

Trunking Between Catalyst 4500/4000, 5500/5000, and 6500/6000 Series Switches Using 802.1Q Encapsulation with Cisco CatOS System Software

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Conventions](#)

[What Is a Trunk?](#)

[Basic Characteristics of 802.1Q Trunking](#)

[Tagging Mechanism](#)

[Spanning Tree Consideration](#)

[Cisco Implementation](#)

[Configure 802.1Q Trunks](#)

[Hardware/Software Requirements](#)

[DTP Modes](#)

[Step-by-Step Example](#)

[Common Errors](#)

[Different Native VLANs](#)

[Different VTP Domains](#)

[Error During an Attempt to Delete Extended-Range VLANs from a Trunk Port](#)

[Trunking Mode Incompatible with the Encapsulation Type](#)

[Commands Used in the Document](#)

[Command Summary](#)

[Related Information](#)

[Introduction](#)

This document introduces the concept of trunking between two Ethernet switches and focuses on the IEEE 802.1Q trunking standard. After a brief description of the 802.1Q trunking mechanism, the document describes the implementation on the Catalyst 4500/4000, 5500/5000, and 6500/6000 series switches. A full example is provided, along with some common errors that relate to the 802.1Q trunking configuration with the use of Catalyst OS (CatOS) system software. For examples of 802.1Q trunking with Cisco IOS® system software, refer to [Configuring 802.1Q Trunking Between a Catalyst 3550/3560/3750 and Catalyst Switches That Run Cisco IOS Software](#).

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

This document is not restricted to specific software and hardware versions.

Conventions

Refer to [Cisco Technical Tips Conventions](#) for more information on document conventions.

What Is a Trunk?

In Cisco terminology, a trunk is a point-to-point link that carries several VLANs. The purpose of a trunk is to save ports when creating a link between two devices that implement VLANs, typically two switches. In this diagram, there are two VLANs that you want to have available on two switches, Sa and Sb. The first easy method to implement is to create two physical links between the devices. The physical links each carry the traffic for a VLAN:

Of course, this solution does not scale. If you want to add a third VLAN, you must sacrifice two additional ports. This design is also inefficient in terms of load sharing; the traffic on some VLANs may not justify a dedicated link. A trunk bundles virtual links over one physical link, as this diagram shows:

Here, the unique physical link between the two switches is able to carry traffic for any VLAN. In order to achieve this, each frame sent on the link is tagged by Sa so that Sb knows the VLAN to which it belongs. Different tagging schemes exist. The most common for Ethernet segments are:

- Inter-Switch Link (ISL) (the original Cisco proprietary ISL protocol)
- 802.1Q (the IEEE standard on which this document focuses)

Basic Characteristics of 802.1Q Trunking

Tagging Mechanism

802.1Q uses an internal tagging mechanism. Internal means that a tag is inserted within the frame:

Note: With ISL, the frame is encapsulated instead.

Note: On an 802.1Q trunk, one VLAN is NOT tagged. This VLAN, named the native VLAN, must be configured the same on each side of the trunk. In this way, you can deduce to which VLAN a frame belongs when you receive a frame with no tag.

The tagging mechanism implies a modification of the frame; the trunking device inserts a 4-byte

tag and recomputes the frame check sequence (FCS):

The EtherType field that identifies the 802.1Q frame is 0x8100. In addition to the 12-bit VLAN-ID, 3 bits are reserved for IEEE 802.1p priority tagging.

Note: Inserting a tag into a frame that already has the maximum Ethernet size creates a 1522-byte frame that can be considered a "baby giant" by the receiving equipment. The IEEE 802.3 committee is extending the maximum standard frame size in order to address this issue.

[Spanning Tree Consideration](#)

The 802.1Q standard is more than just a tagging mechanism. It also defines a unique spanning tree instance that runs on the native VLAN for all the VLANs in the network. Such a Mono Spanning Tree (MST) network lacks some flexibility in comparison to a Per VLAN Spanning Tree (PVST) network that runs one instance of Spanning Tree Protocol (STP) per VLAN. Cisco developed PVST+ in order to allow running several STP instances (even over an 802.1Q network) by using a tunneling mechanism. Although beyond the scope of this document, it can be briefly described as utilizing a Cisco device in order to connect a MST zone (typically the 802.1Q-based network of another vendor) to a PVST zone (typically a Cisco ISL-based network). There is no specific configuration to enter in order to achieve this. Ideally, a mixed environment should look like this diagram:

[Cisco Implementation](#)

In the current implementation, Cisco devices support only VLAN numbers up to 1005. This restriction, introduced to match the number of VLANs that are available with ISL, is allowed by the 802.1Q standard. Cisco implemented a VLAN mapping feature in CatOS 5.1 in order to simplify interoperability with other vendor devices, but it is seldom necessary.

