



## Configuring 802.1Q VLAN Interfaces

This module describes the configuration and management of 802.1Q VLAN interfaces.

The IEEE 802.1Q specification establishes a standard method for tagging Ethernet frames with VLAN membership information, and defines the operation of VLAN bridges that permit the definition, operation, and administration of VLAN topologies within a bridged LAN infrastructure.

The 802.1Q standard is intended to address the problem of how to divide large networks into smaller parts so broadcast and multicast traffic does not use more bandwidth than necessary. The standard also helps provide a higher level of security between segments of internal networks.

### Feature History for Configuring 802.1Q VLAN Interfaces

Release	Modification
Release 3.2	This feature was introduced on the Cisco CRS-1 Router.
Release 3.3.0	<ul style="list-style-type: none"><li>• Support was added for VLAN commands on bundled Ethernet interfaces.</li><li>• Support was added for the <b>dot1q native vlan</b> command on Cisco CRS-1 Router shared port adapters (SPAs).</li></ul>
Release 3.4.0	<ul style="list-style-type: none"><li>• The Layer 2 Virtual Private Network (L2VPN) feature was first supported on Ethernet interfaces on the Cisco CRS-1 Router.</li><li>• Support was added on for the 8-Port 1-Gigabit Ethernet SPA.</li></ul>

- [Prerequisites for Configuring 802.1Q VLAN Interfaces, on page 2](#)
- [Information About Configuring 802.1Q VLAN Interfaces, on page 2](#)
- [How to Configure 802.1Q VLAN Interfaces, on page 4](#)
- [Configuration Examples for VLAN Interfaces, on page 13](#)

## Prerequisites for Configuring 802.1Q VLAN Interfaces

You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Before configuring 802.1Q VLAN interfaces, be sure that the following conditions are met:

- You must have configured a Gigabit Ethernet interface, a 10-Gigabit Ethernet interface, a Fast Ethernet interface, or an Ethernet Bundle.

## Information About Configuring 802.1Q VLAN Interfaces

To configure 802.1Q VLAN interfaces, you must understand the following concepts:

### 802.1Q VLAN Overview

A VLAN is a group of devices on one or more LANs that are configured so that they can communicate as if they were attached to the same wire, when in fact they are located on a number of different LAN segments. Because VLANs are based on logical instead of physical connections, they are very flexible for user and host management, bandwidth allocation, and resource optimization.

The IEEE 802.1Q protocol standard addresses the problem of dividing large networks into smaller parts so broadcast and multicast traffic does not consume more bandwidth than necessary. The standard also helps provide a higher level of security between segments of internal networks.

The 802.1Q specification establishes a standard method for inserting VLAN membership information into Ethernet frames.

Cisco IOS XR software supports VLAN subinterface configuration on Gigabit Ethernet, 10-Gigabit Ethernet, and Fast Ethernet interfaces.

### 802.1Q Tagged Frames

The IEEE 802.1Q tag-based VLAN uses an extra tag in the MAC header to identify the VLAN membership of a frame across bridges. This tag is used for VLAN and quality of service (QoS) priority identification. The VLANs can be created statically by manual entry or dynamically through Generic Attribute Registration Protocol (GARP) VLAN Registration Protocol (GVRP). The VLAN ID associates a frame with a specific VLAN and provides the information that switches must process the frame across the network. A tagged frame is four bytes longer than an untagged frame and contains two bytes of Tag Protocol Identifier (TPID) residing within the type and length field of the Ethernet frame and two bytes of Tag Control Information (TCI) which starts after the source address field of the Ethernet frame.

## CFM on 802.1Q VLAN Interfaces

Configuring Connectivity Fault Management (CFM) for monitoring 802.1Q VLAN interfaces is identical to configuring CFM for monitoring Ethernet interfaces.

For information on configuring CFM for Ethernet interfaces, refer to the following sections in the [Configuring Ethernet OAM](#) module:

## Subinterfaces

Subinterfaces are logical interfaces created on a hardware interface. These software-defined interfaces allow for segregation of traffic into separate logical channels on a single hardware interface as well as allowing for better utilization of the available bandwidth on the physical interface.

Subinterfaces are distinguished from one another by adding an extension on the end of the interface name and designation. For instance, the Ethernet subinterface 23 on the physical interface designated TenGigE 0/1/0/0 would be indicated by TenGigE 0/1/0/0.23.

