JUNIPER NETWORKS | Engineering Simplicity

Ethernet Switching User Guide

Published
2020-06-16

JUNOS

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Ethernet Switching User Guide

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement (“EULA”) posted at https://support.juniper.net/support/eula/. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

About the Documentation

IN THIS SECTION

Use this guide to configure and monitor Layer 2 features.

Documentation and Release Notes

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at https://www.juniper.net/documentation/.

If the information in the latest release notes differs from the information in the documentation, follow the product Release Notes.

Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at https://www.juniper.net/books.

Using the Examples in This Manual

If you want to use the examples in this manual, you can use the **load merge** or the **load merge relative** command. These commands cause the software to merge the incoming configuration into the current candidate configuration. The example does not become active until you commit the candidate configuration.

If the example configuration contains the top level of the hierarchy (or multiple hierarchies), the example is a *full example*. In this case, use the **load merge** command.

If the example configuration does not start at the top level of the hierarchy, the example is a *snippet*. In this case, use the **load merge relative** command. These procedures are described in the following sections.

Merging a Full Example

To merge a full example, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration example into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following configuration to a file and name the file **ex-script.conf**. Copy the **ex-script.conf** file to the **/var/tmp** directory on your routing platform.

   ```
   system {
       scripts {
           commit {
               file ex-script.xsl;
           }
       }
   }
   interfaces {
       fxp0 {
           disable;
           unit 0 {
               family inet {
                   address 10.0.0.1/24;
               }
           }
       }
   }
   ```

2. Merge the contents of the file into your routing platform configuration by issuing the **load merge** configuration mode command:

   ```
   [edit]
   user@host# load merge /var/tmp/ex-script.conf
   load complete
   ```

Merging a Snippet

To merge a snippet, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration snippet into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following snippet to a file and name the file **ex-script-snippet.conf**. Copy the **ex-script-snippet.conf** file to the **/var/tmp** directory on your routing platform.

```
commit {
    file ex-script-snippet.xsl; }
```

2. Move to the hierarchy level that is relevant for this snippet by issuing the following configuration mode command:

```
[edit]
user@host# edit system scripts
[edit system scripts]
```

3. Merge the contents of the file into your routing platform configuration by issuing the **load merge relative** configuration mode command:

```
[edit system scripts]
user@host# load merge relative /var/tmp/ex-script-snippet.conf
load complete
```

For more information about the **load** command, see CLI Explorer.

Documentation Conventions

Table 1 on page xxxiv defines notice icons used in this guide.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.
	Tip	Indicates helpful information.
	Best practice	Alerts you to a recommended use or implementation.

Table 2 on page xxxiv defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the **configure** command: user@host> **configure**
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host>` **show chassis alarms** `No alarms currently active`
Italic text like this	• Introduces or emphasizes important new terms. • Identifies guide names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS CLI User Guide* • RFC 1997, *BGP Communities Attribute*

Table 2: Text and Syntax Conventions *(continued)*

Convention	Description	Examples
Italic text like this	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name** *domain-name*
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	• To configure a stub area, include the **stub** statement at the **[edit protocols ospf area area-id]** hierarchy level. • The console port is labeled **CONSOLE**.
< > (angle brackets)	Encloses optional keywords or variables.	**stub <default-metric *metric*>;**
| (pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	**broadcast | multicast** **(*string1* | *string2* | *string3*)**
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	**rsvp { # Required for dynamic MPLS only**
[ ] (square brackets)	Encloses a variable for which you can substitute one or more values.	**community name members [*community-ids*]**
Indention and braces ({ })	Identifies a level in the configuration hierarchy.	[edit] routing-options { static { route default { nexthop *address*; retain; } } }
; (semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
GUI Conventions		

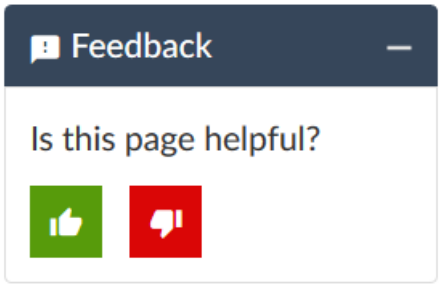
Table 2: Text and Syntax Conventions *(continued)*

Convention	Description	Examples
Bold text like this	Represents graphical user interface (GUI) items you click or select.	• In the Logical Interfaces box, select **All Interfaces**. • To cancel the configuration, click **Cancel**.
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select **Protocols>Ospf**.

Documentation Feedback

We encourage you to provide feedback so that we can improve our documentation. You can use either of the following methods:

- Online feedback system—Click TechLibrary Feedback, on the lower right of any page on the Juniper Networks TechLibrary site, and do one of the following:



  - Click the thumbs-up icon if the information on the page was helpful to you.
  - Click the thumbs-down icon if the information on the page was not helpful to you or if you have suggestions for improvement, and use the pop-up form to provide feedback.
- E-mail—Send your comments to techpubs-comments@juniper.net. Include the document or topic name, URL or page number, and software version (if applicable).

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active Juniper Care or Partner Support Services support contract, or are

covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at https://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf.
- Product warranties—For product warranty information, visit https://www.juniper.net/support/warranty/.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: https://www.juniper.net/customers/support/
- Search for known bugs: https://prsearch.juniper.net/
- Find product documentation: https://www.juniper.net/documentation/
- Find solutions and answer questions using our Knowledge Base: https://kb.juniper.net/
- Download the latest versions of software and review release notes: https://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: https://kb.juniper.net/InfoCenter/
- Join and participate in the Juniper Networks Community Forum: https://www.juniper.net/company/communities/
- Create a service request online: https://myjuniper.juniper.net

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: https://entitlementsearch.juniper.net/entitlementsearch/

Creating a Service Request with JTAC

You can create a service request with JTAC on the Web or by telephone.

- Visit https://myjuniper.juniper.net.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see https://support.juniper.net/support/requesting-support/.

1

CHAPTER

Understanding Layer 2 Networking

Layer 2 Networking

IN THIS SECTION

Overview of Layer 2 Networking

Layer 2, also known as the Data Link Layer, is the second level in the seven-layer OSI reference model for network protocol design. Layer 2 is equivalent to the link layer (the lowest layer) in the TCP/IP network model. Layer2 is the network layer used to transfer data between adjacent network nodes in a wide area network or between nodes on the same local area network.

A *frame* is a protocol data unit, the smallest unit of bits on a Layer 2 network. Frames are transmitted to and received from devices on the same local area network (LAN). Unilke bits, frames have a defined structure and can be used for error detection, control plane activities and so forth. Not all frames carry user data. The network uses some frames to control the data link itself..

At Layer 2, *unicast* refers to sending frames from one node to a single other node, whereas *multicast* denotes sending traffic from one node to multiple nodes, and *broadcasting* refers to the transmission of frames to all nodes in a network. A *broadcast domain* is a logical division of a network in which all nodes of that network can be reached at Layer 2 by a broadcast.

Segments of a LAN can be linked at the frame level using *bridges*. Bridging creates separate broadcast domains on the LAN, creating VLANs, which are independent logical networks that group together related devices into separate network segments. The grouping of devices on a VLAN is independent of where the devices are physically located in the LAN. Without bridging and VLANs, all devices on the Ethernet LAN are in a single broadcast domain, and all the devices detect all the packets on the LAN.

Forwarding is the relaying of packets from one network segment to another by nodes in the network. On a VLAN, a frame whose origin and destination are in the same VLAN are forwarded only within the local

VLAN. A network segment is a portion of a computer network wherein every device communicates using the same physical layer.

Layer 2 contains two sublayers:

- Logical link control (LLC) sublayer, which is responsible for managing communications links and handling frame traffic.
- Media access control (MAC) sublayer, which governs protocol access to the physical network medium. By using the MAC addresses that are assigned to all ports on a switch, multiple devices on the same physical link can uniquely identify one another.

  The ports, or interfaces, on a switch operate in either access mode, tagged-access, or trunk mode:

  - *Access mode* ports connect to a network device such as a desktop computer, an IP telephone, a printer, a file server, or a security camera. The port itself belongs to a single VLAN. The frames transmitted over an access interface are normal Ethernet frames. By default, all ports on a switch are in access mode.
  - *Tagged-Access mode* ports connect to a network device such as a desktop computer, an IP telephone, a printer, a file server, or a security camera. The port itself belongs to a single VLAN. The frames transmitted over an access interface are normal Ethernet frames. By default, all ports on a switch are in access mode. Tagged-access mode accommodates cloud computing, specifically scenarios including virtual machines or virtual computers. Because several virtual computers can be included on one physical server, the packets generated by one server can contain an aggregation of VLAN packets from different virtual machines on that server. To accommodate this situation, tagged-access mode reflects packets back to the physical server on the same downstream port when the destination address of the packet was learned on that downstream port. Packets are also reflected back to the physical server on the downstream port when the destination has not yet been learned. Therefore, the third interface mode, tagged access, has some characteristics of access mode and some characteristics of trunk mode:
  - *Trunk mode* ports handle traffic for multiple VLANs, multiplexing the traffic for all those VLANs over the same physical connection. Trunk interfaces are generally used to interconnect switches to other devices or switches.

    With native VLAN configured, frames that do not carry VLAN tags are sent over the trunk interface. If you have a situation where packets pass from a device to a switch in access mode, and you want to then send those packets from the switch over a trunk port, use native VLAN mode. Configure the single VLAN on the switch's port (which is in access mode) as a native VLAN. The switch's trunk port will then treat those frames differently than the other tagged packets. For example, if a trunk port has three VLANs, 10, 20, and 30, assigned to it with VLAN 10 being the native VLAN, frames on VLAN 10 that leave the trunk port on the other end have no 802.1Q header (tag). There is another native VLAN option. You can have the switch add and remove tags for untagged packets. To do this, you first configure the single VLAN as a native VLAN on a port attached to a device on the edge. Then, assign a VLAN ID tag to the single native VLAN on the port connected to a device. Last, add the VLAN

ID to the trunk port. Now, when the switch receives the untagged packet, it adds the ID you specified and sends and receives the tagged packets on the trunk port configured to accept that VLAN.

Including the sublayers, Layer 2 on the QFX Series supports the following functionality:

- Unicast, multicast, and broadcast traffic.
- Bridging.
- VLAN 802.1Q—Also known as *VLAN tagging*, this protocol allows multiple bridged networks to transparently share the same physical network link by adding VLAN tags to an Ethernet frame.
- Extension of Layer 2 VLANs across multiple switches using Spanning Tree Protocol (STP) prevents looping across the network.
- *MAC learning*, including per-VLAN MAC learning and Layer 2 learning suppression–This process obtains the MAC addresses of all the nodes on a network
- Link aggregation—This process groups of Ethernet interfaces at the physical layer to form a single link layer interface, also known as a *link aggregation group (LAG)* or LAG bundle

  NOTE: Link aggregation is not supported on NFX150 devices.

- Storm control on the physical port for unicast, multicast, and broadcast

  NOTE: Storm control is not supported on NFX150 devices.

- STP support, including 802.1d, RSTP, MSTP, and Root Guard

SEE ALSO

Understanding Bridging and VLANs on Switches | 168

Ethernet Switching and Layer 2 Transparent Mode Overview

Layer 2 transparent mode provides the ability to deploy the firewall without making changes to the existing routing infrastructure. The firewall is deployed as a Layer 2 switch with multiple VLAN segments and provides security services within VLAN segments. Secure wire is a special version of Layer 2 transparent mode that allows bump-in-wire deployment.

A device operates in transparent mode when there are interfaces defined as Layer 2 interfaces. The device operates in route mode (the default mode) if there are no physical interfaces configured as Layer 2 interfaces.

For SRX Series devices, transparent mode provides full security services for Layer 2 switching capabilities. On these SRX Series devices, you can configure one or more VLANs to perform Layer 2 switching. A VLAN is a set of logical interfaces that share the same flooding or broadcast characteristics. Like a virtual LAN (VLAN), a VLAN spans one or more ports of multiple devices. Thus, the SRX Series device can function as a Layer 2 switch with multiple VLANs that participate in the same Layer 2 network.

In transparent mode, the SRX Series device filters packets that traverse the device without modifying any of the source or destination information in the IP packet headers. Transparent mode is useful for protecting servers that mainly receive traffic from untrusted sources because there is no need to reconfigure the IP settings of routers or protected servers.

In transparent mode, all physical ports on the device are assigned to Layer 2 interfaces. Do not route Layer 3 traffic through the device. Layer 2 zones can be configured to host Layer 2 interfaces, and security policies can be defined between Layer 2 zones. When packets travel between Layer 2 zones, security policies can be enforced on these packets.

Table 3 on page 42 lists the security features that are supported and are not supported in transparent mode for Layer 2 switching.

Table 3: Security Features Supported in Transparent Mode

Mode Type	Supported	Not Supported
Transparent mode	• Application Layer Gateways (ALGs) • Firewall User Authentication (FWAUTH) • Intrusion Detection and Prevention (IDP) • Screen • AppSecure • Unified Threat Management (UTM)	• Network Address Translation (NAT) • VPN

NOTE: On SRX300, SRX320, SRX340, SRX345, and SRX550M devices, the DHCP server propagation is not supported in Layer 2 transparent mode.

In addition, the SRX Series devices do not support the following Layer 2 features in Layer 2 transparent mode:

- Spanning Tree Protocol (STP), RSTP, or MSTP—It is the user's responsibility to ensure that no flooding loops exist in the network topology.
- Internet Group Management Protocol (IGMP) snooping—Host-to-router signaling protocol for IPv4 used to report their multicast group memberships to neighboring routers and determine whether group members are present during IP multicasting.
- Double-tagged VLANs or IEEE 802.1Q VLAN identifiers encapsulated within 802.1Q packets (also called "Q in Q" VLAN tagging)—Only untagged or single-tagged VLAN identifiers are supported on SRX Series devices.
- Nonqualified VLAN learning, where only the MAC address is used for learning within the VLAN—VLAN learning on SRX Series devices is qualified; that is, both the VLAN identifier and MAC address are used.

Also, on SRX100, SRX110, SRX210, SRX220, SRX240, SRX300, SRX320, SRX340, SRX345, SRX550, or SRX650 devices, some features are not supported. (Platform support depends on the Junos OS release in your installation.) The following features are not supported for Layer 2 transparent mode on the mentioned devices:

- G-ARP on the Layer 2 interface
- IP address monitoring on any interface
- Transit traffic through IRB
- IRB interface in a routing instance
- IRB interface handling of Layer 3 traffic

> **NOTE:** The IRB interface is a pseudointerface and does not belong to the reth interface and redundancy group.

Layer 2 Transparent Mode on the SRX5000 Line Module Port Concentrator

The SRX5000 line Module Port Concentrator (SRX5K-MPC) supports Layer 2 transparent mode and processes the traffic when the SRX Series device is configured in Layer 2 transparent mode.

When the SRX5K-MPC is operating in Layer 2 mode, you can configure all interfaces on the SRX5K-MPC as Layer 2 switching ports to support Layer 2 traffic.

The security processing unit (SPU) supports all security services for Layer 2 switching functions, and the MPC delivers the ingress packets to the SPU and forwards the egress packets that are encapsulated by the SPU to the outgoing interfaces.

When the SRX Series device is configured in Layer 2 transparent mode, you can enable the interfaces on the MPC to work in Layer 2 mode by defining one or more logical units on a physical interface with the

family address type as **Ethernet switching**. Later you can proceed with configuring Layer 2 security zones and configuring security policies in transparent mode. Once this is done, next-hop topologies are set up to process ingress and egress packets.

Understanding IPv6 Flows in Transparent Mode on Security Devices

In transparent mode, the SRX Series device filters packets that traverse the device without modifying any of the source or destination information in the packet MAC headers. Transparent mode is useful for protecting servers that mainly receive traffic from untrusted sources because there is no need to reconfigure the IP settings of routers or protected servers.

A device operates in transparent mode when all physical interfaces on the device are configured as Layer 2 interfaces. A physical interface is a Layer 2 interface if its logical interface is configured with the **ethernet-switching** option at the [**edit interfaces *interface-name* unit *unit-number* family**] hierarchy level. There is no command to define or enable transparent mode on the device. The device operates in transparent mode when there are interfaces defined as Layer 2 interfaces. The device operates in route mode (the default mode) if all physical interfaces are configured as Layer 3 interfaces.

By default, IPv6 flows are dropped on security devices. To enable processing by security features such as zones, screens, and firewall policies, you must enable flow-based forwarding for IPv6 traffic with the **mode flow-based** configuration option at the [**edit security forwarding-options family inet6**] hierarchy level. You must reboot the device when you change the mode.

In transparent mode, you can configure Layer 2 zones to host Layer 2 interfaces, and you can define security policies between Layer 2 zones. When packets travel between Layer 2 zones, security policies can be enforced on these packets. The following security features are supported for IPv6 traffic in transparent mode:

- Layer 2 security zones and security policies. See “Understanding Layer 2 Security Zones” on page 979 and “Understanding Security Policies in Transparent Mode” on page 982.
- Firewall user authentication. See “Understanding Firewall User Authentication in Transparent Mode” on page 985.
- Layer 2 transparent mode chassis clusters.
- Class of service functions. See *Class of Service Functions in Transparent Mode Overview*.

The following security features are *not* supported for IPv6 flows in transparent mode:

- Logical systems
- IPv6 GTPv2
- J-Web interface
- NAT

- IPsec VPN
- With the exception of DNS, FTP, and TFTP ALGs, all other ALGs are not supported.

Configuring VLANs and Layer 2 logical interfaces for IPv6 flows is the same as configuring VLANs and Layer 2 logical interfaces for IPv4 flows. You can optionally configure an integrated routing and bridging (IRB) interface for management traffic in a VLAN. The IRB interface is the only Layer 3 interface allowed in transparent mode. The IRB interface on the SRX Series device does not support traffic forwarding or routing. The IRB interface can be configured with both IPv4 and IPv6 addresses. You can assign an IPv6 address for the IRB interface with the **address** configuration statement at the [**edit interfaces irb unit *number* family inet6**] hierarchy level. You can assign an IPv4 address for the IRB interface with the **address** configuration statement at the [**edit interfaces irb unit *number* family inet**] hierarchy level.

The Ethernet Switching functions on SRX Series devices are similar to the switching features on Juniper Networks MX Series routers. However, not all Layer 2 networking features supported on MX Series routers are supported on SRX Series devices. See "Ethernet Switching and Layer 2 Transparent Mode Overview" on page 41.

The SRX Series device maintains forwarding tables that contain MAC addresses and associated interfaces for each Layer 2 VLAN. The IPv6 flow processing is similar to IPv4 flows. See "Layer 2 Learning and Forwarding for VLANs Overview" on page 78.

Understanding Layer 2 Transparent Mode Chassis Clusters on Security Devices

A pair of SRX Series devices in Layer 2 transparent mode can be connected in a chassis cluster to provide network node redundancy. When configured in a chassis cluster, one node acts as the primary device and the other as the secondary device, ensuring stateful failover of processes and services in the event of system or hardware failure. If the primary device fails, the secondary device takes over processing of traffic.

> **NOTE:** If the primary device fails in a Layer 2 transparent mode chassis cluster, the physical ports in the failed device become inactive (go down) for a few seconds before they become active (come up) again.

To form a chassis cluster, a pair of the same kind of supported SRX Series devices combines to act as a single system that enforces the same overall security.

Devices in Layer 2 transparent mode can be deployed in active/backup and active/active chassis cluster configurations.

The following chassis cluster features are not supported for devices in Layer 2 transparent mode:

- Gratuitous ARP—The newly elected master in a redundancy group cannot send gratuitous ARP requests to notify network devices of a change in mastership on the redundant Ethernet interface links.
- IP address monitoring—Failure of an upstream device cannot be detected.

A redundancy group is a construct that includes a collection of objects on both nodes. A redundancy group is primary on one node and backup on the other. When a redundancy group is primary on a node, its objects on that node are active. When a redundancy group fails over, all its objects fail over together.

You can create one or more redundancy groups numbered 1 through 128 for an active/active chassis cluster configuration. Each redundancy group contains one or more redundant Ethernet interfaces. A redundant Ethernet interface is a pseudointerface that contains physical interfaces from each node of the cluster. The physical interfaces in a redundant Ethernet interface must be the same kind—either Fast Ethernet or Gigabit Ethernet. If a redundancy group is active on node 0, then the child links of all associated redundant Ethernet interfaces on node 0 are active. If the redundancy group fails over to the node 1, then the child links of all redundant Ethernet interfaces on node 1 become active.

NOTE: In the active/active chassis cluster configuration, the maximum number of redundancy groups is equal to the number of redundant Ethernet interfaces that you configure. In the active/backup chassis cluster configuration, the maximum number of redundancy groups supported is two.

Configuring redundant Ethernet interfaces on a device in Layer 2 transparent mode is similar to configuring redundant Ethernet interfaces on a device in Layer 3 route mode, with the following difference: the redundant Ethernet interface on a device in Layer 2 transparent mode is configured as a Layer 2 logical interface.

The redundant Ethernet interface may be configured as either an access interface (with a single VLAN ID assigned to untagged packets received on the interface) or as a trunk interface (with a list of VLAN IDs accepted on the interface and, optionally, a native-vlan-id for untagged packets received on the interface). Physical interfaces (one from each node in the chassis cluster) are bound as child interfaces to the parent redundant Ethernet interface.

In Layer 2 transparent mode, MAC learning is based on the redundant Ethernet interface. The MAC table is synchronized across redundant Ethernet interfaces and Services Processing Units (SPUs) between the pair of chassis cluster devices.

The IRB interface is used only for management traffic, and it cannot be assigned to any redundant Ethernet interface or redundancy group.

All Junos OS screen options that are available for a single, nonclustered device are available for devices in Layer 2 transparent mode chassis clusters.

NOTE: Spanning Tree Protocols (STPs) are not supported for Layer 2 transparent mode. You must ensure that there are no loop connections in the deployment topology.

Configuring Out-of-Band Management on SRX Devices

You can configure the **fxp0** out-of-band management interface on the SRX Series device as a Layer 3 interface, even if Layer 2 interfaces are defined on the device. With the exception of the **fxp0** interface, you can define Layer 2 and Layer 3 interfaces on the device's network ports.

> **NOTE:** There is no fxp0 out-of-band management interface on the SRX300, SRX320, and SRX550M devices. (Platform support depends on the Junos OS release in your installation.)

Ethernet Switching

Ethernet switching forwards the Ethernet frames within or across the LAN segment (or VLAN) using the Ethernet MAC address information. Ethernet switching on the SRX1500 device is performed in the hardware using ASICs.

Starting in Junos OS Release 15.1X49-D40, use the **set protocols l2-learning global-mode(transparent-bridge | switching)** command to switch between the Layer 2 transparent bridge mode and Ethernet switching mode. After switching the mode, you must reboot the device for the configuration to take effect. Table 4 on page 47 describes the default Layer 2 global mode on SRX Series devices.

Table 4: Default Layer 2 Global Mode on SRX Series Devices

Junos OS Release	Platforms	Default Layer 2 Global Mode	Details
Prior to Junos OS Release 15.1X49-D50 and Junos OS Release 17.3R1 onwards	SRX300, SRX320, SRX340, and SRX345	Switching mode	None
Junos OS Release 15.1X49-D50 to Junos OS Release 15.1X49-D90	SRX300, SRX320, SRX340, and SRX345	Switching mode	When you delete the Layer 2 global mode configuration on a device, the device is in transparent bridge mode.

Table 4: Default Layer 2 Global Mode on SRX Series Devices *(continued)*

Junos OS Release	Platforms	Default Layer 2 Global Mode	Details
Junos OS Release 15.1X49-D100 onwards	SRX300, SRX320, SRX340, SRX345, SRX550, and SRX550M	Switching mode	When you delete the Layer 2 global mode configuration on a device, the device is in switching mode. Configure the **set protocols l2-learning global-mode transparent-bridge** command under the **[edit]** hierarchy level to switch to transparent bridge mode. Reboot the device for the configuration to take effect.
Junos OS Release 15.1X49-D50 onwards	SRX1500	Transparent bridge mode	None

The Layer 2 protocol supported in switching mode is Link Aggregation Control Protocol (LACP).

You can configure Layer 2 transparent mode on a redundant Ethernet interface. Use the following commands to define a redundant Ethernet interface:

- **set interfaces *interface-name* ether-options redundant-parent *reth-interface-name***
- **set interfaces *reth-interface-name* redundant-ether-options redundancy-group *number***

Layer 2 Switching Exceptions on SRX Series Devices

The switching functions on the SRX Series devices are similar to the switching features on Juniper Networks MX Series routers. However, the following Layer 2 networking features on MX Series routers are not supported on SRX Series devices:

- Layer 2 control protocols—These protocols are used on MX Series routers for Rapid Spanning Tree Protocol (RSTP) or Multiple Spanning Tree Protocol (MSTP) in customer edge interfaces of a VPLS routing instance.
- Virtual switch routing instance—The virtual switching routing instance is used on MX Series routers to group one or more VLANs.
- Virtual private LAN services (VPLS) routing instance—The VPLS routing instance is used on MX Series routers for point-to-multipoint LAN implementations between a set of sites in a VPN.

SEE ALSO

global-mode (Protocols) | 1184

l2-learning | 1229

Understanding Unicast

Unicasting is the act of sending data from one node of the network to another. In contrast, multicast transmissions send traffic from one data node to multiple other data nodes.

Unknown unicast traffic consists of unicast frames with unknown destination MAC addresses. By default, the switch floods these unicast frames that are traveling in a VLAN to all interfaces that are members of the VLAN. Forwarding this type of traffic to interfaces on the switch can trigger a security issue. The LAN is suddenly flooded with packets, creating unnecessary traffic that leads to poor network performance or even a complete loss of network service. This is known as a traffic storm.

To prevent a storm, you can disable the flooding of unknown unicast packets to all interfaces by configuring one VLAN or all VLANs to forward any unknown unicast traffic to a specific trunk interface. (This channels the unknown unicast traffic to a single interface.)

SEE ALSO

Understanding Bridging and VLANs on Switches | 168

Understanding Layer 2 Broadcasting on Switches

In a Layer 2 network, *broadcasting* refers to sending traffic to all nodes on a network.

Layer 2 broadcast traffic stays within a local area network (LAN) boundary; known as the *broadcast domain*. Layer 2 broadcast traffic is sent to the broadcast domain using a MAC address of FF:FF:FF:FF:FF:FF. Every device in the broadcast domain recognizes this MAC address and passes the broadcast traffic on to other devices in the broadcast domain, if applicable. Broadcasting can be compared to unicasting (sending traffic to a single node) or multicasting (delivering traffic to a group of nodes simultaneously).

Layer 3 broadcast traffic, however, is sent to all devices in a network using a broadcast network address. For example, if your network address is 10.0.0.0, the broadcast network address is 10.255.255.255. In this case, only devices that belong to the 10.0.0.0 network receive the Layer 3 broadcast traffic. Devices that do not belong to this network drop the traffic.

Broadcasting is used in the following situations:

- Address Resolution Protocol (ARP) uses broadcasting to map MAC addresses to IP addresses. ARP dynamically binds the IP address (the logical address) to the correct MAC address. Before IP unicast packets can be sent, ARP discovers the MAC address used by the Ethernet interface where the IP address is configured.
- Dynamic Host Configuration Protocol (DHCP) uses broadcasting to dynamically assign IP addresses to hosts on a network segment or subnet.
- Routing protocols use broadcasting to advertise routes.

Excessive broadcast traffic can sometimes create a broadcast storm. A broadcast storm occurs when messages are broadcast on a network and each message prompts a receiving node to respond by broadcasting its own messages on the network. This, in turn, prompts further responses that create a snowball effect. The LAN is suddenly flooded with packets, creating unnecessary traffic that leads to poor network performance or even a complete loss of network service.

SEE ALSO

Understanding Bridging and VLANs on Switches | 168

Using the Enhanced Layer 2 Software CLI

IN THIS SECTION

Enhanced Layer 2 Software (ELS) provides a uniform CLI for configuring and monitoring Layer 2 features on QFX Series switches, EX Series switches, and other Juniper Networks devices, such as MX Series routers. With ELS, you configure Layer 2 features in the same way on all these Juniper Networks devices.

This topic explains how to know if your platform is running ELS. It also explains how to perform some common tasks using the ELS style of configuration.

Understanding Which Devices Support ELS

ELS is automatically supported if your device is running a Junos OS release that supports it. You do not need to take any action to enable ELS, and you cannot disable ELS. See Feature Explorer for information about which platforms and releases support ELS.

Understanding How to Configure Layer 2 Features Using ELS

IN THIS SECTION

Because ELS provides a uniform CLI, you can now perform the following tasks on supported devices in the same way:

Configuring a VLAN

You can configure one or more VLANs to perform Layer 2 bridging. The Layer 2 bridging functions include integrated routing and bridging (IRB) for support for Layer 2 bridging and Layer 3 IP routing on the same interface. EX Series and QFX Series switches can function as Layer 2 switches, each with multiple bridging, or broadcast, domains that participate in the same Layer 2 network. You can also configure Layer 3 routing support for a VLAN.

To configure a VLAN:

1. Create the VLAN by setting a unique VLAN name and configuring the VLAN ID:

   ```
   [edit]
   user@host#  set vlans vlan-name vlan-id vlan-id-number
   ```

   Using the VLAN ID list option, you can optionally specify a range of VLAN IDs.

   ```
   [edit]
   user@host#  set vlans vlan-name vlan-id-list vlan-ids | vlan-id--vlan-id
   ```

2. Assign at least one interface to the VLAN:

```
[edit]
user@host# set interface interface-name family ethernet-switching vlan members vlan-name
```

Configuring the Native VLAN Identifier

EX Series and QFX Series switches support receiving and forwarding routed or bridged Ethernet frames with 802.1Q VLAN tags. Typically, trunk ports, which connect switches to each other, accept untagged control packets, but do not accept untagged data packets. You can enable a trunk port to accept untagged data packets by configuring a native VLAN ID on the interface on which you want the untagged data packets to be received.