Note: Refer to [Configuring VLANs](#) for information on the VLAN mapping feature.

Cisco also adapted its Dynamic ISL (DISL) protocol and turned it into Dynamic Trunking Protocol (DTP). DISL can negotiate ISL trunking on a link between two devices; DTP can, in addition, negotiate the type of trunking encapsulation (802.1Q or ISL) that will be used as well. This is an interesting feature as some Cisco devices support only ISL or 802.1Q, whereas some are able to run both.

In Cisco implementation, a trunk is a point-to-point link, although it is possible to use the 802.1Q encapsulation on an Ethernet segment shared by more than two devices. Such a configuration is seldom needed but is still possible with the disablement of DTP negotiation.

[Configure 802.1Q Trunks](#)

[Hardware/Software Requirements](#)

From a software point of view, the first appearance of 802.1Q encapsulation was with CatOS software 4.1. In this release, trunking configuration had to be hard coded; DTP only appeared with CatOS 4.2. See the [DTP Modes](#) section of this document.

Not all Catalyst ports support 802.1Q encapsulation. Currently, while Catalyst 4500/4000 switches only support 802.1Q, ports of the Catalyst 6500/6000 series are able to use 802.1Q or ISL

encapsulation. Depending on the module, Catalyst 5500/5000 trunk-capable ports are able to use 802.1Q encapsulation, ISL encapsulation, or both. The best way to check this out is to use the [show port capabilities](#) command. The trunking capacity is explicitly stated:

```
Sa> (enable) show port capabilities 1/1
Model                WS-X5530
Port                 1/1
Type                 1000BaseSX
Speed                1000
Duplex                full
Trunk encap type     802.1Q, ISL
Trunk mode            on, off, desirable, auto, nonegotiate
Channel              no
Broadcast suppression percentage(0-100)
Flow control          receive-(off, on, desired), send-(off, on, desired)
Security              no
Membership            static
Fast start            yes
Rewrite               no
```

[DTP Modes](#)

When you configure a port for trunking, you can set two parameters: the trunking mode and the encapsulation type (if DTP is supported on that port).

- The **trunking mode** defines how the port will negotiate the setup of a trunk with its peer port. Here is a list of the possible settings: Be careful that some modes (*on*, *nonegotiate*, *off*) explicitly specify in which state the port will end up. A bad configuration can lead to a dangerous, inconsistent state in which one side is trunking and the other side is not. A port in *on*, *auto*, or *desirable* sends DTP frames periodically. A trunking port in *auto* or *desirable* goes back to nontrunking if it does not receive a DTP update from its neighbor within 5 minutes. **Note:** If you run CatOS software 4.1, you must disable any form of negotiation by using the **off** or **nonegotiate** mode when you configure 802.1Q trunking.
- The **encapsulation type** allows the user to specify whether 802.1Q or ISL should be used when setting up the trunk. Of course, the parameter is only relevant if the module that you use is able to use both. The parameter can have three different values:

Refer to the [Results of Possible Fast Ethernet and Gigabit Ethernet Trunk Configurations](#) section of [Configuring VLAN Trunks on Fast Ethernet and Gigabit Ethernet Ports](#) for a list of all the possible resulting configurations.

Note: No negotiation will take place between two switches in different VLAN Trunk Protocol (VTP) domains. Refer to [Configuring VTP](#).

[Step-by-Step Example](#)

[Network Diagram](#)

This example is based on a very simple lab setup that involves two Catalyst 5500/5000 switches that are linked together via trunk-capable ports. You need a [crossover cable](#) in order to interconnect two switches.