Before a subinterface is allowed to pass traffic it must have a valid tagging protocol encapsulation and VLAN identifier assigned. All Ethernet subinterfaces always default to the 802.1Q VLAN encapsulation. However, the VLAN identifier must be explicitly defined.

## Subinterface MTU

The subinterface maximum transmission unit (MTU) is inherited from the physical interface with an additional four bytes allowed for the 802.1Q VLAN tag.

## Native VLAN

Each physical port may have a native VLAN assigned. All untagged frames are assigned to the LAN specified in the PVID parameter. When received packet is tagged with the PVID, that packet is treated as if it was untagged. Therefore, the configuration associated with the native VLAN must be placed on the main interface. The native VLAN allows the coexistence of VLAN-aware bridge or stations with VLAN-unaware bridges or stations.

## VLAN Subinterfaces on Ethernet Bundles

An Ethernet bundle is a group of one or more Ethernet ports that are aggregated together and treated as a single link. Multiple VLAN subinterfaces can be added to a single Ethernet bundle.

The procedure for creating VLAN subinterfaces on an Ethernet bundle is exactly the same as the procedure for creating VLAN subinterfaces on a physical Ethernet interface.

To create a VLAN subinterface on an Ethernet bundle, see the [How to Configure 802.1Q VLAN Interfaces](#) section later in this module.

## Layer 2 VPN on VLANs

The Layer 2 Virtual Private Network (L2VPN) feature enables Service Providers (SPs) to provide layer 2 services to geographically disparate customer sites.

The configuration model for configuring VLAN attachment circuits (ACs) is similar to the model used for configuring basic VLANs, where the user first creates a VLAN subinterface, and then configures that VLAN in subinterface configuration mode. To create an AC, you need to include the **l2transport** keyword in the **interface** command string to specify that the interface is a Layer 2 interface.

VLAN ACs support three modes of L2VPN operation:

- Basic Dot1Q AC—The AC covers all frames that are received and sent with a specific VLAN tag.

- Q-in-Q AC—The AC covers all frames received and sent with a specific outer VLAN tag and a specific inner VLAN tag. Q-in-Q is an extension to Dot1Q that uses a stack of two tags.
- Q-in-Any AC—The AC covers all frames received and sent with a specific outer VLAN tag and any inner VLAN tag, as long as that inner VLAN tag is not L3 terminated. Q-in-Any is an extension to Q-in-Q that uses wildcarding to match any second tag.




---

**Note** Cisco CRS Router does not support Q-in-Q and Dot1Q with the same outer VLAN ID on the same interface/port.

---




---

**Note** The Q-in-Any mode is a variation of the basic Dot1Q mode. In Q-in-Any mode, the frames have a basic Q-in-Q encapsulation; however, in Q-in-Any mode the inner tag is not relevant, except for the fact that a few specific inner VLAN tags are siphoned for specific services. For example, a tag may be used to provide L3 services for general internet access.

---

Each VLAN on a CE-to-PE link can be configured as a separate L2VPN connection (using either VC type 4 or VC type 5). To configure L2VPN on VLANs, see the Configuring an Attachment Circuit on a VLAN section.

Keep the following in mind when configuring L2VPN on a VLAN:

- Cisco IOS XR software supports 4k ACs per LC.
- In a point-to-point connection, the two ACs do not have to be of the same type. For example, a port mode Ethernet AC can be connected to a Dot1Q Ethernet AC.
- Pseudo-wires can run in VLAN mode or in port mode. A pseudo-wire running in VLAN mode has a single Dot1Q tag, while a pseudo-wire running in port mode has no tags. Some interworking is required to connect these different types of circuits together. This interworking takes the form of popping, pushing and rewriting tags. The advantage of Layer 2 VPN is that it simplifies the interworking required to connect completely different media types together.
- The ACs on either side of an MPLS pseudo-wire can be different types. In this case, the appropriate conversion is carried out at one or both ends of the AC to pseudo-wire connection.

Use the **show interfaces** command to display AC and pseudo-wire information.




---

**Note** For detailed information about configuring an L2VPN network, see the module of the *Multiprotocol Label Switching Configuration Guide*.