To configure the native VLAN ID:

1. On the interface on which you want untagged data packets to be received, set the interface mode to **trunk**, which specifies that the interface is in multiple VLANs and can multiplex traffic between different VLANs.

   ```
   [edit interfaces]
   user@host# set interface-name unit logical-unit-number family ethernet-switching interface-mode trunk
   ```

2. Configure the native VLAN ID and assign the interface to the native VLAN ID:

   ```
   [edit interfaces]
   user@host# set interface-name native-vlan-id number
   ```

3. Assign the interface to the native VLAN ID:

   ```
   [edit interfaces]
   user@host# set interface-name unit logical-unit-number family ethernet-switching vlan members native-vlan-id-number
   ```

Configuring Layer 2 Interfaces

To ensure that your high-traffic network is tuned for optimal performance, explicitly configure some settings on the switch's network interfaces.

To configure a Gigabit Ethernet interface or a 10-Gigabit Ethernet interface as a **trunk** interface:

```
[edit]
user@host# set interfaces interface-name unit logical-unit-number family ethernet-switching interface-mode trunk
```

To configure a Gigabit Ethernet interface or a 10-Gigabit Ethernet interface as a **access** interface:

```
[edit]
```

`user@host#` **set interfaces *interface-name* unit *logical-unit-number* family ethernet-switching interface-mode access**

To assign an interface to VLAN:

```
[edit interfaces]
user@host# set interface-name unit logical-unit-number family ethernet-switching vlan members [all | vlan-names | vlan-ids]
```

Configuring Layer 3 Interfaces

To configure a Layer 3 interface, you must assign an IP address to the interface. You assign an address to an interface by specifying the address when you configure the protocol family. For the **inet** or **inet6** family, configure the interface IP address.

You can configure interfaces with a 32-bit IP version 4 (IPv4) address and optionally with a destination prefix, sometimes called a subnet mask. An IPv4 address utilizes a 4-octet dotted decimal address syntax (for example, 192.168.1.1). An IPv4 address with destination prefix utilizes a 4-octet dotted decimal address syntax with a destination prefix appended (for example, 192.168.1.1/16).

To specify an IP4 address for the logical unit:

```
[edit]
user@host# set interfaces interface-name unit logical-unit-number family inet address ip-address
```

You represent IP version 6 (IPv6) addresses in hexadecimal notation by using a colon-separated list of 16-bit values. You assign a 128-bit IPv6 address to an interface.

To specify an IP6 address for the logical unit:

```
[edit]
user@host# set interfaces interface-name unit logical-unit-number family inet6 address ip-address
```

Configuring an IRB Interface

Integrated routing and bridging (IRB) provides support for Layer 2 bridging and Layer 3 IP routing on the same interface. IRB enables you to route packets to another routed interface or to another VLAN that has a Layer 3 protocol configured. IRB interfaces enable the device to recognize packets that are being sent to local addresses so that they are bridged (switched) whenever possible and are routed only when necessary. Whenever packets can be switched instead of routed, several layers of processing are eliminated. An interface named irb functions as a logical router on which you can configure a Layer 3 logical interface for VLAN. For redundancy, you can combine an IRB interface with implementations of the Virtual Router Redundancy Protocol (VRRP) in both bridging and virtual private LAN service (VPLS) environments.

To configure an IRB interface:

1. Create a Layer 2 VLAN by assigning it a name and a VLAN ID:

   ```
   [edit]
   user@host#  set vlans vlan-name vlan-id vlan-id
   ```

2. Create an IRB logical interface:

   ```
   [edit]
   user@host#  set interface irb unit logical-unit-number family inet address ip-address
   ```

3. Associate the IRB interface with the VLAN:

   ```
   [edit]
   user@host#  set vlans vlan-name l3-interface irb.logical-unit-number
   ```

Configuring an Aggregated Ethernet Interface and Configuring LACP on That Interface

Use the link aggregation feature to aggregate one or more links to form a virtual link or link aggregation group (LAG). The MAC client can treat this virtual link as if it were a single link to increase bandwidth, provide graceful degradation as failure occurs, and increase availability.

To configure an aggregated Ethernet interface:

1. Specify the number of aggregated Ethernet interfaces to be created:

   ```
   [edit chassis]
   user@host#  set aggregated-devices ethernet device-count number
   ```

2. Specify the name of the link aggregation group interface:

   ```
   [edit]
   user@host#  set interfaces aex
   ```

3. Specify the minimum number of links for the aggregated Ethernet interface (aex)– that is, the defined bundle– to be labeled *up*:

   ```
   [edit interfaces]
   user@host#  set aex aggregated-ether-options minimum-links number
   ```

4. Specify the link speed for the aggregated Ethernet bundle:

   ```
   [edit interfaces]
   user@host#  set aex aggregated-ether-options link-speed link-speed
   ```

5. Specify the members to be included within the aggregated Ethernet bundle:

```
[edit interfaces]
user@host# set interface-name ether-options 802.3ad aex
user@host# set interface-name ether-options 802.3ad aex
```

6. Specify an interface family for the aggregated Ethernet bundle:

```
[edit interfaces]
user@host# set aex unit 0 family inet address ip-address
```

For aggregated Ethernet interfaces on the device, you can configure the Link Aggregation Control Protocol (LACP). LACP bundles several physical interfaces to form one logical interface. You can configure aggregated Ethernet with or without LACP enabled.

When LACP is enabled, the local and remote sides of the aggregated Ethernet links exchange protocol data units (PDUs), containing information about the state of the link. You can configure Ethernet links to actively transmit PDUs, or you can configure the links to passively transmit them, sending out LACP PDUs only when they receive them from another link. One side of the link must be configured as active for the link to be up.

To configure LACP:

1. Enable one side of the aggregated Ethernet link as active:

```
[edit interfaces]
user@host# set aex aggregated-ether-options lacp active
```

2. Specify the interval at which the interfaces send LACP packets:

```
[edit interfaces]
user@host# set aex aggregated-ether-options lacp periodic interval
```

Understanding ELS Configuration Statement and Command Changes

IN THIS SECTION

ELS was introduced in Junos OS Release 12.3R2 for EX9200 switches. ELS changes the CLI for some of the Layer 2 features on supported EX Series and QFX Series switches.

The following sections provide a list of existing commands that were moved to new hierarchy levels or changed on EX Series switches as part of this CLI enhancement effort. These sections are provided as a high-level reference only. For detailed information about these commands, use the links to the configuration statements provided or see the technical documentation.

Changes to the ethernet-switching-options Hierarchy Level

This section outlines the changes to the **ethernet-switching-options** hierarchy level.

> **NOTE:** The **ethernet-switching-options** hierarchy level has been renamed as **switch-options**.

Table 5: Renaming the ethernet-switching-options hierarchy

Original Hierarchy	Changed Hierarchy
ethernet-switching-options { authentication-whitelist { ... } }	**switch-options** { ... authentication-whitelist { ... } }
ethernet-switching-options { interfaces *interface-name* { no-mac-learning; ... } }	switch-options { interfaces *interface-name* { no-mac-learning; ... } }

Table 5: Renaming the ethernet-switching-options hierarchy *(continued)*

Original Hierarchy	Changed Hierarchy
ethernet-switching-options { **unknown-unicast-forwarding** { (...) } }	switch-options { **unknown-unicast-forwarding** { (...) } }
ethernet-switching-options { **voip** { **interface** (all | [*interface-name* | access-ports]) { **forwarding-class** (assured-forwarding | best-effort | expedited-forwarding | network-control); **vlan** *vlan-name*; ... } } }	**switch-options** { **voip** { **interface** (all | [*interface-name* | access-ports]) { **forwarding-class** (assured-forwarding | best-effort | expedited-forwarding | network-control); **vlan** *vlan-name*; ... } } }

Table 6: RTG Statements

Original Hierarchy	Changed Hierarchy
ethernet-switching-options { redundant-trunk-group { group *name* { **description**; **interface** *interface-name* { **primary**; } preempt-cutover-timer *seconds*; ... } } }	switch-options { redundant-trunk-group { **group *name*** { **description**; **interface *interface-name*** { **primary**; } **preempt-cutover-timer** *seconds*; ... } } }

Table 7: Deleted Statements

Original Hierarchy	Changed Hierarchy
ethernet-switching-options { **mac-notification** { notification-interval *seconds*; ... } }	The statements have been removed from the **switch-options** hierarchy.
ethernet-switching-options { **traceoptions** { **file** *filename* <files *number*> <no-stamp> <replace> <size *size*> <world-readable | no-world-readable>; **flag** *flag* <disable>; ... } }	The statements have been removed from the **switch-options** hierarchy.
ethernet-switching-options { **port-error-disable** { **disable-timeout** ***timeout***; ... } }	**NOTE:** The **port-error-disable** statement has been replaced with a new statement. **interfaces** *interface-name* family **ethernet-switching** { **recovery-timeout** *seconds*; }

Changes to the Port Mirroring Hierarchy Level

> **NOTE:** Statements have moved from the **ethernet-switching-options** hierarchy level to the **forwarding-options** hierarchy level.

Table 8: Port Mirroring hierarchy

Original Hierarchy	Changed Hierarchy
ethernet-switching-options { **analyzer (Port Mirroring)** { *name* { ... } } }	forwarding-options { **analyzer (Port Mirroring)** { *name* { ... } } }

Changes to the Layer 2 Control Protocol Hierarchy Level

The Layer 2 control protocol statements have moved from the **ethernet-switching-options** hierarchy to the **protocols** hierarchy.

Table 9: Layer 2 Control Protocol

Original Hierarchy	Changed Hierarchy
ethernet-switching-options { bpdu-block { ... } }	protocols { layer2-control { **bpdu-block** { ... } } }

Changes to the dot1q-tunneling Statement

The **dot1q-tunneling** statement has been replaced with a new statement and moved to a different hierarchy level.

Table 10: dot1q-tunneling

Original Hierarchy	Changed Hierarchy
ethernet-switching-options { **dot1q-tunneling** { **ether-type** (0x8100 | 0x88a8 | 0x9100); ... } }	interfaces *interface-name* { ether-options { ethernet-switch-profile { tag-protocol-id [*tpids*]; } } } interfaces *interface-name* { aggregated-ether-options { ethernet-switch-profile { tag-protocol-id [*tpids*]; } } }

Changes to the L2 Learning Protocol

The **mac-table-aging-time** statement has been replaced with a new statement and moved to a different hierarchy level.

Table 11: mac-table-aging-time statement

Original Hierarchy	Changed Hierarchy
ethernet-switching-options { mac-table-aging-time *seconds*; ... }	protocols { l2-learning { global-mac-table-aging-time *seconds*; ... } }

Changes to Nonstop Bridging

The **nonstop-bridging** statement has moved to a different hierarchy level.

Table 12: Nonstop Bridging statement

Original Hierarchy	Changed Hierarchy
ethernet-switching-options { **nonstop-bridging**; }	protocols { layer2-control { **nonstop-bridging** { } } }

Changes to Port Security and DHCP Snooping

Port security and DHCP snooping statements have moved to different hierarchy levels.

> **NOTE:** The statement **examine-dhcp** does not exist in the changed hierarchy. DHCP snooping is now enabled automatically when other DHCP security features are enabled on a VLAN. See *Configuring Port Security (ELS)* for additional information.

Table 13: Port Security statements

Original Hierarchy	Changed Hierarchy
ethernet-switching-options { **secure-access-port** { interface (all | *interface-name*) { (**dhcp-trusted** | no-dhcp-trusted); **static-ip** *ip-address* { mac *mac-address*; vlan *vlan-name*; } } vlan (all | *vlan-name*) { (**arp-inspection** | no-arp-inspection); **dhcp-option82** { disable; circuit-id { prefix *hostname*; use-interface-description; use-vlan-id; } remote-id { prefix (*hostname* | mac | none); use-interface-description; use-string *string*; } vendor-id [*string*]; } (**examine-dhcp** | no-examine-dhcp); } (**ip-source-guard** | no-ip-source-guard); } }	vlans *vlan-name* forwarding-options{ **dhcp-security** { **arp-inspection;** **group *group-name*** { **interface***interface-name* { **static-ip** *ip-address* { mac *mac-address*; } } **overrides** { **no-option82;** **trusted**; } } **ip-source-guard**; **no-dhcp-snooping**; **option-82** { circuit-id { prefix { host-name; routing-instance-name; } use-interface-description (device | logical); use-vlan-id; } remote-id { host-name; use-interface-description (device | logical); use-string *string*; } vendor-id { use-string *string*; } } }

TIP: For allowed mac configuration, the original hierarchy statement **set ethernet-switching-options secure-access-port interface ge-0/0/2 allowed-mac 00:05:85:3A:82:8** is replaced by the ELS command **set interfaces ge-0/0/2 unit 0 accept-source-mac mac-address 00:05:85:3A:82:8**

NOTE: DHCP snooping statements have moved to a different hierarchy level.

Table 14: DHCP Snooping Statements

Original Hierarchy	Changed Hierarchy
ethernet-switching-options { **secure-access-port** { **dhcp-snooping-file** { location *local_pathname* | *remote_URL*; timeout *seconds*; write-interval *seconds*; }	system [processes [**dhcp-service** **dhcp-snooping-file** *local_pathname* | *remote_URL*; write-interval *interval*; } }

Changes to Configuring VLANs

The statements for configuring VLANs have moved to a different hierarchy level.

NOTE: Starting with Junos OS Release 14.1X53-D10 for EX4300 and EX4600 switches, when enabling *x*STP, you can enable it on some or all interfaces included in a VLAN. For example, if you configure VLAN 100 to include interfaces ge-0/0/0, ge-0/0/1, and ge-0/0/2, and you want to enable MSTP on interfaces ge-0/0/0 and ge-0/0/2, you can specify the **set protocols mstp interface ge-0/0/0** and **set protocols mstp interface ge-0/0/2** commands. In this example, you did not explicitly enable MSTP on interface ge-0/0/1; therefore, MSTP is not enabled on this interface.

Table 15: VLAN hierarchy

Original Hierarchy	Changed Hierarchy
ethernet-switching-options { **secure-access-port** vlan (all | *vlan-name*{ **mac-move-limit** }	vlans *vlan-name* switch-options { **mac-move-limit** }

Table 15: VLAN hierarchy *(continued)*

Original Hierarchy	Changed Hierarchy
ethernet-switching-options { **static** { **vlan** *vlan-id* { mac *mac-address* next-hop *interface-name*; ... } } }	**NOTE:** Statement is replaced with a new statement and has moved to a different hierarchy level. vlans { *vlan-name* { switch-options { interface *interface-name* { static-mac *mac-address*; ... } } } }
vlans { *vlan-name* { interface *interface-name* { **egress**; **ingress**; **mapping** (native (push | swap) | policy | *tag* (push | swap)); **pvlan-trunk**; ... } } }	These statements have been removed. You can assign interfaces to a VLAN using the **[edit interfaces *interface-name* unit *logical-unit-number* family ethernet-switching vlan members *vlan-name*]** hierarchy.
vlans { *vlan-name* { isolation-id *id-number*; ... } }	Statements have been removed.
vlans { *vlan-name* { interface **vlan**.*logical-interface-number*; ... } }	**NOTE:** Syntax is changed. vlans { *vlan-name* { interface **irb**.*logical-interface-number*; ... } }

Table 15: VLAN hierarchy *(continued)*

Original Hierarchy	Changed Hierarchy
vlans { *vlan-name* { l3-interface-ingress-counting *layer-3-interface-name*; ... } }	Statement is removed. Ingress traffic is automatically tracked.
vlans { *vlan-name* { no-local-switching; ... } }	Statement is removed.
vlans { *vlan-name* { **no-mac-learning**; ... } }	Statement has been moved to different hierarchy. vlans { *vlan-name* { switch-options { no-mac-learning *limit* ... } } }
vlans { *vlan-name* { **primary-vlan** *vlan-name*; ... } }	Statement has been removed.
vlans { *vlan-name* { vlan-prune; ... } }	Statement is removed.

Table 15: VLAN hierarchy *(continued)*

Original Hierarchy	Changed Hierarchy
vlans { *vlan-name* { **vlan-range** *vlan-id-low-vlan-id-high*; ... } }	**NOTE:** Statement has been replaced with a new statement. vlans { *vlan-name* { vlan-id-list [*vlan-id-numbers*]; ... } }
vlans { *vlan-name* { l3-interface **vlan**.*logical-interface-number*; ... } }	**NOTE:** Syntax is changed. vlans { *vlan-name* { interface **irb**.*logical-interface-number*; ... } }

Table 16: Statements Moved to a Different Hierarchy

Original Hierarchy	Changed Hierarchy
vlans { *vlan-name* { **dot1q-tunneling** { **customer-vlans** (*id* | native | *range*); **layer2-protocol-tunneling** all | *protocol-name* { drop-threshold *number*; shutdown-threshold *number*; ... } } } }	For **dot1q-tunneling**: interface *interface-name* { encapsulation extended-vlan-bridge; flexible-vlan-tagging; native-vlan-id *number*; unit *logical-unit-number* { input-vlan-map *action*; output-vlan-map *action*; vlan-id *number*; vlan-id-list [*vlan-id vlan-id–vlan-id*]; } } For **layer2-protocol-tunneling** (MAC rewrite enabled on an interface): protocols { layer2-control { mac-rewrite { interface *interface-name* { protocol { ... } } } } }
vlans { *vlan-name* { filter{ **input** *filter-name* **output** *filter-name*; ... } } }	vlans { *vlan-name* { forwarding-options { filter{ **input** *filter-name* **output** *filter-name*; ... } } } }

Table 16: Statements Moved to a Different Hierarchy *(continued)*

Original Hierarchy	Changed Hierarchy
vlans { *vlan-name* { mac-limit *limit* **action** *action*; ... } }	vlans { *vlan-name* { switch-options { interface-mac-limit *limit* { packet-action *action*; ... } } } } vlans { *vlan-name* { switch-options { interface *interface-name* { interface-mac-limit *limit* { packet-action *action*; ... } } } } }
vlans { *vlan-name* { mac-table-aging-time *seconds*; ... } }	protocols { l2-learning { global-mac-table-aging-time *seconds*; ... } }

Changes to Storm Control Profiles

Storm control is configured in two steps. The first step is to create a storm control profile at the **[edit forwarding-options]** hierarchy level, and the second step is to bind the profile to a logical interface at the **[edit interfaces]** hierarchy level. See *Example: Configuring Storm Control to Prevent Network Outages on EX Series Switches* for the changed procedure.

Table 17: Changes to the Storm Control Profile hierarchy level

Original Hierarchy	Changed Hierarchy
ethernet-switching-options { **storm-control** { (...) } }	forwarding-options { **storm-control-profiles** *profile-name* { (...) } } interfaces *interface-name* unit *number* family ethernet-switching { **storm-control** *storm-control-profile*; }

Changes to the Interfaces Hierarchy

> **NOTE:** Statements have been moved to a different hierarchy.

Table 18: Changes to the Interfaces hierarchy

Original Hierarchy	Changed Hierarchy
interfaces *interface-name* { ether-options { **link-mode** *mode*; **speed** (auto-negotiation | *speed*) } }	interfaces *interface-name* { **link-mode** *mode*; **speed** *speed*) }
interfaces *interface-name* { unit *logical-unit-number* { family ethernet-switching { **native-vlan-id** *vlan-id* } } }	interfaces *interface-name* { native-vlan-id *vlan-id* }

Table 18: Changes to the Interfaces hierarchy *(continued)*

Original Hierarchy	Changed Hierarchy
interfaces *interface-name* { unit *logical-unit-number* { family ethernet-switching { port-mode *mode* } } }	**NOTE:** Statement has been replaced with a new statement. interfaces *interface-name* { unit *logical-unit-number* { family ethernet-switching { interface-mode *mode* } } }
interfaces vlan	**NOTE:** Statement has been replaced with a new statement. **interfaces irb**

Changes to IGMP Snooping

Table 19: IGMP Snooping hierarchy

Original Hierarchy	Changed Hierarchy
protocols { igmp-snooping { traceoptions { file *filename* <files *number*> <no-stamp> <replace> <size *maximum-file-size*> <world-readable | no-world-readable>; flag *flag* <*flag-modifier*> <disable>; } vlan (all | *vlan-identifier*) { disable; data-forwarding { receiver { install; source-vlans *vlan-name*; } source { groups *ip-address*; } } immediate-leave; interface (all | *interface-name*) { multicast-router-interface; static { group *multicast-ip-address*; } } proxy { source-address *ip-address*; } robust-count *number*; } } }	

Table 19: IGMP Snooping hierarchy *(continued)*

Original Hierarchy	Changed Hierarchy
	protocols { igmp-snooping { **vlan** *vlan-name* { data-forwarding { receiver { install; source-list *vlan-name*; translate; } source { groups *ip-address*; } } immediate-leave; interface *(all | interface-name)* { group-limit <1..65535> host-only-interface multicast-router-interface; immediate-leave; static { group *multicast-ip-address* { source <> } } } } **l2-querier {** source-address *ip-address*; } proxy { source-address *ip-address*; } query-interval *number*; query-last-member-interval *number*; query-response-interval *number*; robust-count *number*; traceoptions { file *filename* <files *number*> <no-stamp> <replace> <size *maximum-file-size*> <world-readable | no-world-readable>; flag *flag* <*flag-modifier*>; }

Table 19: IGMP Snooping hierarchy *(continued)*

Original Hierarchy	Changed Hierarchy
	} } }

Enhanced Layer 2 CLI Configuration Statement and Command Changes for Security Devices

Starting in Junos OS Release 15.1X49-D10 and Junos OS Release 17.3R1, some Layer 2 CLI configuration statements are enhanced, and some commands are changed. Table 20 on page 72 and Table 21 on page 73 provide lists of existing commands that have been moved to new hierarchies or changed on SRX Series devices as part of this CLI enhancement effort. The tables are provided as a high-level reference only. For detailed information about these commands, see CLI Explorer.

Table 20: Enhanced Layer 2 Configuration Statement Changes

Original Hierarchy	Changed Hierarchy	Hierarchy Level	Change Description
bridge-domains ***bridge-domain--name*** **{** ... } }	**vlans *vlans-name* {** ... } }	[edit]	Hierarchy renamed.
bridge-domains *bridge-domain--name* { **vlan-id-list [*vlan-id*]** ; }	vlans *vlans-name* { **vlan members [*vlan-id*]** ; }	[edit vlans *vlans-name*]	Statement renamed.

Table 20: Enhanced Layer 2 Configuration Statement Changes *(continued)*

Original Hierarchy	Changed Hierarchy	Hierarchy Level	Change Description
bridge-options { interface *interface-name* { encapsulation-type; ignore-encapsulation-mismatch; pseudowire-status-tlv; static-mac *mac-address* { vlan-id *vlan-id*; } } mac-table-aging-time *seconds*; mac-table-size { *number*; packet-action drop; } }	**switch-options** { interface *interface-name* { encapsulation-type; ignore-encapsulation-mismatch; pseudowire-status-tlv; static-mac *mac-address* { vlan-id *vlan-id*; } } mac-table-aging-time *seconds*; mac-table-size { *number*; packet-action drop; } }	[edit vlans *vlans-name*]	Statement renamed.
bridge { block-non-ip-all; bpdu-vlan-flooding; bypass-non-ip-unicast; no-packet-flooding { no-trace-route; } }	**ethernet-switching** { block-non-ip-all; bpdu-vlan-flooding; bypass-non-ip-unicast; no-packet-flooding { no-trace-route; } }	[edit security flow]	Statement renamed.
family { **bridge** { bridge-domain-type (svlan| bvlan); ...	family { **ethernet-switching** { ...	[edit interfaces *interface-name*] unit *unit-number*	Hierarchy renamed.
... **routing-interface** irb.0; ...	... **l3-interface** irb.0; ...	[edit vlans *vlans-name*]	Statement renamed.

Table 21: Enhanced Layer 2 Operational Command Changes

Original Operational Command	Modified Operational Command
clear bridge mac-table	clear ethernet-switching table

Table 21: Enhanced Layer 2 Operational Command Changes *(continued)*

Original Operational Command	Modified Operational Command
clear bridge mac-table persistent-learning	clear ethernet-switching table persistent-learning
show bridge domain	show vlans
show bridge mac-table	show ethernet-switching table
show l2-learning interface	show ethernet-switching interface

NOTE: There is no fxp0 out-of-band management interface on the SRX300, SRX320, and SRX500HM devices. (Platform support depends on the Junos OS release in your installation.)

SEE ALSO

Understanding Switching Modes on Security Devices | 988

Layer 2 Next Generation Mode for ACX Series

The Layer 2 Next Generation mode, also called Enhanced Layer 2 Software (ELS), is supported on ACX5048, ACX5096, and ACX5448 routers for configuring Layer 2 features. The Layer 2 CLI configurations and show commands for ACX5048, ACX5096, and ACX5448 routers differ from those for other ACX Series routers (ACX1000, ACX1100, ACX2000, ACX2100, ACX2200, and ACX4000) and MX Series routers.

Table 22 on page 74 shows the differences in CLI hierarchy for configuring Layer 2 features in Layer 2 next generation mode.

Table 22: Differences in CLI Hierarchy for Layer 2 Features in Layer 2 Next Generation Mode

Feature	ACX1000, ACX1100, ACX2000, ACX2100, ACX2200, ACX4000, and MX Series Routers	ACX5048, ACX5096, and ACX5448 Routers
Bridge Domain	[**edit bridge-domains *bridge-domain-name***]	[**edit vlans *vlan-name***]
Family **bridge**	[**edit interfaces *interface-name* unit *unit-number* family bridge**]	[**edit interfaces *interface-name* unit *unit-number* family ethernet-switching**]

Table 22: Differences in CLI Hierarchy for Layer 2 Features in Layer 2 Next Generation Mode *(continued)*

Feature	ACX1000, ACX1100, ACX2000, ACX2100, ACX2200, ACX4000, and MX Series Routers	ACX5048, ACX5096, and ACX5448 Routers
Layer 2 options	[**edit bridge-domains *bridge-domain-name* bridge-options**]	[**edit vlans *vlan-name* switch-options**]
Ethernet options	[**edit interfaces *interface-name* gigether-options**]	[**edit interfaces *interface-name* ether-options**]
Integrated routing and bridging (IRB)	[**edit bridge-domains *bridge-domain-name***] **routing-interface irb.*unit*;**	[**edit vlans *vlan-name***] **l3-interface irb.*unit*;**
Storm control	[**edit vlans *vlan-name* forwarding-options flood filter *filter-name***]	[**edit forwarding-options storm-control-profiles**] [**edit interfaces *interface-name* ether-options**] **storm-control *name*;** **recovery-timeout *interval*;**
Internet Group Management Protocol (IGMP) snooping	[**edit bridge-domains *bridge-domain-name* protocols igmp-snooping**]	[**edit protocols igmp-snooping vlan *vlan-name***]
Family **bridge** firewall filter	[**edit firewall family bridge**]	[**edit firewall family ethernet-switching**]

Table 23 on page 75 shows the differences in **show** commands for Layer 2 features in Layer 2 next generation mode.

Table 23: Differences in show Commands for Layer 2 Features in Layer 2 Next Generation Mode

Feature	ACX1000, ACX1100, ACX2000, ACX2100, ACX2200, ACX4000, and MX Series Routers	ACX5048, ACX5096, and ACX5448 Routers
VLAN	**show bridge-domain**	**show vlans**
MAC table	**show bridge mac-table**	**show ethernet-switching table**
MAC table options	**show bridge mac-table** (MAC address, bridge-domain name, interface, VLAN ID, and instance)	**show ethernet-switching table**

Table 23: Differences in show Commands for Layer 2 Features in Layer 2 Next Generation Mode *(continued)*

Feature	ACX1000, ACX1100, ACX2000, ACX2100, ACX2200, ACX4000, and MX Series Routers	ACX5048, ACX5096, and ACX5448 Routers
Switch port listing with VLAN assignments	**show l2-learning interface**	**show ethernet-switching interfaces**
Kernel state of flush database	**show route forwarding-table family bridge**	**show route forwarding-table family ethernet-switching**

SEE ALSO

- *Storm Control on ACX Series Routers Overview*
- *Layer 2 Bridge Domains on ACX Series Overview*
- *Guidelines for Configuring Firewall Filters*
- *IGMP Snooping and Bridge Domains*
- *Understanding Ethernet Link Aggregation on ACX Series Routers*

2
CHAPTER

Configuring Layer 2 Forwarding Tables

Layer 2 Forwarding Tables

IN THIS SECTION

Layer 2 Learning and Forwarding for VLANs Overview

Understanding Layer 2 Forwarding Tables on Switches, Routers and NFX Series Devices

You can configure Layer 2 MAC address and VLAN learning and forwarding properties in support of Layer 2 bridging. Unicast media access control (MAC) addresses are learned to avoid flooding the packets to all the ports in a VLAN. A source MAC entry is created in its source and destination MAC tables for each MAC address learned from packets received on ports that belong to the VLAN.

When you configure a VLAN, Layer 2 address learning is enabled by default. The VLAN learns unicast media access control (MAC) addresses to avoid flooding the packets to all the ports in the VLAN. Each VLAN creates a source MAC entry in its source and destination MAC tables for each source MAC address learned from packets received on the ports that belong to the VLAN.

> **NOTE:** Traffic is not flooded back onto the interface on which it was received. However, because this “split horizon” occurs at a late stage, the packet statistics displayed by commands such as **show interfaces queue** will include flood traffic.