Minimal Setup of a 802.1Q Trunk with Connectivity Tests

Complete these steps:

1. Check that the statuses of the ports are up but not trunking. Connect a terminal to the console of your switches. Refer to the document [Connecting a Terminal to the Console Port on Catalyst Switches](#) if necessary. First, check the status of the port that is involved in the setup. Use the command [show port 5/24](#) on Sa ([show port 2/24](#) on Sb) and check that the status is connected:

```
Sa> (enable) show port 5/24
Port Name                Status      Vlan      Level Duplex Speed Type
-----
 5/24                    connected  1         normal a-full a-100 10/100BaseTX
!--- Output suppressed.
```

You have the default value for that kind of port. It came when negotiating 100-MB full duplex, and it is assigned to VLAN 1. Issue the **show trunk 5/24** command in order to clearly see that the port is not trunking and has a default mode auto and encapsulation negotiate.

```
Sa> (enable) show trunk 5/24
Port      Mode      Encapsulation  Status      Native vlan
-----
 5/24     auto     negotiate      not-trunking  1
!--- Output suppressed.
```

2. Set an IP address on the sc0 management interfaces. Use the [set interface sc0 10.0.0.1](#) command on switch Sa and the [set interface sc0 10.0.0.2](#) command on switch Sb in order to assign an IP address to the two switches. The [show interface](#) command confirms that the management interface is now correctly set in the default VLAN 1:

```
Sa> (enable) set interface sc0 10.0.0.1
Interface sc0 IP address set.

Sa> (enable) show interface
sl0: flags=51<,POINTOPOINT,RUNNING>
      slip 0.0.0.0 dest 0.0.0.0
sc0: flags=63<UP,BROADCAST,RUNNING>
      vlan 1 inet 10.0.0.1 netmask 255.0.0.0 broadcast 10.255.255.255
Sa> (enable)
```

If you have the output of a **show interface** command from your Cisco device, you can use [Output Interpreter](#) ([registered](#) customers only) to display potential issues and fixes.

3. Check connectivity between Sa and Sb. Issue the [ping 10.0.0.2](#) command from switch Sa in order to prove that switch Sb can now be reached:

```
Sa> (enable) ping 10.0.0.2
10.0.0.2 is alive
Sa> (enable)
```

4. Configure the same VTP domain on both switches. Now assign the same VTP domain to both switches. As you saw, having the same VTP domain is mandatory in order to use DTP negotiation. Issue the [set vtp domain cisco](#) command on both switches in order to configure them with the domain name "cisco":

```
Sa> (enable) set vtp domain cisco
VTP domain cisco modified
Sa> (enable)
```

5. Create a VLAN 2 in each switch. Issue the [set vlan 2](#) command on both switches in order to create the VLAN 2. If the switches were already linked by a trunk, you would only need to issue the command on one switch, and the other switch would learn it automatically via VTP. As you do not have a trunk yet, there is no VTP communication between Sa and Sb:

```
Sa> (enable) set vlan 2
Vlan 2 configuration successful
```

```
Sa> (enable)
```

6. Change the management interfaces to VLAN 2. You now move the management interface of both switches into VLAN 2. In this way, you show that there is no communication between Sa and Sb before a trunk is established. Issue the [set interface sc0 2](#) command on each switch in order to move the sc0 interface in VLAN 2. Issue the [show interface](#) command in order to check that the command is effective:

```
Sa> (enable) set interface sc0 2
Interface sc0 vlan set.
Sa> (enable) show interface
sl0: flags=51<UP,POINTOPOINT,RUNNING>
    slip 0.0.0.0 dest 0.0.0.0
sc0: flags=63<UP,BROADCAST,RUNNING>
    vlan 2 inet 10.0.0.1 netmask 255.0.0.0 broadcast 10.255.255.255
Sa> (enable)
```

7. Check if connectivity is broken between the two switches. Now the [ping 10.0.0.2](#) to Sb fails from Sa, which proves that there is no connectivity in VLAN 2 between the switches:

```
Sa> (enable) ping 10.0.0.2
no answer from 10.0.0.2
Sa> (enable)
```

8. Check the port capabilities. Before you start to configure a trunk, you can check with the [show port capabilities](#) command that both ports are able to implement 802.1Q trunking:

```
Sa> (enable) show port capabilities 5/24
Model                WS-X5225R
Port                 5/24
Type                 10/100BaseTX
Speed                auto,10,100
Duplex                half,full
Trunk encap type     802.1Q,ISL
Trunk mode            on,off,desirable,auto,nonegotiate
Channel              5/23-24,5/21-24
Broadcast suppression percentage(0-100)
Flow control         receive-(off,on),send-(off,on)
Security              yes
Membership            static,dynamic
Fast start            yes
Rewrite               yes
Sa> (enable)
```