---

## How to Configure 802.1Q VLAN Interfaces

This section contains the following procedures:

## Configuring 802.1Q VLAN Subinterfaces

This task explains how to configure 802.1Q VLAN subinterfaces. To remove these subinterfaces, see the “Removing an 802.1Q VLAN Subinterface” section.

### SUMMARY STEPS

1. **configure**
2. **interface** {GigabitEthernet | TenGigE | Bundle-Ether} *interface-path-id.subinterface*
3. **encapsulation dot1q**
4. **ipv4 address** *ip-address mask*
5. **exit**
6. Repeat Step 2 through Step 5 to define the rest of the VLAN subinterfaces.
7. **end** or **commit**
8. **show vlan interface** [*type interface-path-id*][**location instance**]
9. **show vlan trunks** [**brief**] [**location instance**] [{GigabitEthernet | TenGigE | Bundle-Ether | } *interface-path-id*] [**summary**]

### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>configure</b> <b>Example:</b> <pre>/CPU0:router# configure</pre>	Enters global configuration mode.
Step 2	<b>interface</b> {GigabitEthernet   TenGigE   Bundle-Ether} <i>interface-path-id.subinterface</i> <b>Example:</b> <pre>RP/0/RP0/CPU0:router(config)# interface TenGigE 0/2/0/4.10</pre>	Enters subinterface configuration mode and specifies the interface type, location, and subinterface number. <ul style="list-style-type: none"> <li>• Replace the <i>interface-path-id</i> argument with one of the following instances:</li> <li>• Physical Ethernet interface instance, or with an Ethernet bundle instance. Naming notation is <i>rack/slot/module/port</i>, and a slash between values is required as part of the notation.</li> <li>• Ethernet bundle instance. Range is from 1 through 65535.</li> <li>• Replace the <i>subinterface</i> argument with the subinterface value. Range is from 0 through 4095.</li> <li>• Naming notation is <i>interface-path-id.subinterface</i>, and a period between arguments is required as part of the notation.</li> </ul>
Step 3	<b>encapsulation dot1q</b> <b>Example:</b>	Sets the Layer 2 encapsulation of an interface.

	Command or Action	Purpose
	<pre>RP/0/RP0/CPU0:router(config-subif)# encapsulation dot1q 100, untagged</pre>	<p><b>Note</b></p> <ul style="list-style-type: none"> <li>The <b>dot1q vlan</b> command is replaced by the <b>encapsulation dot1q</b> command on the Cisco ASR 9000 Series Router. It is still available for backward-compatibility, but only for Layer 3 interfaces.</li> </ul>
<b>Step 4</b>	<p><b>ipv4 address</b> <i>ip-address mask</i></p> <p><b>Example:</b></p> <pre>RP/0/RP0/CPU0:router(config-subif)# ipv4 address 178.18.169.23/24</pre>	<p>Assigns an IP address and subnet mask to the subinterface.</p> <ul style="list-style-type: none"> <li>Replace <i>ip-address</i> with the primary IPv4 address for an interface.</li> <li>Replace <i>mask</i> with the mask for the associated IP subnet. The network mask can be specified in either of two ways: <ul style="list-style-type: none"> <li>The network mask can be a four-part dotted decimal address. For example, 255.0.0.0 indicates that each bit equal to 1 means that the corresponding address bit belongs to the network address.</li> <li>The network mask can be indicated as a slash (/) and number. For example, /8 indicates that the first 8 bits of the mask are ones, and the corresponding bits of the address are network address.</li> </ul> </li> </ul>
<b>Step 5</b>	<p><b>exit</b></p> <p><b>Example:</b></p> <pre>RP/0/RP0/CPU0:router(config-subif)# exit</pre>	<p>(Optional) Exits the subinterface configuration mode.</p> <ul style="list-style-type: none"> <li>The <b>exit</b> command is not explicitly required.</li> </ul>
<b>Step 6</b>	Repeat Step 2 through Step 5 to define the rest of the VLAN subinterfaces.	—
<b>Step 7</b>	<p><b>end</b> or <b>commit</b></p> <p><b>Example:</b></p> <pre>RP/0/RP0/CPU0:router(config)# end</pre> <p>OR</p> <pre>RP/0/RP0/CPU0:router(config)# commit</pre>	<p>Saves configuration changes.</p> <ul style="list-style-type: none"> <li>When you issue the <b>end</b> command, the system prompts you to commit changes: <pre>Uncommitted changes found, commit them before   exiting (yes/no/cancel)? [cancel]:</pre> <ul style="list-style-type: none"> <li>Entering <b>yes</b> saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</li> <li>Entering <b>no</b> exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</li> <li>Entering <b>cancel</b> leaves the router in the current configuration session without exiting or committing the configuration changes.</li> </ul> </li> </ul>