You can optionally disable MAC learning either for the entire device or for a specific VLAN or logical interface. You can also configure the following Layer 2 learning and forwarding properties:

- Timeout interval for MAC entries
- Static MAC entries for logical interfaces only

- Limit to the number of MAC addresses learned from a specific logical interface or from all the logical interfaces in a VLAN
- Size of the MAC address table for the VLAN
- MAC accounting for a VLAN

Understanding Layer 2 Forwarding Tables on Security Devices

The SRX Series device maintains forwarding tables that contain MAC addresses and associated interfaces for each Layer 2 VLAN. When a packet arrives with a new source MAC address in its frame header, the device adds the MAC address to its forwarding table and tracks the interface at which the packet arrived. The table also contains the corresponding interface through which the device can forward traffic for a particular MAC address.

If the destination MAC address of a packet is unknown to the device (that is, the destination MAC address in the packet does not have an entry in the forwarding table), the device duplicates the packet and floods it on all interfaces in the VLAN other than the interface on which the packet arrived. This is known as *packet flooding* and is the default behavior for the device to determine the outgoing interface for an unknown destination MAC address. Packet flooding is performed at two levels: packets are flooded to different zones as permitted by configured Layer 2 security policies, and packets are also flooded to different interfaces with the same VLAN identifier within the same zone. The device learns the forwarding interface for the MAC address when a reply with that MAC address arrives at one of its interfaces.

You can specify that the SRX Series device use ARP queries and traceroute requests (which are ICMP echo requests with the time-to-live values set to 1) instead of packet flooding to locate an unknown destination MAC address. This method is considered more secure than packet flooding because the device floods ARP queries and traceroute packets—not the initial packet—on all interfaces. When ARP or traceroute flooding is used, the original packet is dropped. The device broadcasts an ARP or ICMP query to all other devices on the same subnetwork, requesting the device at the specified destination IP address to send back a reply. Only the device with the specified IP address replies, which provides the requestor with the MAC address of the responder.

ARP allows the device to discover the destination MAC address for a unicast packet if the destination IP address is in the same subnetwork as the ingress IP address. (The ingress IP address refers to the IP address of the last device to send the packet to the device. The device might be the source that sent the packet or a router forwarding the packet.) Traceroute allows the device to discover the destination MAC address even if the destination IP address belongs to a device in a subnetwork beyond that of the ingress IP address.

When you enable ARP queries to locate an unknown destination MAC address, traceroute requests are also enabled. You can also optionally specify that traceroute requests not be used; however, the device can then discover destination MAC addresses for unicast packets only if the destination IP address is in the same subnetwork as the ingress IP address.

Whether you enable ARP queries and traceroute requests or ARP-only queries to locate unknown destination MAC addresses, the SRX Series device performs the following series of actions:

1. The device notes the destination MAC address in the initial packet. The device adds the source MAC address and its corresponding interface to its forwarding table, if they are not already there.

2. The device drops the initial packet.

3. The device generates an ARP query packet and optionally a traceroute packet and floods those packets out all interfaces except the interface on which the initial packet arrived.

   ARP packets are sent out with the following field values:

   - Source IP address set to the IP address of the IRB
   - Destination IP address set to the destination IP address of the original packet
   - Source MAC address set to the MAC address of the IRB
   - Destination MAC address set to the broadcast MAC address (all **0xf**)

   Traceroute (ICMP echo request or ping) packets are sent out with the following field values:

   - Source IP address set to the IP address of the original packet
   - Destination IP address set to the destination IP address of the original packet
   - Source MAC address set to the source MAC address of the original packet
   - Destination MAC address set to the destination MAC address of the original packet
   - Time-to-live (TTL) set to **1**

4. Combining the destination MAC address from the initial packet with the interface leading to that MAC address, the device adds a new entry to its forwarding table.

5. The device forwards all subsequent packets it receives for the destination MAC address out the correct interface to the destination.

Layer 2 Learning and Forwarding for VLANs Acting as a Switch for a Layer 2 Trunk Port

Layer 2 learning is enabled by default. A set of VLANs, configured to function as a switch with a Layer 2 trunk port, learns unicast media access control (MAC) addresses to avoid flooding packets to the trunk port.

NOTE: Traffic is not flooded back onto the interface on which it was received. However, because this "split horizon" occurs at a late stage, the packet statistics displayed by commands such as **show interfaces queue** will include flood traffic.

You can optionally disable Layer 2 learning for the entire set of VLANs as well as modify the following Layer 2 learning and forwarding properties:

- Limit the number of MAC addresses learned from the Layer 2 trunk port associated with the set of VLANs
- Modify the size of the MAC address table for the set of VLANs
- Enable MAC accounting for the set of VLANs

Understanding the Unified Forwarding Table

IN THIS SECTION

Benefits of Unified Forwarding Tables

Traditionally, forwarding tables have been statically defined and have supported only a fixed number of entries for each type of address. The unified forward table provides the following benefits:

- Enables you to allocate forwarding table resources to optimize the memory available for different address types based on the needs of your network.
- Enables you to allocate a higher percentage of memory for one type of address or another.

Using the Unified Forwarding Table to Optimize Address Storage

On the QFX5100, EX4600, EX4650, QFX5110, QFX5200, and QFX5120 switches, you can control the allocation of forwarding table memory available to store the following:

- MAC addresses—In a Layer 2 environment, the switch learns new MAC addresses and stores them in a MAC address table
- Layer 3 host entries–In a Layer 2 and Layer 3 environment, the switch learns which IP addresses are mapped to which MAC addresses; these key-value pairs are stored in the Layer 3 host table.
- Longest prefix match (LPM) table entries—In a Layer 3 environment, the switch has a routing table and the most specific route has an entry in the forwarding table to associate a prefix or netmask to a next hop. Note, however, that all IPv4 /32 prefixes and IPv6 /128 prefixes are stored in the Layer 3 host table.

UFT essentially combines the three distinct forwarding tables to create one table with flexible resource allocation. You can select one of five forwarding table profiles that best meets your network needs. Each profile is configured with different maximum values for each type of address. For example, for a switch that handles a great deal of Layer 2 traffic, such as a virtualized network with many servers and virtualized machines, you would likely choose a profile that allocates a higher percentage of memory to MAC addresses. For a switch that operates in the core of a network, participates in an IP fabric, you probably want to maximize the number of routing table entries it can store. In this case, you would choose a profile that allocates a higher percentage of memory to longest match prefixes. The QFX5200 switch supports a custom profile that allows you to partition the four available shared memory banks with a total of 128,000 entries among MAC addresses, Layer 3 host addresses, and LPM prefixes.

> **NOTE:** Support for QFX5200 switches was introduced in Junos OS Release 15.1x53-D30. The QFX5200 switch is not supported on Junos OS Release 16.1R1.

Understanding the Allocation of MAC Addresses and Host Addresses

All five profiles are supported, each of which allocates different amounts of memory for Layer 2 or Layer 3 entries, enabling you choose one that best suits the needs of your network. The QFX5200 and QFX5210 switches, however, supports different maximum values for each profile from the other switches. For more information about the custom profile, see “Configuring the Unified Forwarding Table on Switches” on page 94.

NOTE: The default profile is **l2-profile-three**, which allocates equal space for MAC Addresses and Layer 3 host addresses. On QFX5100, EX4600, QFX5110, and QFX5200 switches, the space is equal to 16,000 IPv4 entries for the LPM table, and on QFX5210 switches, the space is equal to 32,000 IPv4 entries for the LPM table. For the **lpm-profile** the LPM table size is equal to 256,000 IPv4 entries.

NOTE: Starting with Junos OS Release 18.1R1 on the QFX5210-64C switch, for all these profiles, except for the **lpm-profile** the longest prefix match (LPM) table size is equal to 32,000 IPv4 entries.

NOTE: Starting with Junos OS Release 18.3R1 on the QFX5120 and EX4650 switches, for all these profiles, except for the **lpm-profile** the longest prefix match (LPM) table size is equal to 32,000 IPv4 entries.

NOTE: On QFX5100, EX4600, EX4650, QFX5110, QFX5200, QFX5120, and QFX5210-64C switches, IPv4 and IPv6 host routes with ECMP next hops are stored in the host table.

BEST PRACTICE: If the host or LPM table stores the maximum number of entries for any given type of entry, the entire shared table is full and is unable to accommodate *any* entries of any other type. Different entry types occupy different amounts of memory. For example, an IPv6 unicast address occupies twice as much memory as an IPv4 unicast address, and an IPv6 multicast address occupies four times as much memory as an IPv4 unicast address.

Table 24 on page 83 lists the profiles you can choose and the associated maximum values for the MAC address and host table entries on QFX5100 and EX4600 switches.

Table 24: Unified Forwarding Table Profiles on QFX5100 and EX4600 Switches

Profile Name	MAC Table	Host Table (unicast and multicast addresses)					
	MAC Addresses	IPv4 unicast	IPv6 unicast	IPv4 (*, G)	IPv4 (S, G)	IPv6 (*, G)	IPv6 (S, G)

Table 24: Unified Forwarding Table Profiles on QFX5100 and EX4600 Switches *(continued)*

Profile Name	MAC Table	Host Table (unicast and multicast addresses)					
l2-profile-one	288K	16K	8K	8K	8K	4K	4K
l2-profile-two	224K	80K	40K	40K	40K	20K	20K
l2-profile-three (default)	160K	144K	72K	72K	72K	36K	36K
l3-profile	96K	208K	104K	104K	104K	52K	52K
lpm-profile	32K	16K	8K	8K	8K	4K	4K
lpm-profilewith **unicast-in-lpm** option	32K	(stored in LPM table)	(stored in LPM table)	8K	8K	4K	4K

Table 25 on page 84 lists the profiles you can choose and the associated maximum values for the MAC address and host table entries on QFX5110 switches.

Table 25: Unified Forwarding Table Profiles on QFX5110 Switches

Profile Name	MAC Table	Host Table (unicast and multicast addresses)					
	MAC Addresses	IPv4 unicast	IPv6 unicast	IPv4 (*, G)	IPv4 (S, G)	IPv6 (*, G)	IPv6 (S, G)
l2-profile-one	288K	16K	8K	8K	8K	4K	4K
l2-profile-two	224K	80K	40K	40K	40K	20K	20K
l2-profile-three (default)	160K	144K	72K	72K	72K	36K	36K
l3-profile	96K	208K	104K	104K	104K	52K	52K

Table 26 on page 84 lists the LPM table size variations for the QFX5110 switch depending on the prefix entries.

Table 26: LPM Table Size Variations on QFX5110 Switches

Profile Name	Prefix Entries		
num-65-127-prefix	**IPv4 LPM<= /32**	**IPv6 LPM <= /64**	**IPv6 LPM > /64**

Table 26: LPM Table Size Variations on QFX5110 Switches *(continued)*

Profile Name	Prefix Entries		
0	16K	8K	0K
1	12K	6K	1K
2	8K	4K	2K
3	4K	2K	3K
4	0K	0K	4K

Table 27 on page 85 lists the profiles you can choose and the associated maximum values for the MAC address and host table entries on QFX5200-32C switches.

Table 27: Unified Forwarding Table Profiles on QFX5200-32C Switches

Profile Name	MAC Table	Host Table (unicast and multicast addresses)						
	MAC Addresses	IPv4 unicast	IPv6 unicast	IPv4 (*, G)	IPv4 (S, G)	IPv6 (*, G)	IPv6 (S, G)	Exact Match
l2-profile-one	136K	8K	4K	4K	4K	2K	2K	0
l2-profile-two	104K	40K	20K	20K	20K	10K	10K	0
l2-profile-three (default)	72K	72K	36K	36K	36K	18K	18K	0
l3-profile	40K	104K	52K	52K	52K	26K	26K	0
lpm-profile	8K	8K	4K	4K	4K	2K	2K	0

Table 28 on page 85 lists the profiles you can choose and the associated maximum values for the MAC address and host table entries on QFX5200-48Y switches.

Table 28: Unified Forwarding Table Profiles on QFX5200-48Y Switches

Profile Name	MAC Table	Host Table (unicast and multicast addresses)					
	MAC Addresses	IPv4 unicast	IPv6 unicast	IPv4 (*, G)	IPv4 (S, G)	IPv6 (*, G)	IPv6 (S, G)

Table 28: Unified Forwarding Table Profiles on QFX5200-48Y Switches *(continued)*

Profile Name	MAC Table	Host Table (unicast and multicast addresses)					
l2-profile-one	136K	8K	4K	4K	4K	2K	2K
l2-profile-two	104K	40K	20K	20K	20K	10K	10K
l2-profile-three (default)	72K	72K	36K	36K	36K	18K	18K
l3-profile	40K	104K	52K	52K	52K	26K	26K
lpm-profile	8K	8K	4K	4K	4K	2K	2K

Table 29 on page 86 lists the LPM table size variations for the QFX5200-48Y switch depending on the prefix entries.

Table 29: LPM Table Size Variations on QFX5200-48Y Switches

Profile Name	Prefix Entries		
num-65-127-prefix	**IPv4 LPM<= /32**	**IPv6 LPM <= /64**	**IPv6 LPM > /64**
0	16K	8K	0K
1	12K	6K	1K
2	8K	4K	2K
3	40K	2K	3K
4	0K	0K	4K

Table 30 on page 86 lists the profiles you can choose and the associated maximum values for the MAC address and host table entries on QFX5210-64C switches.

Table 30: Unified Forwarding Table Profiles on QFX5210-64C Switches

Profile Name	MAC Table	Host Table (unicast and multicast addresses)						
	MAC Addresses	**IPv4 unicast**	**IPv6 unicast**	**IPv4 (*, G)**	**IPv4 (S, G)**	**IPv6 (*, G)**	**IPv6 (S, G)**	**Exact Match**

Table 30: Unified Forwarding Table Profiles on QFX5210-64C Switches *(continued)*

Profile Name	MAC Table	Host Table (unicast and multicast addresses)						
l2-profile-one	264K	8K	4K	4K	4K	2K	2K	0K
l2-profile-two	200K	72K	36K	36K	36K	18K	18K	0K
l2-profile-three (default)	136K	136K	72K	72K	72K	36K	36K	0K
l3-profile	72K	200K	100K	100K	100K	50K	50K	0K

Table 31 on page 87 lists the profiles you can choose and the associated maximum values for the MAC address and host table entries on QFX5120 and EX4650 switches.

Table 31: Unified Forwarding Table Profiles on QFX5120 and EX4650 Switches

Profile Name	MAC Table	Host Table (unicast and multicast addresses)					
	MAC Addresses	IPv4 unicast	IPv6 unicast	IPv4 (*, G)	IPv4 (S, G)	IPv6 (*, G)	IPv6 (S, G)
l2-profile-one	288K	16K	8K	8K	8K	4K	4K
l2-profile-two	224K	80K	40K	40K	40K	20K	20K
l2-profile-three (default)	160K	144K	72K	72K	72K	36K	36K
l3-profile	96K	208K	104K	104K	104K	52K	52K

Table 32 on page 87 lists the LPM table size variations for the QFX5210-64C switch depending on the prefix entries.

Table 32: LPM Table Size Variations on QFX5210-64C Switches

Profile Name	Prefix Entries		
num-65-127-prefix	**IPv4 LPM<= /32**	**IPv6 LPM <= /64**	**IPv6 LPM > /64**
0	32K	16K	0K
1	28K	14K	1K

Table 32: LPM Table Size Variations on QFX5210-64C Switches *(continued)*

Profile Name	Prefix Entries		
2	24K	12K	2K
3	20K	10K	3K
4	0K	0K	4K

Table 33 on page 88 lists the Layer 3 Defip table size variations for the QFX5120 and EX4650 switches depending on the changing IPv6/128 prefix entries.

Table 33: LPM Table Size Variations on QFX5210-64C and EX4650 Switches

Profile Name	Prefix Entries		
num-65-127-prefix	IPv4 LPM<= /32	IPv6 LPM <= /64	IPv6 LPM > /64
0	32K	16K	0K
2	24K	12K	2K
4	16K	8K	4K
6	8K	4K	6K
8	0K	0K	8K

Understanding Ternary Content Addressable Memory (TCAM) and Longest Prefix Match Entries

You can further customize non-LPM profiles by configuring the space available for ternary content addressable memory (TCAM) to allocate more memory for longest prefix match entries. You can change the number of entries allocated to these IPv6 addresses, essentially allocating more or less space for LPM IPv4 entries with any prefix length or IPv6 entries with prefix lengths of 64 of shorter. For more information about how to change the default parameters of the TCAM memory space for LPM entries, see “Configuring the Unified Forwarding Table on Switches” on page 94.

NOTE: The option to adjust TCAM space is not supported on the longest prefix match (LPM) or custom profiles. However, for the LPM profile, you can configure TCAM space not to allocate any memory for IPv6 entries with prefix lengths of 65 or longer, thereby allocating that memory space only for IPv4 routes or IP routes with prefix lengths equal to or less than 64 or a combination of the two types of prefixes.

NOTE: Starting with Junos OS Release 18.1R1 on QFX5210 switches, you can configure TCAM space to allocate a maximum of 8,000 IPv6 entries with prefix lengths of 65 or longer. The default value is 2,000 entries. Starting with Junos OS Release 13.2X51-D15, you can configure TCAM space to allocate a maximum of 4,000 IPv6 entries with prefix lengths of 65 or longer. The default value is 1,000 entries. Previous to Junos OS Release 13.2X51-D15, you could allocate only a maximum of 2,048 entries for IPv6 the IPv6 prefixes with lengths in the range /65 to /127 range. The default value was 16 entries for these types of IPv6 prefixes.

On Junos OS Releases 13.2x51-D10 and 13.2x52D10, the procedure to change the default value of 16 entries differs from later releases, where the maximum and default values are higher. For more information about that procedure, see "Configuring the Unified Forwarding Table on Switches" on page 94

Host Table Example for Profile with Heavy Layer 2 Traffic

Table 34 on page 89 lists various valid combinations that the host table can store if you use the **l2-profile-one** profile on QFX5100 and EX4600 switches. This profile allocates the percentage of memory to Layer 2 addresses. Note that the default values might be different on other switches. Each row in the table represents a case in which the host table is full and cannot accommodate any more entries.

Table 34: Example Host Table Combinations Using l2-profile-one on QFX5100 and EX4600 Switches

IPv4 unicast	IPv6 unicast	IPv4 multicast (*, G)	IPv4 multicast (S, G)	IPv6 multicast (*, G)	IPv6 multicast (S, G)
16K	0	0	0	0	0
12K	2K	0	0	0	0
12K	0	2K	2K	0	0
8K	4K	0	0	0	0
4K	2K	2K	2K	0	0

Table 34: Example Host Table Combinations Using l2-profile-one on QFX5100 and EX4600 Switches *(continued)*

IPv4 unicast	IPv6 unicast	IPv4 multicast (*, G)	IPv4 multicast (S, G)	IPv6 multicast (*, G)	IPv6 multicast (S, G)
0	4K	0	0	1K	1K

Example: Configuring a Unified Forwarding Table Custom Profile

IN THIS SECTION

Traditionally, forwarding tables have been statically defined and have supported only a fixed number of entries for each type of address. The Unified Forwarding Table (UFT) feature enables you to optimize how forwarding-table memory is allocated to best suit the needs of your network. This example shows how to configure a Unified Forwarding Table profile that enables you to partition four shared hash memory banks among three different types of forwarding-table entries: MAC addresses, Layer 3 host addresses, and longest prefix match (LPM).

The UFT feature also supports five profiles that each allocate a specific maximum amount of memory for each type of forwarding table entry. Some profiles allocate more memory to Layer 2 entries, while other profiles allocate more memory to Layer 3 or LPM entries. The maximum values for each type of entry are fixed in these profiles. With the custom profile, you can designate one or more shared memory banks to store a specific type of forwarding-table entry. You can configure as few as one or as many as four memory banks in a custom profile. The custom profile thus provides even more flexibility in enabling you to allocate forwarding-table memory for specific types of entries.

Requirements

This example uses the following hardware and software components:

- One QFX5200 switch

- Junos OS Release 15.1x53-D30 or later.

Before you configure a custom profile, be sure you have:

- Configured interfaces

Overview

The Unified Forwarding Table custom profile enables you to allocate forwarding-table entries among four banks of shared hash tables with a total memory equal to 128,000 unicast IPv4 addresses, or 32,000 entries for each bank. Specifically, you can allocate one or more of these shared banks to store a specific type of forwarding-table entry. The custom profile does not affect the dedicated hash tables. Those tables remain fixed with 8,000 entries allocated to Layer 2 addresses, the equivalent of 8,000 entries allocated to IPv4 addresses, and the equivalent of 16,000 entries allocated to longest prefix match (LPM) addresses.

In this example, you allocate two memory banks to Layer 3 host addresses, and two memory banks to LPM entries. This means that no shared hash table memory is allocated for Layer 2 addresses. Only the dedicated hash table memory is allocated for Layer 2 addresses in this scenario.

Configuration

IN THIS SECTION

To configure a custom profile for the Unified Forwarding Table feature on a QFX5200 switch that allocates two shared memory banks for Layer 3 host address and two shared memory banks for LPM entries, perform these tasks:

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode. A commit check is performed to ensure that you have allocated forwarding-table space for no more than four memory banks.

CAUTION: When you configure and commit a profile, the Packet Forwarding Engine restarts and all the data interfaces on the switch go down and come back up.

```
user@switch# set chassis forwarding-options custom-profile
user@switch# set chassis forwarding-options custom-profile l2-entries num-banks 0
user@switch# set chassis forwarding-options custom-profile l3-entries num-banks 2
user@switch# set chassis forwarding-options custom-profile lpm-entries num-banks 2
```

Configuring the Custom Profile

Step-by-Step Procedure

To create the custom profile:

1. Specify the **custom-profile** option.

```
[edit chassis forwarding-options]
user@switch# set custom-profile
```

Configuring the Allocation of Shared Memory Banks

Step-by-Step Procedure

To allocate memory for specific types of entries for the shared memory banks:

1. Specify to allocate no shared bank memory for Layer 2 entries.

```
[edit chassis forwarding-options custom-profile]
user@switch# set l2-entries num-banks 0
```

2. Specify to allocate two shared memory banks (or the equivalent of 64,000 IPv4 entries) for Layer 3 host entries.

```
[edit chassis forwarding-options custom-profile]
user@switch# set l3-entries num-banks 2
```

3. Specify to allocate two shared memory banks (or the equivalent of 64,000 IPv4 entries) for LPM entries.

```
[edit chassis forwarding-options custom-profile]
user@switch# set lpm-entries numer-banks 2
```

Results

From configuration mode, confirm your configuration by entering the show chassis forwarding-options command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@switch# show chassis forwarding-profile
custom-profile {
    l2-entries {
        num-banks 0;
    }
    l3-entries {
        num-banks 2;
    }
    lpm-entries {
        num-banks 2
    }
}
```

If you are done configuring the switch, enter **commit** from configuration mode

CAUTION: The Packet Forwarding Engine will restart and all the data interfaces on the switch will go down and come back up.

Verification

IN THIS SECTION

- Checking the Parameters of the Custom Profile | 93

Confirm that the configuration is working properly.

Checking the Parameters of the Custom Profile

Purpose

Verify that the custom profile is enabled.

Action

```
user@switch> show chassis forwarding-options

UFT Configuration:
custom-profile
Configured custom scale:
Entry type                Total scale(K)
L2(mac)                   8
L3 (unicast & multicast)        72
Exact Match                0
Longest Prefix Match (lpm)        80
num-65-127-prefix = 1K
-------------Bank details for various types of entries------------
Entry type                  Dedicated Bank Size(K)      Shared Bank Size(K)
L2 (mac)                    8                         32 * num shared banks
L3 (unicast & multicast         8                          32 * num shared banks
Exact match                  0                           16 * num shared banks
Longest Prefix match(lpm)           16                      32 * num shared banks
```

Meaning

The output shows that the custom profile is enabled as configured with two shared memory banks designated for Layer 3 host entries; two shared memory banks designated for LPM entries; and no shared memory allocated for Layer 2 entries.

The total scale(K) field shows the total allocation of memory, that is, the amount allocated through the shared memory banks plus the amount allocated through the dedicated hash tables. The amount allocated through the dedicated hash tables is fixed and cannot be changed. Therefore, Layer 2 entries have 8K of memory allocated only through the dedicated hash table. Layer 3 host entries have 64K of memory allocated through two shared memory banks plus 8K through the dedicated hash table, for a total of 72K of memory. LPM entries have 64K of memory allocated through two shared memory banks plus 16K through the dedicated hash table, for a total of 80K of memory.

Configuring the Unified Forwarding Table on Switches

IN THIS SECTION

Traditionally, forwarding tables have been statically defined and have supported only a fixed number of entries for each type of address stored in the tables. The Unified Forwarding Table feature lets you optimize how your switch allocates forwarding-table memory for different types of addresses. You can choose one of five unified forwarding table profiles. Each profile allocates a different maximum amount of memory for Layer 2, Layer 3 host, and longest prefix match (LPM) entries. In addition to selecting a profile, you can also select how much additional memory to allocate for LPM entries.

Two profiles allocate higher percentages of memory to Layer 2 addresses. A third profile allocates a higher percentage of memory to Layer 3 host address, while a fourth profile allocates a higher percentage of memory to LPM entries. There is a default profile configured that allocates an equal amount of memory to Layer 2 and Layer 3 host addresses with the remainder allocated to LPM entries. For a switch in a virtualized network that handles a great deal of Layer 2 traffic, you would choose a profile that allocates a higher percentage of memory to Layer 2 addresses. For a switch that operates in the core of the network, you would choose a profile that allocates a higher percentage of memory to LPM entries.

On QFX5200 and QFX5210-64C switches only, you can also configure a custom profile that allows you to partition shared memory banks among the different types of forwarding table entries. On QFX5200 switches, these shared memory banks have a total memory equal to 128,000 IPv4 unicast addresses. On QFX5210 switches, these shared memory banks have a total memory equal to 256,000 IPv4 unicast addresses. For more information about configuring the custom profile, see “Example: Configuring a Unified Forwarding Table Custom Profile” on page 90.

Configuring a Unified Forwarding Table Profile

To configure a unified forwarding table profile:

Specify a forwarding-table profile.

```
[edit chassis forwarding-options]
user@switch# set profile-name
```

For example, to specify the profile that allocates the highest percentage of memory to Layer 2 traffic:

```
[edit chassis forwarding-options]
user@switch# set l2-profile-one
```

CAUTION: When you configure and commit a profile, in most cases the Packet Forwarding Engine automatically restarts and all the data interfaces on the switch go down and come back up (the management interfaces are unaffected).

Starting with Junos OS Releases 14.1X53-D40, 15.1R5, and 16.1R3, for a Virtual Chassis or Virtual Chassis Fabric (VCF) comprised of EX4600 or QFX5100 switches, the Packet Forwarding Engine in member switches does not automatically restart upon configuring and committing a unified forwarding table profile change. This behavior avoids Virtual Chassis or VCF instability after the change propagates to member switches and multiple Packet Forwarding Engines automatically restart at the same time. Instead, a message is displayed at the CLI prompt and logged to the switch's system log to notify you that the profile change does not take effect until the next time you reboot the Virtual Chassis or VCF. We recommend that you plan to make profile changes only when you can perform a Virtual Chassis or VCF system reboot immediately after committing the configuration update. Otherwise, the Virtual Chassis or VCF could become inconsistent if one or more members have a problem and restart with the new configuration before a planned system reboot activates the change on all members.

NOTE: You can configure only one profile for the entire switch.

NOTE: The **l2-profile-three** is configured by default.

NOTE: If the host table stores the maximum number of entries for any given type, the entire table is full and is unable to accommodate *any* entries of any other type. Keep in mind that an IPv6 unicast address occupies twice as much memory as an IPv4 unicast address, and an IPv6 multicast address occupies four times as much memory as an IPv4 unicast address..

Configuring the Memory Allocation for Longest Prefix Match Entries

IN THIS SECTION

In addition to choosing a profile, you can further optimize memory allocation for longest prefix match (LPM) entries by configuring how many IPv6 prefixes to store with lengths from /65 through /127. The switch uses LPM entries during address lookup to match addresses to the most-specific (longest) applicable prefix. Prefixes of this type are stored in the space for ternary content addressable memory (TCAM). Changing the default parameters makes this space available for LPM entries. Increasing the amount of memory available for these IPv6 prefixes reduces by the same amount how much memory is available to store IPv4 unicast prefixes and IPv6 prefixes with lengths equal to or less than 64.

The procedures for configuring the LPM table are different, depending on which version of Junos OS you are using. In the initial releases that UFT is supported, Junos OS Releases 13.2X51-D10 and 13.2X52-10, you can only increase the amount of memory allocated to IPv6 prefixes with lengths from /65 through /127 for any profile, except for **lpm-profile**. Starting with Junos OS Release 13.2X51-D15, you can also allocate either less or no memory for IPv6 prefixes with lengths in the range /65 through /127, depending on which profile is configured. For the **lpm-profie**, however, the only change you can make to the default parameters is to allocate no memory for these types of prefixes.

Configuring the LPM Table With Junos OS Releases 13.2X51-D10 and 13.2X52-D10

In Junos OS Releases 13.2x51-D10 and 13.2X52-D10, by default, the switch allocates memory for 16 IPv6 with prefixes with lengths in the range /65 through /127. You can configure the switch to allocate more memory for IPv6 prefixes with lengths in the range /65 through /127.

To allocate more memory for IPv6 prefixes in the range /65 through /127:

1. Choose a forwarding table profile.

```
[edit chassis forwarding-options]
user@swtitch# set profile-name
```

For example, to specify the profile that allocates the highest percentage of memory to Layer 2 traffic:

```
[edit chassis forwarding-options]
user@swtitch# set l2-profile-one
```

2. Select how much memory to allocate for IPv6 prefixes in the range /65 thorugh 127.