9. Configure the trunk encapsulation to be 802.1Q. Now the trunk on Sa must be configured. You saw in Step 1 that both ports were in the default trunking mode auto, encapsulation type negotiate. A combination auto-auto does not bring a trunk up. This is normal; each side is willing to become trunk, but will only do it if the remote requests it. With consideration of the default configuration: You just need to change the trunk mode to desirable on one side in order to bring the trunk up. This is because a port in desirable mode notifies its neighbor that it wants to go trunking. As the remote (in auto mode) goes to trunking if prompted to, this is enough to bring the trunk up. If you configure the encapsulation dot1q on a subinterface, this means that that VLAN cannot be used again in the system since internally, the 6500 or 7600 allocate the VLAN and then make that subinterface the only member of it. So it is not possible to have a VLAN and then try to use it in a subinterface or vice versa. In order to fix that issue, instead of subinterfaces, create trunk ports and that way the VLAN can be seen in all interfaces. If subinterfaces are required, then the VLANs added in the subinterfaces can not be used in other ports. You also need to specify which encapsulation you want to use. This is because both ports are ISL capable, and this encapsulation is chosen first when both ends are in negotiate mode. The syntax of the command is: **set trunk module/port [on | off | desirable | auto | nonegotiate] [vlan_range] [isl | dot1q | negotiate]**. Issue the [set trunk 5/24 dot1q desirable](#) command on switch Sa:

```

Sa> (enable) set trunk 5/24 dot1q desirable
Port(s) 5/24 trunk mode set to desirable.
Port(s) 5/24 trunk type set to dot1q.
1997 May 07 17:32:01 %DTP-5-TRUNKPORTON:Port 5/24 has become dot1q trunk
1997 May 07 17:32:02 %PAGP-5-PORTFROMSTP:Port 5/24 left bridge port 5/24
1997 May 07 17:32:13 %PAGP-5-PORTTOSTP:Port 5/24 joined bridge port 5/24

```

10. Verify that the trunk is up. The console log of the previous command clearly shows that the port moved to trunking, but you can also issue the [show trunk 5/24](#) command on Sa and the [show trunk 2/24](#) command on Sb in order to check. You can see a subtle difference between the two outputs: The port on Sa is in desirable mode, whereas the Sb port is in auto mode. More interesting, the encapsulation is dot1q on Sa whereas it is **n-dot1q** on Sb. This is to show that Sb negotiated its encapsulation to dot1q. If you did not specify an encapsulation on Sa, both ports would have ended up in n-isl encapsulation:

```

Sa> (enable) show trunk 5/24
Port      Mode      Encapsulation  Status      Native vlan
-----
5/24      desirable dot1q           trunking    1

Port      Vlans allowed on trunk
-----
5/24      1-1005

Port      Vlans allowed and active in management domain
-----
5/24      1-2

Port      Vlans in spanning tree forwarding state and not pruned
-----
5/24      1-2

```

```

Sa> (enable)
Sb> (enable) show trunk 2/24
Port      Mode      Encapsulation  Status      Native vlan
-----
2/24      auto      n-dot1q        trunking    1

```

!--- Output suppressed.

If you have the output of a **show trunk** command from your Cisco device, you can use [Output Interpreter](#) (registered customers only) to display potential issues and fixes.

11. Check connectivity. You can check that VLAN 2 is now going through your trunk by simply pinging Sb from Sa:

```

Sa> (enable) ping 10.0.0.2
10.0.0.2 is alive
Sa> (enable)

```

[Set the Native VLAN](#)

Complete these steps:

1. Issue the **set vlan** command. The command [set vlan 2 5/24](#) is used to assign a port to a specific VLAN. In the case of a trunking port, it changes the native VLAN to VLAN 2. Of course, you need to do the same on Sb with [set vlan 2 2/24](#):

```

Sa> (enable) set vlan 2 5/24
VLAN 2 modified.
VLAN 1 modified.
VLAN Mod/Ports
-----
2      5/24

