

"Добавление, изменение и удаление VLAN на Catalyst с помощью SNMP"

Содержание

[Введение](#)

[Предварительные условия](#)

[Требования](#)

[Компоненты](#)

[Условные обозначения](#)

[Общие сведения](#)

[Подробные данные переменных MIB — включая идентификаторы объекта \(OID\)](#)

[Добавьте VLAN к коммутатору Cisco Catalyst с SNMP](#)

[Пошаговые инструкции](#)

[Добавьте VLAN к коммутатору Cisco Catalyst с SNMP](#)

[Инструкции по шагу](#)

[Удалите VLAN из коммутатора Cisco Catalyst с SNMP](#)

[Пошаговые инструкции](#)

[Добавьте порт к VLAN на коммутаторе Cisco Catalyst с SNMP](#)

[Как изменить порт от одной VLAN до другой VLAN](#)

[Дополнительные сведения](#)

Введение

В этом документе описывается создание и удаление VLAN на коммутаторе Cisco Catalyst с использованием протокола SNMP. Также описан способ добавления портов в сеть VLAN с помощью SNMP.

Предварительные условия

Требования

Прежде чем вы будете использовать сведения в этом документе, будете гарантировать, что понимаете:

- Как работают ifTable и ifIndex
- Как VLAN работают на коммутаторы Cisco Catalyst
- Как просмотреть сведения о виртуальной локальной сети (VLAN) на коммутаторах Cisco Catalyst
- Общее использование **SNMP get**, набора и команд **walk**

Компоненты

Этот документ для Коммутаторов Catalyst, которые выполняют обычный Catalyst OS или Catalyst IOS, которые поддерживают IF-MIB, CISCO-VTP-MIB и CISCO-VLAN-MEMBERSHIP-MIB. Сведения, содержащиеся в данном документе, касаются следующих версий программного обеспечения и обслуживания:

- Catalyst 3524XL запускает CatIOS 12.0(5)WC5a
- [Версия NET-SNMP 5.0.6 доступна по адресу http://www.net-snmp.org/](http://www.net-snmp.org/)

Сведения, содержащиеся в данном документе, были получены с устройств в специальной лабораторной среде. Все устройства, описанные в данном документе, были запущены с конфигурацией по умолчанию. Если вы работаете в действующей сети перед использованием любой команды, удостоверьтесь, что вы понимаете потенциальное воздействие любой команды.

Условные обозначения

[Дополнительные сведения об условных обозначениях см. в документе Технические рекомендации Cisco. Условные обозначения.](#)

Общие сведения

[Подробные данные переменных MIB — включая идентификаторы объекта \(OID\)](#)

```
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 }

[Добавьте VLAN к коммутатору Cisco Catalyst с SNMP](#)

[Пошаговые инструкции](#)

В примере, показанном ниже, VLAN 11 добавлен к коммутатору:

1. Для проверки, какие VLAN в настоящее время настраиваются на коммутаторе, выполняют **snmpwalk** на OID **vtpVlanState**: **Примечание:** Последний номер в OID является номером виртуальной локальной сети (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. Проверьте, используется ли выпуск другой Станцией NMS или устройством. Если вы

видите это сообщение, выпуск не используется: no MIB objects contained under subtree:
`snmpwalk -c public crumpy vtpVlanEditTable` no MIB objects contained under subtree.

- Выпуск не используется, таким образом, безопасно начать редактировать. Установите **vtpVlanEditOperation** в состояние копии (целое число 2). Это позволяет вам создавать VLAN.

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

- Для создания текущего владельца разрешений редактирования видимым, можно установить владельца при выдаче команды, **vtpVlanEditBufferOwner**.

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

- Данный пример показывает, как проверить, что существует таблица:

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

- Данный пример является VLAN 11 и показывает вам, как создать строку и установить тип и название:

```
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
```

- Установите **vtpVlanEditDot10Said**. Это - номер виртуальной локальной сети (VLAN) + 100000 преобразованных к шестнадцатеричному. Данный пример создает VLAN 11, таким образом, **vtpVlanEditDot10Said** должен быть: $11 + 100000 = 100011 \rightarrow$ Hex: 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 -- -- -- --
-- -- -- -- -- .....
```

- Когда у вас есть созданная виртуальная локальная сеть VLAN 11, необходимо применить модификации. Используйте **vtpVlanEditOperation** OID снова. На этот раз используйте **Применение** для подтверждения параметров настройки:

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

- Проверьте, что VLAN была создана успешно. Используйте **статус применения VTP VLAN для OID**. Проверьте процесс до чтений статуса: succeeded:

```
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
```

- Последнее действие должно передать модификации и освободить разрешения так, чтобы другие пользователи могли добавить, модифицировать или удалить VLAN из своего NMS.