```
[edit chassis forwarding-options profile-name]
user@swtitch# set num-65-127-prefix number
```

For example, to specify to allocate memory for 32 IPv6 prefixes in the range /65 through 127:

```
[edit chassis forwarding-options l2-profile-one]
user@switch# set num-65-127-prefix 2
```

NOTE: When you configure and commit the **num-65-127-prefix *number*** statement, all the data interfaces on the switch restart. The management interfaces are unaffected.

The **num-65-127-prefix *number*** statement is not supported on the **lpm-profile**.

Configuring the LPM Table With Junos OS Release 13.2x51-D15 and Later

IN THIS SECTION

Configuring Layer 2 and Layer 3 Profiles With Junos OS Release 13.2x51-D15 or Later

Starting in Junos OS Release 13.2X51-D15, you can configure the switch to allocate forwarding table memory for as many as 4,000 IPv6 prefixes with lengths in the range /65 through /127 for any profile other than the **lpm-profile** or **custom-profile**. You can also specify to allocate no memory for these IPv6

entries. The default is 1,000 entries for IPv6 prefixes with lengths in the range /65 through /127. Previously, the maximum you could configure was for 2,048 entries for IPv6 prefixes with lengths in the range /65 through /127. The minimum number of entries was previously 16, which was the default.

To specify how much forwarding table memory to allocate for IPv6 prefixes with length in the range /65 through /127:

1. Choose a forwarding table profile.

   ```
   [edit chassis forwarding-options]
   user@swtitch# set profile-name
   ```

   For example, to specify the profile that allocates the highest percentage of memory to Layer 2 traffic:

   ```
   [edit chassis forwarding-options]
   user@swtitch# set l2-profile-one
   ```

2. Select how much memory to allocate for IPv6 prefixes in the range /65 thorugh 127.

   ```
   [edit chassis forwarding-options profile-name]
   user@swtitch# set num-65-127-prefix number
   ```

   For example, to specify to allocate memory for 2,000 IPv6 prefixes in the range /65 through 127:

   ```
   [edit chassis forwarding-options l2-profile-one]
   user@switch# set num-65-127-prefix 2
   ```

Starting with Junos OS Release 13.2X51-D15, you can use the **num-65-127-prefix** statement to allocate entries. Table 35 on page 99 shows the numbers of entries that you can allocate. Each row represents a case in which the table is full and cannot accommodate any more entries.

Table 35: LPM Table Combinations for L2 and L3 profiles With Junos OS 13.2X51-D15 and Later

num-65-127-prefix Value	IPv4 Entries	IPv6 Entries (Prefix <= 64)	IPv6 Entries (Prefix >= 65)
0	16K	8K	0K
1 (default)	12K	6K	1K
2	8K	4K	2K
3	4K	2K	3K

Table 35: LPM Table Combinations for L2 and L3 profiles With Junos OS 13.2X51-D15 and Later *(continued)*

4	0K	0K	4K

CAUTION: When you configure and commit a profile change with the **num-65-127-prefix *number*** statement, the Packet Forwarding Engine automatically restarts and all the data interfaces on the switch go down and come back up (the management interfaces are unaffected).

However, starting with Junos OS Releases 14.1X53-D40, 15.1R5, and 16.1R3, Packet Forwarding Engines on switches in a Virtual Chassis or Virtual Chassis Fabric (VCF) do not automatically restart upon configuring a unified forwarding table profile change. This behavior avoids Virtual Chassis or VCF instability after the change propagates to member switches and multiple Packet Forwarding Engines automatically restart at the same time. Instead, a message is displayed at the CLI prompt and logged to the switch's system log to notify you that the profile change does not take effect until the next time you reboot the Virtual Chassis or VCF. We recommend that you plan to make profile changes only when you can perform a Virtual Chassis or VCF system reboot immediately after committing the configuration update. Otherwise, the Virtual Chassis or VCF could become inconsistent if one or more members have a problem and restart with the new configuration before a planned system reboot activates the change on all members.

Configuring the lpm-profile With Junos OS Release 13.2x51-D15 and Later

Starting with Junos OS Release 13.2X51-D15 you can configure the **lpm-profile** profile not to allocate any memory for IPv6 entries with prefix lengths from /65 through /127. These are the default maximum values allocated for LPM memory for the **lpm-profile** by address type:

- 128K of IPv4 prefixes
- 16K of IPv6 prefixes (all lengths)

NOTE: The memory allocated for each address type represents the maximum default value for all LPM memory.

To configure the **lpm-profile** not to allocate forwarding-table memory for IPv6 entries with prefixes from /65 through /127, thus allocating more memory for IPv4:

Specify to disable forwarding-table memory for IPv6 prefixes with lengths in the range /65 through /127.

```
[edit chassis forwarding-options lpm-profile]
user@switch# set prefix-65-127-disable
```

For example, on the QFX5100 and EX4600 switches only, if you use the **prefix-65-127-disable** option, each of the following combinations are valid:

- 100K IPv4 and 28K IPv6 /64 or shorter prefixes.
- 64K IPv4 and 64K IPv6 /64 or shorter prefixes.
- 128K IPv4 and 0K IPv6 /64 or shorter prefixes.
- 0K IPv4 and 128K IPv6 /64 or shorter prefixes.

NOTE: On the QFX5200 switches, when you configure the **prefix-65-127-disable** statement, the maximum number of IPv6 entries with prefixes equal to or shorter than 64 is 98,000.

Configuring the lpm-profile With Junos OS Release 14.1x53-D30 and Later

Starting in Junos OS Release 15.1X53-D30, you can configure the **lpm-profile** profile to store unicast IPv4 and IPv6 host addresses in the LPM table , thereby freeing memory in the host table. Unicast IPv4 and IPv6 addresses are stored in the LPM table instead of the host table, as shown in Table 36 on page 101 for QFX5100 and EX4600 switches. (Platform support depends on the Junos OS release in your installation.) You can use this option in conjunction with the option to allocate no memory in the LPM table for IPv6 entries with prefix lengths in the range /65 through /127. Together, these options maximize the amount of memory available for IPv4 unicast entries and IPv6 entries with prefix lengths equal to or less than 64.

Table 36: lpm-profile with unicast-in-lpm Option for QFX5100 and EX4600 Switches

prefix-65-127-disable	MAC Table	Host Table (multicast addresses)						LPM Table unicast addresses)		
	MAC	IPv4 unicast	IPv6 unicast	IPv4 (*, G)	IPv4 (S, G)	IPv6 (*, G)	IPv6 (S, G)	IPv4 unicast	IPv6 unicast (</65)	IPv6 unicast (>/64)
No	32K	0	0	8K	8K	4K	4K	128K	16K	16K
Yes	32K	0	0	8K	8K	4K	4K	128K	128K	0

Starting with Junos Release 18.1R1, you cannot set configure a prefix for the **num-65-127-prefix** statement on non-LPM profiles. You can only enable or disable the **prefix-65-127-disable** statement for the **lpm-profile**.

Table 37 on page 102 lists the situations in which the **prefix-65-127-disable** statement should be enabled or disabled.

Table 37: LPM Table Size Variations on QFX5200-48Y Switches

Profile Name	Prefix Entries		
num-65-127-prefix	**IPv4 <= /32**	**IPv6 <= /64**	**IPv6 > /64**
Enabled	> 128K (minimum guaranteed)	98K	0K
Disabled	128K	16K	16K

On QFX5120 and EX4600 switches, you cannot set configure a prefix for the **num-65-127-prefix** statement on non-LPM profiles. You can only enable or disable the **prefix-65-127-disable** statement for the **lpm-profile**

Table 38 on page 102 lists the situations in which the **prefix-65-127-disable** statement should be enabled or disabled.

Table 38: LPM Table Size Variations on QFX5120 and EX4650 Switches

Profile Name	Prefix Entries		
prefix-65-127-disable	**IPv4 <= /32**	**IPv6 <= /64**	**IPv6 > /64**
Enabled	351K (360,000 approximate)	168K (172,000 approximate)	0K
Disabled	168K (172,000 approximate)	64K (65,524 approximate)	64K (65,524 approximate)

Note that all entries in each table share the same memory space. If a table stores the maximum number of entries for any given type, the entire shared table is full and is unable to accommodate any entries of any other type. For example, if you use the the **unicast-in-lpm** option and there are 128K IPv4 unicast addresses stored in the LPM table, the entire LPM table is full and no IPv6 addresses can be stored. Similarly, if you use the **unicast-in-lpm** option but do not use the **prefix-65-127-disable** option, and 16K IPv6 addresses with prefixes shorter than /65 are stored, the entire LPM table is full and no additional addresses (IPv4 or IPv6) can be stored.

To configure the **lpm-profile** to store unicast IPv4 entries and IPv6 entries with prefix lengths equal to or less than 64 in the LPM table:

1. Specify the option to store these entries in the LPM table.

```
[edit chassis forwarding-options lpm-profile]
user@switch# set unicast-in-lpm
```

2. (Optional) Specify to allocate no memory for in the LPM table for IPv6 prefixes with length in the range /65 through /127:

```
[edit chassis forwarding-options lpm-profile]
user@switch# set prefix-65-127-disable
```

Configuring Non-LPM Profiles on QFX5120 and EX4650 Switches

For non-LPM profiles, each profile provides the option of reserving a portion of the 16K L3-defip table to store IPv6 Prefixes > 64. Because these are 128-bit prefixes, you can have maximum of 8k IPv6/128 entries in the l3-defip table.

1. Choose a forwarding table profile.

```
[edit chassis forwarding-options]
user@swtitch# set profile-name
```

For example, to specify the profile that allocates the highest percentage of memory to Layer 3 traffic:

```
[edit chassis forwarding-options]
user@swtitch# set l3-profile
```

2. Select how much memory to allocate for IPv6 prefixes in the range /65 thorugh 127.

```
[edit chassis forwarding-options profile-name]
user@swtitch# set num-65-127-prefix number
```

For example, to specify to allocate memory for 2,000 IPv6 prefixes in the range /65 through 127:

You can choose between 0 and 4, 1 being the default.

```
[edit chassis forwarding-options l3-profile]
user@switch# set num-65-127-prefix 1
```

Configuring Forwarding Mode on Switches

By default, packets packets are forwarded using store-and-forward mode. You can configure all the interfaces to use cut-through mode instead.

To enable cut-through switching mode, enter the following statement:

```
[edit forwarding-options]
user@switch# set cut-through
```

SEE ALSO

cut-through | 1104

Disabling Layer 2 Learning and Forwarding

Disabling dynamic MAC learning on an MX Series router or an EX Series switch prevents all the logical interfaces on the router or switch from learning source and destination MAC addresses.

To disable MAC learning for an MX Series router or an EX Series switch, include the **global-no-mac-learning** statement at the **[edit protocols l2-learning]** hierarchy level:

```
[edit protocols l2-learning]
global-no-mac-learning;
```

For information about how to configure a virtual switch, see *Configuring a Layer 2 Virtual Switch* .

SEE ALSO

Understanding Layer 2 Learning and Forwarding

Configuring the MAC Table Timeout Interval

Enabling MAC Accounting

Limiting the Number of MAC Addresses Learned from Each Logical Interface

3
CHAPTER

Configuring MAC Addresses

MAC Addresses

IN THIS SECTION

Introduction to the Media Access Control (MAC) Layer 2 Sublayer

This topic provides an introduction to the MAC sublayer of the data link layer (Layer 2).

In Layer 2 of a network, the Media Access Control (MAC) sublayer provides addressing and channel access control mechanisms that enable several terminals or network nodes to communicate in a network.

The MAC sublayer acts as an interface between the logical link control (LLC) Ethernet sublayer and Layer 1 (the physical layer). The MAC sublayer emulates a full-duplex logical communication channel in a multipoint network. This channel may provide unicast, multicast, or broadcast communication service. The MAC sublayer uses MAC protocols to prevent collisions.

In Layer 2, multiple devices on the same physical link can uniquely identify one another at the data link layer, by using the MAC addresses that are assigned to all ports on a switch. A MAC algorithm accepts as input a secret key and an arbitrary-length message to be authenticated, and outputs a MAC address.

A MAC address is a 12-digit hexadecimal number (48 bits in long). MAC addresses are usually written in one of these formats:

- MM:MM:MM:SS:SS:SS
- MM-MM-MM-SS-SS-SS

The first half of a MAC address contains the ID number of the adapter manufacturer. These IDs are regulated by an Internet standards body. The second half of a MAC address represents the serial number assigned to the adapter by the manufacturer.

Contrast MAC addressing, which works at Layer 2, with IP addressing, which runs at Layer 3 (networking and routing). One way to remember the difference is that the MAC addresses apply to a physical or virtual node, whereas IP addresses apply to the software implementation of that node. MAC addresses are typically fixed on a per-node basis, whereas IP addresses change when the node moves from one part of the network to another.

IP networks maintain a mapping between the IP and MAC addresses of a node using the Address Resolution Protocol (ARP) table. DHCP also typically uses MAC addresses when assigning IP addresses to nodes.

SEE ALSO

Overview of Layer 2 Networking | 39

Understanding MAC Learning | 117

Understanding MAC Address Assignment on an EX Series Switch

This topic describes MAC address assignment for interfaces on standalone Juniper Networks EX Series Ethernet Switches. For information regarding MAC address assignments in a Virtual Chassis, see *Understanding MAC Address Assignment on a Virtual Chassis*.

MAC addresses are used to identify network devices at Layer 2. Because all Layer 2 traffic decisions are based on an interface's MAC address, understanding MAC address assignment is important to understanding how network traffic is forwarded and received by the switch. For additional information on how a network uses MAC addresses to forward and receive traffic, see "Understanding Bridging and VLANs on Switches" on page 168.

A MAC address comprises six groups of two hexadecimal digits, with each group separated from the next group by a colon—for instance, aa:bb:cc:dd:ee:00. The first five groups of hexadecimal digits are derived from the switch and are the same for all interfaces on the switch.

The assignment of a unique MAC address to each network interface helps ensure that functions that require MAC address differentiation—such as redundant trunk groups (RTGs), Link Aggregation Control Protocol (LACP), and general monitoring functions—can properly function.

On switches that use line cards, this MAC addressing scheme differentiates the Layer 2 interfaces on different line cards in the switch.

For EX Series switches, the first five groups of hexadecimal digits are determined when the switch is manufactured. The switch then assigns a unique MAC address to each interface by assigning a unique identifier as the last group of hexadecimal digits. The assignment depends on how the interface is configured. The switch uses a different pattern to distinguish between an interface that is configured as any of a routed

VLAN interface (RVI), a virtual management Ethernet (VME) interface, or an aggregated Ethernet interface or is not configured as any of an RVI, a VME, or as an aggregated Ethernet interface.

For aggregated Ethernet interfaces, the MAC address assignment remains constant regardless of whether the configuration of the interface is Layer 2 or Layer 3.

> **NOTE:** In Junos OS Release 11.3 and later releases through Release 12.1, the MAC address assignment for aggregated Ethernet interfaces changes if the interface is changed from Layer 2 to Layer 3 or the reverse. Starting with Junos Release 12.2, the MAC address assignment for aggregated Ethernet interfaces remains constant regardless of whether the interface is Layer 2 or Layer 3.

> **NOTE:** Prior to Junos OS Release 11.3, MAC addresses for Layer 2 interfaces could be shared between interfaces and RVIs on different line cards in the same switch. However, if you upgrade from Junos OS Release 11.2 or earlier to Junos OS Release 11.3 or later on a switch that supports line cards, the MAC addresses of these interfaces will change.

MAC addresses are assigned to interfaces automatically—no user configuration is possible or required. You can view MAC addresses assigned to interfaces using the **show interfaces** command.

SEE ALSO

Interfaces Overview for Switches

Configuring MAC Move Parameters

When a MAC address appears on a different physical interface or within a different unit of the same physical interface and this behavior occurs frequently, it is considered a MAC move. You can configure the router to report a MAC address move based on the following parameters: the number of times a MAC address move occurs, a specified period of time over which the MAC address move occurs, and specified number of times a MAC address move occurs in one second. You can only configure the **global-mac-move** statement at the global hierarchy level.

To globally disable the MAC move action feature, include the **disable-action** statement at the **[edit protocols l2-learning global-mac-move]**. This disables the MAC move action feature, while MAC move detection exists.

To configure the time duration after which the port will be unblocked, include the **reopen-time** statement at the **[edit protocols l2-learning global-mac-move]**. The default reopen timer is 180 second.

To configure MAC address move reporting if the MAC address moves at least a specified number of times in one second, include the **threshold-time** statement at the **[edit protocols l2-learning global-mac-move]** hierarchy level. The default threshold time is 1 second.

To configure reporting of a MAC address move if the MAC address moves for a specified period of time, include the **notification-time** statement at the **[edit protocols l2-learning global-mac-move]** hierarchy level. The default notification timer is 1 second.

To configure reporting of a MAC address move if the MAC address moves a specified number of times, include the **threshold-count** statement at the **[edit protocols l2-learning global-mac-move]** hierarchy level. The default threshold count is 50 moves.

Use the **show l2-learning mac-move-buffer** command to view the actions as a result of MAC address move feature.

Use the **show l2-learning mac-move-buffer active** command to view the set of IFLs blocked as a result of MAC move action.

Use the **exclusive-mac** command exclude a MAC address from the MAC move limit algorithm, preventing a MAC address from being tracked.

Use the **clear l2-learning mac-move-buffer active** command to unblock the IFBDs that were blocked by MAC move action feature. This allows the user to keep the **reopen-time** configured to a large value, but when the looping error is fixed, user can manually release the blocking.

The following example sets the notification time for MAC moves to 1 second, the threshold time to 1 second, reopen-time to 180 seconds and the threshold count to 50 moves.

```
[edit protocols l2-learning]
global-mac-move {
   notification-time 1;
   reopen-time 180;
   threshold-count 50;
   threshold-time 1;
}
```

Configuring MAC Limiting (ELS)

IN THIS SECTION

This topic describes different ways of configuring a limitation on MAC addresses in packets that are received and forwarded by the switch.

> **NOTE:** The tasks presented in the first section uses Junos OS for EX Series switches and QFX3500 and QFX3600 switches with support for the Enhanced Layer 2 Software (ELS) configuration style. See "Using the Enhanced Layer 2 Software CLI" on page 50 for more information about ELS configurations.
>
> - For information on configuring an interface to automatically recover from a shutdown caused by MAC limiting, see *Configuring Autorecovery for Port Security Events*If you do not configure the switch for autorecovery from the disabled condition, you can bring up the disabled interfaces by running the **clear ethernet-switching recovery-timeout** command.

The different ways of setting a MAC limit are described in the following sections:

Limiting the Number of MAC Addresses Learned by an Interface

To secure a port, you can set the maximum number of MAC addresses that can be learned by an interface:

- Set the MAC limit on an interface, and specify an action that the switch takes after the specified limit is exceeded:

  ```
  [edit switch-options]
  user@switch# set interface interface-name interface-mac-limit limit packet-action action
  ```

  After you set a new MAC limit for the interface, the system clears existing entries in the MAC address forwarding table associated with the interface.

Limiting the Number of MAC Addresses Learned by a VLAN

To limit the number of MAC addresses learned by a VLAN, perform both of the following steps:

1. Set the maximum number of MAC addresses that can be learned by a VLAN, and specify an action that the switch takes after the specified limit is exceeded:

   ```
   [edit vlans]
   user@switch# set vlan-name switch-options mac-table-size limit packet-action action
   ```

2. Set the maximum number of MAC addresses that can be learned by one *or* all interfaces in the VLAN, and specify an action that the switch takes after the specified limit is exceeded:

   > **NOTE:** If you specify a MAC limit and packet action for all interfaces in the VLAN *and* a specific interface in the VLAN, the MAC limit and packet action specified at the specific interface level takes precedence. Also, at the VLAN interface level, only the **drop** and **drop-and-log** options are supported.

   ```
   [edit vlans]
   user@switch# set vlan-name switch-options interface interface-name
   interface-mac-limit limit packet-action action
   ```

   ```
   [edit vlans]
   user@switch# set vlan-name switch-options interface-mac-limit limit packet-action action
   ```

After you set new MAC limits for a VLAN by using the **mac-table-size** statement or for interfaces associated with a VLAN by using the **interface-mac-limit** statement, the system clears the corresponding existing entries in the MAC address forwarding table.

> **NOTE:** On a QFX Series Virtual Chassis, if you include the **shutdown** option at the **[edit vlans *vlan-name* switch-options interface *interface-name* interface-mac-limit packet-action]** hierarchy level and issue the **commit** operation, the system generates a commit error. The system does not generate an error if you include the **shutdown** option at the **[edit switch-options interface *interface-name* interface-mac-limit packet-action]** hierarchy level.

Adding a Static MAC Address Entry to the Ethernet Switching Table on a Switch with ELS Support

> **NOTE:** This task uses Junos OS for EX Series switches and Junos OS for QFX3500 and QFX3600 switches with support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does not support ELS, see "Adding a Static MAC Address Entry to the Ethernet Switching Table" on page 113. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50.

The Ethernet switching table, also known as the forwarding table, specifies the known locations of VLAN nodes and the addresses of devices within those nodes. There are two ways to populate the Ethernet switching table on a switch. The easiest method is to let the switch update the table with MAC addresses.

The second way to populate the Ethernet switching table is to manually insert addresses into the table. You can do this to reduce flooding and speed up the switch's automatic learning process.

Before configuring a static MAC address, be sure that you have:

- Set up the VLAN. See *Configuring VLANs for EX Series Switches with ELS Support (CLI Procedure)*.

To configure an interface to have a static MAC address:

```
[edit vlans vlan-name switch-options interface interface-name]
user@switch# set static-mac mac-address
```

Adding a Static MAC Address Entry to the Ethernet Switching Table

> **NOTE:** This task uses Junos OS for EX Series switches and Junos OS for QFX3500 and QFX3600 switches that does not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see "Adding a Static MAC Address Entry to the Ethernet Switching Table on a Switch with ELS Support" on page 112. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50.

The Ethernet switching table, also known as the forwarding table, specifies the known locations of VLAN nodes. There are two ways to populate the Ethernet switching table on a switch. The easiest method is to let the switch update the table with MAC addresses.

The second way to populate the Ethernet switching table is to manually insert a VLAN node location into the table. You can do this to reduce flooding and speed up the switch's automatic learning process. To further optimize the switching process, indicate the next hop (next interface) packets will use after leaving the node.

Before configuring a static MAC address, be sure that you have:

- Set up the VLAN. See "Configuring VLANs for EX Series Switches" on page 183 or"Configuring VLANs on Switches" on page 182.

To add a MAC address to the Ethernet switching table:

1. Specify the MAC address to add to the table:

   ```
   [edit ethernet-switching-options]
   set static vlan vlan-name mac mac-address
   ```

2. Indicate the next hop MAC address for packets sent to the indicated MAC address:

   ```
   [edit ethernet-switching-options]
   ```

set static vlan *vlan-name* **mac** *mac-address* **next-hop** *interface*

Example: Configuring the Default Learning for Unknown MAC Addresses

IN THIS SECTION

This example shows how to configure the device to use only ARP requests to learn the outgoing interfaces for unknown destination MAC addresses.

Requirements

Before you begin, determine the MAC addresses and associated interfaces of the forwarding table. See "Layer 2 Learning and Forwarding for VLANs Overview" on page 78.

Overview

In this example, you configure the device to use only ARP queries without traceroute requests.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set security flow ethernet-switching no-packet-flooding no-trace-route
```

Step-by-Step Procedure

To configure the device to use only ARP requests to learn unknown destination MAC addresses:

1. Enable the device.

```
[edit]
user@host# set security flow ethernet-switching no-packet-flooding no-trace-route
```

2. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show security flow** command.

4

CHAPTER

Configuring MAC Learning

MAC Learning

IN THIS SECTION

Understanding MAC Learning

MAC learning is the process of obtaining the MAC addresses of all the nodes on a network.

When a node is first connected to an Ethernet LAN or VLAN, it has no information about the other nodes on the network. As data is sent through the network, data packets include a data frame listing their source and destination MAC addresses. The data frame is forwarded to a target port, which is connected to the second device. The MAC address is learned locally at the target port, which facilitates communications for frames that later enter the target port and contain addresses previously learned from a received frame.

By default, MAC learning is enabled on the QFX and NFX Series.

Disabling MAC Learning on Devices with ELS Support

By default, MAC learning is globally enabled on all nodes. This topic describes how to disable MAC learning, as well as how to reenable and verify that MAC learning has been enabled or disabled.

> **NOTE:** This task supports the Enhanced Layer 2 Software (ELS) configuration style. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50 If your switch runs software that does not support ELS, see "Disabling MAC Learning on QFX Switches" on page 118.

Disabling dynamic MAC learning prevents a node from learning source and destination MAC addresses.

- To disable MAC learning:

```
[edit vlans vlan-name switch-options interface interface-name]
user@switch# set no-mac-learning
```

- To enable MAC learning:

```
[edit vlans vlan-name switch-options interface interface-name]
user@switch# delete no-mac-learning
user@switch# deactivate no-mac-learning
```

- To verify the status of MAC learning, view the Ethernet MAC learning statistics in operational mode.

```
user@switch> show ethernet-switching table
Ethernet-switching table: 2 entries, 1 learned
  VLAN              MAC address       Type         Age Interfaces
  default           *                 Flood          - All-members
  default           00:1f:12:39:90:80 Learn         29 xe-/0/0.0
```

Disabling MAC Learning on QFX Switches

By default, MAC learning is globally enabled on all nodes in a device. This topic describes how to disable MAC learning, as well as how to reenable and verify that MAC learning has been enabled or disabled.

Disabling dynamic MAC learning on the device prevents a node from learning source and destination MAC addresses.

> **NOTE:** This task uses Junos OS for QFX3500 and QFX3600 switches and does not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see “Disabling MAC Learning on Devices with ELS Support” on page 117.

- To disable MAC learning on the QFX Series:

```
[edit ethernet-switching-options interfaces interface]
user@switch# set no-mac-learning
```

- To enable MAC learning on the QFX Series:

```
[edit ethernet-switching-options interfaces interface]
```

```
user@switch# delete no-mac-learning
user@switch# deactivate no-mac-learning
```

- To verify the status of MAC learning on the QFX Series, view the Ethernet MAC learning statistics in operational mode.

```
user@switch> show ethernet-switching table
```

```
Ethernet-switching table: 2 entries, 1 learned
  VLAN              MAC address       Type         Age Interfaces
  default           *                 Flood          - All-members
  default           00:1f:12:39:90:80 Learn         29 xe-/0/0.0
```

Disabling MAC Learning in a VLAN on a QFX Switch

By default, MAC learning is enabled on a VLAN. This topic describes how to disable MAC learning in a VLAN, as well as how to reenable and verify that MAC learning has been enabled or disabled.

Disabling dynamic MAC learning in a VLAN on a QFX Series product prevents a node from learning source and destination MAC addresses.

- To disable MAC learning in a VLAN:

```
[edit vlans vlan-name]
user@switch# set no-mac-learning
```

- To reenable MAC learning in a VLAN, use either of the following two commands:

```
[edit vlans vlan-name]
user@switch# delete no-mac-learning
user@switch# deactivate no-mac-learning
```

- To verify the status of MAC learning on the QFX series:

```
user@switch> show ethernet-switching table
```

Disabling MAC Learning for a VLAN or Logical Interface

You can disable MAC learning for all logical interfaces in a specified VLAN, or for a specific logical interface in a VLAN. Disabling dynamic MAC learning prevents the specified interfaces from learning source MAC addresses.

To disable MAC learning for all logical interfaces in a VLAN in a virtual switch, include the **no-mac-learning** statement at the **[edit vlans *vlan-name* switch-options]** hierarchy level:

```
[edit]
vlans {
    vlan-name {
        domain-type bridge;
        interface interface-name;
        switch-options {
            no-mac-learning;
        }
    }
}
```

To disable MAC learning for a specific logical interface in a VLAN, include the **no-mac-learning** statement at the **[edit vlans *vlan-name* switch-options interface *interface-name*]** hierarchy level.

```
[edit]
vlans {
    vlan-name {
        domain-type bridge;
        interface interface-name;
        switch-options {
            interface interface-name {
                no-mac-learning;
            }
        }
    }
}
```

NOTE: When you disable MAC learning, source MAC addresses are not dynamically learned, and any packets sent to these source addresses are flooded into the VLAN.

NOTE: When you gather interfaces into a VLAN, the **no-mac-learn-enable** statement at the **[edit interfaces *interface-name* ether-options ethernet-switch-profile]** hierarchy level is not supported. You must use the **no-mac-learning** statement at the **[edit vlans *vlan-name* switch-options interface *interface-name*]** hierarchy level to disable MAC learning on an interface in a VLAN.

NOTE: When MAC learning is disabled for a VPLS routing instance, traffic is not load balanced and only one of the equal-cost next hops is used.

Disabling MAC Learning for a Set of VLANs

You can disable MAC learning for a set of VLANs. Disabling dynamic MAC learning prevents the Layer 2 trunk port associated with the set of VLANs from learning source and destination MAC addresses. When you disable MAC learning, source MAC addresses are not dynamically learned, and any packets sent to these source addresses are flooded into the switch.

To disable MAC learning for a set of VLANs, include the **no-mac-learning** statement at the **[edit switch-options]** hierarchy level:

```
[edit switch-options]
no-mac-learning;
```

5
CHAPTER

Configuring MAC Accounting

MAC Accounting

IN THIS SECTION

Enabling MAC Accounting on a Device

By default, MAC accounting is disabled on the device. You can enable packet accounting either for a device as a whole or for a specific VLAN. After you enable packet accounting, the Junos OS maintains packet counters for each MAC address learned.