	Command or Action	Purpose
		<ul style="list-style-type: none"> <li>Use the <b>commit</b> command to save the configuration changes to the running configuration file and remain within the configuration session.</li> </ul>
<b>Step 8</b>	<b>show vlan interface</b> [ <i>type interface-path-id</i> ][ <b>location instance</b> ] <b>Example:</b> <pre>RP/0/RP0/CPU0:router# show vlan interface 5</pre>	(Optional) Displays the interface configuration. <ul style="list-style-type: none"> <li>To display the configuration for a particular port, use the <b>location</b> keyword.</li> <li>To display the configuration for the specified interface or subinterface, use the <b>interface</b> keyword.</li> </ul>
<b>Step 9</b>	<b>show vlan trunks</b> [ <b>brief</b> ] [ <b>location instance</b> ] [ <b>GigabitEthernet</b>   <b>TenGigE</b>   <b>Bundle-Ether</b>   } <i>interface-path-id</i> ] [ <b>summary</b> ] <b>Example:</b> <pre>RP/0/RP0/CPU0:router# show vlan trunk summary</pre>	(Optional) Displays summary information about each of the VLAN trunk interfaces. <ul style="list-style-type: none"> <li>The keywords have the following meanings:</li> <li><b>brief</b>—Displays a brief summary.</li> <li><b>summary</b>—Displays a full summary.</li> <li><b>location</b>—Displays information about the VLAN trunk interface on the given port.</li> <li><b>interface</b>—Displays information about the specified interface or subinterface.</li> </ul>

## Configuring Native VLAN

This task explains how to configure the native, or default, VLAN on an interface.

### SUMMARY STEPS

- configure**
- interface** {**GigabitEthernet** | **TenGigE** | **Bundle-Ether**} *interface-path-id*
- dot1q native vlan** *number*
- end** or **commit**
- show ethernet trunk bundle-ether** *instance* **show vlan trunks** [**brief**] [**location instance**] [**GigabitEthernet** | **TenGigE** | **Bundle-Ether**] *interface-path-id*] [**summary**]

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>configure</b> <b>Example:</b> <pre>RP/0/RP0/CPU0:router# configure</pre>	Enters global configuration mode.

	Command or Action	Purpose
<b>Step 2</b>	<p><b>interface</b> {<b>GigabitEthernet</b>   <b>TenGigE</b>   <b>Bundle-Ether</b>} <i>interface-path-id</i></p> <p><b>Example:</b></p> <pre>RP/0/RP0/CPU0:router(config)# interface TenGigE 0/2/0/4</pre>	<p>Enters interface configuration mode and specifies the Ethernet interface name and designation.</p> <p>Replace the <i>interface-path-id</i> argument with one of the following instances:</p> <ul style="list-style-type: none"> <li>Physical Ethernet interface instance or with an Ethernet bundle instance. Naming notation is <i>rack/slot/module/port</i>, and a slash between values is required as part of the notation.</li> <li>Ethernet bundle instance. Range is from 1 through 65535.</li> </ul>
<b>Step 3</b>	<p><b>dot1q native vlan</b> <i>number</i></p> <p><b>Example:</b></p> <pre>RP/0/RP0/CPU0:router(config-if)# dot1q native vlan 1</pre>	<p>Defines the default, or Native VLAN, associated with an 802.1Q trunk interface.</p> <ul style="list-style-type: none"> <li>The number argument is the ID of the trunk interface.</li> <li>Range is from 1 through 4094 inclusive (0 and 4095 are reserved).</li> </ul>
<b>Step 4</b>	<p><b>end</b> or <b>commit</b></p> <p><b>Example:</b></p> <pre>RP/0/RP0/CPU0:router(config-if)# end</pre> <p>or</p> <pre>RP/0/RP0/CPU0:router(config-if)# commit</pre>	<p>Saves configuration changes.</p> <ul style="list-style-type: none"> <li>When you issue the <b>end</b> command, the system prompts you to commit changes:</li> </ul> <pre>Uncommitted changes found, commit them before   exiting(yes/no/cancel)? [cancel]:</pre> <ul style="list-style-type: none"> <li>- Entering <b>yes</b> saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</li> <li>- Entering <b>no</b> exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</li> <li>- Entering <b>cancel</b> leaves the router in the current configuration session without exiting or committing the configuration changes.</li> </ul> <ul style="list-style-type: none"> <li>Use the <b>commit</b> command to save the configuration changes to the running configuration file and remain within the configuration session.</li> </ul>
<b>Step 5</b>	<p><b>show ethernet trunk bundle-ether</b> <i>instance</i> <b>show vlan trunks</b> [<b>brief</b>] [<b>location</b> <i>instance</i>] [{<b>GigabitEthernet</b>   <b>TenGigE</b>   <b>Bundle-Ether</b>} <i>interface-path-id</i>] [<b>summary</b>]</p> <p><b>Example:</b></p>	<p>(Optional) Displays summary information about each of the interface configuration.VLAN trunk interfaces.</p> <p>The Ethernet bundle instance range is from 1 through 65535.</p> <ul style="list-style-type: none"> <li>The keywords have the following meanings:</li> </ul>

