



Virtual LAN Commands

This section describes the commands used to configure virtual LANs in Layer 2 VPNs.



Note All commands applicable for the Cisco NCS 5500 Series Router are also supported on the Cisco NCS 540 Series Router that is introduced from Cisco IOS XR Release 6.3.2. References to earlier releases in Command History tables apply to only the Cisco NCS 5500 Series Router.



Note

- Starting with Cisco IOS XR Release 6.6.25, all commands applicable for the Cisco NCS 5500 Series Router are also supported on the Cisco NCS 560 Series Routers.
- Starting with Cisco IOS XR Release 6.3.2, all commands applicable for the Cisco NCS 5500 Series Router are also supported on the Cisco NCS 540 Series Router.
- References to releases before Cisco IOS XR Release 6.3.2 apply to only the Cisco NCS 5500 Series Router.
- Cisco IOS XR Software Release 7.0.1 specific updates are not applicable for the following variants of Cisco NCS 540 Series Routers:
 - N540-28Z4C-SYS-A
 - N540-28Z4C-SYS-D
 - N540X-16Z4G8Q2C-A
 - N540X-16Z4G8Q2C-D
 - N540-12Z20G-SYS-A
 - N540-12Z20G-SYS-D
 - N540X-12Z16G-SYS-A
 - N540X-12Z16G-SYS-D

For detailed information about concepts and configuration, see the Configure Virtual LANs in Layer 2 VPNs chapter in the *L2VPN and Ethernet Services Configuration Guide for Cisco NCS 5500 Series Routers* and *Ethernet Services Configuration Guide for Cisco NCS 540 Series Routers*.

- [encapsulation dot1q](#), on page 3
- [encapsulation dot1ad](#), on page 5
- [encapsulation dot1q second-dot1q](#), on page 7
- [encapsulation dot1ad dot1q](#), on page 9
- [rewrite ingress tag](#), on page 11
- [show interfaces](#), on page 13

encapsulation dot1q

To define the matching criteria to map 802.1Q frames ingress on an interface to the appropriate service instance, use the **encapsulation dot1q** command in the interface configuration mode. To delete the matching criteria to map 802.1Q frames ingress on an interface to the appropriate service instance, use the **no** form of this command.

encapsulation dot1q { any | *vlan-id* [*,vlan-id* [-*vlan-id*]] } **second-dot1q** *vlan-id*
no encapsulation dot1q { any | *vlan-id* [*,vlan-id* [-*vlan-id*]] } **second-dot1q** *vlan-id*

Syntax Description

vlan-id VLAN ID, can be given as single ID.

From Release 6.6.2 onwards, VLAN ID can be given as ranges also.

Command Default

No matching criteria are defined.

Command Modes

Interface configuration

Command History

Release	Modification
Release 6.0.1	This command was introduced.
Release 6.6.2	VLAN ID ranges are introduced for inner and outer VLAN tags.

Usage Guidelines

Only one encapsulation statement can be applied to a sub-interface. Encapsulation statements cannot be applied to main interfaces.

A single encapsulation dot1q statement specifies matching for frames with a single VLAN ID.

Examples

The following example shows how to map 802.1Q frames ingress on an interface to the appropriate service instance:

```
Router(config-if)# encapsulation dot1q 10
```

The following example shows how to map 802.1Q frames ingress on an l2transport sub-interface:

```
Router# configure
Router(config)# interface TenGigE 0/1/0/3.10 l2transport
Router(config-subif)# encapsulation dot1q 10
```

Related Commands

Command	Description
encapsulation dot1q second-dot1q, on page 7	Defines the matching criteria to map Q-in-Q ingress frames on an interface to the appropriate service instance.
encapsulation dot1ad, on page 5	Defines the matching criteria to map 802.1ad frames ingress on an interface to the appropriate service instance.

Command	Description
rewrite ingress tag, on page 11	Specifies the encapsulation adjustment that is to be performed on the frame ingress to the service instance.
dot1q tunneling ethertype	Configures the Ethertype, used by peer devices when implementing QinQ VLAN tagging, to be 0x9100.

encapsulation dot1ad

To define the matching criteria to map 802.1ad frames ingress on an interface to the appropriate service instance, use the **encapsulation dot1ad** command in the interface configuration mode. To delete the matching criteria to map 802.1ad frames ingress on an interface to the appropriate service instance, use the **no** form of this command.

```
encapsulation dot1ad vlan-id [{second-dot1ad vlan-id}]
no encapsulation dot1ad
```

Syntax Description

vlan-id VLAN ID, can be given as single ID.