To enable MAC accounting, include the **global-mac-statistics** statement at the **[edit protocols l2-learning]** hierarchy level:

```
[edit protocols l2-learning]
global-mac-statistics;
```

Enabling MAC Accounting for a VLAN

By default, MAC accounting is disabled. You can enable packet counting for a VLAN. When you enable packet accounting, the Junos OS maintains packet counters for each MAC address learned on the interfaces in the VLAN.

To enable MAC accounting for a VLAN, include the **mac-statistics** statement at the **[edit vlans *vlan-name* switch-options]** hierarchy level:

```
[edit vlans vlan-name switch-options]
mac-statistics;
```

Enabling MAC Accounting for a Set of VLANs

By default, MAC accounting is disabled. You can enable packet counting for a set of VLANs. After you enable packet accounting, the Junos OS maintains packet counters for each MAC address learned on the trunk port associated with the set of VLANs.

To enable MAC accounting for a set of VLANs, include the **mac-statistics** statement at the **[edit switch-options]** hierarchy level:

```
[edit switch-options]
mac-statistics;
```

Verifying That MAC Accounting Is Working

Purpose

Verify that MAC accounting is enabled and the system is counting packets and collecting statistics.

Action

1. Verify that MAC accounting is enabled.

```
user@switch> show ethernet-switching table
```

```
MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
        SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags               interface
    VLAN101             88:e0:f3:bb:07:f0   D,SE           -    ae20.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
       SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags               interface
    VLAN102             88:e0:f3:bb:07:f0   D,SE           -    ae20.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
```

```
       SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags               interface
    VLAN103             88:e0:f3:bb:07:f0   D,SE            -   ae20.0
[...output truncated...]
```

2. Display MAC accounting statistics for all VLANs associated with an interface.

   `user@switch>` **show ethernet-switching statistics**

```
 Local interface: ae20.0, Index: 1039
   Broadcast packets:                   115
   Broadcast bytes  :                  6900
   Multicast packets:                395113
   Multicast bytes  :              61622869
   Flooded packets  :                     0
   Flooded bytes    :                     0
   Unicast packets  :                  1419
   Unicast bytes    :                117924
   Current MAC count:                     4 (Limit 8192)
[...output truncated...]
```

3. Display MAC accounting statistics for each address in the MAC address table.

   `user@switch>` **show ethernet-switching table extensive**

```
MAC address: 88:e0:f3:bb:07:f0
  Routing instance: default-switch
VLAN ID: 101
   Learning interface: ae20.0
   Layer 2 flags: in_hash,in_ifd,in_ifl,in_vlan,in_rtt,acct,kernel,in_ifbd
   Epoch: 6                            Sequence number: 13
   Learning mask: 0x00000020
MAC address used as destination:
Packet count:                     0  Byte count:                     0
MAC address used as source:
Packet count:                     9  Byte count:                  1116

MAC address: 88:e0:f3:bb:07:f0
  Routing instance: default-switch
VLAN ID: 102
```

```
   Learning interface: ae20.0
   Layer 2 flags: in_hash,in_ifd,in_ifl,in_vlan,in_rtt,acct,kernel,in_ifbd
   Epoch: 6                          Sequence number: 13
   Learning mask: 0x00000020
MAC address used as destination:
Packet count:                     0  Byte count:                     0
MAC address used as source:
Packet count:                     9  Byte count:                  1116

MAC address: 88:e0:f3:bb:07:f0
  Routing instance: default-switch
VLAN ID: 103
   Learning interface: ae20/0
   Layer 2 flags: in_hash,in_ifd,in_ifl,in_vlan,in_rtt,acct,kernel,in_ifbd
   Epoch: 6                          Sequence number: 13
   Learning mask: 0x00000020
MAC address used as destination:
Packet count:                     0  Byte count:                     0
MAC address used as source:
Packet count:                     9  Byte count:                  1116
[...output truncated...]
```

Meaning

In the output for **show ethernet-switching table**, the MAC flag **SE** indicates that MAC accounting is enabled for VLANs 101, 102, and 103, which are all associated with the **default-switch** routing instance.

The output for **show ethernet-switching statistics** displays packet statistics and the current number of MAC addresses learned by the VLANs associated with aggregated Ethernet interface **ae20.0**.

The output for **show ethernet-switching table extensive** shows information for each address in the MAC address table. In particular, it displays the number of packets sent to and received by an interface, which is identified by a MAC address.

The output from the three commands demonstrates that MAC accounting is working properly. That is, MAC accounting is enabled on VLANs 101, 102, and 103, and as a result, you can view statistics for each of these VLANs, aggregated Ethernet interface **ae20.0**, and each MAC address.

CHAPTER 6

Configuring MAC Notification

MAC Notification

IN THIS SECTION

Understanding MAC Notification on EX Series Switches

Juniper Networks EX Series Switches track clients on a network by storing Media Access Control (MAC) addresses in the Ethernet switching table on the switch. When switches learn or unlearn a MAC address, SNMP notifications can be sent to the network management system at regular intervals to record the addition or removal of the MAC address. This process is known as MAC notification.

The MAC Notification MIB controls MAC notification for the network management system. For general information on the MAC Notification MIB, see the *Junos OS Network Management Configuration Guide*.

The MAC notification interval defines how often these SNMP notifications are sent to the network management system. The MAC notification interval works by tracking all of the MAC address additions or removals on the switch over a period of time and then sending all of the tracked MAC address additions or removals to the network management server at the end of the interval. For instance, if the MAC notification interval is set to 10, all of the MAC address addition and removal SNMP notifications are sent to the network management system every 10 seconds.

Enabling MAC notification allows users to monitor the addition and removal of MAC addresses from the Ethernet switching table remotely using a network management system. The advantage of setting a high MAC notification interval is that the amount of network traffic is reduced because updates are sent less frequently. The advantage of setting a low MAC notification interval is that the network management system is better synchronized with the switch.

MAC notification is disabled by default. When MAC notification is enabled, the default MAC notification interval is 30 seconds.

Configuring MAC Notification on Switches with ELS Support

IN THIS SECTION

NOTE: This task uses the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does not support ELS, see "Configuring Non-ELS MAC Notification" on page 130 or "Configuring Non-ELS MAC Notification" on page 130. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50.

When a switch learns or unlearns a MAC address, SNMP notifications can be sent to the network management system at regular intervals to record the addition or removal of the MAC address. This process is known as MAC notification.

The MAC notification interval defines how often Simple Network Management Protocol (SNMP) notifications logging the addition or removal of MAC addresses on the switch are sent to the network management system.

MAC notification is disabled by default. When MAC notification is enabled, the default MAC notification interval is 30 seconds.

To enable or disable MAC notification, or to set the MAC notification interval, perform these tasks:

Enabling MAC Notification

MAC notification is disabled by default. You need to perform this procedure to enable MAC notification.

To enable MAC notification on the switch with the default MAC notification interval of 30 seconds:

```
[edit switch-options]
user@switch# set mac-notification
```

To enable MAC notification on the switch with any other MAC notification interval (here, the MAC notification interval is set to 60 seconds):

```
[edit switch-options]
user@switch# set mac-notification notification-interval 60
```

Disabling MAC Notification

MAC notification is disabled by default. Perform this procedure only if MAC notification was previously enabled on your switch.

To disable MAC notification on the switch:

```
[edit switch-options]
user@switch# delete mac-notification
```

To disable MAC notification on a specific interface (here, the interface is ge-0/0/3):

```
[edit switch-options]
user@switch# set interface ge-0/0/3 no-mac-notification
```

Setting the MAC Notification Interval

The default MAC notification interval is 30 seconds. The procedure to change the MAC notification interval to a different interval is identical to the procedure to enable MAC notification on the switch with a nondefault value for the MAC notification interval.

To set the MAC notification interval on the switch (here, the MAC notification interval is set to 5 seconds):

```
[edit switch-options]
user@switch# set mac-notification notification-interval 5
```

Configuring Non-ELS MAC Notification

IN THIS SECTION

NOTE: This task uses Junos OS for EX Series switches that do not support Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see "Configuring MAC Notification on Switches with ELS Support" on page 129. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50.

When a switch learns or unlearns a MAC address, SNMP notifications can be sent to the network management system at regular intervals to record the addition or removal of the MAC address. This process is known as MAC notification.

The MAC notification interval defines how often Simple Network Management Protocol (SNMP) notifications logging the addition or removal of MAC addresses on the switch are sent to the network management system.

MAC notification is disabled by default. When MAC notification is enabled, the default MAC notification interval is 30 seconds.

To enable or disable MAC notification, or to set the MAC notification interval, perform these tasks:

Enabling MAC Notification

MAC notification is disabled by default. You need to perform this procedure to enable MAC notification.

To enable MAC notification on the switch with the default MAC notification interval of 30 seconds:

```
[edit ethernet-switching-options]
user@switch# set mac-notification
```

To enable MAC notification on the switch with any other MAC notification interval (here, the MAC notification interval is set to 60 seconds):

```
[edit ethernet-switching-options]
user@switch# set mac-notification notification-interval 60
```

Disabling MAC Notification

MAC Notification is disabled by default. Perform this procedure only if MAC notification was previously enabled on your switch.

To disable MAC notification on the switch:

```
[edit ethernet-switching-options]
user@switch# delete mac-notification
```

Setting the MAC Notification Interval

The default MAC notification interval is 30 seconds. The procedure to change the MAC notification interval to a different interval is identical to the procedure to enable MAC notification on the switch with a nondefault value for the MAC notification interval.

To set the MAC notification interval on the switch (here, the MAC notification interval is set to 5 seconds):

```
[edit ethernet-switching-options]
user@switch# set mac-notification notification-interval 5
```

Verifying That MAC Notification Is Working Properly

Purpose

Verify that MAC notification is enabled or disabled, and that the MAC notification interval is set to the specified value.

Action

To verify that MAC notification is enabled or disabled on a QFX Series switch or an EX4600, and also to verify the MAC notification interval setting:

`user@switch>` **show ethernet-switching mac-notification**

```
Notification Status: Enabled
Notification Interval: 60
Notifications Sent          : 0
Notifications Table Maxsize : 256
```

The output in the **Notification Status** field shows that MAC notification is enabled. The output in the **Notification Status** field would display **Disabled** if MAC notification was disabled.

The **Notification Interval** field output shows that the MAC notification interval is set to 60 seconds.

To verify that MAC notification is enabled on an EX Series switch while also verifying the MAC notification interval setting:

`user@switch>` **show ethernet-switching mac-notification**

```
Notification Status: Enabled
Notification Interval: 30
```

The output in the **Notification Status** field shows that MAC notification is enabled. The output in the **Notification Status** field would display **Disabled** if MAC notification was disabled.

The **Notification Interval** field output shows that the MAC notification interval is set to 30 seconds.

CHAPTER 7

Configuring MAC Table Aging

MAC Table Aging

IN THIS SECTION

Understanding MAC Table Aging

Juniper Networks EX Series Ethernet Switches store MAC addresses in the Ethernet switching table, also called the *MAC table*. When the aging time for a MAC address in the table expires, the address is removed.

If your switch runs Juniper Networks Junos operating system (Junos OS) for EX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style, you can configure the MAC table aging time on all VLANs on the switch. If your switch runs Junos OS that does not support ELS, you can configure the MAC table aging time on all VLANs on the switch or on specified VLANs, as well as configure aging time to be unlimited, either on all VLANs or on specified VLANs, so that MAC addresses never age out of the table.

To learn MAC addresses, the switch reads all packets that it detects on the LAN or on the local VLAN, looking for MAC addresses of sending nodes. It places these addresses into its Ethernet switching table, along with two other pieces of information—the interface on which the traffic was received and the time when the address was learned.

When the switch receives traffic on an interface, it searches the Ethernet switching table for the MAC address of the destination. If the MAC address is not found, the traffic is flooded out all of the other interfaces associated with the VLAN. For example, if traffic is received on an interface that is associated with VLAN v-10 and there is no entry in the Ethernet switching table for VLAN v-10 (the Ethernet switching table is organized by VLAN), then the traffic is flooded to all access and trunk interfaces that are members of VLAN v-10.

Flooding allows the switch to learn about destinations that are not yet in its Ethernet switching table. If a particular destination MAC address is not in the Ethernet switching table, the switch floods the traffic to all interfaces except the interface on which it was received. When the destination node receives the flooded traffic, it sends an acknowledgment packet back to the switch, allowing the switch to learn the MAC address of the node and to add the address to its Ethernet switching table.

The switch uses a mechanism called aging to keep the Ethernet switching table current. For each MAC address in the Ethernet switching table, the switch records a timestamp of when the information about

the network node was learned. Each time the switch detects traffic from a MAC address that is in its Ethernet switching table, it updates the timestamp of that MAC address. A timer on the switch periodically checks the timestamp, and if the MAC address of a node is older than the value set, the switch removes that MAC address from the Ethernet switching table. This aging process ensures that the switch tracks only active MAC addresses on the network and that it is able to flush out from the Ethernet switching table MAC addresses that are no longer available.

You configure how long MAC addresses remain in the Ethernet switching table by:

- (On switches that run Junos OS with support for the ELS configuration style) Using the **global-mac-table-aging-time** statement in the **[edit protocols l2-learning]** hierarchy.
- (On switches that run Junos OS that does not support ELS) Using the **mac-table-aging-time** statement in either the **[edit ethernet-switching-options]** or the **[edit vlans]** hierarchy, depending on whether you want to configure it for the entire switch or only for specific VLANs.

For example, in a topology with EX switches that run Junos OS that does not support ELS, if you have a printer VLAN, you might choose to configure the aging time for that VLAN to be considerably longer than for other VLANs so that MAC addresses of printers on this VLAN age out less frequently. Because the MAC addresses remain in the table, even if a printer has been idle for some time before traffic arrives for it, the switch still finds the MAC address and does not need to flood the traffic to all other interfaces.

Similarly, in a data center environment where the list of servers connected to the switch is fairly stable, you might choose to increase MAC address aging time, or even set it to unlimited, to increase the efficiency of the utilization of network bandwidth by reducing flooding.

SEE ALSO

> *Controlling Authentication Session Timeouts (CLI Procedure)*

Configuring MAC Table Aging on Switches

MAC table aging ensures that a switch tracks only active nodes on the network and that it is able to flush out network nodes that are no longer available.

To manage MAC entries more efficiently, you can configure an entry's aging time, which is the maximum time that an entry can remain in the MAC address table before it is deleted because it has reached its maximum age.

The following example uses Junos OS for Junos OS for QFX3500 and QFX3600 switches with no support for the Enhanced Layer 2 Software (ELS) configuration style. Use the **set-mac-table-aging-time** command to configure how long entries remain in the Ethernet switching table before expiring. Here the VLAN is **employee-vlan**:

```
[edit vlans employee-vlan]
user@switch# set mac-table-aging-time 200
```

> **NOTE:** This command applies to all VLANs configured for the switch. You cannot configure separate MAC table aging times for specific VLANs.

The following example uses Junos OS for QFX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style. Use the **global-mac-table-aging-time** command to configure how long entries remain in the Ethernet switching table before expiring, as follows:

```
[edit protocols l2-learning]
user@switch# set global-mac-table-aging-time 200
```

> **NOTE:** This command applies to all VLANs configured for the switch. You cannot configure separate MAC table aging times for specific VLANs.

The following example uses Junos OS for EX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style.

The Ethernet switching table (or MAC table) aging process ensures that the EX Series switch tracks only active MAC addresses on the network and is able to flush out MAC addresses that are no longer used.

You can configure the MAC table aging time, the maximum time that an entry can remain in the Ethernet Switching table before it *ages out*, on all VLANs on the switch. This setting can influence efficiency of network resource use by affecting the amount of traffic that is flooded to all interfaces because when traffic is received for MAC addresses no longer in the Ethernet switching table, the switch floods the traffic to all interfaces.

```
[edit]
user@switch# set protocols l2-learning global-mac-table-aging-time seconds
```

The following example uses Junos OS for EX Series switches that do not support the Enhanced Layer 2 Software (ELS) configuration style.

The Ethernet switching table (or MAC table) aging process ensures that the EX Series switch tracks only active MAC addresses on the network and is able to flush out MAC addresses that are no longer used.

You can configure the MAC table aging time, the maximum time that an entry can remain in the Ethernet Switching table before it “ages out,” either on all VLANs on the switch or on particular VLANs. This setting can influence efficiency of network resource use by affecting the amount of traffic that is flooded to all interfaces because when traffic is received for MAC addresses no longer in the Ethernet switching table, the switch floods the traffic to all interfaces.

To configure the MAC table aging time on all VLANs on the switch:

```
[edit]
user@switch# set ethernet-switching-options mac-table-aging-time seconds
```

To configure the MAC table aging time on a VLAN:

```
[edit]
user@switch# set vlans vlan-name mac-table-aging-time seconds
```

> **NOTE:** You can set the MAC table aging time to unlimited. If you specify the value as **unlimited**, entries are never removed from the table. Generally, use this setting only if the switch or the VLAN has a fairly static number of end devices; otherwise the table will eventually fill up. You can use this setting to minimize traffic loss and flooding that might occur when traffic arrives for MAC addresses that have been removed from the table.

8

CHAPTER

Configuring Learning and Forwarding

Layer 2 Forwarding Tables

IN THIS SECTION

Layer 2 Learning and Forwarding for VLANs Overview

Understanding Layer 2 Forwarding Tables on Switches, Routers and NFX Series Devices

You can configure Layer 2 MAC address and VLAN learning and forwarding properties in support of Layer 2 bridging. Unicast media access control (MAC) addresses are learned to avoid flooding the packets to all the ports in a VLAN. A source MAC entry is created in its source and destination MAC tables for each MAC address learned from packets received on ports that belong to the VLAN.

When you configure a VLAN, Layer 2 address learning is enabled by default. The VLAN learns unicast media access control (MAC) addresses to avoid flooding the packets to all the ports in the VLAN. Each VLAN creates a source MAC entry in its source and destination MAC tables for each source MAC address learned from packets received on the ports that belong to the VLAN.

> **NOTE:** Traffic is not flooded back onto the interface on which it was received. However, because this “split horizon” occurs at a late stage, the packet statistics displayed by commands such as **show interfaces queue** will include flood traffic.

You can optionally disable MAC learning either for the entire device or for a specific VLAN or logical interface. You can also configure the following Layer 2 learning and forwarding properties:

- Timeout interval for MAC entries
- Static MAC entries for logical interfaces only

- Limit to the number of MAC addresses learned from a specific logical interface or from all the logical interfaces in a VLAN
- Size of the MAC address table for the VLAN
- MAC accounting for a VLAN

Understanding Layer 2 Forwarding Tables on Security Devices

The SRX Series device maintains forwarding tables that contain MAC addresses and associated interfaces for each Layer 2 VLAN. When a packet arrives with a new source MAC address in its frame header, the device adds the MAC address to its forwarding table and tracks the interface at which the packet arrived. The table also contains the corresponding interface through which the device can forward traffic for a particular MAC address.

If the destination MAC address of a packet is unknown to the device (that is, the destination MAC address in the packet does not have an entry in the forwarding table), the device duplicates the packet and floods it on all interfaces in the VLAN other than the interface on which the packet arrived. This is known as *packet flooding* and is the default behavior for the device to determine the outgoing interface for an unknown destination MAC address. Packet flooding is performed at two levels: packets are flooded to different zones as permitted by configured Layer 2 security policies, and packets are also flooded to different interfaces with the same VLAN identifier within the same zone. The device learns the forwarding interface for the MAC address when a reply with that MAC address arrives at one of its interfaces.

You can specify that the SRX Series device use ARP queries and traceroute requests (which are ICMP echo requests with the time-to-live values set to 1) instead of packet flooding to locate an unknown destination MAC address. This method is considered more secure than packet flooding because the device floods ARP queries and traceroute packets—not the initial packet—on all interfaces. When ARP or traceroute flooding is used, the original packet is dropped. The device broadcasts an ARP or ICMP query to all other devices on the same subnetwork, requesting the device at the specified destination IP address to send back a reply. Only the device with the specified IP address replies, which provides the requestor with the MAC address of the responder.

ARP allows the device to discover the destination MAC address for a unicast packet if the destination IP address is in the same subnetwork as the ingress IP address. (The ingress IP address refers to the IP address of the last device to send the packet to the device. The device might be the source that sent the packet or a router forwarding the packet.) Traceroute allows the device to discover the destination MAC address even if the destination IP address belongs to a device in a subnetwork beyond that of the ingress IP address.

When you enable ARP queries to locate an unknown destination MAC address, traceroute requests are also enabled. You can also optionally specify that traceroute requests not be used; however, the device can then discover destination MAC addresses for unicast packets only if the destination IP address is in the same subnetwork as the ingress IP address.

Whether you enable ARP queries and traceroute requests or ARP-only queries to locate unknown destination MAC addresses, the SRX Series device performs the following series of actions:

1. The device notes the destination MAC address in the initial packet. The device adds the source MAC address and its corresponding interface to its forwarding table, if they are not already there.

2. The device drops the initial packet.

3. The device generates an ARP query packet and optionally a traceroute packet and floods those packets out all interfaces except the interface on which the initial packet arrived.

   ARP packets are sent out with the following field values:

   - Source IP address set to the IP address of the IRB
   - Destination IP address set to the destination IP address of the original packet
   - Source MAC address set to the MAC address of the IRB
   - Destination MAC address set to the broadcast MAC address (all **0xf**)

   Traceroute (ICMP echo request or ping) packets are sent out with the following field values:

   - Source IP address set to the IP address of the original packet
   - Destination IP address set to the destination IP address of the original packet
   - Source MAC address set to the source MAC address of the original packet
   - Destination MAC address set to the destination MAC address of the original packet
   - Time-to-live (TTL) set to **1**

4. Combining the destination MAC address from the initial packet with the interface leading to that MAC address, the device adds a new entry to its forwarding table.

5. The device forwards all subsequent packets it receives for the destination MAC address out the correct interface to the destination.

Layer 2 Learning and Forwarding for VLANs Acting as a Switch for a Layer 2 Trunk Port

Layer 2 learning is enabled by default. A set of VLANs, configured to function as a switch with a Layer 2 trunk port, learns unicast media access control (MAC) addresses to avoid flooding packets to the trunk port.

> **NOTE:** Traffic is not flooded back onto the interface on which it was received. However, because this "split horizon" occurs at a late stage, the packet statistics displayed by commands such as **show interfaces queue** will include flood traffic.

You can optionally disable Layer 2 learning for the entire set of VLANs as well as modify the following Layer 2 learning and forwarding properties:

- Limit the number of MAC addresses learned from the Layer 2 trunk port associated with the set of VLANs
- Modify the size of the MAC address table for the set of VLANs
- Enable MAC accounting for the set of VLANs

Understanding the Unified Forwarding Table

IN THIS SECTION

Benefits of Unified Forwarding Tables

Traditionally, forwarding tables have been statically defined and have supported only a fixed number of entries for each type of address. The unified forward table provides the following benefits:

- Enables you to allocate forwarding table resources to optimize the memory available for different address types based on the needs of your network.
- Enables you to allocate a higher percentage of memory for one type of address or another.

Using the Unified Forwarding Table to Optimize Address Storage

On the QFX5100, EX4600, EX4650, QFX5110, QFX5200, and QFX5120 switches, you can control the allocation of forwarding table memory available to store the following:

- MAC addresses—In a Layer 2 environment, the switch learns new MAC addresses and stores them in a MAC address table
- Layer 3 host entries–In a Layer 2 and Layer 3 environment, the switch learns which IP addresses are mapped to which MAC addresses; these key-value pairs are stored in the Layer 3 host table.
- Longest prefix match (LPM) table entries—In a Layer 3 environment, the switch has a routing table and the most specific route has an entry in the forwarding table to associate a prefix or netmask to a next hop. Note, however, that all IPv4 /32 prefixes and IPv6 /128 prefixes are stored in the Layer 3 host table.

UFT essentially combines the three distinct forwarding tables to create one table with flexible resource allocation. You can select one of five forwarding table profiles that best meets your network needs. Each profile is configured with different maximum values for each type of address. For example, for a switch that handles a great deal of Layer 2 traffic, such as a virtualized network with many servers and virtualized machines, you would likely choose a profile that allocates a higher percentage of memory to MAC addresses. For a switch that operates in the core of a network, participates in an IP fabric, you probably want to maximize the number of routing table entries it can store. In this case, you would choose a profile that allocates a higher percentage of memory to longest match prefixes. The QFX5200 switch supports a custom profile that allows you to partition the four available shared memory banks with a total of 128,000 entries among MAC addresses, Layer 3 host addresses, and LPM prefixes.

> **NOTE:** Support for QFX5200 switches was introduced in Junos OS Release 15.1x53-D30. The QFX5200 switch is not supported on Junos OS Release 16.1R1.

Understanding the Allocation of MAC Addresses and Host Addresses

All five profiles are supported, each of which allocates different amounts of memory for Layer 2 or Layer 3 entries, enabling you choose one that best suits the needs of your network. The QFX5200 and QFX5210 switches, however, supports different maximum values for each profile from the other switches. For more information about the custom profile, see “Configuring the Unified Forwarding Table on Switches” on page 94.

NOTE: The default profile is **l2-profile-three**, which allocates equal space for MAC Addresses and Layer 3 host addresses. On QFX5100, EX4600, QFX5110, and QFX5200 switches, the space is equal to 16,000 IPv4 entries for the LPM table, and on QFX5210 switches, the space is equal to 32,000 IPv4 entries for the LPM table. For the **lpm-profile** the LPM table size is equal to 256,000 IPv4 entries.

NOTE: Starting with Junos OS Release 18.1R1 on the QFX5210-64C switch, for all these profiles, except for the **lpm-profile** the longest prefix match (LPM) table size is equal to 32,000 IPv4 entries.

NOTE: Starting with Junos OS Release 18.3R1 on the QFX5120 and EX4650 switches, for all these profiles, except for the **lpm-profile** the longest prefix match (LPM) table size is equal to 32,000 IPv4 entries.

NOTE: On QFX5100, EX4600, EX4650, QFX5110, QFX5200, QFX5120, and QFX5210-64C switches, IPv4 and IPv6 host routes with ECMP next hops are stored in the host table.

BEST PRACTICE: If the host or LPM table stores the maximum number of entries for any given type of entry, the entire shared table is full and is unable to accommodate *any* entries of any other type. Different entry types occupy different amounts of memory. For example, an IPv6 unicast address occupies twice as much memory as an IPv4 unicast address, and an IPv6 multicast address occupies four times as much memory as an IPv4 unicast address.

Table 24 on page 83 lists the profiles you can choose and the associated maximum values for the MAC address and host table entries on QFX5100 and EX4600 switches.

Table 39: Unified Forwarding Table Profiles on QFX5100 and EX4600 Switches

Profile Name	MAC Table	Host Table (unicast and multicast addresses)					
	MAC Addresses	IPv4 unicast	IPv6 unicast	IPv4 (*, G)	IPv4 (S, G)	IPv6 (*, G)	IPv6 (S, G)

Table 39: Unified Forwarding Table Profiles on QFX5100 and EX4600 Switches *(continued)*

Profile Name	MAC Table	Host Table (unicast and multicast addresses)					
l2-profile-one	288K	16K	8K	8K	8K	4K	4K
l2-profile-two	224K	80K	40K	40K	40K	20K	20K
l2-profile-three (default)	160K	144K	72K	72K	72K	36K	36K
l3-profile	96K	208K	104K	104K	104K	52K	52K
lpm-profile	32K	16K	8K	8K	8K	4K	4K
lpm-profilewith **unicast-in-lpm** option	32K	(stored in LPM table)	(stored in LPM table)	8K	8K	4K	4K

Table 25 on page 84 lists the profiles you can choose and the associated maximum values for the MAC address and host table entries on QFX5110 switches.

Table 40: Unified Forwarding Table Profiles on QFX5110 Switches

Profile Name	MAC Table	Host Table (unicast and multicast addresses)					
	MAC Addresses	IPv4 unicast	IPv6 unicast	IPv4 (*, G)	IPv4 (S, G)	IPv6 (*, G)	IPv6 (S, G)
l2-profile-one	288K	16K	8K	8K	8K	4K	4K
l2-profile-two	224K	80K	40K	40K	40K	20K	20K
l2-profile-three (default)	160K	144K	72K	72K	72K	36K	36K
l3-profile	96K	208K	104K	104K	104K	52K	52K

Table 26 on page 84 lists the LPM table size variations for the QFX5110 switch depending on the prefix entries.

Table 41: LPM Table Size Variations on QFX5110 Switches

Profile Name	Prefix Entries		
num-65-127-prefix	**IPv4 LPM<= /32**	**IPv6 LPM <= /64**	**IPv6 LPM > /64**

Table 41: LPM Table Size Variations on QFX5110 Switches *(continued)*

Profile Name	Prefix Entries		
0	16K	8K	0K
1	12K	6K	1K
2	8K	4K	2K
3	4K	2K	3K
4	0K	0K	4K

Table 27 on page 85 lists the profiles you can choose and the associated maximum values for the MAC address and host table entries on QFX5200-32C switches.

Table 42: Unified Forwarding Table Profiles on QFX5200-32C Switches

Profile Name	MAC Table	Host Table (unicast and multicast addresses)						
	MAC Addresses	IPv4 unicast	IPv6 unicast	IPv4 (*, G)	IPv4 (S, G)	IPv6 (*, G)	IPv6 (S, G)	Exact Match
l2-profile-one	136K	8K	4K	4K	4K	2K	2K	0
l2-profile-two	104K	40K	20K	20K	20K	10K	10K	0
l2-profile-three (default)	72K	72K	36K	36K	36K	18K	18K	0
l3-profile	40K	104K	52K	52K	52K	26K	26K	0
lpm-profile	8K	8K	4K	4K	4K	2K	2K	0

Table 28 on page 85 lists the profiles you can choose and the associated maximum values for the MAC address and host table entries on QFX5200-48Y switches.