	Command or Action	Purpose
	<pre>RP/0/RP0/CPU0:router# show ethernet vlan trunk bundle-ether 5summary</pre>	<ul style="list-style-type: none"> <li>• <b>brief</b>—Displays a brief summary.</li> <li>• <b>summary</b>—Displays a full summary.</li> <li>• <b>location</b>— Displays information about the VLAN trunk interface on the given port.</li> <li>• <b>interface</b>—Displays information about the specified interface or subinterface.</li> </ul>

## Configuring an Attachment Circuit on a VLAN

Use the following procedure to configure an attachment circuit on a VLAN.

### SUMMARY STEPS

1. **configure**
2. **interface** [**GigabitEthernet** | **TenGigE** | **Bundle-Ether** | **TenGigE**] *interface-path* *id.subinterface* **l2transport**
3. **dot1q vlan** *vlan-id* **encapsulation dot1q**
4. **l2protocol** {**cdp** | **pvst** | **stp** | **vtp**} {[**forward** | **tunnel**][**experimental bits**]}**drop**}
5. **end** or **commit**
6. **show interfaces** [**GigabitEthernet** | **TenGigE**] *interface-path-id.subinterface*

### DETAILED STEPS

	Command or Action	Purpose
<p><b>Step 1</b></p>	<p><b>configure</b></p> <p><b>Example:</b></p> <pre>RP/0/RP0/CPU0:router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p><b>Step 2</b></p>	<p><b>interface</b> [<b>GigabitEthernet</b>   <b>TenGigE</b>   <b>Bundle-Ether</b>   <b>TenGigE</b>] <i>interface-path</i> <i>id.subinterface</i> <b>l2transport</b></p> <p><b>Example:</b></p> <pre>RP/0/RP0/CPU0:router(config)# interface TenGigE 0/1/0/0.1 l2transport</pre>	<p>Enters subinterface configuration and specifies the interface type, location, and subinterface number.</p> <ul style="list-style-type: none"> <li>• Replace the <i>interface-path-id</i> argument with one of the following instances:</li> <li>• Physical Ethernet interface instance or Ethernet bundle instance. Naming notation is <i>rack/slot/module/port</i>, and a slash between values is required as part of the notation.</li> <li>• Ethernet bundle instance. Range is from 1 through 65535.</li> <li>• Replace the <i>subinterface</i> argument with the subinterface value. Range is from 0 through 4095.</li> </ul>