Command Default

No matching criteria are defined.

Command Modes

Interface configuration

Command History

Release	Modification
Release 6.0.1	This command was introduced.

Usage Guidelines

Only one encapsulation statement can be applied to a sub-interface. Encapsulation statements cannot be applied to main interfaces.

A single encapsulation dot1ad statement specifies matching for frames with a single VLAN ID.

Examples

The following example shows how to map 802.1ad frames ingress on an interface to the appropriate service instance:

```
Router(config-if)# encapsulation dot1ad 10
```

The following example shows how to map 802.1ad frames ingress on an l2transport sub-interface:

```
Router# configure
Router(config)# interface TenGigE 0/1/0/3.10 l2transport
Router(config-subif)# encapsulation dot1ad 10
```

Related Commands

Command	Description
encapsulation dot1q second-dot1q, on page 7	Defines the matching criteria to map Q-in-Q ingress frames on an interface to the appropriate service instance.
encapsulation dot1q, on page 3	Defines the matching criteria to map 802.1Q frames ingress on an interface to the appropriate service instance.
rewrite ingress tag, on page 11	Specifies the encapsulation adjustment that is to be performed on the frame ingress to the service instance.

Command	Description
dot1q tunneling ethertype	Configures the Ethertype, used by peer devices when implementing QinQ VLAN tagging, to be 0x9100.

encapsulation dot1q second-dot1q

To define the matching criteria to map Q-in-Q ingress frames on an interface to the appropriate service instance, use the **encapsulation dot1q second-dot1q** command in the interface configuration mode. To remove the configuration, use the **no** form of this command.

encapsulation dot1q { any | *vlan-id* [*,vlan-id* [-*vlan-id*]] } **second-dot1q** *vlan-id* [*,vlan-id* [-*vlan-id*]]
no encapsulation dot1q { any | *vlan-id* [*,vlan-id* [-*vlan-id*]] } **second-dot1q** *vlan-id* [*,vlan-id* [-*vlan-id*]]

Syntax Description	<i>vlan-id</i>	VLAN ID, can be given as single ID. From Release 6.6.2 onwards, VLAN ID can be given as ranges also.
	second-dot1q	(Optional) Specifies IEEE 802.1Q VLAN tagged packets.

Command Default No matching criteria are defined.

Command Modes Interface configuration

Command History	Release	Modification
		Release 6.0.1
	Release 6.6.2	VLAN ID ranges are introduced for inner and outer VLAN tags.

Usage Guidelines

The following restrictions are applicable for this command:

- The outer tag must be unique and the inner tag may be a single VLAN.
- QinQ service instance, allows single or multiple on second-dot1q.
- Only one encapsulation command must be configured per service instance.
- Overlapping inner VLAN ranges are not supported.
- VLAN ID ranges cannot be used for both outer and inner tags, simultaneously.

For example:

encaps dot1q 10-20 second-dot1q 30-40, is not allowed.

But either **dot1q 10-20 second-dot1q 30** or **dot1q 10 second-dot1q 30-40** is allowed.

Examples

The following example shows how to map ingress frames to a service instance:

```
Router(config-if)# encapsulation dot1q 10 second-dot1q 20
```

The following example shows how to map ingress frames to a service instance, using VLAN ID ranges:

```
Router(config-if)# encapsulation dot1q 10-20 second-dot1q 30
```

Related Commands

Command	Description
encapsulation dot1q, on page 3	Defines the matching criteria to map 802.1Q frames ingress on an interface to the appropriate service instance.
encapsulation dot1ad, on page 5	Defines the matching criteria to map 802.1ad frames ingress on an interface to the appropriate service instance.
rewrite ingress tag, on page 11	Specifies the encapsulation adjustment that is to be performed on the frame ingress to the service instance.
dot1q tunneling ethertype	Configures the Ethertype, used by peer devices when implementing QinQ VLAN tagging, to be 0x9100.

