

Hinzufügen, Ändern und Entfernen von VLANs in einem Catalyst mithilfe von SNMP

Inhalt

[Einführung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Komponenten](#)

[Konventionen](#)

[Hintergrund](#)

[Details der MIB-Variablen, einschließlich OIDs \(Object Identifiers\)](#)

[Hinzufügen eines VLANs zu einem Cisco Catalyst Switch mit SNMP](#)

[Schrittweise Anleitung](#)

[Hinzufügen eines VLANs zu einem Cisco Catalyst Switch mit SNMP](#)

[Anweisungen in einem Schritt](#)

[Löschen eines VLANs von einem Cisco Catalyst Switch mit SNMP](#)

[Schrittweise Anleitung](#)

[Hinzufügen eines Ports zu einem VLAN auf einem Cisco Catalyst Switch mit SNMP](#)

[Ändern eines Ports von einem VLAN in ein anderes VLAN](#)

[Zugehörige Informationen](#)

[Einführung](#)

In diesem Dokument wird beschrieben, wie Sie VLANs auf einem Cisco Catalyst Switch erstellen und löschen, der das Simple Network Management Protocol (SNMP) verwendet. Außerdem wird beschrieben, wie einem VLAN mit SNMP Ports hinzugefügt werden.

[Voraussetzungen](#)


[Anforderungen](#)

Bevor Sie die Informationen in diesem Dokument verwenden, vergewissern Sie sich, dass Sie Folgendes verstehen:

- Funktionsweise von ifTable und ifIndexes
- Funktionsweise von VLANs auf Cisco Catalyst Switches
- Anzeigen von VLAN-Informationen auf Cisco Catalyst Switches
- Die allgemeine Verwendung von SNMP-Befehlen **Get**, **Set** und **walk**

[Komponenten](#)

Dieses Dokument gilt nur für Catalyst Switches mit regulärem Catalyst OS oder Catalyst IOS, die IF-MIB, CISCO-VTP-MIB und CISCO-VLAN-MEMBERSHIP-MIB unterstützen. Die Informationen in diesem Dokument basieren auf den folgenden Software- und Hardwareversionen:

- Catalyst 3524XL mit CatIOS 12.0(5)WC5a
- NET-SNMP Version 5.0.6 verfügbar unter <http://www.net-snmp.org/> 

Die in diesem Dokument enthaltenen Informationen wurden aus Geräten in einer bestimmten Laborumgebung erstellt. Alle in diesem Dokument verwendeten Geräte haben mit einer leeren (Standard-)Konfiguration begonnen. Wenn Sie in einem Live-Netzwerk arbeiten, vergewissern Sie sich vor der Verwendung eines Befehls, dass Sie die potenziellen Auswirkungen eines Befehls verstehen.

Konventionen

Weitere Informationen zu Dokumentkonventionen finden Sie in den [Cisco Technical Tips Conventions](#).

Hintergrund

Details der MIB-Variablen, einschließlich OIDs (Object Identifiers)

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

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 }

```

[Hinzufügen eines VLANs zu einem Cisco Catalyst Switch mit SNMP](#)

[Schrittweise Anleitung](#)

Im Beispiel unten wird VLAN 11 zum Switch hinzugefügt:

1. Um zu überprüfen, welche VLANs derzeit auf dem Switch konfiguriert sind, führen Sie einen **snmpwalk** auf der **vtpVlanState**-OID aus:**Hinweis:** Die letzte Nummer in der OID ist die

VLAN-Nummer.

```
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
```

- Überprüfen Sie, ob die Edition von einer anderen NMS-Station oder einem anderen Gerät verwendet wird. Die Edition wird nicht verwendet, wenn die folgende Meldung angezeigt

wird: Keine MIB-Objekte in Unterstruktur enthalten:

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

- Die Edition ist nicht in Gebrauch, daher ist es sicher zu beginnen, zu bearbeiten. Legen Sie **vtpVlanEditOperation** auf den Kopierzustand (Ganzzahl 2) fest. Dadurch können Sie das VLAN erstellen.

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

- Um den aktuellen Besitzer der Bearbeitungsberechtigung sichtbar zu machen, können Sie beim Ausgeben des Befehls **vtpVlanEditBufferOwner** den Besitzer festlegen.

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

- In diesem Beispiel wird veranschaulicht, wie überprüft wird, ob die Tabelle vorhanden ist:

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

- In diesem Beispiel wird VLAN 11 veranschaulicht, wie Sie eine Zeile erstellen und Typ und Namen festlegen:

```
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
```

- Legen Sie **vtpVlanEditDot10Said** fest. Dies ist die VLAN-Nummer + 100000, die in hexadezimal übersetzt wurde. In diesem Beispiel wird VLAN 11 erstellt. Das **vtpVlanEditDot10Said** sollte daher wie folgt lauten: $11 + 100000 = 100011 \rightarrow$ Hex: 000186AB

```
snmpset -c private crumpy vtpVlanEditDot10Said.1.11 octetstringhex 000186AB  
cisco.ciscoMgmt.ciscoVtpMIB.vtpMIBObjects.vlanEdit.vtpVlanEditTable.vtpVlanEdi
```