Table 43: Unified Forwarding Table Profiles on QFX5200-48Y Switches

Profile Name	MAC Table	Host Table (unicast and multicast addresses)					
	MAC Addresses	IPv4 unicast	IPv6 unicast	IPv4 (*, G)	IPv4 (S, G)	IPv6 (*, G)	IPv6 (S, G)

Table 43: Unified Forwarding Table Profiles on QFX5200-48Y Switches *(continued)*

Profile Name	MAC Table	Host Table (unicast and multicast addresses)					
l2-profile-one	136K	8K	4K	4K	4K	2K	2K
l2-profile-two	104K	40K	20K	20K	20K	10K	10K
l2-profile-three (default)	72K	72K	36K	36K	36K	18K	18K
l3-profile	40K	104K	52K	52K	52K	26K	26K
lpm-profile	8K	8K	4K	4K	4K	2K	2K

Table 29 on page 86 lists the LPM table size variations for the QFX5200-48Y switch depending on the prefix entries.

Table 44: LPM Table Size Variations on QFX5200-48Y Switches

Profile Name	Prefix Entries		
num-65-127-prefix	**IPv4 LPM<= /32**	**IPv6 LPM <= /64**	**IPv6 LPM > /64**
0	16K	8K	0K
1	12K	6K	1K
2	8K	4K	2K
3	40K	2K	3K
4	0K	0K	4K

Table 30 on page 86 lists the profiles you can choose and the associated maximum values for the MAC address and host table entries on QFX5210-64C switches.

Table 45: Unified Forwarding Table Profiles on QFX5210-64C Switches

Profile Name	MAC Table	Host Table (unicast and multicast addresses)						
	MAC Addresses	IPv4 unicast	IPv6 unicast	IPv4 (*, G)	IPv4 (S, G)	IPv6 (*, G)	IPv6 (S, G)	Exact Match

Table 45: Unified Forwarding Table Profiles on QFX5210-64C Switches *(continued)*

Profile Name	MAC Table	Host Table (unicast and multicast addresses)						
l2-profile-one	264K	8K	4K	4K	4K	2K	2K	0K
l2-profile-two	200K	72K	36K	36K	36K	18K	18K	0K
l2-profile-three (default)	136K	136K	72K	72K	72K	36K	36K	0K
l3-profile	72K	200K	100K	100K	100K	50K	50K	0K

Table 31 on page 87 lists the profiles you can choose and the associated maximum values for the MAC address and host table entries on QFX5120 and EX4650 switches.

Table 46: Unified Forwarding Table Profiles on QFX5120 and EX4650 Switches

Profile Name	MAC Table	Host Table (unicast and multicast addresses)					
	MAC Addresses	IPv4 unicast	IPv6 unicast	IPv4 (*, G)	IPv4 (S, G)	IPv6 (*, G)	IPv6 (S, G)
l2-profile-one	288K	16K	8K	8K	8K	4K	4K
l2-profile-two	224K	80K	40K	40K	40K	20K	20K
l2-profile-three (default)	160K	144K	72K	72K	72K	36K	36K
l3-profile	96K	208K	104K	104K	104K	52K	52K

Table 32 on page 87 lists the LPM table size variations for the QFX5210-64C switch depending on the prefix entries.

Table 47: LPM Table Size Variations on QFX5210-64C Switches

Profile Name	Prefix Entries		
num-65-127-prefix	**IPv4 LPM<= /32**	**IPv6 LPM <= /64**	**IPv6 LPM > /64**
0	32K	16K	0K
1	28K	14K	1K

Table 47: LPM Table Size Variations on QFX5210-64C Switches *(continued)*

Profile Name	Prefix Entries		
2	24K	12K	2K
3	20K	10K	3K
4	0K	0K	4K

Table 33 on page 88 lists the Layer 3 Defip table size variations for the QFX5120 and EX4650 switches depending on the changing IPv6/128 prefix entries.

Table 48: LPM Table Size Variations on QFX5210-64C and EX4650 Switches

Profile Name	Prefix Entries		
num-65-127-prefix	IPv4 LPM<= /32	IPv6 LPM <= /64	IPv6 LPM > /64
0	32K	16K	0K
2	24K	12K	2K
4	16K	8K	4K
6	8K	4K	6K
8	0K	0K	8K

Understanding Ternary Content Addressable Memory (TCAM) and Longest Prefix Match Entries

You can further customize non-LPM profiles by configuring the space available for ternary content addressable memory (TCAM) to allocate more memory for longest prefix match entries. You can change the number of entries allocated to these IPv6 addresses, essentially allocating more or less space for LPM IPv4 entries with any prefix length or IPv6 entries with prefix lengths of 64 of shorter. For more information about how to change the default parameters of the TCAM memory space for LPM entries, see “Configuring the Unified Forwarding Table on Switches” on page 94.

NOTE: The option to adjust TCAM space is not supported on the longest prefix match (LPM) or custom profiles. However, for the LPM profile, you can configure TCAM space not to allocate any memory for IPv6 entries with prefix lengths of 65 or longer, thereby allocating that memory space only for IPv4 routes or IP routes with prefix lengths equal to or less than 64 or a combination of the two types of prefixes.

NOTE: Starting with Junos OS Release 18.1R1 on QFX5210 switches, you can configure TCAM space to allocate a maximum of 8,000 IPv6 entries with prefix lengths of 65 or longer. The default value is 2,000 entries. Starting with Junos OS Release 13.2X51-D15, you can configure TCAM space to allocate a maximum of 4,000 IPv6 entries with prefix lengths of 65 or longer. The default value is 1,000 entries. Previous to Junos OS Release 13.2X51-D15, you could allocate only a maximum of 2,048 entries for IPv6 the IPv6 prefixes with lengths in the range /65 to /127 range. The default value was 16 entries for these types of IPv6 prefixes.

On Junos OS Releases 13.2x51-D10 and 13.2x52D10, the procedure to change the default value of 16 entries differs from later releases, where the maximum and default values are higher. For more information about that procedure, see "Configuring the Unified Forwarding Table on Switches" on page 94

Host Table Example for Profile with Heavy Layer 2 Traffic

Table 34 on page 89 lists various valid combinations that the host table can store if you use the **l2-profile-one** profile on QFX5100 and EX4600 switches. This profile allocates the percentage of memory to Layer 2 addresses. Note that the default values might be different on other switches. Each row in the table represents a case in which the host table is full and cannot accommodate any more entries.

Table 49: Example Host Table Combinations Using l2-profile-one on QFX5100 and EX4600 Switches

IPv4 unicast	IPv6 unicast	IPv4 multicast (*, G)	IPv4 multicast (S, G)	IPv6 multicast (*, G)	IPv6 multicast (S, G)
16K	0	0	0	0	0
12K	2K	0	0	0	0
12K	0	2K	2K	0	0
8K	4K	0	0	0	0
4K	2K	2K	2K	0	0

Table 49: Example Host Table Combinations Using l2-profile-one on QFX5100 and EX4600 Switches *(continued)*

IPv4 unicast	IPv6 unicast	IPv4 multicast (*, G)	IPv4 multicast (S, G)	IPv6 multicast (*, G)	IPv6 multicast (S, G)
0	4K	0	0	1K	1K

Example: Configuring a Unified Forwarding Table Custom Profile

IN THIS SECTION

Traditionally, forwarding tables have been statically defined and have supported only a fixed number of entries for each type of address. The Unified Forwarding Table (UFT) feature enables you to optimize how forwarding-table memory is allocated to best suit the needs of your network. This example shows how to configure a Unified Forwarding Table profile that enables you to partition four shared hash memory banks among three different types of forwarding-table entries: MAC addresses, Layer 3 host addresses, and longest prefix match (LPM).

The UFT feature also supports five profiles that each allocate a specific maximum amount of memory for each type of forwarding table entry. Some profiles allocate more memory to Layer 2 entries, while other profiles allocate more memory to Layer 3 or LPM entries. The maximum values for each type of entry are fixed in these profiles. With the custom profile, you can designate one or more shared memory banks to store a specific type of forwarding-table entry. You can configure as few as one or as many as four memory banks in a custom profile. The custom profile thus provides even more flexibility in enabling you to allocate forwarding-table memory for specific types of entries.

Requirements

This example uses the following hardware and software components:

- One QFX5200 switch

- Junos OS Release 15.1x53-D30 or later.

Before you configure a custom profile, be sure you have:

- Configured interfaces

Overview

The Unified Forwarding Table custom profile enables you to allocate forwarding-table entries among four banks of shared hash tables with a total memory equal to 128,000 unicast IPv4 addresses, or 32,000 entries for each bank. Specifically, you can allocate one or more of these shared banks to store a specific type of forwarding-table entry. The custom profile does not affect the dedicated hash tables. Those tables remain fixed with 8,000 entries allocated to Layer 2 addresses, the equivalent of 8,000 entries allocated to IPv4 addresses, and the equivalent of 16,000 entries allocated to longest prefix match (LPM) addresses.

In this example, you allocate two memory banks to Layer 3 host addresses, and two memory banks to LPM entries. This means that no shared hash table memory is allocated for Layer 2 addresses. Only the dedicated hash table memory is allocated for Layer 2 addresses in this scenario.

Configuration

IN THIS SECTION

To configure a custom profile for the Unified Forwarding Table feature on a QFX5200 switch that allocates two shared memory banks for Layer 3 host address and two shared memory banks for LPM entries, perform these tasks:

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode. A commit check is performed to ensure that you have allocated forwarding-table space for no more than four memory banks.

CAUTION: When you configure and commit a profile, the Packet Forwarding Engine restarts and all the data interfaces on the switch go down and come back up.

```
user@switch# set chassis forwarding-options custom-profile
user@switch# set chassis forwarding-options custom-profile l2-entries num-banks 0
user@switch# set chassis forwarding-options custom-profile l3-entries num-banks 2
user@switch# set chassis forwarding-options custom-profile lpm-entries num-banks 2
```

Configuring the Custom Profile

Step-by-Step Procedure

To create the custom profile:

1. Specify the **custom-profile** option.

```
[edit chassis forwarding-options]
user@switch# set custom-profile
```

Configuring the Allocation of Shared Memory Banks

Step-by-Step Procedure

To allocate memory for specific types of entries for the shared memory banks:

1. Specify to allocate no shared bank memory for Layer 2 entries.

```
[edit chassis forwarding-options custom-profile]
user@switch# set l2-entries num-banks 0
```

2. Specify to allocate two shared memory banks (or the equivalent of 64,000 IPv4 entries) for Layer 3 host entries.

```
[edit chassis forwarding-options custom-profile]
user@switch# set l3-entries num-banks 2
```

3. Specify to allocate two shared memory banks (or the equivalent of 64,000 IPv4 entries) for LPM entries.

```
[edit chassis forwarding-options custom-profile]
user@switch# set lpm-entries numer-banks 2
```

Results

From configuration mode, confirm your configuration by entering the show chassis forwarding-options command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@switch# show chassis forwarding-profile
custom-profile {
    l2-entries {
        num-banks 0;
    }
    l3-entries {
        num-banks 2;
    }
    lpm-entries {
        num-banks 2
    }
}
```

If you are done configuring the switch, enter **commit** from configuration mode

CAUTION: The Packet Forwarding Engine will restart and all the data interfaces on the switch will go down and come back up.

Verification

IN THIS SECTION

- Checking the Parameters of the Custom Profile | 155

Confirm that the configuration is working properly.

Checking the Parameters of the Custom Profile

Purpose

Verify that the custom profile is enabled.

Action

```
user@switch> show chassis forwarding-options
```

```
UFT Configuration:
custom-profile
Configured custom scale:
Entry type                Total scale(K)
L2(mac)                   8
L3 (unicast & multicast)        72
Exact Match                0
Longest Prefix Match (lpm)        80
num-65-127-prefix = 1K
-------------Bank details for various types of entries------------
Entry type                  Dedicated Bank Size(K)      Shared Bank Size(K)
L2 (mac)                    8                         32 * num shared banks
L3 (unicast & multicast         8                          32 * num shared banks
Exact match                  0                           16 * num shared banks
Longest Prefix match(lpm)          16                       32 * num shared banks
```

Meaning

The output shows that the custom profile is enabled as configured with two shared memory banks designated for Layer 3 host entries; two shared memory banks designated for LPM entries; and no shared memory allocated for Layer 2 entries.

The total scale(K) field shows the total allocation of memory, that is, the amount allocated through the shared memory banks plus the amount allocated through the dedicated hash tables. The amount allocated through the dedicated hash tables is fixed and cannot be changed. Therefore, Layer 2 entries have 8K of memory allocated only through the dedicated hash table. Layer 3 host entries have 64K of memory allocated through two shared memory banks plus 8K through the dedicated hash table, for a total of 72K of memory. LPM entries have 64K of memory allocated through two shared memory banks plus 16K through the dedicated hash table, for a total of 80K of memory.

Configuring the Unified Forwarding Table on Switches

IN THIS SECTION

Traditionally, forwarding tables have been statically defined and have supported only a fixed number of entries for each type of address stored in the tables. The Unified Forwarding Table feature lets you optimize how your switch allocates forwarding-table memory for different types of addresses. You can choose one of five unified forwarding table profiles. Each profile allocates a different maximum amount of memory for Layer 2, Layer 3 host, and longest prefix match (LPM) entries. In addition to selecting a profile, you can also select how much additional memory to allocate for LPM entries.

Two profiles allocate higher percentages of memory to Layer 2 addresses. A third profile allocates a higher percentage of memory to Layer 3 host address, while a fourth profile allocates a higher percentage of memory to LPM entries. There is a default profile configured that allocates an equal amount of memory to Layer 2 and Layer 3 host addresses with the remainder allocated to LPM entries. For a switch in a virtualized network that handles a great deal of Layer 2 traffic, you would choose a profile that allocates a higher percentage of memory to Layer 2 addresses. For a switch that operates in the core of the network, you would choose a profile that allocates a higher percentage of memory to LPM entries.

On QFX5200 and QFX5210-64C switches only, you can also configure a custom profile that allows you to partition shared memory banks among the different types of forwarding table entries. On QFX5200 switches, these shared memory banks have a total memory equal to 128,000 IPv4 unicast addresses. On QFX5210 switches, these shared memory banks have a total memory equal to 256,000 IPv4 unicast addresses. For more information about configuring the custom profile, see “Example: Configuring a Unified Forwarding Table Custom Profile” on page 90.

Configuring a Unified Forwarding Table Profile

To configure a unified forwarding table profile:

Specify a forwarding-table profile.

```
[edit chassis forwarding-options]
user@switch# set profile-name
```

For example, to specify the profile that allocates the highest percentage of memory to Layer 2 traffic:

```
[edit chassis forwarding-options]
user@switch# set l2-profile-one
```

CAUTION: When you configure and commit a profile, in most cases the Packet Forwarding Engine automatically restarts and all the data interfaces on the switch go down and come back up (the management interfaces are unaffected).

Starting with Junos OS Releases 14.1X53-D40, 15.1R5, and 16.1R3, for a Virtual Chassis or Virtual Chassis Fabric (VCF) comprised of EX4600 or QFX5100 switches, the Packet Forwarding Engine in member switches does not automatically restart upon configuring and committing a unified forwarding table profile change. This behavior avoids Virtual Chassis or VCF instability after the change propagates to member switches and multiple Packet Forwarding Engines automatically restart at the same time. Instead, a message is displayed at the CLI prompt and logged to the switch's system log to notify you that the profile change does not take effect until the next time you reboot the Virtual Chassis or VCF. We recommend that you plan to make profile changes only when you can perform a Virtual Chassis or VCF system reboot immediately after committing the configuration update. Otherwise, the Virtual Chassis or VCF could become inconsistent if one or more members have a problem and restart with the new configuration before a planned system reboot activates the change on all members.

NOTE: You can configure only one profile for the entire switch.

NOTE: The **l2-profile-three** is configured by default.

NOTE: If the host table stores the maximum number of entries for any given type, the entire table is full and is unable to accommodate *any* entries of any other type. Keep in mind that an IPv6 unicast address occupies twice as much memory as an IPv4 unicast address, and an IPv6 multicast address occupies four times as much memory as an IPv4 unicast address..

Configuring the Memory Allocation for Longest Prefix Match Entries

IN THIS SECTION

In addition to choosing a profile, you can further optimize memory allocation for longest prefix match (LPM) entries by configuring how many IPv6 prefixes to store with lengths from /65 through /127. The switch uses LPM entries during address lookup to match addresses to the most-specific (longest) applicable prefix. Prefixes of this type are stored in the space for ternary content addressable memory (TCAM). Changing the default parameters makes this space available for LPM entries. Increasing the amount of memory available for these IPv6 prefixes reduces by the same amount how much memory is available to store IPv4 unicast prefixes and IPv6 prefixes with lengths equal to or less than 64.

The procedures for configuring the LPM table are different, depending on which version of Junos OS you are using. In the initial releases that UFT is supported, Junos OS Releases 13.2X51-D10 and 13.2X52-10, you can only increase the amount of memory allocated to IPv6 prefixes with lengths from /65 through /127 for any profile, except for **lpm-profile**. Starting with Junos OS Release 13.2X51-D15, you can also allocate either less or no memory for IPv6 prefixes with lengths in the range /65 through /127, depending on which profile is configured. For the **lpm-profie**, however, the only change you can make to the default parameters is to allocate no memory for these types of prefixes.

Configuring the LPM Table With Junos OS Releases 13.2X51-D10 and 13.2X52-D10

In Junos OS Releases 13.2x51-D10 and 13.2X52-D10, by default, the switch allocates memory for 16 IPv6 with prefixes with lengths in the range /65 through /127. You can configure the switch to allocate more memory for IPv6 prefixes with lengths in the range /65 through /127.

To allocate more memory for IPv6 prefixes in the range /65 through /127:

1. Choose a forwarding table profile.

```
[edit chassis forwarding-options]
user@swtitch# set profile-name
```

For example, to specify the profile that allocates the highest percentage of memory to Layer 2 traffic:

```
[edit chassis forwarding-options]
user@swtitch# set l2-profile-one
```

2. Select how much memory to allocate for IPv6 prefixes in the range /65 thorugh 127.

```
[edit chassis forwarding-options profile-name]
user@swtitch# set num-65-127-prefix number
```

For example, to specify to allocate memory for 32 IPv6 prefixes in the range /65 through 127:

```
[edit chassis forwarding-options l2-profile-one]
user@switch# set num-65-127-prefix 2
```

NOTE: When you configure and commit the **num-65-127-prefix *number*** statement, all the data interfaces on the switch restart. The management interfaces are unaffected.

The **num-65-127-prefix *number*** statement is not supported on the **lpm-profile**.

Configuring the LPM Table With Junos OS Release 13.2x51-D15 and Later

IN THIS SECTION

Configuring Layer 2 and Layer 3 Profiles With Junos OS Release 13.2x51-D15 or Later

Starting in Junos OS Release 13.2X51-D15, you can configure the switch to allocate forwarding table memory for as many as 4,000 IPv6 prefixes with lengths in the range /65 through /127 for any profile other than the **lpm-profile** or **custom-profile**. You can also specify to allocate no memory for these IPv6

entries. The default is 1,000 entries for IPv6 prefixes with lengths in the range /65 through /127. Previously, the maximum you could configure was for 2,048 entries for IPv6 prefixes with lengths in the range /65 through /127. The minimum number of entries was previously 16, which was the default.

To specify how much forwarding table memory to allocate for IPv6 prefixes with length in the range /65 through /127:

1. Choose a forwarding table profile.

   ```
   [edit chassis forwarding-options]
   user@swtitch# set profile-name
   ```

   For example, to specify the profile that allocates the highest percentage of memory to Layer 2 traffic:

   ```
   [edit chassis forwarding-options]
   user@swtitch# set l2-profile-one
   ```

2. Select how much memory to allocate for IPv6 prefixes in the range /65 thorugh 127.

   ```
   [edit chassis forwarding-options profile-name]
   user@swtitch# set num-65-127-prefix number
   ```

   For example, to specify to allocate memory for 2,000 IPv6 prefixes in the range /65 through 127:

   ```
   [edit chassis forwarding-options l2-profile-one]
   user@switch# set num-65-127-prefix 2
   ```

Starting with Junos OS Release 13.2X51-D15, you can use the **num-65-127-prefix** statement to allocate entries. Table 35 on page 99 shows the numbers of entries that you can allocate. Each row represents a case in which the table is full and cannot accommodate any more entries.

Table 50: LPM Table Combinations for L2 and L3 profiles With Junos OS 13.2X51-D15 and Later

num-65-127-prefix Value	IPv4 Entries	IPv6 Entries (Prefix <= 64)	IPv6 Entries (Prefix >= 65)
0	16K	8K	0K
1 (default)	12K	6K	1K
2	8K	4K	2K
3	4K	2K	3K

Table 50: LPM Table Combinations for L2 and L3 profiles With Junos OS 13.2X51-D15 and Later *(continued)*

4	0K	0K	4K

CAUTION: When you configure and commit a profile change with the **num-65-127-prefix *number*** statement, the Packet Forwarding Engine automatically restarts and all the data interfaces on the switch go down and come back up (the management interfaces are unaffected).

However, starting with Junos OS Releases 14.1X53-D40, 15.1R5, and 16.1R3, Packet Forwarding Engines on switches in a Virtual Chassis or Virtual Chassis Fabric (VCF) do not automatically restart upon configuring a unified forwarding table profile change. This behavior avoids Virtual Chassis or VCF instability after the change propagates to member switches and multiple Packet Forwarding Engines automatically restart at the same time. Instead, a message is displayed at the CLI prompt and logged to the switch's system log to notify you that the profile change does not take effect until the next time you reboot the Virtual Chassis or VCF. We recommend that you plan to make profile changes only when you can perform a Virtual Chassis or VCF system reboot immediately after committing the configuration update. Otherwise, the Virtual Chassis or VCF could become inconsistent if one or more members have a problem and restart with the new configuration before a planned system reboot activates the change on all members.

Configuring the lpm-profile With Junos OS Release 13.2x51-D15 and Later

Starting with Junos OS Release 13.2X51-D15 you can configure the **lpm-profile** profile not to allocate any memory for IPv6 entries with prefix lengths from /65 through /127. These are the default maximum values allocated for LPM memory for the **lpm-profile** by address type:

- 128K of IPv4 prefixes
- 16K of IPv6 prefixes (all lengths)

NOTE: The memory allocated for each address type represents the maximum default value for all LPM memory.

To configure the **lpm-profile** not to allocate forwarding-table memory for IPv6 entries with prefixes from /65 through /127, thus allocating more memory for IPv4:

Specify to disable forwarding-table memory for IPv6 prefixes with lengths in the range /65 through /127.

```
[edit chassis forwarding-options lpm-profile]
user@switch# set prefix-65-127-disable
```

For example, on the QFX5100 and EX4600 switches only, if you use the **prefix-65-127-disable** option, each of the following combinations are valid:

- 100K IPv4 and 28K IPv6 /64 or shorter prefixes.
- 64K IPv4 and 64K IPv6 /64 or shorter prefixes.
- 128K IPv4 and 0K IPv6 /64 or shorter prefixes.
- 0K IPv4 and 128K IPv6 /64 or shorter prefixes.

NOTE: On the QFX5200 switches, when you configure the **prefix-65-127-disable** statement, the maximum number of IPv6 entries with prefixes equal to or shorter than 64 is 98,000.

Configuring the lpm-profile With Junos OS Release 14.1x53-D30 and Later

Starting in Junos OS Release 15.1X53-D30, you can configure the **lpm-profile** profile to store unicast IPv4 and IPv6 host addresses in the LPM table , thereby freeing memory in the host table. Unicast IPv4 and IPv6 addresses are stored in the LPM table instead of the host table, as shown in Table 36 on page 101 for QFX5100 and EX4600 switches. (Platform support depends on the Junos OS release in your installation.) You can use this option in conjunction with the option to allocate no memory in the LPM table for IPv6 entries with prefix lengths in the range /65 through /127. Together, these options maximize the amount of memory available for IPv4 unicast entries and IPv6 entries with prefix lengths equal to or less than 64.

Table 51: lpm-profile with unicast-in-lpm Option for QFX5100 and EX4600 Switches

prefix-65-127-disable	MAC Table	Host Table (multicast addresses)						LPM Table unicast addresses)		
	MAC	IPv4 unicast	IPv6 unicast	IPv4 (*, G)	IPv4 (S, G)	IPv6 (*, G)	IPv6 (S, G)	IPv4 unicast	IPv6 unicast (</65)	IPv6 unicast (>/64)
No	32K	0	0	8K	8K	4K	4K	128K	16K	16K
Yes	32K	0	0	8K	8K	4K	4K	128K	128K	0

Starting with Junos Release 18.1R1, you cannot set configure a prefix for the **num-65-127-prefix** statement on non-LPM profiles. You can only enable or disable the **prefix-65-127-disable** statement for the **lpm-profile**.

Table 37 on page 102 lists the situations in which the **prefix-65-127-disable** statement should be enabled or disabled.

Table 52: LPM Table Size Variations on QFX5200-48Y Switches

Profile Name	Prefix Entries		
num-65-127-prefix	**IPv4 <= /32**	**IPv6 <= /64**	**IPv6 > /64**
Enabled	> 128K (minimum guaranteed)	98K	0K
Disabled	128K	16K	16K

On QFX5120 and EX4600 switches, you cannot set configure a prefix for the **num-65-127-prefix** statement on non-LPM profiles. You can only enable or disable the **prefix-65-127-disable** statement for the **lpm-profile**

Table 38 on page 102 lists the situations in which the **prefix-65-127-disable** statement should be enabled or disabled.

Table 53: LPM Table Size Variations on QFX5120 and EX4650 Switches

Profile Name	Prefix Entries		
prefix-65-127-disable	**IPv4 <= /32**	**IPv6 <= /64**	**IPv6 > /64**
Enabled	351K (360,000 approximate)	168K (172,000 approximate)	0K
Disabled	168K (172,000 approximate)	64K (65,524 approximate)	64K (65,524 approximate)

Note that all entries in each table share the same memory space. If a table stores the maximum number of entries for any given type, the entire shared table is full and is unable to accommodate any entries of any other type. For example, if you use the the **unicast-in-lpm** option and there are 128K IPv4 unicast addresses stored in the LPM table, the entire LPM table is full and no IPv6 addresses can be stored. Similarly, if you use the **unicast-in-lpm** option but do not use the **prefix-65-127-disable** option, and 16K IPv6 addresses with prefixes shorter than /65 are stored, the entire LPM table is full and no additional addresses (IPv4 or IPv6) can be stored.

To configure the **lpm-profile** to store unicast IPv4 entries and IPv6 entries with prefix lengths equal to or less than 64 in the LPM table:

1. Specify the option to store these entries in the LPM table.

```
[edit chassis forwarding-options lpm-profile]
user@switch# set unicast-in-lpm
```

2. (Optional) Specify to allocate no memory for in the LPM table for IPv6 prefixes with length in the range /65 through /127:

```
[edit chassis forwarding-options lpm-profile]
user@switch# set prefix-65-127-disable
```

Configuring Non-LPM Profiles on QFX5120 and EX4650 Switches

For non-LPM profiles, each profile provides the option of reserving a portion of the 16K L3-defip table to store IPv6 Prefixes > 64. Because these are 128-bit prefixes, you can have maximum of 8k IPv6/128 entries in the l3-defip table.

1. Choose a forwarding table profile.

```
[edit chassis forwarding-options]
user@swtitch# set profile-name
```

For example, to specify the profile that allocates the highest percentage of memory to Layer 3 traffic:

```
[edit chassis forwarding-options]
user@swtitch# set l3-profile
```

2. Select how much memory to allocate for IPv6 prefixes in the range /65 thorugh 127.

```
[edit chassis forwarding-options profile-name]
user@swtitch# set num-65-127-prefix number
```

For example, to specify to allocate memory for 2,000 IPv6 prefixes in the range /65 through 127:

You can choose between 0 and 4, 1 being the default.