encapsulation dot1ad dot1q

To define the matching criteria to be used in order to map single-tagged 802.1ad frames ingress on an interface to the appropriate service instance, use the **encapsulation dot1ad dot1q** command in sub-interface configuration mode. To remove the configuration, use the **no** form of this command.

```
encapsulation dot1ad vlan-id dot1q vlan-id
no encapsulation dot1ad vlan-id dot1q vlan-id
```

Syntax Description

dot1ad Indicates that the IEEE 802.1ad provider bridges encapsulation type is used for the outer tag.

dot1q Indicates that the IEEE 802.1q standard encapsulation type is used for the inner tag.

vlan-id VLAN ID, can be given as single ID.

Command Default

No matching criteria are defined.

Command Modes

Sub-interface configuration

Command History

Release	Modification
Release 6.0.1	This command was introduced.

Usage Guidelines

The outer VLAN tag is an 802.1ad VLAN tag, instead of an 802.1Q tag. An 802.1ad tag has an ethertype value of 0x88A8, instead of 0x8100 that 802.1Q uses.

Some of the fields in the 802.1ad VLAN header are interpreted differently per 802.1ad standard.

A **tunneling ethertype** command applied to the main interface does not apply to an 802.1ad sub-interface. An interface with encapsulation dot1ad causes the router to categorize the interface as an 802.1ad interface. This causes special processing for certain protocols and other features:

- MSTP uses the IEEE 802.1ad MAC STP address instead of the STP MAC address.
- Certain QoS functions may use the Drop Eligibility (DE) bit of the IEEE 802.1ad tag.

Examples

The following example shows how to map single-tagged 802.1ad ingress frames to a service instance:

```
Router(config-subif)# encapsulation dot1ad 100 dot1q 20
```

Related Commands

Command	Description
encapsulation dot1q second-dot1q, on page 7	Defines the matching criteria to map Q-in-Q ingress frames on an interface to the appropriate service instance.
encapsulation dot1ad, on page 5	Defines the matching criteria to map 802.1ad frames ingress on an interface to the appropriate service instance.
rewrite ingress tag, on page 11	Specifies the encapsulation adjustment that is to be performed on the frame ingress to the service instance.

Command	Description
dot1q tunneling ethertype	Configures the Ethertype, used by peer devices when implementing QinQ VLAN tagging, to be 0x9100.

rewrite ingress tag

To specify the encapsulation adjustment that is to be performed on the frame ingress to the service instance, use the **rewrite ingress tag** command in the interface configuration mode. To delete the encapsulation adjustment that is to be performed on the frame ingress to the service instance, use the **no** form of this command.

```
rewrite ingress tag {push {dot1q vlan-id} | pop {1} | translate {1-to-1 {dot1q vlan-id} | 1-to-2
{dot1q vlan-id } | 2-to-2 {dot1q vlan-id dot1q vlan-id} | 2-to-1 dot1q vlan-id}} [symmetric]
no rewrite ingress tag {push {dot1q vlan-id} | pop {1} | translate {1-to-1 {dot1q vlan-id} | 1-to-2
{dot1q vlan-id } | 2-to-2 {dot1q vlan-id dot1q vlan-id} | 2-to-1 dot1q vlan-id}} [symmetric]
```

Syntax Description		
<i>vlan-id</i>		VLAN ID, can be given as single ID.
push dot1q <i>vlan-id</i>		Pushes one 802.1Q tag with <i>vlan-id</i> .
pop {1}		One tag is removed from the packet. This command can be combined with a push (pop N and subsequent push <i>vlan-id</i>).
translate 1-to-1 dot1q <i>vlan-id</i>		Replaces the incoming tag (defined in the encapsulation command) into a different 802.1Q tag at the ingress service instance.
translate 1-to-2 dot1q <i>vlan-id</i> dot1q <i>vlan-id</i>		Replaces the incoming tag defined by the encapsulation command by a pair of 802.1Q tags.
translate 2-to-2 dot1q <i>vlan-id</i> second-dot1q <i>vlan-id</i>		Replaces the pair of tags defined by the encapsulation command by a pair of VLANs defined by this rewrite.
symmetric		(Optional) A rewrite operation is applied on both ingress and egress. The operation on egress is the inverse operation as ingress. Note Symmetric is the default behavior. Hence, it cannot be disabled.

