Cómo agregar, modifique, y quite los VLA N en un Catalyst usando el SNMP

Contenido

Introducción

Este documento describe cómo crear y eliminar las VLAN en un switch Cisco Catalyst que use Simple Network Management Protocol (SNMP). También describe cómo agregar puertos a una VLAN mediante SNMP.

prerrequisitos

Requisitos

Antes de que usted utilice la información en este documento, asegúrese de que usted entienda:

- Cómo el ifTable y los ifIndexes trabajan
- Cómo los VLA N trabajan en el Switches del Cisco Catalyst
- Cómo ver la información de VLAN en el Switches de los Catalyst de Cisco
- El uso general de los **comandos get, set, y walk** SNMP

Componentes

Este documento está para los switches de Catalyst que funcionan con el Catalyst regular OS o IOS de Catalyst que soporta el IF-MIB, el CISCO-VTP-MIB y el CISCO-VLAN-MEMBERSHIP-MIB. La información que contiene este documento se basa en las siguientes versiones de software y hardware.

- Catalyst 3524XL que ejecuta CatIOS 12.0(5)WC5a
- La versión 5.0.6 de NET-SNMP está disponible en http://www.net-snmp.org/

La información que se presenta en este documento se originó a partir de dispositivos dentro de un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si usted está trabajando en una red en funcionamiento, antes de que usted utilice el comando any aseegurese que usted entiende el impacto potencial del comando any.

Convenciones

Para obtener más información sobre las convenciones del documento, consulte Convenciones de Consejos Técnicos de Cisco.

Antecedente

Detalles de las variables MIB — Incluyendo los 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 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 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 }
```

Agregue un VLA N a un Switch del Cisco Catalyst con el SNMP

Instrucciones Paso a Paso

En el ejemplo mostrado abajo, el VLAN 11 se agrega al Switch:

1. Para marcar qué VLA N se configuran actualmente en el Switch, publique un **snmpwalk** en el **vtpVlanState** OID:**Nota:** El número más reciente del OID es el número 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 si la edición es funcionando por otra estación NMS o dispositivo. La edición es parada si usted ve este mensaje: `ningunos objetos de MIB contenidos bajo la sub-estructura: snmpwalk -c public crumpy vtpVlanEditTable no MIB objects contained under subtree.`
3. La edición es parada, así que es seguro comenzar a editar. Fije el **vtpVlanEditOperation al** estado de la copia (número entero 2). Esto permite que usted cree el VLA N.
   ```
   snmpset -c private crumpy vtpVlanEditOperation.1 integer 2
   cisco.ciscoMgmt.ciscoVtpMIB.vtpMIBObjects.vlanEdit.vtpEditControlTable.vtpEditControlEntry.
   ```

```
vtpVlanEditOperation.1 : INTEGER: copy
```

4. Para hacer al propietario actual del permiso del editar visible, usted puede fijar al propietario cuando usted publica el 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 ejemplo muestra cómo verificar que existe la tabla:
```
snmpwalk -c public crumpy vtpVlanEditTable vtpVlanEditState.1.1 : INTEGER: operational
vtpVlanEditState.1.2 : INTEGER: operational vtpVlanEditState.1.3 : INTEGER: operational ..
```
6. Este ejemplo es VLAN 11 y le muestra cómo crear una fila y fijar el tipo y el nombre:
```
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. Fije el **vtpVlanEditDot10Said**. Éste es el número VLAN + 100000 traducidos al hexadecimal. Este ejemplo crea el VLAN 11, así que el **vtpVlanEditDot10Said** debe ser: 11 + 100000 = 100011 - > hex.: 000186AB
```
snmpset -c private crumpy vtpVlanEditDot10Said.1.11 octetstringhex 000186AB
cisco.ciscoMgmt.ciscoVtpMIB.vtpMIBObjects.vlanEdit.vtpVlanEditTable.vtpVlanEdi
ntry.vtpVlanEditDot10Said.1.11 : OCTET STRING- (hex): length = 4 0: 00 01 86 ab -- -- -- --
-- -- -- -- -- -- -- -- ................
```
8. Cuando usted tiene VLAN creado 11, usted debe aplicar las modificaciones. Utilice el **vtpVlanEditOperation** OID otra vez. Este uso del tiempo la **aplicación** de confirmar las configuraciones:
```
snmpset -c private crumpy vtpVlanEditOperation.1 integer 3
cisco.ciscoMgmt.ciscoVtpMIB.vtpMIBObjects.vlanEdit.vtpEditControlTable.vtpEditControlEntry.
vtpVlanEditOperation.1 : INTEGER: apply
```
9. Verifique que el VLA N fuera creado con éxito. Utilice IOD vtpVlanApplyStatus. Marque el proceso hasta que el estatus lea: `tenido éxito`:
```
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. La acción más reciente es confiar las modificaciones y liberar los permisos de modo que otros usuarios puedan agregar, modificar, o borrar los VLA N de su NMS.
```
snmpset -c private crumpy vtpVlanEditOperation.1 integer 4 vtpVlanEditOperation.1 :
INTEGER: release
```
11. Verifique que el buffer esté vacío:
```
snmpwalk –c public crumpy vtpVlanEditTable no MIB objects contained under subtree.
```
12. Verifique que el VLAN 11 fuera creado en el Switch con la **demostración del** comando CLI **vlan** o con un **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 …
```