```

```
Sa> (enable)
```

Before you change the native VLAN on Sb, there is now an inconsistency between the Sa and Sb configuration. The two ends of the trunk do not have the same native VLAN configuration. Here, some warning messages are displayed on the Sb console. **Note:** The switch that reports the inconsistency may vary, which depends on which one is the root bridge for VLANs 1 and 2.

```
Sb> (enable) 2000 Dec 07 16:31:24 %SPANTREE-2-RX_1QPVIDERR: Rcvd
pvid_inc BPDU on 1Q port 2/24 vlan 1.
2000 Dec 07 16:31:24 %SPANTREE-2-TX_BLKPORTPVID: Block 2/24 on xmtting
vlan 2 for inc peer vlan.
2000 Dec 07 16:31:24 %SPANTREE-2-RX_BLKPORTPVID: Block 2/24 on rcving
vlan 1 for inc peer vlan 2.
```

```
Sb> (enable)
```

```
Sb> (enable) set vlan 2 2/24
```

```
VLAN 2 modified.
```

```
VLAN 1 modified.
```

```
VLAN Mod/Ports
```

```
-----
```

```
2      2/24
```

```
Sb> (enable) 2000 Dec 07 16:31:46 %SPANTREE-2-PORTUNBLK: Unblock
previously inc port 2/24 on vlan 1.
```

```
2000 Dec 07 16:31:48 %SPANTREE-2-PORTUNBLK: Unblock previously inc
port 2/24 on vlan 2.
```

The native VLAN mismatch has been corrected and everything goes back to normal.

2. Check the result. Now simply check the result of these commands on your trunk with use of the [show trunk 5/24](#) command:

```
Sa> (enable) show trunk 5/24
```

Port	Mode	Encapsulation	Status	Native vlan
5/24	desirable	dot1q	trunking	2

```
<
```

[Specify VLANs That Are Allowed on the Trunk](#)

Complete these steps:

1. Create additional VLANs. When you create a new trunk, it carries by default all the existing VLANs in the network. You will see how to restrict the list of allowed VLANs on a trunk. First, you must create two additional VLANs (3 and 4). You can issue the [set vlan 3](#) command and the [set vlan 4](#) command on Sa, for instance, in order to create the additional VLANs. You only need to enter the command on one switch; VTP propagates this information to the other switch. **Note:** This part of the configuration is absolutely the same whether 802.1Q or ISL encapsulation is used.

```
Sa> (enable) set vlan 3
```

```
Vlan 3 configuration successful
```

```
Sa> (enable) set vlan 4
```

```
Vlan 4 configuration successful
```

2. Remove VLANs from the trunk. The command **clear trunk module/port vlan-list** allows you to remove one or several VLANs from a given trunk. Here, the four VLANs that you created were defined on your trunk. Remove VLAN 2 and VLAN 3 with use of the [clear trunk 5/24 2-3](#) command on Sa and the [clear trunk 2/24 2-3](#) command on Sb. You can check the result of the **clear** command with use of the [show trunk 5/24](#) command. Only VLANs 1 and 4 now cross the trunk between Sa and Sb. A ping between Sa and Sb now fails:


```

Sa> (enable) clear trunk 5/24 2-3
Removing Vlan(s) 2-3 from allowed list.
Port 5/24 allowed vlans modified to 1,4-1005.
Sa> (enable) show trunk 5/24
Port      Mode           Encapsulation  Status      Native vlan
-----  -
5/24     desirable     dot1q          trunking    2

Port      Vlans allowed on trunk
-----  -
5/24     1,4-1005

Port      Vlans allowed and active in management domain
-----  -
5/24     1,4

Port      Vlans in spanning tree forwarding state and not pruned
-----  -
5/24     1,4

```

3. Reactivate a VLAN. In order to add a VLAN back on a trunk, use the [set trunk module/port vlan-list](#) command.

```

Sa> (enable) set trunk 5/24 2
Adding vlans 2 to allowed list.
Port(s) 5/24 allowed vlans modified to 1-2,4-1005.
Sa> (enable) show trunk
Port      Mode           Encapsulation  Status      Native vlan
-----  -
5/24     desirable     dot1q          trunking    2

Port      Vlans allowed on trunk
-----  -
5/24     1-2,4-1005

Port      Vlans allowed and active in management domain
-----  -
5/24     1-2,4

Port      Vlans in spanning tree forwarding state and not pruned
-----  -
5/24     1-2,4

```

VLAN 2 is now flowing again on the trunk. A ping from Sa to Sb is possible.