Command Default The frame is left intact on ingress.

Command Modes Interface configuration

Command History	Release	Modification
	Release 6.0.1	This command was introduced.

Usage Guidelines The **symmetric** keyword is accepted only when a single VLAN is configured in encapsulation. If a list of VLANs is configured in encapsulation, the **symmetric** keyword is accepted only for push rewrite operations; all other rewrite operations are rejected.

The **pop** command assumes the elements being popped are defined by the encapsulation type.

The **rewrite ingress tag translate** command assume the tags being translated from are defined by the encapsulation type. In the 2-to-1 option, the “2” means 2 tags of a type defined by the **encapsulation** command.

The translation operation requires at least “from” tag in the original packet. If the original packet contains more tags than the ones defined in the “from”, then the operation should be done beginning on the outer tag.

Examples

The following example shows how to specify the encapsulation adjustment that is to be performed on the frame ingress to the service instance:

```
Router(config-if)# rewrite ingress tag push dot1q 200
```

Related Commands

Command	Description
encapsulation dot1q, on page 3	Defines the matching criteria to map 802.1Q frames ingress on an interface to the appropriate service instance.
encapsulation dot1ad, on page 5	Defines the matching criteria to map 802.1ad frames ingress on an interface to the appropriate service instance.
encapsulation dot1q second-dot1q, on page 7	Defines the matching criteria to map Q-in-Q ingress frames on an interface to the appropriate service instance.
encapsulation dot1ad dot1q, on page 9	Defines the matching criteria to be used in order to map single-tagged 802.1ad frames ingress on an interface to the appropriate service instance.
dot1q tunneling ethertype	Configures the Ethertype, used by peer devices when implementing QinQ VLAN tagging, to be 0x9100.

show interfaces

To display the operational information for Ethernet interfaces, use the **show interfaces** in the EXEC mode.

show interfaces *type interface-path-id*

Syntax Description	<i>type interface-path-id</i> Specifies the type of interface for which you want to display statistics.				
Command Default	None				
Command Modes	EXEC				
Command History	<table border="1"> <thead> <tr> <th>Release</th> <th>Modification</th> </tr> </thead> <tbody> <tr> <td>Release 6.5.1</td> <td>This command was introduced.</td> </tr> </tbody> </table>	Release	Modification	Release 6.5.1	This command was introduced.
Release	Modification				
Release 6.5.1	This command was introduced.				
Usage Guidelines	None				
Task ID	<table border="1"> <thead> <tr> <th>Task ID</th> <th>Operations</th> </tr> </thead> <tbody> <tr> <td>interface</td> <td>read</td> </tr> </tbody> </table>	Task ID	Operations	interface	read
Task ID	Operations				
interface	read				

Examples

This example shows the output from the **show interfaces** command:

```
Router# show interfaces GigabitEthernet 0/0/0/0
Wed May 23
08:07:39.869 PDT
GigabitEthernet0/0/0/0 is up, line protocol is up
  Interface state transitions: 1
  Hardware is GigabitEthernet, address is 02fe.08cb.26c5 (bia
02fe.08cb.26c5)
  Internet address is Unknown
  MTU 1518 bytes, BW 1000000 Kbit (Max: 1000000 Kbit)
    reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation 802.1Q Virtual LAN,
  vlan-switched trunk port: Dot1ad VLAN 1-100
  Full-duplex, 1000Mb/s, unknown, link type is force-up
  output flow control is off, input flow control is off
  Carrier delay (up) is 10 msec
  loopback not set,
  Last link flapped 00:05:32
  Last input never, output never
  Last clearing of "show interface" counters never
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    0 packets input, 0 bytes, 0 total input drops
    0 drops for unrecognized upper-level protocol
  Received 0 broadcast packets, 0 multicast packets
    0 runts, 0 giants, 0 throttles, 0 parity
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 packets output, 0 bytes, 0 total output drops
  Output 0 broadcast packets, 0 multicast packets
  0 output errors, 0 underruns, 0 applique, 0 resets
```

```
0 output buffer failures, 0 output buffers swapped out  
1 carrier transitions
```