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

- Проверьте, что буфер пуст:

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

- Проверьте, что VLAN 11 был создан на коммутаторе с **show vlan** команды CLI или с **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 ...
```

[Добавьте VLAN к коммутатору Cisco Catalyst с SNMP](#)

[Инструкции по шагу](#)

Один процесс шага использует номера OID вместо названий OID как предыдущий пошаговый процесс. Посмотрите [подробности MIB для преобразования](#). Данный пример создает 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
```

Примечание: Определенные версии SNMP требуют, чтобы вы использовали (.) перед OID в командах SNMP SET.

[Удалите VLAN из коммутатора Cisco Catalyst с SNMP](#)

[Пошаговые инструкции](#)

В данном примере VLAN 48 удален из коммутатора. См. [Добавление VLAN к Cisco Catalyst с SNMP](#) для получения дополнительной информации. Различие между этим разделом, где вы удаляете VLAN и ту, где вы добавляете VLAN, - то, что вы используете **уничтожение** вместо команды **CreateAndGo** для **vtpVlanEditRowStatus**:

1. Выполните команду для удаления VLAN 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.vtpVlanEditRowStatus.1.48 : INTEGER: destroy
```

2. Чтобы проверить, что VLAN 48 был удален, используйте **vtpVlanState** или **show vlan** на 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
```

Добавьте порт к VLAN на коммутаторе Cisco Catalyst с SNMP

Данный пример показывает, как добавить Fast Ethernet порта 0/5 к VLAN 48.

1. Для проверки, который имеет ifIndex Быстрый Eth 0/5 выполните **snmpwalk ifDescr**:

```
snmpwalk -c public crumpy ifDescr ... interfaces.ifTable.ifEntry.ifDescr.6 : DISPLAY STRING-  
(ascii): FastEthernet0/5 ...
```
2. Так как вы знаете, что порт, Быстрый Eth 0/5 имеет ifIndex 6, добавляет порт к VLAN 48:

```
snmpset -c private crumpy vmVlan.6 integer 48  
cisco.ciscoMgmt.ciscoVlanMembershipMIB.ciscoVlanMembershipMIBObjects.vmMembership.vmMembers  
hipTable.vmMembershipEntry.vmVlan.6 : INTEGER: 48
```
3. Проверьте, что порт был правильно добавлен путем запроса того же OID снова.

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

 Можно также проверить это на коммутаторе:

```
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
```

Как изменить порт от одной VLAN до другой VLAN

Данный пример демонстрирует, как порт, Быстрый Eth 0/3 принадлежит VLAN 48 и как переместить его в (виртуальную локальную сеть (VLAN) по умолчанию) VLAN 1:

1. Для проверки, который имеет ifIndex Быстрый Eth 0/3 выполните **snmpwalk ifDescr**:

```
snmpwalk -c public crumpy ifDescr ... interfaces.ifTable.ifEntry.ifDescr.4 : DISPLAY STRING-  
(ascii): FastEthernet0/3 ...
```
2. Так как вы знаете, что порт, Быстрый Eth 0/3 имеет ifIndex 4, можно проверить, которой VLAN в настоящее время принадлежит порт:

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

```
snmpset -c private crumpy vmVlan.4 integer 1  
cisco.ciscoMgmt.ciscoVlanMembershipMIB.ciscoVlanMembershipMIBObjects.vmMembership.vmMembers  
hipTable.vmMembershipEntry.vmVlan.4 : INTEGER: 1
```
4. Для перемещения порта от VLAN 48 до VLAN 1 выполните **snmpset vmVlan**.
5. Проверить, был ли порт изменен на другую VLAN, запрос **vmVlan** снова:

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

 Можно также проверить это на самом коммутаторе:

```
Перед изменением: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
```

 После изменения:

```
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
```

Примечание: Можно внести другие изменения, такие как название VLAN, владелец, и многое другое. См. весь MIB для получения дополнительной информации на OID.

Дополнительные сведения

- [Техническая поддержка - Cisco Systems](#)