```
[edit chassis forwarding-options l3-profile]
user@switch# set num-65-127-prefix 1
```

Configuring Forwarding Mode on Switches

By default, packets packets are forwarded using store-and-forward mode. You can configure all the interfaces to use cut-through mode instead.

To enable cut-through switching mode, enter the following statement:

```
[edit forwarding-options]
user@switch# set cut-through
```

SEE ALSO

cut-through | 1104

Disabling Layer 2 Learning and Forwarding

Disabling dynamic MAC learning on an MX Series router or an EX Series switch prevents all the logical interfaces on the router or switch from learning source and destination MAC addresses.

To disable MAC learning for an MX Series router or an EX Series switch, include the **global-no-mac-learning** statement at the **[edit protocols l2-learning]** hierarchy level:

```
[edit protocols l2-learning]
global-no-mac-learning;
```

For information about how to configure a virtual switch, see *Configuring a Layer 2 Virtual Switch* .

SEE ALSO

Understanding Layer 2 Learning and Forwarding

Configuring the MAC Table Timeout Interval

Enabling MAC Accounting

Limiting the Number of MAC Addresses Learned from Each Logical Interface

CHAPTER 9

Configuring Bridging and VLANs

Bridging and VLANs

IN THIS SECTION

Understanding Bridging and VLANs on Switches

IN THIS SECTION

Network switches use Layer 2 bridging protocols to discover the topology of their LAN and to forward traffic toward destinations on the LAN. This topic explains the following concepts regarding bridging and VLANs:

> **NOTE:** For Ethernet, Fast Ethernet, Tri-Rate Ethernet copper, Gigabit Ethernet, 10-Gigabit Ethernet, and aggregated Ethernet interfaces supporting VPLS, the Junos OS supports a subset of the IEEE 802.1Q standard for channelizing an Ethernet interface into multiple logical interfaces, allowing many hosts to be connected to the same Gigabit Ethernet switch, but preventing them from being in the same routing or bridging domain.

Benefits of Using VLANs

In addition to reducing traffic and thereby speeding up the network, VLANs have the following advantages:

- VLANs provide segmentation services traditionally provided by routers in LAN configurations, thereby reducing hardware equipment costs.
- Packets coupled to a VLAN can be reliably identified and sorted into different domains. You can contain broadcasts within parts of the network, thereby freeing up network resources. For example, when a DHCP server is plugged into a switch and starts broadcasting its presence, you can prevent some hosts from accessing it by using VLANs to split up the network.
- For security issues, VLANs provide granular control of the network because each VLAN is identified by a single IP subnetwork. All packets passing in and out of a VLAN are consistently tagged with the VLAN ID of that VLAN, thereby providing easy identification, because a VLAN ID on a packet cannot be altered. (For a switch that runs Junos OS that does not support ELS, we recommend that you avoid using 1 as a VLAN ID, because that ID is a default value.)

- VLANs react quickly to host relocation—this is also due to the persistent VLAN tag on packets.
- On an Ethernet LAN, all network nodes must be physically connected to the same network. In VLANs, the physical location of nodes is not important—you can group network devices in any way that makes sense for your organization, such as by department or business function, types of network nodes, or physical location.

History of VLANs

Ethernet LANs were originally designed for small, simple networks that primarily carried text. However, over time, the type of data carried by LANs grew to include voice, graphics, and video. This more complex data, when combined with the ever-increasing speed of transmission, eventually became too much of a load for the original Ethernet LAN design. Multiple packet collisions were significantly slowing down the larger LANs.

The IEEE 802.1D-2004 standard helped evolve Ethernet LANs to cope with the higher data and transmission requirements by defining the concept of *transparent bridging* (generally called simply *bridging*). Bridging divides a single physical LAN (now called a single *broadcast domain*) into two or more virtual LANs, or VLANs. Each *VLAN* is a collection of some of the LAN nodes grouped together to form individual broadcast domains.

When VLANs are grouped logically by function or organization, a significant percentage of data traffic stays within the VLAN. This relieves the load on the LAN because all traffic no longer has to be forwarded to all nodes on the LAN. A VLAN first transmits packets within the VLAN, thereby reducing the number of packets transmitted on the entire LAN. Because packets whose origin and destination are in the same VLAN are forwarded only within the local VLAN, packets that are not destined for the local VLAN are the only ones forwarded to other broadcast domains. This way, bridging and VLANs limit the amount of traffic flowing across the entire LAN by reducing the possible number of collisions and packet retransmissions within VLANs and on the LAN as a whole.

How Bridging of VLAN Traffic Works

Because the objective of the IEEE 802.1D-2004 standard was to reduce traffic and therefore reduce potential transmission collisions for Ethernet, a system was implemented to reuse information. Instead of having a switch go through a location process every time a frame is sent to a node, the transparent bridging protocol allows a switch to record the location of known nodes. When packets are sent to nodes, those destination node locations are stored in address-lookup tables called *Ethernet switching tables*. Before sending a packet, a switch using bridging first consults the switching tables to see if that node has already been located. If the location of a node is known, the frame is sent directly to that node.

Transparent bridging uses five mechanisms to create and maintain Ethernet switching tables on the switch:

- Learning
- Forwarding

- Flooding
- Filtering
- Aging

The key bridging mechanism used by LANs and VLANs is *learning*. When a switch is first connected to an Ethernet LAN or VLAN, it has no information about other nodes on the network. As packets are sent, the switch learns the embedded MAC addresses of the sending nodes and stores them in the Ethernet switching table, along with two other pieces of information—the interface (or port) on which the traffic was received on the destination node and the time the address was learned.

Learning allows switches to then do *forwarding*. By consulting the Ethernet switching table to see whether the table already contains the frame's destination MAC address, switches save time and resources when forwarding packets to the known MAC addresses. If the Ethernet switching table does not contain an entry for an address, the switch uses flooding to learn that address.

Flooding finds a particular destination MAC address without using the Ethernet switching table. When traffic originates on the switch and the Ethernet switching table does not yet contain the destination MAC address, the switch first floods the traffic to all other interfaces within the VLAN. When the destination node receives the flooded traffic, it can send an acknowledgment packet back to the switch, allowing it to learn the MAC address of the node and add the address to its Ethernet switching table.

Filtering, the fourth bridging mechanism, is how broadcast traffic is limited to the local VLAN whenever possible. As the number of entries in the Ethernet switching table grows, the switch pieces together an increasingly complete picture of the VLAN and the larger LAN—it learns which nodes are in the local VLAN and which are on other network segments. The switch uses this information to filter traffic. Specifically, for traffic whose source and destination MAC addresses are in the local VLAN, filtering prevents the switch from forwarding this traffic to other network segments.

To keep entries in the Ethernet switching table current, the switch uses a fifth bridging mechanism, *aging*. Aging is the reason that the Ethernet switching table entries include timestamps. Each time the switch detects traffic from a MAC address, it updates the timestamp. A timer on the switch periodically checks the timestamp, and if it is older than a user-configured value, the switch removes the node's MAC address from the Ethernet switching table. This aging process eventually flushes unavailable network nodes out of the Ethernet switching table.

Packets Are Either Tagged or Untagged

When an Ethernet LAN is divided into VLANs, each VLAN is identified by a unique 802.1Q ID. The number of available VLANs and VLAN IDs are listed below:

- On a switch running ELS software, you can configure 4093 VLANs using VLAN IDs 1 through 4094, while VLAN IDs 0 and 4095 are reserved by Junos OS and cannot be assigned.
- On a switch running non-ELS software, you can configure 4091 VLANs using VLAN IDs 1-4094.

Ethernet packets include a tag protocol identifier (TPID) EtherType field, which identifies the protocol being transported. When a device within a VLAN generates a packet, this field includes a value of 0x8100, which indicates that the packet is a VLAN-tagged packet. The packet also has a VLAN ID field that includes the unique 802.1Q ID, which identifies the VLAN to which the packet belongs.

Junos OS switches support the TPID value 0x9100 for Q-in-Q on switches. In addition to the TPID EtherType value of 0x8100, EX Series switches that do not support the Enhanced Layer 2 Software (ELS) configuration style also support values of 0x88a8 (Provider Bridging and Shortest Path Bridging) and 0x9100 (Q-inQ).

For a simple network that has only a single VLAN, all packets include a default 802.1Q tag, which is the only VLAN membership that does not mark the packet as tagged. These packets are untagged packets.

> **NOTE:** Q-in-Q tunnelling is not supported on NFX150 devices.

Switch Interface Modes—Access, Trunk, or Tagged Access

Ports, or interfaces, on a switch operate in one of three modes:

- Access mode
- Trunk mode
- Tagged-access mode

Access Mode

An interface in access mode connects a switch to a single network device, such as a desktop computer, an IP telephone, a printer, a file server, or a security camera. Access interfaces accept only untagged packets.

By default, when you boot a switch that runs Junos OS that does not support ELS and use the factory default configuration, or when you boot such a switch and do not explicitly configure a port mode, all interfaces on the switch are in access mode and accept only untagged packets from the VLAN named **default**. You can optionally configure another VLAN and use that VLAN instead of **default**.

On a switch that supports ELS, the VLAN named **default** is not supported. Therefore, on such switches, you must explicitly configure at least one VLAN, even if your network is simple and you want only one broadcast domain to exist. After you assign an interface to a VLAN, the interface functions in access mode.

For switches that run either type of software, you can also configure a trunk port or interface to accept untagged packets from a user-configured VLAN. For details about this concept (native VLAN), see “Trunk Mode and Native VLAN” on page 173.

Trunk Mode

Trunk mode interfaces are generally used to connect switches to one another. Traffic sent between switches can then consist of packets from multiple VLANs, with those packets multiplexed so that they can be sent over the same physical connection. Trunk interfaces usually accept only tagged packets and use the VLAN ID tag to determine both the packets' VLAN origin and VLAN destination.

On a switch that runs software that does not support ELS, an untagged packet is not recognized on a trunk port unless you configure additional settings on that port.

On a switch that runs Junos OS that supports ELS, a trunk port recognizes untagged control packets for protocols such as the Link Aggregation Control Protocol (LACP) and the Link Layer Discovery Protocol (LLDP). However, the trunk port does not recognize untagged data packets unless you configure additional settings on that port.

> **NOTE:** LACP is not supported on NFX150 devices.

In the rare case where you want untagged packets to be recognized by a trunk port on switches that run either type of software, you must configure the single VLAN on a trunk port as a *native VLAN*. For more information about native VLANs, see "Trunk Mode and Native VLAN" on page 173.

Trunk Mode and Native VLAN

On a switch that runs Junos OS that does not support ELS, a trunk port does not recognize packets that do not include VLAN tags, which are also known an untagged packets. On a switch that runs Junos OS that supports ELS, a trunk port recognizes untagged control packets, but it does not recognize untagged data packets. With native VLAN configured, untagged packets that a trunk port normally does not recognize are sent over the trunk interface. In a situation where packets pass from a device, such as an IP phone or printer, to a switch in access mode, and you want those packets sent from the switch over a trunk port, use native VLAN mode. Create a native VLAN by configuring a VLAN ID for it, and specify that the trunk port is a member of the native VLAN.

The switch's trunk port will then treat those packets differently than the other tagged packets. For example, if a trunk port has three VLANs, 10, 20, and 30, assigned to it with VLAN 10 being the native VLAN, packets on VLAN 10 that leave the trunk port on the other end have no 802.1Q header (tag).

There is another native VLAN option for switches that do not support ELS. You can have the switch add and remove tags for untagged packets. To do this, you first configure the single VLAN as a native VLAN on a port attached to a device on the edge. Then, assign a VLAN ID tag to the single native VLAN on the port connected to a device. Last, add the VLAN ID to the trunk port. Now, when the switch receives the untagged packet, it adds the ID you specified and sends and receives the tagged packets on the trunk port configured to accept that VLAN.

Tagged-Access Mode

Only switches that run Junos OS not using the ELS configuration style support tagged-access mode. Tagged-access mode accommodates cloud computing, specifically scenarios including virtual machines or virtual computers. Because several virtual computers can be included on one physical server, the packets generated by one server can contain an aggregation of VLAN packets from different virtual machines on that server. To accommodate this situation, tagged-access mode reflects packets back to the physical server on the same downstream port when the destination address of the packet was learned on that downstream port. Packets are also reflected back to the physical server on the downstream port when the destination has not yet been learned. Therefore, the third interface mode, tagged access, has some characteristics of access mode and some characteristics of trunk mode:

- Like access mode, tagged-access mode connects the switch to an access layer device. Unlike access mode, tagged-access mode is capable of accepting VLAN tagged packets.
- Like trunk mode, tagged-access mode accepts VLAN tagged packets from multiple VLANs. Unlike trunk port interfaces, which are connected at the core/distribution layer, tagged-access port interfaces connect devices at the access layer.

  Like trunk mode, tagged-access mode also supports native VLAN.

> **NOTE:** Control packets are never reflected back on the downstream port.

Maximum VLANs and VLAN Members Per Switch

Starting in Junos OS Release 17.3 on QFX10000 switches, the number of vmembers has increased to 256k for integrated routing and bridging interfaces and aggregated Ethernet interfaces.

The number of VLANs supported per switch varies for each switch. Use the configuration-mode command **set vlans *vlan-name* vlan-id ?** to determine the maximum number of VLANs allowed on a switch. You cannot exceed this VLAN limit because you have to assign a specific ID number when you create a VLAN—you could overwrite one of the numbers, but you cannot exceed the limit.

You can, however, exceed the recommended VLAN member maximum for a switch.

On a switch that runs Junos OS that does not support the ELS configuration style, the maximum number of VLAN members allowed on the switch is eight times the maximum number of VLANs that the switch supports (vmember limit = vlan max * 8). If the configuration of the switch exceeds the recommended VLAN member maximum, a warning message appears when you commit the configuration. If you commit the configuration despite the warning, the commit succeeds, but there is a risk of the Ethernet switching process (eswd) failing as a result of memory allocation failure.

On most switches running Junos OS that supports ELS, the maximum number of VLAN members allowed on the switch is 24 times the maximum number of VLANs that the switch supports (vmember limit = vlan

max * 24). If the configuration of the switch exceeds the recommended VLAN member maximum, a warning message appears in the system log (syslog).

On an EX Series switch that runs Junos OS that supports ELS, the maximum number of VLAN members allowed on the switch is as follows:

- EX4300—24 times the maximum number of VLANs that the switch supports (vmember limit = *vlan max* * 24)
- EX3400—16 times the maximum number of VLANs that the switch supports (vmember limit = *vlan max* * 16)
- EX2300—8 times the maximum number of VLANs that the switch supports (vmember limit = *vlan max* * 8)

A QFabric system supports up to 131,008 VLAN members (vmembers) on a single network node group, server node group, or redundant server node group. The number of vmembers is calculated by multiplying the maximum number of VLANs by 32.

For example, to calculate how many interfaces are required to support 4,000 VLANs, divide the maximum number of vmembers (128,000) by the number of configured VLANs (4,000). In this case, 32 interfaces are required.

On network Node groups and server Node groups, you can configure link aggregation groups (LAGs) across multiple interfaces. Each LAG and VLAN combination is considered a vmember.

> **NOTE:** LAG is not supported on NFX150 devices.

A Virtual Chassis Fabric supports up to 512,000 vmembers. The number of vmembers is based on the number of VLANs, and the number of interfaces configured in each VLAN.

A Default VLAN Is Configured on Most Switches

Some switches running Junos OS that do not support the ELS configuration style are preconfigured with a VLAN named **default** that does not tag packets and operates only with untagged packets. On these switches, each interface already belongs to the VLAN named **default** and all traffic uses this VLAN until you configure more VLANs and assign traffic to those VLANs.

EX Series switches that run Junos OS with the ELS configuration style do not support a default VLAN. The following EX Series switches running Junos OS not supporting the ELS configuration style are not preconfigured to belong to **default** or any other VLAN:

- Modular switches, such as the EX8200 switches and EX6200 switches
- Switches that are part of a Virtual Chassis

The reason that these switches are not preconfigured is that the physical configuration in both situations is flexible. There is no way of knowing which line cards have been inserted in either the EX8200 switch or EX6200 switch. There is also no way of knowing which switches are included in the Virtual Chassis. Switch interfaces in these two cases must first be defined as Ethernet switching interfaces. After an interface is defined as an Ethernet switching interface, the default VLAN appears in the output from the ? help and other commands.

NOTE: When a Juniper Networks EX4500 Ethernet Switch, EX4200 Ethernet Switch, EX3300 Ethernet Switch, QFX3500 or QFX3600 switch is interconnected with other switches in a Virtual Chassis configuration, each individual switch that is included as a member of the configuration is identified with a member ID. The member ID functions as an FPC slot number. When you are configuring interfaces for a Virtual Chassis configuration, you specify the appropriate member ID (0 through 9) as the slot element of the interface name. The default factory settings for a Virtual Chassis configuration include FPC 0 as a member of the default VLAN because FPC 0 is configured as part of the ethernet-switching family. In order to include FPC 1 through FPC 9 in the default VLAN, add the ethernet-switching family to the configurations for those interfaces.

NOTE: You cannot configure a default VLAN on NFX150 devices.

Assigning Traffic to VLANs

You can assign traffic on any switch to a particular VLAN by referencing either the interface port of the traffic or the MAC addresses of devices sending traffic.

NOTE: Two logical interfaces that are configured on the same physical interface cannot be mapped to the same VLAN.

Assign VLAN Traffic According to the Interface Port Source

This method is most commonly used to assign traffic to VLANs. In this case, you specify that all traffic received on a particular switch interface is assigned to a specific VLAN. You configure this VLAN assignment when you configure the switch, by using either the VLAN number (called a VLAN ID) or by using the VLAN name, which the switch then translates into a numeric VLAN ID. This method is referred to simply as creating a VLAN because it is the most commonly used method.

Assign VLAN Traffic According to the Source MAC Address

In this case, all traffic received from a specific MAC address is forwarded to a specific egress interface (next hop) on the switch. MAC-based VLANs are either static (named MAC addresses configured one at a time) or dynamic (configured using a RADIUS server).

To configure a static MAC-based VLAN on a switch that supports ELS, see "Adding a Static MAC Address Entry to the Ethernet Switching Table" on page 113.To configure a static MAC-based VLAN on a switch that does not support ELS, see "Adding a Static MAC Address Entry to the Ethernet Switching Table" on page 113.

For information about using 802.1X authentication to authenticate end devices and allow access to dynamic VLANs configured on a RADIUS server, see *Understanding Dynamic VLAN Assignment Using RADIUS Attributes*. You can optionally implement this feature to offload the manual assignment of VLAN traffic to automated RADIUS server databases.

Forwarding VLAN Traffic

To pass traffic within a VLAN, the switch uses Layer 2 forwarding protocols, including IEEE 802.1Q spanning-tree protocols.

To pass traffic between two VLANs, the switch uses standard Layer 3 routing protocols, such as static routing, OSPF, and RIP. The same interfaces that support Layer 2 bridging protocols also support Layer 3 routing protocols, providing multilayer switching.

To pass traffic from a single device on an access port to a switch and then pass those packets on a trunk port, use the native mode configuration previously discussed under "Trunk Mode" on page 173.

VLANs Communicate with Integrated Routing and Bridging Interfaces or Routed VLAN Interfaces

Traditionally, switches sent traffic to hosts that were part of the same broadcast domain (VLAN) but routers were needed to route traffic from one broadcast domain to another. Also, only routers performed other Layer 3 functions such as traffic engineering.

Switches that run Junos OS that supports the ELS configuration style perform inter-VLAN routing functions using an integrated routing and bridging (IRB) interface named irb, while switches that run Junos OS that does not support ELS perform these functions using a routed VLAN interface (RVI) named vlan. These interfaces detect both MAC addresses and IP addresses and route data to Layer 3 interfaces, thereby frequently eliminating the need to have both a switch and a router.

VPLS Ports

You can configure VPLS ports in a virtual switch instead of a dedicated routing instance of type **vpls** so that the logical interfaces of the Layer 2 VLANs in the virtual switch can handle VPLS routing instance

traffic. Packets received on a Layer 2 trunk interface are forwarded within a VLAN that has the same VLAN identifier.

SEE ALSO

- *Understanding FCoE*
- *Interfaces Overview for Switches*
- Understanding Multiple VLAN Registration Protocol (MVRP) | **787**
- Understanding Integrated Routing and Bridging | **728**

Configuring VLANs on Switches with Enhanced Layer 2 Support

Switches use VLANs to make logical groupings of network nodes with their own broadcast domains. You can use VLANs to limit the traffic flowing across the entire LAN and reduce collisions and packet retransmissions.

> **NOTE:** This task supports the Enhanced Layer 2 Software (ELS) configuration style. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50. If your switch runs software that does not support ELS, see "Configuring VLANs on Switches" on page 182.

> **NOTE:** Starting with Junos OS Release 17.1R3, on QFX10000 switches, you cannot configure an interface with both **family ethernet-switching** and **flexible-vlan-tagging**. This configuration is not supported, and a warning will be issued if you try to commit this configuration.

> **NOTE:** Two logical interfaces that are configured on the same physical interface cannot be mapped to the same VLAN.

For each endpoint on the VLAN, configure the following VLAN parameters on the corresponding interface:

1. Specify the description of the VLAN:

   ```
   [edit interfaces interface-name unit 0]
   user@switch# set description vlan-description
   ```

2. Specify the unique name of the VLAN:

   > **NOTE:** Switches that run Junos OS with the ELS configuration style do not support a default VLAN. Therefore, on such switches, you must explicitly configure at least one VLAN, even if your network is simple and you want only one broadcast domain to exist.

   > **NOTE:** On QFX5100 switches running Junos OS Release 14.1X53-D46 or earlier, when you configure an interface under a VLAN but do not specify the name of the VLAN, the system will not issue a commit error.

```
[edit interfaces interface-name unit 0]
user@switch# set family ethernet-switching vlan members vlan-name
```

3. Create the subnet for the VLAN:

```
[edit interfaces]
user@switch# set vlan unit 0 family inet address ip-address
```

> **NOTE:** The **family inet** option is not supported on NFX150 devices.

4. Configure the VLAN tag ID or VLAN ID list for the VLAN:

```
[edit vlans]
user@switch# set vlan-name vlan-id vlan-id-number
```

or

```
[edit vlans]
user@switch# set vlan-name vlan-id-list [vlan-ids | vlan-id--vlan-id-]
```

5. Specify a VLAN firewall filter to be applied to incoming or outgoing packets:

```
[edit vlans]
user@switch# set vlan-name filter (input | output) filter-name
```

SEE ALSO

Example: Setting Up Basic Bridging and a VLAN on Switches | 203

Configuring IRB Interfaces on Switches | 735

Configuring a VLAN

A VLAN must include a set of logical interfaces that participate in Layer 2 learning and forwarding. You can optionally configure a VLAN identifier and a Layer 3 interface for the VLAN to also support Layer 3 IP routing.

To enable a VLAN, include the following statements:

```
[edit]
vlans {
    vlan-name {
        domain-type bridge:
        interface interface-name;
        l3-interface interface-name;
        vlan-id (none | all | number);
        vlan-id-list [ vlan-id-numbers ];
        vlan-tags outer number inner number);
    }
}
```

You cannot use the slash (/) character in VLAN names. If you do, the configuration does not commit and an error is generated.

For the **vlan-id** statement, you can specify either a valid VLAN identifier or the **none** or **all** options.

To include one or more logical interfaces in the VLAN, specify an ***interface-name*** for an Ethernet interface you configured at the **[edit interfaces]** hierarchy level.

> **NOTE:** A maximum of 4096 active logical interfaces are supported for a VLAN or on each mesh group in a virtual private LAN service (VPLS) instance configured for Layer 2 bridging.

By default, each VLAN maintains a Layer 2 forwarding database that contains media access control (MAC) addresses learned from packets received on the ports that belong to the VLAN. You can modify Layer 2 forwarding properties, for example, disabling MAC learning for the entire system or a VLAN, adding static MAC addresses for specific logical interfaces, and limiting the number of MAC addresses learned by the entire system, the VLAN, or a logical interface.

You can also configure spanning tree protocols to prevent forwarding loops.

Configuring VLANs on Switches

Switches use VLANs to make logical groupings of network nodes with their own broadcast domains. You can use VLANs to limit the traffic flowing across the entire LAN and reduce collisions and packet retransmissions.

> **NOTE:** This task uses Junos OS for the QFX Series that does not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see "Configuring VLANs on Switches with Enhanced Layer 2 Support" on page 179.

For each endpoint on the VLAN, configure the following VLAN parameters on the corresponding interface:

1. Specify the description of the VLAN:

   ```
   [edit interfaces interface-name unit 0]
   user@switch# set description vlan-description
   ```

2. Specify the unique name of the VLAN:

   > **NOTE:** In a QFabric system, do not configure "default" as the name of a VLAN. Though the QFabric system will allow you to configure and commit a VLAN with the name "default" in the current software with no commit errors, it will not work. Junos OS 12.2 and onwards will not allow you to commit a VLAN with the name "default."

   ```
   [edit interfaces interface-name unit 0]
   user@switch# set family ethernet-switching vlan members vlan-name
   ```

3. Create the subnet for the VLAN:

   ```
   [edit interfaces]
   user@switch# set vlan unit 0 family inet address ip-address
   ```

4. Configure the VLAN tag ID or VLAN ID range for the VLAN:

   ```
   [edit vlans]
   user@switch# set vlan-name vlan-id vlan-id-number
   ```

   or

   ```
   [edit vlans]
   ```

```
user@switch# set vlan-name vlan-range vlan-id-low-vlan-id-high
```

5. Specify a VLAN firewall filter to be applied to incoming or outgoing packets:

```
[edit vlans]
user@switch# set vlan-name filter (input | output) filter-name
```

SEE ALSO

- Example: Setting Up Basic Bridging and a VLAN on Switches | 203
- Configuring IRB Interfaces on Switches | 735
- Creating a Series of Tagged VLANs | 355

Configuring VLANs for EX Series Switches

IN THIS SECTION

NOTE: This task uses Junos OS for EX Series switches that do not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see *Configuring VLANs for EX Series Switches with ELS Support (CLI Procedure)*. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50.

EX Series switches use VLANs to make logical groupings of network nodes with their own broadcast domains. VLANs limit the traffic flowing across the entire LAN and reduce collisions and packet retransmissions.

Why Create a VLAN?

Some reasons to create VLANs are:

- A LAN has more than 200 devices.
- A LAN has a large amount of broadcast traffic.
- A group of clients requires that a higher-than-average level of security be applied to traffic entering or exiting the group's devices.
- A group of clients requires that the group's devices receive less broadcast traffic than they are currently receiving, so that data speed across the group is increased.

Create a VLAN Using the Minimum Procedure

Two steps are required to create a VLAN:

- Uniquely identify the VLAN. You do this by assigning either a name or an ID (or both) to the VLAN. When you assign just a VLAN name, an ID is generated by Junos OS.
- Assign at least one switch port interface to the VLAN for communication. All interfaces in a single VLAN are in a single broadcast domain, even if the interfaces are on different switches. You can assign traffic on any switch to a particular VLAN by referencing either the interface sending traffic or the MAC addresses of devices sending traffic.

The following example creates a VLAN using only the two required steps. The VLAN is created with the name employee-vlan. Then, three interfaces are assigned to that VLAN so that the traffic is transmitted among these interfaces.

> **NOTE:** In this example, you could alternatively assign an ID number to the VLAN. The requirement is that the VLAN have a unique ID.

```
[edit] set vlans employee-vlan
[edit] set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members employee-vlan
[edit] set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan members employee-vlan
[edit] set interfaces ge-0/0/3 unit 0 family ethernet-switching vlan members employee-vlan
```

In the example, all users connected to the interfaces ge-0/0/1, ge-0/0/2, and ge-0/0/3 can communicate with each other, but not with users on other interfaces in this network. To configure communication between VLANs, you must configure a routed VLAN interface (RVI). See *Configuring Routed VLAN Interfaces on Switches (CLI Procedure)*.

Create a VLAN Using All of the Options

To configure a VLAN, follow these steps:

1. In configuration mode, create the VLAN by setting the unique VLAN name:

   ```
   [edit]user@switch# set vlans vlan-name
   ```

2. Configure the VLAN tag ID or VLAN ID range for the VLAN. (If you assigned a VLAN name, you do not have to do this, because a VLAN ID is assigned automatically, thereby associating the name of the VLAN to an ID number. However, if you want to control the ID numbers, you can assign both a name and an ID.)

   ```
   [edit]user@switch# set vlans vlan-name vlan-id vlan-id-number
   ```

   or

   ```
   [edit]user@switch# set vlans vlan-name vlan-range (vlan-id-low) - (vlan-id-high)
   ```

3. Assign at least one interface to the VLAN:

   ```
   [edit]user@switch# set vlans vlan-name interface interface-name
   ```

   > **NOTE:** You can also specify that a trunk interface is a member of all the VLANs that are configured on this switch. When a new VLAN is configured on the switch, this trunk interface automatically becomes a member of the VLAN.

4. (Optional) Create a subnet for the VLAN because all computers that belong to a subnet are addressed with a common, identical, most-significant-bit group in their IP address. This makes it easy to identify VLAN members by their IP addresses. To create the subnet for the VLAN:

   ```
   [edit interfaces]user@switch# set vlan unit logical-unit-number family inet address ip-address
   ```

5. (Optional) Specify the description of the VLAN:

   ```
   [edit]user@switch# set vlans vlan-name description text-description
   ```

6. (Optional) To avoid exceeding the maximum number of members allowed in a VLAN, specify the maximum time that an entry can remain in the forwarding table before it ages out:

   ```
   [edit]user@switch#  set vlans vlan-name mac-table-aging-time time
   ```

7. (Optional) For security purposes, specify a VLAN firewall filter to be applied to incoming or outgoing packets:

   ```
   [edit]user@switch#  set vlans vlan-name  filter input-or-output  filter-name
   ```

8. (Optional) For accounting purposes, enable a counter to track the number of times this VLAN is accessed:

   ```
   [edit]user@switch#  set vlans vlan-name l3-interface ingress-counting l3-interface-name
   ```

9. (Optional) For Virtual Chassis bandwidth management purposes, enable VLAN Pruning to ensure all broadcast, multicast, and unknown unicast traffic entering the Virtual Chassis on the VLAN uses the shortest possible path through the Virtual Chassis:

   ```
   [edit]
   user@switch#  set vlans vlan-name  vlan-prune
   ```

Configuration Guidelines for VLANs

Two steps are required to create a VLAN. You must uniquely identify the VLAN and you must assign at least one switch port interface to the VLAN for communication.

After creating a VLAN, all users all users connected to the interfaces assigned to the VLAN can communicate with each other but not with users on other interfaces in the network. To configure communication between VLANs, you must configure a routed VLAN interface (RVI). See *Configuring Routed VLAN Interfaces on Switches (CLI Procedure)* to create an RVI.

The number of VLANs supported per switch varies for each switch type. Use the command **set vlans id vlan-id ?** to discover the maximum number of VLANs allowed on a switch. You cannot exceed this VLAN limit because each VLAN is assigned an ID number when it is created. You can, however, exceed the recommended VLAN member maximum . To determine the maximum number of VLAN members allowed on a switch, multiply the VLAN maximum obtained using **set vlans id vlan-id ?** times 8.

If a switch configuration exceeds the recommended VLAN member maximum, you see a warning message when you commit the configuration. If you ignore the warning and commit such a configuration, the configuration succeeds but you run the risk of crashing the Ethernet switching process (eswd) due to memory allocation failure.

NOTE: When EX2300 and EX3400 ERPS switches have a VLAN-ID configured with a name under an interface hierarchy, a commit error occurs. Avoid this by configuring VLAN-IDs using numbers when they are under an interface hierarchy with ERPS configured in the switch.

SEE ALSO

Configuring VLANs for EX Series Switches (J-Web Procedure)

Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch | 226

Creating a Series of Tagged VLANs on EX Series Switches (CLI Procedure) | 357

Understanding Integrated Routing and Bridging | 728

Example: Configuring VLANs on Security Devices

IN THIS SECTION

This example shows you how to configure a VLAN.

Requirements

Before you begin:

- Determine which interfaces to use and verify that they are in switching mode. See *Understanding VLANs*.
- Determine what ports to use on the device and how to segment your network. See "Ethernet Ports Switching Overview for Security Devices" on page 989.

Overview

In this example, you create a new VLAN and then configure its attributes. You can configure one or more VLANs to perform Layer 2 switching. The Layer 2 switching functions include integrated routing and bridging (IRB) for support for Layer 2 switching and Layer 3 IP routing on the same interface. SRX Series devices can function as Layer 2 switches, each with multiple switching or broadcast domains that participate in the same Layer 2 network.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set interfaces ge-0/0/1 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members v10
set vlans v10 vlan-id 10
set vlans v10 l3-interface irb.10
set interfaces irb unit 10 family inet address 198.51.100.0/24
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure a VLAN:

1. Configure a Gigabit Ethernet interface or a 10-Gigabit Ethernet interface as a access interface:

   ```
   [edit interfaces]
   user@host# set ge-0/0/1 unit 0 family ethernet-switching interface-mode access
   ```

2. Assign an interface to the VLAN by specifying the logical interface (with the unit statement) and specifying the VLAN name as the member.

   ```
   [edit interfaces]
   user@host# set ge-0/0/1 unit 0 family ethernet-switching vlan members v10
   ```

3. Create the VLAN by setting the unique VLAN name and configuring the VLAN ID.

   ```
   [edit]
   ```

```
user@host# set vlans v10 vlan-id 10
```

4. Bind a Layer 3 interface with the VLAN.