	Command or Action	Purpose
		<ul style="list-style-type: none"> <li>Naming notation is <i>instance.subinterface</i>, and a period between arguments is required as part of the notation.</li> <li>You must include the <b>I2transport</b> keyword in the command string; otherwise, the configuration creates a Layer 3 subinterface rather than an AC.</li> </ul>
<b>Step 3</b>	<p><b>dot1q vlan <i>vlan-id</i> encapsulation dot1q</b></p> <p><b>Example:</b></p> <pre>RP/0/RP0/CPU0:router(config-subif)# dot1q vlan 10 vlan any</pre>	<p>Assigns a VLAN AC to the subinterface.</p> <ul style="list-style-type: none"> <li>Replace the <i>vlan-id</i> argument with a subinterface identifier. Range is from 1 to 4094 inclusive (0 and 4095 are reserved). To configure a basic Dot1Q AC, use the following syntax:</li> </ul> <pre><b>dot1q vlan <i>vlan-id</i></b></pre> <ul style="list-style-type: none"> <li>To configure a Q-in-Q AC, use the following syntax:</li> </ul> <pre><b>dot1q vlan <i>vlan-id</i> vlan <i>vlan-id</i></b></pre> <ul style="list-style-type: none"> <li>To configure a Q-in-Any AC, use the following syntax:</li> </ul> <pre><b>dot1q vlan <i>vlan-id</i> vlan any</b></pre>
<b>Step 4</b>	<p><b>l2protocol {<i>cdp</i>   <i>pvst</i>   <i>stp</i>   <i>vtp</i>} {[<i>forward</i>   <i>tunnel</i>][<i>experimental bits</i>][<i>drop</i>]</b></p> <p><b>Example:</b></p> <pre>RP/0/RP0/CPU0:router(config-if-12)# l2protocol stp tunnel</pre>	<p>Configures Layer 2 protocol tunneling and protocol data unit (PDU) filtering on an interface.</p> <p>Possible protocols and options are:</p> <ul style="list-style-type: none"> <li><b>cdp</b>—Cisco Discovery Protocol (CDP) tunneling and data unit parameters.</li> <li><b>pvst</b>—Configures VLAN spanning tree protocol tunneling and data unit parameters.</li> <li><b>stp</b>—spanning tree protocol tunneling and data unit parameters.</li> <li><b>vtp</b>—VLAN trunk protocol tunneling and data unit parameters.</li> <li><b>tunnel</b>—(Optional) Tunnels the packets associated with the specified protocol.</li> <li><b>experimental <i>bits</i></b>—(Optional) Modifies the MPLS experimental bits for the specified protocol.</li> <li><b>drop</b>—(Optional) Drop packets associated with the specified protocol.</li> </ul>
<b>Step 5</b>	<p><b>end</b> or <b>commit</b></p> <p><b>Example:</b></p>	Saves configuration changes.

	Command or Action	Purpose
	<pre>RP/0/RP0/CPU0:router(config-if-12)# end or RP/0/RP0/CPU0:router(config-if-12)# commit</pre>	<ul style="list-style-type: none"> <li>When you issue the <b>end</b> command, the system prompts you to commit changes: <pre>Uncommitted changes found, commit them before   exiting(yes/no/cancel)? [cancel]:</pre> <ul style="list-style-type: none"> <li>Entering <b>yes</b> saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</li> <li>Entering <b>no</b> exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</li> <li>Entering <b>cancel</b> leaves the router in the current configuration session without exiting or committing the configuration changes.</li> </ul> </li> <li>Use the <b>commit</b> command to save the configuration changes to the running configuration file and remain within the configuration session.</li> </ul>
<p><b>Step 6</b></p>	<p><b>show interfaces</b> [<b>GigabitEthernet</b>   <b>TenGigE</b>] <i>interface-path-id.subinterface</i></p> <p><b>Example:</b></p> <pre>RP/0/RP0/CPU0:router# show interfaces TenGigE 0/3/0/0.1</pre>	<p>(Optional) Displays statistics for interfaces on the router.</p>

**What to do next**

- To configure a point-to-point pseudowire cross connect on the AC, see the “Implementing MPLS Layer 2 VPNs” VPNs module of the Cisco IOS XR Multiprotocol Label Switching Configuration Guide.
- To attach Layer 3 service policies, such as Multiprotocol Label Switching (MPLS) or Quality of Service (QoS), to the VLAN, refer to the appropriate Cisco IOS XR software configuration guide.

## Removing an 802.1Q VLAN Subinterface

This task explains how to remove 802.1Q VLAN subinterfaces that have been previously configured using the Configuring 802.1Q VLAN subinterfaces section in this module.