Common Errors

Different Native VLANs

This is a frequent configuration error. The native VLAN that is configured on each end of a 802.1Q trunk must be the same. Remember that a switch receiving a nontagged frame assigns it to the native VLAN of the trunk. If one end is configured for native VLAN 1 and the other to native VLAN 2, a frame that is sent in VLAN 1 on one side is received on VLAN 2 on the other. This results in the merge of VLAN 1 and 2. There is no reason that you would want that, and it may imply some connectivity issues in your network.

A Cisco device usually warns you of a native VLAN mismatch. See Step 1 of the section [Set the Native VLAN](#) for the kind of error messages that you get on the console in this case. Always check that the native VLAN is the same on the trunk configuration of your switches.

Different VTP Domains

When you create a trunk between two switches and you use DTP negotiation, double check that the VTP domain that is configured on both switches is the same. Negotiation does not take place between two switches that are in different VTP domains. The example in this section takes the working trunking configuration that is described above.

Note: Even if two switches are in different VTP domains, you can make these switches communicate with each other if you add VLANs manually on each switch. Although there is a VTP domain mismatch, the VLAN communication works fine. However, VTP updates are not propagated through this link on that VLAN because the domains are different.

- Sa in trunking mode desirable, encapsulation dot1q
- Sb in trunking mode auto, encapsulation negotiate
- The same native VLAN, and the same VLANs allowed on each side

The only difference is that you assign VTP domain "c" on Sa and VTP domain "**cisco**" on Sb:

```
Sa> (enable) show trunk
No ports trunking.
Sa> (enable) show trunk 5/24
Port      Mode           Encapsulation  Status        Native vlan
-----  -
5/24     desirable     dot1q          not-trunking  1

Port      Vlans allowed on trunk
-----  -
5/24     1-1005

Port      Vlans allowed and active in management domain
-----  -
5/24     1

Port      Vlans in spanning tree forwarding state and not pruned
-----  -
5/24

Sb> (enable) show trunk
No ports trunking.
Sb> (enable) show trunk 2/24
Port      Mode           Encapsulation  Status        Native vlan
-----  -
2/24     auto          negotiate      not-trunking  1

Port      Vlans allowed on trunk
-----  -
2/24     1-1005

Port      Vlans allowed and active in management domain
-----  -
2/24     1

Port      Vlans in spanning tree forwarding state and not pruned
-----  -
2/24
Sb> (enable)
```

You can see that the trunk did not come up. When you see that kind of issue, check the VTP

domain that is configured on the switches. Issue the [show vtp domain](#) command:

```
Sa> (enable) show vtp domain
Domain Name                Domain Index VTP Version Local Mode Password
-----
c                          1           2           server      -

Vlan-count Max-vlan-storage Config Revision Notifications
-----
8          1023           0           disabled

Last Updater   V2 Mode Pruning PruneEligible on Vlans
-----
10.0.0.1      disabled disabled 2-1000
```

```
Sb> (enable) show vtp domain
Domain Name                Domain Index VTP Version Local Mode Password
-----
cisco              1           2           server      -

Vlan-count Max-vlan-storage Config Revision Notifications
-----
8          1023           20          disabled

Last Updater   V2 Mode Pruning PruneEligible on Vlans
-----
10.0.0.1      disabled disabled 2-1000
```

Now put switch Sa in VTP domain "cisco" with use of the [set vtp domain cisco](#) command. After a few seconds, the trunk is negotiated and up again:

```
Sa> (enable) set vtp domain cisco
VTP domain cisco modified
Sa> (enable) 1997 May 13 13:59:22 %DTP-5-TRUNKPORTON:Port 5/24 has become dot1q trunk
1997 May 13 13:59:22 %PAGP-5-PORTFROMSTP:Port 5/24 left bridge port 5/24
1997 May 13 13:59:33 %PAGP-5-PORTTOSTP:Port 5/24 joined bridge port 5/24
```

If you want to keep different VTP domains but still create a trunk between two switches, you must hard code trunking on each side of the trunk (with use of `nonegotiate/on`).

[Error During an Attempt to Delete Extended-Range VLANs from a Trunk Port](#)

When you try to delete the extended-range VLANs from a trunk port with use of the [clear trunk](#) command, this error is sometimes shown on the switch console:

```
Sa> (enable) set vtp domain cisco
VTP domain cisco modified
Sa> (enable) 1997 May 13 13:59:22 %DTP-5-TRUNKPORTON:Port 5/24 has become dot1q trunk
1997 May 13 13:59:22 %PAGP-5-PORTFROMSTP:Port 5/24 left bridge port 5/24
1997 May 13 13:59:33 %PAGP-5-PORTTOSTP:Port 5/24 joined bridge port 5/24
```

Note: The term *extended range* includes any VLAN from 1025 to 4094. The term *default extended range* includes all VLANs from 1025 to 4094. If you try to clear any VLAN in the range from 1025 to 4094, the VLAN becomes *non-default extended range*. The maximum number of trunks which

pass *non-default extended range* is 64. This includes both inactive and active trunks.