Agregue un VLA N a un Switch del Cisco Catalyst con el SNMP

Instrucciones de un paso

El un paso de proceso utiliza los números OID en vez de los nombres OID como el proceso gradual anterior. Vea a los detalles de MIB para traducir. Este ejemplo crea el VLA N 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
```

Nota: Ciertas versiones de SNMP le requieren utilizar a (.) antes del OID en los comandos SNMP SET.

Borre un VLA N de un Switch del Cisco Catalyst con el SNMP

Instrucciones Paso a Paso

En este ejemplo el VLA N 48 se borra del Switch. Refiera al agregar un VLA N a un Cisco Catalyst con el SNMP para más información. La diferencia entre esta sección donde usted borra un VLA N y el donde usted agrega un VLA N es que usted utiliza la **destrucción** en vez del **comando CreateAndGo** para el **vtpVlanEditRowStatus**:

1. Publique el comando de borrar el VLA N 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. Para verificar que el VLA N 48 fuera borrado, utilice el **vtpVlanState** o **muestre vlan** en el 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 …
   ```

Agregue un puerto a un VLA N en un Switch del Cisco Catalyst con el SNMP

Este ejemplo muestra cómo agregar un fast ethernet 0/5 del puerto al VLA N 48.

1. Para verificar que el Eth rápido 0/5 del ifIndex tiene, publique un **snmpwalk del ifDescr**:
   ```
   snmpwalk -c public crumpy ifDescr … interfaces.ifTable.ifEntry.ifDescr.6 : DISPLAY STRING-
   (ascii): FastEthernet0/5 …
   ```
2. Puesto que usted sabe que el Eth rápido 0/5 del puerto tiene un ifIndex de 6, agregue el

puerto al VLA N 48:
snmpset -c private crumpy vmVlan.6 integer 48
cisco.ciscoMgmt.ciscoVlanMembershipMIB.ciscoVlanMembershipMIBObjects.vmMembership.vmMembershipTable.vmMembershipEntry.vmVlan.6 : INTEGER: 48

3. Verifique que el puerto fuera agregado correctamente preguntando el mismo OID otra vez.
snmpget -c public crumpy vmVlan.6
cisco.ciscoMgmt.ciscoVlanMembershipMIB.ciscoVlanMembershipMIBObjects.vmMembership.vmMembershipTable.vmMembershipEntry.vmVlan.6 : INTEGER: 48 Usted puede también verificar esto en el Switch: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

Cómo cambiar un puerto a partir de un VLA N a otro VLA N

Este ejemplo demuestra cómo el Eth rápido 0/3 del puerto pertenece al VLA N 48 y cómo moverlo al VLAN1 (VLAN predeterminado):

1. Para verificar que el Eth rápido 0/3 del ifIndex tiene, publique un **snmpwalk del ifDescr**:
snmpwalk -c public crumpy ifDescr … interfaces.ifTable.ifEntry.ifDescr.4 : DISPLAY STRING-(ascii): FastEthernet0/3 …
2. Puesto que usted sabe que el Eth rápido 0/3 del puerto tiene un ifIndex de 4, usted puede verificar a qué VLA N pertenece el puerto actualmente:
snmpget -c public crumpy vmVlan.4
cisco.ciscoMgmt.ciscoVlanMembershipMIB.ciscoVlanMembershipMIBObjects.vmMembership.vmMembershipTable.vmMembershipEntry.vmVlan.4 : INTEGER: 48
3. El puerto pertenece al VLA N 48.
snmpset -c private crumpy vmVlan.4 integer 1
cisco.ciscoMgmt.ciscoVlanMembershipMIB.ciscoVlanMembershipMIBObjects.vmMembership.vmMembershipTable.vmMembershipEntry.vmVlan.4 : INTEGER: 1
4. Para mover el puerto desde el VLA N 48 al VLAN1, publique un **snmpset de vmVlan**.
5. Para verificar si el puerto fue cambiado al otro VLA N, pregunte **vmVlan** otra vez:
snmpget -c public crumpy vmVlan.4
cisco.ciscoMgmt.ciscoVlanMembershipMIB.ciscoVlanMembershipMIBObjects.vmMembership.vmMembershipTable.vmMembershipEntry.vmVlan.4 : INTEGER: 1 Usted puede también verificar esto en el Switch sí mismo:Antes del cambio: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 Después del cambio:
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
Nota: Usted puede realizar otros cambios, tales como el nombre del VLA N, el propietario, y mucho más. Refiera al MIB entero para más detalles en el OID.

Información Relacionada

- Soporte Técnico - Cisco Systems