

Configuring VLAN Identifiers for Bridge Domains and VPLS Routing Instances

You can configure VLAN identifiers for a bridge domain or a VPLS routing instance in the following ways:

- By using the `input-vlan-map` and the `output-vlan-map` statements at the [edit interfaces *interface-name*] or [edit logical-systems *logical-system-name* interfaces *interface-name*] hierarchy level to configure VLAN mapping. For information about configuring input and output VLAN maps to stack and rewrite VLAN tags in incoming or outgoing frames, see the *Junos Network Interfaces Configuration Guide*.
- By using either the `vlan-id` statement or the `vlan-tags` statement to configure a normalizing VLAN identifier. This topic describes how normalizing VLAN identifiers are processed and translated in a bridge domain or a VPLS routing instance.

The `vlan-id` and `vlan-tags` statements are used to specify the normalizing VLAN identifier under the bridge domain or VPLS routing instance. The normalizing VLAN identifier is used to perform the following functions:

- Translate, or normalize, the VLAN tags of received packets received into a learn VLAN identifier.
- Create multiple learning domains that each contain a learn VLAN identifier. A learning domain is a MAC address database to which MAC addresses are added based on the learn VLAN identifier.



NOTE: You cannot configure VLAN mapping using the `input-vlan-map` and `output-vlan-map` statements if you configure a normalizing VLAN identifier for a bridge domain or VPLS routing instance using the `vlan-id` or `vlan-tags` statements.

To configure a VLAN identifier for a bridge domain, include either the `vlan-id` or the `vlan-tags` statement at the [edit interfaces *interface-name* unit *logic-unit-number* family bridge] or [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logic-unit-number* family bridge] hierarchy level, and then include that logical interface in the bridge domain configuration. For more information about configuring a bridge domain, see *Configuring a Bridge Domain*.

For a VPLS routing instance, include either the `vlan-id` or `vlan-tags` statement at the [edit interfaces *interface-name* unit *logic-unit-number*] or [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logic-unit-number*] hierarchy level, and then include that logical interface in the VPLS routing instance configuration. For more information about configuring a VPLS routing instance, see the *Junos VPNs Configuration Guide*.



NOTE: For a single bridge domain or VPLS routing instance, you can include either the `vlan-id` or the `vlan-tags` statement, but not both.

The VLAN tags associated with the inbound logical interface are compared with the normalizing VLAN identifier. If the tags are different, they are rewritten as described in Table 1. The source MAC address of a received packet is learned based on the normalizing VLAN identifier.



NOTE: You do not have to specify a VLAN identifier for a bridge domain that is performing Layer 2 switching only. To support Layer 3 IP routing, you must specify either a VLAN identifier or a pair of VLAN tags. However, you cannot specify the same VLAN identifier for more than one bridge domain within a routing instance. Each bridge domain must have a unique VLAN identifier.

If the VLAN tags associated with the outbound logical interface and the normalizing VLAN identifier are different, the normalizing VLAN identifier is rewritten to match the VLAN tags of the outbound logical interface, as described in Table 2.

For the packets sent over the VPLS routing instance to be tagged by the normalizing VLAN identifier, include one of the following configuration statements:

- `vlan-id number` to tag all packets that are sent over the VPLS virtual tunnel (VT) interfaces with the VLAN identifier.
- `vlan-tags outer number inner number` to tag all packets sent over the VPLS VT interfaces with dual outer and inner VLAN tags.

Use the `vlan-id none` statement to have the VLAN tags removed from packets associated with an inbound logical interface when those packets are sent over VPLS VT interfaces. Note that those packets might still be sent with other customer VLAN tags.

The `vlan-id all` statement enables you to configure bridging for several VLANs with a minimum amount of configuration. Configuring this statement creates a learning domain for:

- Each inner VLAN, or learn VLAN, identifier of a logical interface configured with two VLAN tags
 - Each VLAN, or learn VLAN, identifier of a logical interface configured with one VLAN tag
-



NOTE: If you configure the `vlan-id all` statement in a VPLS routing instance, we recommend using the `input-vlan-map pop` and `output-vlan-map push` statements on the logical interface to pop the service VLAN ID on input and push the service VLAN ID on output and in this way limit the impact of doubly-tagged frames on scaling.