```
ntrypVlanEditDot10Said.1.11 : OCTET STRING- (hex): length = 4
0: 00 01 86 ab -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- .....
```

8. Wenn Sie VLAN 11 erstellt haben, müssen Sie die Änderungen übernehmen. Verwenden Sie die OID **vtpVlanEditOperation** erneut. Verwenden Sie dieses Mal **Apply**, um die Einstellungen zu bestätigen:

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

9. Überprüfen Sie, ob das VLAN erfolgreich erstellt wurde. Verwenden Sie die OID **vtpVlanApplyStatus**. Überprüfen Sie den Vorgang, bis der Status lautet: **erfolgreich**:

```
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. Die letzte Aktion besteht darin, die Änderungen zu bestätigen und die Berechtigungen freizugeben, damit andere Benutzer VLANs aus ihrem NMS hinzufügen, ändern oder löschen können.

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

11. Stellen Sie sicher, dass der Puffer leer ist:

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

12. Vergewissern Sie sich, dass VLAN 11 auf dem Switch mit dem CLI-Befehl **show vlan** oder mit **snmpwalk** erstellt wurde:

```
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
...
```

[Hinzufügen eines VLANs zu einem Cisco Catalyst Switch mit SNMP](#)

[Anweisungen in einem Schritt](#)

Bei einem Schritt werden die OID-Nummern anstelle der OID-Namen wie im vorherigen Schritt verwendet. Die [MIB-Details](#) zur Übersetzung finden Sie hier. In diesem Beispiel wird VLAN 6 erstellt:

```
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
```

Hinweis: Bei einigen SNMP-Versionen muss in den SNMP SET-Befehlen ein (.) vor der OID verwendet werden.

[Löschen eines VLANs von einem Cisco Catalyst Switch mit SNMP](#)

[Schrittweise Anleitung](#)

In diesem Beispiel wird VLAN 48 vom Switch gelöscht. Weitere Informationen finden Sie unter [Hinzufügen eines VLAN zu einem Cisco Catalyst mit SNMP](#). Der Unterschied zwischen diesem Abschnitt, in dem Sie ein VLAN löschen, und dem VLAN, in dem Sie ein VLAN hinzufügen, besteht darin, dass Sie anstelle des **CreateAndGo**-Befehls für den **VtpVlanEditRowStatus** die **Zerstörung** verwenden:

1. Geben Sie den Befehl zum Löschen von VLAN 48 aus:

```
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. Um zu überprüfen, ob VLAN 48 gelöscht wurde, verwenden Sie **vtpVlanState** oder **show vlan** in der 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
...
```

[Hinzufügen eines Ports zu einem VLAN auf einem Cisco Catalyst Switch mit SNMP](#)

Dieses Beispiel zeigt, wie Sie einen Port Fast Ethernet 0/5 zu VLAN 48 hinzufügen.

1. Um zu überprüfen, welche Eigenschaft bei Index Fast Eth 0/5 vorhanden ist, führen Sie einen Snapwalk von **ifDescr** aus:

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

2. Da Sie wissen, dass Port Fast Eth 0/5 einen ifIndex von 6 hat, fügen Sie den Port zu VLAN 48 hinzu:

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

3. Stellen Sie sicher, dass der Port korrekt hinzugefügt wurde, indem Sie erneut dieselbe OID abfragen.

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

Sie können dies auch auf dem Switch überprüfen:

```
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

Ändern eines Ports von einem VLAN in ein anderes VLAN

Dieses Beispiel veranschaulicht, wie Port Fast Eth 0/3 zu VLAN 48 gehört und in VLAN 1 (Standard-VLAN) verschoben wird:

1. Um zu überprüfen, welche Eigenschaft bei Index Fast Eth 0/3 vorhanden ist, führen Sie einen Snapwalk von **ifDescr** aus:

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

2. Da Sie wissen, dass Port Fast Eth 0/3 einen ifIndex von 4 aufweist, können Sie überprüfen, zu welchem VLAN der Port derzeit gehört:

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

3. Der Port gehört zu VLAN 48.

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

```
hipTable.vmMembershipEntry.vmVlan.4 : INTEGER: 1
```

- Um den Port von VLAN 48 zu VLAN 1 zu verschieben, müssen Sie einen **Satz vmVLAN** eingeben.
- Um zu überprüfen, ob der Port in das andere VLAN geändert wurde, stellen Sie **vmVlan** erneut **ab**:

```
snmpget -c public crumpy vmVlan.4
```

```
cisco.ciscoMgmt.ciscoVlanMembershipMIB.ciscoVlanMembershipMIBObjects.vmMembership.vmMembers  
hipTable.vmMembershipEntry.vmVlan.4 : INTEGER: 1
```

Sie können dies auch auf dem Switch selbst überprüfen: Vor der Änderung:

```
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

Nach der Änderung:

```
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	

Hinweis: Sie können weitere Änderungen vornehmen, z. B. den VLAN-Namen, den Besitzer und vieles mehr. Weitere Informationen zu OID finden Sie in der gesamten MIB.

[Zugehörige Informationen](#)

- [Technischer Support - Cisco Systems](#)