**SUMMARY STEPS**

- configure**
- no interface** {**GigabitEthernet** | **TenGigE** | **Bundle-Ether**} *interface-path-id.subinterface*
- Repeat Step 2 to remove other VLAN subinterfaces.
- end** or **commit**

5. **show vlan trunks** [**brief**] [**location** *instance*] [{**GigabitEthernet** | **TenGigE** | **Bundle-Ether** | **fastethernet**} *interface-path-id*] [**summary**]

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<p><b>configure</b></p> <p><b>Example:</b></p> <pre>/CPU0:router# configure</pre>	Enters global configuration mode.
<b>Step 2</b>	<p><b>no interface</b> {<b>GigabitEthernet</b>   <b>TenGigE</b>   <b>Bundle-Ether</b>} <i>interface-path-id.subinterface</i></p> <p><b>Example:</b></p> <pre>RP/0/RP0/CPU0:router(config)# no interface TenGigE 0/2/0/4.10</pre>	<p>Removes the subinterface, which also automatically deletes all the configuration applied to the subinterface.</p> <ul style="list-style-type: none"> <li>• Replace the <i>instance</i> argument with one of the following instances:</li> <li>• Physical Ethernet interface instance, or with an Ethernet bundle instance. Naming notation is <i>rack/slot/module/port</i>, and a slash between values is required as part of the notation.</li> <li>• Ethernet bundle instance. Range is from 1 through 65535.</li> <li>• Replace the <i>subinterface</i> argument with the subinterface value. Range is from 0 through 4095.</li> </ul> <p>Naming notation is <i>instance.subinterface</i>, and a period between arguments is required as part of the notation.</p>
<b>Step 3</b>	Repeat Step 2 to remove other VLAN subinterfaces.	—
<b>Step 4</b>	<p><b>end</b> or <b>commit</b></p> <p><b>Example:</b></p> <pre>RP/0/RP0/CPU0:router(config)# end</pre> <p>or</p> <pre>RP/0/RP0/CPU0:router(config)# commit</pre>	<p>Saves configuration changes.</p> <ul style="list-style-type: none"> <li>• When you issue the <b>end</b> command, the system prompts you to commit changes:</li> </ul> <pre>Uncommitted changes found, commit them before   exiting (yes/no/cancel)? [cancel]:</pre> <ul style="list-style-type: none"> <li>- Entering <b>yes</b> saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</li> <li>- Entering <b>no</b> exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</li> <li>- Entering <b>cancel</b> leaves the router in the current configuration session without exiting or committing the configuration changes.</li> </ul>

	Command or Action	Purpose
		<ul style="list-style-type: none"> <li>Use the <b>commit</b> command to save the configuration changes to the running configuration file and remain within the configuration session.</li> </ul>
<b>Step 5</b>	<p><b>show vlan trunks [brief] [location <i>instance</i>] [{GigabitEthernet   TenGigE   Bundle-Ether   fastethernet} <i>interface-path-id</i>] [summary]</b></p> <p><b>Example:</b></p> <pre>RP/0/RP0/CPU0:router# show vlan trunk summary</pre>	<p>(Optional) Displays summary information about each of the VLAN trunk interfaces.</p> <ul style="list-style-type: none"> <li>The keywords have the following meanings:</li> <li><b>brief</b>—Displays a brief summary.</li> <li><b>summary</b>—Displays a full summary.</li> <li><b>location</b>—Displays information about the VLAN trunk interface on the given port.</li> <li><b>interface</b>—Displays information about the specified interface or subinterface.</li> </ul>

## Configuration Examples for VLAN Interfaces

This section contains the following example:

### VLAN Subinterfaces: Example

The following example shows how to create three VLAN subinterfaces at one time:

```
RP/0/RP00/CPU0:router# configure
RP/0/RP00/CPU0:router(config)# interface TenGigE 0/2/0/4.1
RP/0/RP0/CPU0:router(config-subif)# dot1q vlan 10
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 100
RP/0/RP00/CPU0:router(config-subif)# ipv4 address 10.0.10.1/24
RP/0/RP00/CPU0:router(config-subif)# interface TenGigE0/2/0/4.2
RP/0/RP0/CPU0:router(config-subif)# dot1q vlan 20
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 101
RP/0/RP00/CPU0:router(config-subif)# ipv4 address 10.0.20.1/24
RP/0/RP00/CPU0:router(config-subif)# interface TenGigE0/2/0/4.3
RP/0/RP0/CPU0:router(config-subif)# dot1q vlan 30
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 102
RP/0/RP00/CPU0:router(config-subif)# ipv4 address 10.0.30.1/24
RP/0/RP00/CPU0:router(config-subif)# commit
RP/0/RP00/CPU0:router(config-subif)# exit
RP/0/RP00/CPU0:router(config)# exit

RP/0/RP00/CPU0:router# show vlan trunks summary
Trunk                               Sub types          Sub states
VLAN trunks: 1,
  1 are 802.1Q (Ether)
Sub-interfaces: 3,
```

```

3 are up.
802.1Q VLANs: 3,
3 have VLAN Ids,