The `vlan-id-list [vlan-id-numbers]` statement enables you to configure bridging for multiple VLANs on a trunk interface. Configuring this statement creates a learning domain for:

- Each VLAN listed: `vlan-id-list [100 200 300]`
- Each VLAN in a range: `vlan-id-list [100-200]`
- Each VLAN in a list and range combination: `vlan-id-list [50, 100-200, 300]`

The following steps outline the process for bridging a packet received over a Layer 2 logical interface when you specify a normalizing VLAN identifier using either the `vlan-id number` or `vlan-tags` statement for a bridge domain or a VPLS routing instance:

1. When a packet is received on a physical port, it is accepted only if the VLAN identifier of the packet matches the VLAN identifier of one of the logical interfaces configured on that port.
2. The VLAN tags of the received packet are then compared with the normalizing VLAN identifier. If the VLAN tags of the packet are different from the normalizing VLAN identifier, the VLAN tags are rewritten as described in Table 1.
3. If the source MAC address of the received packet is not present in the source MAC table, it is learned based on the normalizing VLAN identifier.
4. The packet is then forwarded toward one or more outbound Layer 2 logical interfaces based on the destination MAC address. A packet with a known unicast destination MAC address is forwarded only to one outbound logical interface. For each outbound Layer 2 logical interface, the normalizing VLAN identifier configured for the bridge domain or VPLS routing instance is compared with the VLAN tags configured on that logical interface. If the VLAN tags associated with an outbound logical interface do not match the normalizing VLAN identifier configured for the bridge domain or VPLS routing instance, the VLAN tags are rewritten as described in Table 2.

The tables below show how VLAN tags are applied for traffic sent to and from the bridge domain, depending on how the `vlan-id` and `vlan-tags` statements are configured for the bridge domain and on how VLAN identifiers are configured for the logical interfaces in a bridge domain or VPLS routing instance. Depending on your configuration, the following rewrite operations are performed on VLAN tags:

- `pop`—Remove a VLAN tag from the top of the VLAN tag stack.
- `pop-pop`—Remove both the outer and inner VLAN tags of the frame.
- `pop-swap`—Remove the outer VLAN tag of the frame and replace the inner VLAN tag of the frame.
- `swap`—Replace the VLAN tag of the frame.
- `push`—Add a new VLAN tag to the top of the VLAN stack.
- `push-push`—Push two VLAN tags in front of the frame.
- `swap-push`—Replace the VLAN tag of the frame and add a new VLAN tag to the top of the VLAN stack.
- `swap-swap`—Replace both the outer and inner VLAN tags of the frame.

Table 1 shows specific examples of how the VLAN tags for packets sent to the bridge domain are processed and translated, depending on your configuration. “–” means that the statement is not supported for the specified logical interface VLAN identifier. “No operation” means that the VLAN tags of the received packet are not translated for the specified input logical interface.

Table 1: Statement Usage and Input Rewrite Operations for VLAN Identifiers for a Bridge Domain

VLAN Identifier of Logical Interface	VLAN Configurations for Bridge Domain			
	vlan-id none	vlan-id 200	vlan-id all	vlan tags outer 100 inner 300
none	No operation	push 200	–	push 100, push 300
200	pop 200	No operation	No operation	swap 200 to 300, push 100
1000	pop 1000	swap 1000 to 200	No operation	swap 1000 to 300, push 100
vlan-tags outer 2000 inner 300	pop 2000, pop 300	pop 2000, swap 300 to 200	pop 2000	swap 2000 to 100
vlan-tags outer 100 inner 400	pop 100, pop 400	pop 100, swap 400 to 200	pop 100	swap 400 to 300
vlan-id-range 10-100	–	–	No operation	–
vlan-tags outer 200 inner-range 10-100	–	–	pop 200	–

Table 2 shows specific examples of how the VLAN tags for packets sent from the bridge domain are processed and translated, depending on your configuration. “–” means that the statement is not supported for the specified logical interface VLAN identifier. “No operation” means that the VLAN tags of the outbound packet are not translated for the specified output logical interface.

Table 2: Statement Usage and Output Rewrite Operations for VLAN Identifiers for a Bridge Domain

VLAN Identifier of Logical Interface	VLAN Configurations for Bridge Domain			
	vlan-id none	vlan-id 200	vlan-id all	vlan tags outer 100 inner 300
none	no operation	pop 200	–	pop 100, pop 300
200	push 200	No operation	No operation	pop 100, swap 300 to 200
1000	push 1000	swap 200 to 1000	No operation	pop 100, swap 300 to 1000

Table 2: Statement Usage and Output Rewrite Operations for VLAN Identifiers for a Bridge Domain *(continued)*

VLAN Identifier of Logical Interface	VLAN Configurations for Bridge Domain			
	vlan-id none	vlan-id 200	vlan-id all	vlan tags outer 100 inner 300
vlan-tags outer 2000 inner 300	push 2000, push 300	swap 200 to 300, push 2000	push 2000	swap 100 to 2000
vlan-tags outer 100 inner 400	push 100, push 400	swap 200 to 400, push 100	push 100	swap 300 to 400
vlan-id-range 10-100	–	–	No operation	–
vlan-tags outer 200 inner-range 10-100	–	–	push 200	–

- Related Topics**
- Configuring Integrated Routing and Bridging for Bridge Domains
 - Layer 2 Learning and Forwarding Overview
 - Layer 2 Learning and Forwarding for Bridge Domains Overview
 - Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports

Published: 2010-05-11