This error and the limitation of 64 trunks come from the NVRAM block which is used to store nondefault configurations for extended-range VLANs. If you issue the [show trunk extended-range](#) command, you can see all the trunks that are configured with nondefault extended ranges. By default, the entire configuration is stored in NVRAM. NVRAM has different "blocks" for saving the nondefault configurations. The blocks are placed into different categories, such as global or module. The block that holds the nondefault configuration for extended ranges has a limitation of 64 trunks.

There are two workarounds to reduce the number of nondefault extended-range trunks. The first method is to set any of the nonactive/unused trunk ports back to the default allowed VLANs. Use the [set trunk mod/port 1025-4094](#) command. Then the **clear trunk mod/port 1025-4094** command should work for the extended VLANs. The second workaround is to change the configuration mode from binary (default) to text mode. Use the [set config mode text](#) command in order to change the configuration mode to text mode. Text mode typically uses less NVRAM or Flash memory space than binary configuration mode uses.

Note: When operating in text file configuration mode, most user settings are not immediately saved to NVRAM; configuration changes are only written to DRAM. You must issue the [write memory](#) command in order to store the configuration in nonvolatile storage. Use the **set config mode text auto-save** command in order to save the text configuration in NVRAM automatically.

[Trunking Mode Incompatible with the Encapsulation Type](#)

This is a common issue that began to be raised to [Cisco Technical Support](#) when the first modules that were able to support both 802.1Q and ISL shipped. People were used to the configuration of a trunk with use of the **set trunk module/port on** command or the **set trunk module/port nonegotiate** command. The problem is that, by default, the encapsulation type is set to negotiate. The negotiate encapsulation type is only supported by auto or desirable trunking modes. The on and nonegotiate encapsulation types do not perform any negotiations between switches and must be hard set to ISL or 802.1Q encapsulation when they are configured. Here is a log of what happens on the switch in this case:

```
Sa> (enable) set trunk 5/24 on
Failed to set port 5/24 to trunk mode on.
Trunk mode 'on' not allowed with trunk encapsulation type 'negotiate'.
Sa> (enable) set trunk 5/24 nonegotiate
Failed to set port 5/24 to trunk mode nonegotiate.
Trunk mode 'nonegotiate' not allowed with trunk encapsulation type
'negotiate'.
Sa> (enable)
```

This makes sense because if you do not negotiate with the remote, how would you know which kind of encapsulation (802.1Q or ISL) to use in order to bring up the trunk? There are two possibilities:

- Use the desirable mode. In this case, you negotiate the encapsulation mode with the remote:

```
Sa> (enable) set trunk 5/24 desirable
Port(s) 5/24 trunk mode set to desirable.
Sa> (enable) 1997 May 09 17:49:19 %DTP-5-TRUNKPORTON:Port 5/24 has become
isl trunk
```

- Specify the encapsulation that you want to use:

```
Sa> (enable) set trunk 5/24 isl on
Port(s) 5/24 trunk mode set to on.
Port(s) 5/24 trunk type set to isl.
Sa> (enable) 1997 May 09 17:50:16 %DTP-5-TRUNKPORTON:Port 5/24 has become
isl trunk
```

[Commands Used in the Document](#)

[Command Summary](#)

- [ping](#)
- [set interface](#)
- [set trunk](#)
- [set vlan](#)
- [set vtp domain](#)
- [show interface](#)
- [show port](#)
- [show port capabilities](#)
- [show trunk](#)
- [show vtp domain](#)

[Related Information](#)

- [Configuring ISL Trunking on Catalyst 5500/5000 and 6500/6000 Family Switches](#)
- [Configuring VLAN Trunks on Fast Ethernet and Gigabit Ethernet Ports](#)
- [Understanding and Configuring VLAN Trunk Protocol \(VTP\)](#)
- [LAN Product Support](#)
- [LAN Switching Technology Support](#)
- [Technical Support & Documentation - Cisco Systems](#)