relacionadas](#)

- [Suporte Técnico - Cisco Systems](#)