```

```

RP/0/RP0/CPU0:router# show vlan interfaceinterface          encapsulation  vlan-id
intf-state
Te0/2/0/4.1          802.1Q          10  up
Te0/2/0/4.2          802.1Q          20  up
Te0/2/0/4.3          802.1Q          30  up
RP/0/RP0/CPU0:router# show vlan trunks briefinterface          encapsulations
intf-state
Summary              1000            0   1000    1000      0      0
Te0/2/0/4            802.1Q (Ether)  up

```

The following example shows how to create two VLAN subinterfaces on an Ethernet bundle:

```

RP/0/RP00/CPU0:router# configure
RP/0/RP00/CPU0:router(config)# interface bundle-ether 2
RP/0/RP00/CPU0:router(config-if)# ipv4 address 192.168.2.1/24
RP/0/RP00/CPU0:router(config-if)# exit
RP/0/RP00/CPU0:router(config)# interface bundle-ether 2.1
RP/0/RP00/CPU0:router(config-subif)# dot1q vlan 10
RP/0/RP00/CPU0:router(config-subif)# encapsulation dot1q 100
RP/0/RP00/CPU0:router(config-subif)# ipv4 address 192.168.100.1/24
RP/0/RP00/CPU0:router(config-subif)# exit
RP/0/RP00/CPU0:router(config)# interface bundle-ether 2.2
RP/0/RP00/CPU0:router(config-subif)# dot1q vlan 20
RP/0/RP00/CPU0:router(config-subif)# encapsulation dot1q 200
RP/0/RP00/CPU0:router(config-subif)# ipv4 address 192.168.200.1/24
RP/0/RP00/CPU0:router(config-subif)# exit
RP/0/RP00/CPU0:router(config)# commit

```

The following example shows how to create a basic dot1Q AC:

```

RP/0/RP00/CPU0:router# configure
RP/0/RP00/CPU0:router(config)# interface GigabitEthernet 0/0/0/0.1
RP/0/RP00/CPU0:router(config-subif)# l2transport
RP/0/RP00/CPU0:router(config-subif)# dot1q vlan 20
RP/0/RP00/CPU0:router(config-subif)# encapsulation dot1q 100
RP/0/RP00/CPU0:router(config-subif)# commit
RP/0/RP00/CPU0:router(config-subif)# exit
RP/0/RP00/CPU0:router(config)# exit

```

The following example shows how to create a Q-in-Q AC:

```

RP/0/RP00/CPU0:router# configure
RP/0/RP00/CPU0:router(config)# interface GigabitEthernet 0/0/0/0.2
RP/0/RP00/CPU0:router(config-subif)# l2transport
RP/0/RP00/CPU0:router(config-subif)# dot1q vlan 20 vlan 10
RP/0/RP00/CPU0:router(config-subif)# encapsulation dot1q 200 second-dot1q 201

```

```
RP/0/RP00/CPU0:router(config-subif)# commit  
RP/0/RP00/CPU0:router(config-subif)# exit  
RP/0/RP00/CPU0:router(config)# exit
```

The following example shows how to create a Q-in-Any AC:

```
RP/0/RP00/CPU0:router# configure  
RP/0/RP00/CPU0:router(config)# interface GigabitEthernet 0/0/0/0.3  
RP/0/RP00/CPU0:router(config-subif)# l2transport  
RP/0/0/CPU0:router(config-subif)# dot1q vlan 30 vlan any  
RP/0/RP00/CPU0:router(config-subif)# encapsulation dot1q 300 second-dot1q any  
RP/0/RP00/CPU0:router(config-subif)# commit  
RP/0/RP00/CPU0:router(config-subif)# exit  
RP/0/RP00/CPU0:router(config)# exit
```