```
[edit]
user@host# set vlans v10 l3-interface irb.10
```

5. Create the subnet for the VLAN's broadcast domain.

```
[edit interfaces]
user@host# set irb unit 10 family inet address 198.51.100.0/24
```

Results

From configuration mode, confirm your configuration by entering the **show vlans** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show vlans
v10 {
    vlan-id 10;
    l3-interface irb.10;
}
user@host# show interfaces
ge-0/0/1 {
    unit 0 {
        family ethernet-switching {
            interface-mode access;
            vlan {
                members v10;
            }
        }
    }
}
irb {
    unit 10 {
        family inet {
            address 198.51.100.0/24;
        }
    }
```

```
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Verifying VLANs

Purpose

Verify that VLANs are configured and assigned to the interfaces.

Action

From operational mode, enter the **show vlans** command.

```
user@host> show vlans
```

```
Routing instance        VLAN name             Tag         Interfaces
default-switch          default               1

default-switch          v10                   10
                                                          ge-0/0/1.0
```

Meaning

The output shows the VLAN is configured and assigned to the interface.

Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch with ELS Support

IN THIS SECTION

NOTE: This example uses Junos OS for EX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs Junos OS that does not support ELS, see "Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch" on page 226. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50.

EX Series switches use bridging and virtual LANs (VLANs) to connect network devices in a LAN—desktop computers or laptops, IP telephones, printers, file servers, wireless access points, and others—and to segment the LAN into smaller broadcast domains.

This example describes how to configure basic bridging and a VLAN on an EX Series switch:

Requirements

This example uses the following hardware and software components:

- One EX Series switch
- Junos OS Release 13.2X50-D10 or later for EX Series switches

Before you set up bridging and a VLAN, be sure you have:

- Installed your EX Series switch. See the installation instructions for your switch.
- Performed the initial switch configuration. See *Connecting and Configuring an EX Series Switch (CLI Procedure)*.

Overview and Topology

EX Series switches connect network devices in an office LAN or a data center LAN to provide sharing of common resources such as printers and file servers and to enable wireless devices to connect to the LAN through wireless access points. Without bridging and VLANs, all devices on the Ethernet LAN are in a single broadcast domain, and all the devices detect all the packets on the LAN. Bridging creates separate broadcast domains on the LAN, creating VLANs, which are independent logical networks that group together related devices into separate network segments. The grouping of devices on a VLAN is independent of where the devices are physically located in the LAN.

To use an EX Series switch to connect network devices on a LAN, you must, at a minimum, explicitly configure at least one VLAN, even if your network is simple and you want only one broadcast domain to exist, as is the case with this example. You must also assign all needed interfaces to the VLAN, after which the interfaces function in access mode. After the VLAN is configured, you can plug access devices—such as desktop or laptop computers, IP telephones, file servers, printers, and wireless access points—into the switch, and they are joined immediately into the VLAN, and the LAN is up and running.

The topology used in this example consists of one EX4300-24P switch, which has a total of 24 ports. All ports support Power over Ethernet (PoE), which means they provide both network connectivity and electric power for the device connecting to the port. To these ports, you can plug in devices requiring PoE, such as Avaya VoIP telephones, wireless access points, and some IP cameras. (Avaya phones have a built-in hub that allows you to connect a desktop PC to the phone, so the desktop and phone in a single office require only one port on the switch.) Table 54 on page 192 details the topology used in this configuration example.

Table 54: Components of the Basic Bridging Configuration Topology

Property	Settings
Switch hardware	EX4300-24P switch, with 24 Gigabit Ethernet ports: in this example, 8 ports are used as PoE ports (ge-0/0/0 through ge-0/0/7) and 16 ports used as non-PoE ports (ge-0/0/8 through ge-0/0/23)
VLAN name	employee-vlan
VLAN ID	10
Connection to wireless access point (requires PoE)	ge-0/0/0
Connections to Avaya IP telephone—with integrated hub, to connect phone and desktop PC to a single port (requires PoE)	ge-0/0/1 through ge-0/0/7
Direct connections to desktop PCs and laptops (no PoE required)	ge-0/0/8 through ge-0/0/12
Connections to file servers (no PoE required)	ge-0/0/17 and ge-0/0/18
Connections to integrated printer/fax/copier machines (no PoE required)	ge-0/0/19 through ge-0/0/20
Unused ports (for future expansion)	ge-0/0/13 through ge-0/0/16, and ge-0/0/21 through ge-0/0/23

Configuration

To set up basic bridging and a VLAN:

CLI Quick Configuration

To quickly configure a VLAN, copy the following commands and paste them into the switch terminal window:

```
[edit]
set vlans employee-vlan vlan-id 10
set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces ge-0/0/3 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces ge-0/0/4 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces ge-0/0/5 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces ge-0/0/6 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces ge-0/0/7 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces ge-0/0/8 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces ge-0/0/9 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces ge-0/0/10 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces ge-0/0/11 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces ge-0/0/12 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces ge-0/0/17 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces ge-0/0/18 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces ge-0/0/19 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces ge-0/0/20 unit 0 family ethernet-switching vlan members employee-vlan
```

You must then plug the wireless access point into PoE-enabled port **ge-0/0/0** and the Avaya IP phones into the PoE-enabled ports **ge-0/0/1** through **ge-0/0/7**. Also, plug the PCs, file servers, and printers into ports **ge-0/0/8** through **ge-0/0/12** and **ge-0/0/17** through **ge-0/0/20**.

Step-by-Step Procedure

To set up basic bridging and a VLAN:

1. Create a VLAN named employee-vlan and specify the VLAN ID of 10 for it:

   ```
   [edit vlans]
   user@switch# set employee-vlan vlan-id 10
   ```

2. Assign interfaces ge-0/0/0 through ge-0/0/12, and ge-0/0/17 through ge-0/0/20 to the employee-vlan VLAN:

   ```
   [edit interface]
   user@switch# set ge-0/0/0 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set ge-0/0/1 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set ge-0/0/2 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set ge-0/0/3 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set ge-0/0/4 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set ge-0/0/5 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set ge-0/0/6 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set ge-0/0/7 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set ge-0/0/8 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set ge-0/0/9 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set ge-0/0/10 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set ge-0/0/11 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set ge-0/0/12 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set ge-0/0/17 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set ge-0/0/18 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set ge-0/0/19 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set ge-0/0/20 unit 0 family ethernet-switching vlan members employee-vlan
   ```

3. Connect the wireless access point to switch port ge-0/0/0.

4. Connect the seven Avaya phones to switch ports ge-0/0/1 through ge-0/0/7.

5. Connect the five PCs to ports ge-0/0/8 through ge-0/0/12.

6. Connect the two file servers to ports ge-0/0/17 and ge-0/0/18.

7. Connect the two printers to ports ge-0/0/19 and ge-0/0/20.

Results

Check the results of the configuration:

```
user@switch> show configuration
```

```
    ge-0/0/0 {
        unit 0 {
            family ethernet-switching {
                vlan {
                    members employee-vlan;
                }
            }
        }
    }
    ge-0/0/1 {
        unit 0 {
            family ethernet-switching {
                vlan {
                    members employee-vlan;
                }
            }
        }
    }
    ge-0/0/2 {
        unit 0 {
            family ethernet-switching {
                vlan {
                    members employee-vlan;
                }
            }
        }
    }
    ge-0/0/3 {
        unit 0 {
            family ethernet-switching {
                vlan {
                    members employee-vlan;
                }
            }
        }
    }
    ge-0/0/4 {
        unit 0 {
            family ethernet-switching {
```

```
            vlan {
                members employee-vlan;
            }
        }
    }
}
ge-0/0/5 {
    unit 0 {
        family ethernet-switching {
            vlan {
                members employee-vlan;
            }
        }
    }
}
ge-0/0/6 {
    unit 0 {
        family ethernet-switching {

            vlan {
                members employee-vlan;
            }
        }
    }
}
ge-0/0/7 {
    unit 0 {
        family ethernet-switching {
             vlan {
                members employee-vlan;
            }
        }
    }
}
ge-0/0/8 {
    unit 0 {
        family ethernet-switching {
            vlan {
                members employee-vlan;
            }
        }
    }
}
ge-0/0/9 {
```

```
    unit 0 {
        family ethernet-switching {
            vlan {
                members employee-vlan;
            }
        }
    }
}
ge-0/0/10 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
ge-0/0/11 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
ge-0/0/12 {
    unit 0 {
        family ethernet-switching {
            vlan {
                members employee-vlan;
            }
        }
    }
}
ge-0/0/17 {
    unit 0 {
        family ethernet-switching {
            vlan {
                members employee-vlan;
            }
        }
    }
}
```

```
ge-0/0/18 {
    unit 0 {
        family ethernet-switching {
            vlan {
                members employee-vlan;
            }
        }
    }
}
ge-0/0/19 {
    unit 0 {
        family ethernet-switching {
            vlan {
                members employee-vlan;
            }
        }
    }
}
ge-0/0/20 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
```

Verification

IN THIS SECTION

To verify that switching is operational and that **employee-vlan** has been created, perform these tasks:

Verifying That the VLAN Has Been Created

Purpose

Verify that the VLAN named **employee-vlan** has been created on the switch.

Action

List all VLANs configured on the switch:

```
user@switch>  show vlans
```

```
Routing instance        VLAN name             Tag         Interfaces
default-switch          employee-vlan            10
                                                            ge-0/0/0.0
                                                            ge-0/0/1.0
                                                            ge-0/0/2.0
                                                            ge-0/0/3.0
                                                            ge-0/0/4.0
                                                            ge-0/0/5.0
                                                            ge-0/0/6.0
                                                            ge-0/0/7.0
                                                            ge-0/0/8.0
                                                            ge-0/0/9.0
                                                            ge-0/0/10.0
                                                            ge-0/0/11.0
                                                            ge-0/0/12.0
                                                            ge-0/0/17.0
                                                            ge-0/0/18.0
                                                            ge-0/0/19.0
                                                            ge-0/0/20.0
...
```

Meaning

The **show vlans** command lists the VLANs configured on the switch. This output shows that the VLAN **employee-vlan** has been created.

Verifying That Interfaces Are Associated with the Proper VLANs

Purpose

Verify that Ethernet switching is enabled on switch interfaces and that all interfaces are included in the VLAN.

Action

List all interfaces on which switching is enabled:

```
user@switch>  show ethernet-switching interfaces
```

```
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
ge-0/0/0.0                    65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
ge-0/0/1.0                    65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
ge-0/0/2.0                    65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
ge-0/0/3.0                    65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
ge-0/0/4.0                    65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
```

```
interface    members          limit    state        interface flags
ge-0/0/5.0                    65535                                  untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                          LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
ge-0/0/6.0                    65535                                  untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                          LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
ge-0/0/7.0                    65535                                  untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                          LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
ge-0/0/8.0                    65535                                  untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                          LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
ge-0/0/9.0                    65535                                  untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                          LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
ge-0/0/10.0                   65535                                  untagged
             employee-vlan 10
                              65535    Discarding
```

```
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
ge-0/0/11.0                   65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
ge-0/0/12.0                   65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
ge-0/0/17.0                   65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
ge-0/0/18.0                   65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
ge-0/0/19.0                   65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
```

```
interface    members          limit    state        interface flags
ge-0/0/20.0                   65535                                 untagged
             employee-vlan 10
                              65535    Discarding
...
```

Meaning

The **show ethernet-switching interfaces** command lists all interfaces on which switching is enabled (in the **Logical interface** column), along with the VLANs that are active on the interfaces (in the **VLAN members** column). The output in this example shows all the connected interfaces, ge-0/0/0 through ge-0/0/12 and ge-0/0/17 through ge-0/0/20 and that they are all part of VLAN **employee-vlan**. Notice that the interfaces listed are the logical interfaces, not the physical interfaces. For example, the output shows ge-0/0/0.0 instead of ge-0/0/0. This is because Junos OS creates VLANs on logical interfaces, not directly on physical interfaces.

SEE ALSO

Example: Connecting Access Switches with ELS Support to a Distribution Switch with ELS Support | 251

Example: Setting Up Basic Bridging and a VLAN on Switches

IN THIS SECTION

The QFX Series products use bridging and virtual LANs (VLANs) to connect network devices—storage devices, file servers, and other LAN components—in a LAN and to segment the LAN into smaller bridging domains.

To segment traffic on a LAN into separate broadcast domains, you create separate virtual LANs (VLANs) on a switch. Each VLAN is a collection of network nodes. When you use VLANs, frames whose origin and destination are in the same VLAN are forwarded only within the local VLAN, and only frames not destined

for the local VLAN are forwarded to other broadcast domains. VLANs thus limit the amount of traffic flowing across the entire LAN, reducing the possible number of collisions and packet retransmissions within the LAN.

> **NOTE:** You cannot configure more than one logical interface that belongs to the same physical interface in the same bridge domain.

This example describes how to configure basic bridging and VLANs for the QFX Series:

Requirements

This example uses the following software and hardware components:

- Junos OS Release 11.1 or later for the QFX Series
- A configured and provisioned QFX Series product

Overview and Topology

To use a switch to connect network devices on a LAN, you must at a minimum configure bridging and VLANs. By default, bridging is enabled on all switch interfaces, all interfaces are in access mode, and all interfaces belong to a VLAN called **employee-vlan**, which is automatically configured. When you plug in access devices—such as desktop computers, file servers, and printers—they are joined immediately into the **employee-vlan** VLAN, and the LAN is up and running.

The topology used in this example consists of a single QFX3500 switch, with a total of 48 10-Gbps Ethernet ports. (For the purposes of this example, the QSFP+ ports Q0-Q3, which are ports xe-0/1/0 through xe-0/1/15, are excluded.) You use the ports to connect devices that have their own power sources. Table 1 details the topology used in this configuration example.

Table 55: Components of the Basic Bridging Configuration Topology

Property	Settings
Switch hardware	QFX3500 switch, with 48 10-Gbps Ethernet ports
VLAN name	**employee-vlan**
VLAN ID	10
Connections to file servers	**xe-0/0/17** and **xe-0/0/18**
Direct connections to desktop PCs and laptops	**xe-0/0/0** through **xe-0/0/16**

Table 55: Components of the Basic Bridging Configuration Topology *(continued)*

Property	Settings
Connections to integrated printer/fax/copier machines	**xe-0/0/19** through **xe-0/0/40**
Unused ports	**xe-0/0/41** through **xe-0/0/47**

Configuration

CLI Quick Configuration

To quickly configure a VLAN, copy the following commands and paste them into the switch terminal window:

```
[edit]
```

set vlans employee-vlan vlan-id 10

set interfaces xe-0/0/0 unit 0 family ethernet-switching vlan members employee-vlan

set interfaces xe-0/0/1 unit 0 family ethernet-switching vlan members employee-vlan

set interfaces xe-0/0/2 unit 0 family ethernet-switching vlan members employee-vlan

set interfaces xe-0/0/3 unit 0 family ethernet-switching vlan members employee-vlan

set interfaces xe-0/0/4 unit 0 family ethernet-switching vlan members employee-vlan

set interfaces xe-0/0/5 unit 0 family ethernet-switching vlan members employee-vlan

set interfaces xe-0/0/6 unit 0 family ethernet-switching vlan members employee-vlan

set interfaces xe-0/0/7 unit 0 family ethernet-switching vlan members employee-vlan

set interfaces xe-0/0/8 unit 0 family ethernet-switching vlan members employee-vlan

set interfaces xe-0/0/9 unit 0 family ethernet-switching vlan members employee-vlan

set interfaces xe-0/0/10 unit 0 family ethernet-switching vlan members employee-vlan

set interfaces xe-0/0/11 unit 0 family ethernet-switching vlan members employee-vlan

set interfaces xe-0/0/12 unit 0 family ethernet-switching vlan members employee-vlan

set interfaces xe-0/0/13 unit 0 family ethernet-switching vlan members employee-vlan

set interfaces xe-0/0/14 unit 0 family ethernet-switching vlan members employee-vlan

```
set interfaces xe-0/0/15 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/16 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/17 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/18 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/19 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/20 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/21 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/22 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/23 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/24 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/25 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/26 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/27 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/28 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/29 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/30 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/31 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/32 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/33 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/34 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/35 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/36 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/37 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/38 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/39 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/40 unit 0 family ethernet-switching vlan members employee-vlan
```

Step-by-Step Procedure

To set up basic bridging and a VLAN:

1. Create a VLAN named employee-vlan and specify the VLAN ID of 10 for it:

```
[edit vlans]
user@switch# set employee-vlan vlan-id 10
```

2. Assign interfaces xe-0/0/0 through xe-0/0/40 to the employee-vlan VLAN:

```
[edit interface]
user@switch# set xe-0/0/0 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/1 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/2 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/3 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/4 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/5 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/6 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/7 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/8 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/9 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/10 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/11 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/12 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/13 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/14 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/15 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/16 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/17 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/18 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/19 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/20 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/21 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/22 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/23 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/24 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/25 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/26 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/27 unit 0 family ethernet-switching vlan members employee-vlan
```

```
user@switch# set xe-0/0/28 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/29 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/30 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/31 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/32 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/33 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/34 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/35 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/36 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/37 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/38 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/39 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/40 unit 0 family ethernet-switching vlan members employee-vlan
```

3. Connect the two file servers to ports xe-0/0/17 and xe-0/0/18.

4. Connect the desktop PCs and laptops to ports xe-0/0/0 through xe-0/0/16.

5. Connect the integrated printer/fax/copier machines to ports xe-0/0/19 through xe-0/0/40.

Results

Check the results of the configuration:

```
user@switch> show configuration
```

```
    xe-0/0/0 {
        unit 0 {
            family ethernet-switching {
                vlan {
                    members employee-vlan;
                }
            }
        }
    }
    xe-0/0/1 {
        unit 0 {
            family ethernet-switching {
                vlan {
                    members employee-vlan;
                }
```

```
        }
    }
}
xe-0/0/2 {
    unit 0 {
        family ethernet-switching {
            vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/3 {
    unit 0 {
        family ethernet-switching {
            vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/4 {
    unit 0 {
        family ethernet-switching {
            vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/5 {
    unit 0 {
        family ethernet-switching {
            vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/6 {
    unit 0 {
        family ethernet-switching {

            vlan {
```

```
                members employee-vlan;
            }
        }
    }
}
xe-0/0/7 {
    unit 0 {
        family ethernet-switching {
             vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/8 {
    unit 0 {
        family ethernet-switching {
            vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/9 {
    unit 0 {
        family ethernet-switching {
            vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/10 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/11 {
    unit 0 {
        family ethernet-switching {
```

```
                    vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/12 {
    unit 0 {
        family ethernet-switching {
            vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/13 {
    unit 0 {
        family ethernet-switching {
            vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/14 {
    unit 0 {
        family ethernet-switching {
            vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/15 {
    unit 0 {
        family ethernet-switching {
            vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/16 {
    unit 0 {
```

```
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/17 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/18 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/19 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/20 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/21 {
```

```
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/22 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/23 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/24 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/25 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
```

```
xe-0/0/26 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/27 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/28 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/29 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/30 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
```

```
}
xe-0/0/31 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/32 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/33 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/34 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/35 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
```

```
    }
}
xe-0/0/36 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/37 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/38 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/39 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/40 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
```

```
            }
        }
    }
```

Verification

IN THIS SECTION

To verify that switching is operational and that **employee-vlan** has been created, perform these tasks:

Verifying That the VLAN Has Been Created

Purpose

Verify that the VLAN named **employee-vlan** has been created on the switch.

Action

List all VLANs configured on the switch:

```
user@switch> show vlans
```

```
Routing instance        VLAN name             Tag         Interfaces
default-switch          employee-vlan            10
                                                            xe-0/0/0.0
                                                            xe-0/0/1.0
                                                            xe-0/0/2.0
                                                            xe-0/0/3.0
                                                            xe-0/0/4.0
                                                            xe-0/0/5.0
                                                            xe-0/0/6.0
                                                            xe-0/0/7.0
                                                            xe-0/0/8.0
                                                            xe-0/0/9.0
                                                            xe-0/0/10.0
                                                            xe-0/0/11.0
                                                            xe-0/0/12.0
```

```
                                                    xe-0/0/13.0
                                                    xe-0/0/14.0
                                                    xe-0/0/15.0
                                                    xe-0/0/16.0
                                                    xe-0/0/17.0
                                                    xe-0/0/18.0
                                                    xe-0/0/19.0
                                                    xe-0/0/20.0
                                                    xe-0/0/21.0
                                                    xe-0/0/22.0
                                                    xe-0/0/23.0
                                                    xe-0/0/24.0
                                                    xe-0/0/25.0
                                                    xe-0/0/26.0
                                                    xe-0/0/27.0
                                                    xe-0/0/28.0
                                                    xe-0/0/29.0
                                                    xe-0/0/30.0
                                                    xe-0/0/31.0
                                                    xe-0/0/32.0
                                                    xe-0/0/33.0
                                                    xe-0/0/34.0
                                                    xe-0/0/35.0
                                                    xe-0/0/36.0
                                                    xe-0/0/37.0
                                                    xe-0/0/38.0
                                                    xe-0/0/39.0
                                                    xe-0/0/40.0
...
```

Meaning

The **show vlans** command lists the VLANs configured on the switch. This output shows that the VLAN **employee-vlan** has been created.

Verifying That Interfaces Are Associated with the Proper VLANs

Purpose

Verify that Ethernet switching is enabled on switch interfaces and that all interfaces are included in the VLAN.

Action

List all interfaces on which switching is enabled:

```
user@switch>  show ethernet-switching interfaces
```

```
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/0.0                    65535                                untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/1.0                    65535                                untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/2.0                    65535                                untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/3.0                    65535                                untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/4.0                    65535                                untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
```

```
interface    members          limit    state        interface flags
xe-0/0/5.0                    65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                          LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/6.0                    65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                          LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/7.0                    65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                          LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/8.0                    65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                          LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/9.0                    65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                          LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/10.0                   65535                                 untagged
             employee-vlan 10
                              65535    Discarding
```

```
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/11.0                   65535                                untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/12.0                   65535                                untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/13.0                   65535                                untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/14.0                   65535                                untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/15.0                   65535                                untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
```

```
interface    members          limit    state        interface flags
xe-0/0/16.0                   65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                          LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/17.0                   65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                          LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/18.0                   65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                          LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/19.0                   65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                          LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/20.0                   65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                          LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/21.0                   65535                                 untagged
             employee-vlan 10
                              65535    Discarding
```

```
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/22.0                   65535                                untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/23.0                   65535                                untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/24.0                   65535                                untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/25.0                   65535                                untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/26.0                   65535                                untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
```

```
interface    members          limit    state        interface flags
xe-0/0/27.0                   65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                          LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/28.0                   65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                          LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/29.0                   65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                          LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/30.0                   65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                          LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/31.0                   65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                          LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/32.0                   65535                                 untagged
             employee-vlan 10
                              65535    Discarding
```

```
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/33.0                   65535                                untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/34.0                   65535                                untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/35.0                   65535                                untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/36.0                   65535                                untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/37.0                   65535                                untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
```

```
interface    members          limit    state        interface flags
xe-0/0/38.0                   65535                                untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                          LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/39.0                   65535                                untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                          LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/40.0                   65535                                untagged
             employee-vlan 10
                              65535    Discarding
...
```

Meaning

The **show ethernet-switching interfaces** command lists all interfaces on which switching is enabled (in the **Logical interface** column), along with the VLANs that are active on the interfaces (in the **VLAN members** column). The output in this example shows all the connected interfaces, xe-0/0/0 through xe-0/0/40, are all part of VLAN **employee-vlan**. Notice that the interfaces listed are the logical interfaces, not the physical interfaces. For example, the output shows xe-0/0/0.0 instead of xe-0/0/0. This is because Junos OS creates VLANs on logical interfaces, not directly on physical interfaces.

Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch

IN THIS SECTION

NOTE: This example uses Junos OS for EX Series switches that does not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see "Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch with ELS Support" on page 190. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50

EX Series switches use bridging and virtual LANs (VLANs) to connect network devices in a LAN—desktop computers, IP telephones, printers, file servers, wireless access points, and others—and to segment the LAN into smaller bridging domains. The switch's default configuration provides a quick setup of bridging and a single VLAN.

This example describes how to configure basic bridging and VLANs for an EX Series switch:

Requirements

This example uses the following software and hardware components:

- Junos OS Release 9.0 or later for EX Series switches
- One EX4200 Virtual Chassis switch

Before you set up bridging and a VLAN, be sure you have:

- Performed the initial switch configuration. See *Connecting and Configuring an EX Series Switch (J-Web Procedure)*.

Overview and Topology

EX Series switches connect network devices in an office LAN or a data center LAN to provide sharing of common resources such as printers and file servers and to enable wireless devices to connect to the LAN through wireless access points. Without bridging and VLANs, all devices on the Ethernet LAN are in a single broadcast domain, and all the devices detect all the packets on the LAN. Bridging creates separate broadcast domains on the LAN, creating VLANs, which are independent logical networks that group together related devices into separate network segments. The grouping of devices on a VLAN is independent of where the devices are physically located in the LAN.

To use an EX Series switch to connect network devices on a LAN, you must, at a minimum, configure bridging and VLANs. If you simply power on the switch and perform the initial switch configuration using the factory-default settings, bridging is enabled on all the switch's interfaces, all interfaces are in access mode, and all interfaces belong to a VLAN called **default**, which is automatically configured. When you plug access devices—such as desktop computers, Avaya IP telephones, file servers, printers, and wireless access points—into the switch, they are joined immediately into the **default** VLAN and the LAN is up and running.

The topology used in this example consists of one EX4200-24T switch, which has a total of 24 ports. Eight of the ports support Power over Ethernet (PoE), which means they provide both network connectivity and electric power for the device connecting to the port. To these ports, you can plug in devices requiring PoE, such as Avaya VoIP telephones, wireless access points, and some IP cameras. (Avaya phones have a built-in hub that allows you to connect a desktop PC to the phone, so the desktop and phone in a single office require only one port on the switch.) The remaining 16 ports provide only network connectivity. You use them to connect devices that have their own power sources, such as desktop and laptop computers, printers, and servers. Table 56 on page 228 details the topology used in this configuration example.

Table 56: Components of the Basic Bridging Configuration Topology

Property	Settings
Switch hardware	EX4200-24T switch, with 24 Gigabit Ethernet ports: 8 PoE ports (**ge-0/0/0** through **ge-0/0/7**) and 16 non-PoE ports (**ge-0/0/8** through **ge-0/0/23**)
VLAN name	**default**
Connection to wireless access point (requires PoE)	**ge-0/0/0**
Connections to Avaya IP telephone—with integrated hub, to connect phone and desktop PC to a single port (requires PoE)	**ge-0/0/1** through **ge-0/0/7**
Direct connections to desktop PCs (no PoE required)	**ge-0/0/8** through **ge-0/0/12**
Connections to file servers (no PoE required)	**ge-0/0/17** and **ge-0/0/18**
Connections to integrated printer/fax/copier machines (no PoE required)	**ge-0/0/19** through **ge-0/0/20**
Unused ports (for future expansion)	**ge-0/0/13** through **ge-0/0/16**, and **ge-0/0/21** through **ge-0/0/23**

Configuration

CLI Quick Configuration

By default, after you perform the initial configuration on the EX4200 switch, switching is enabled on all interfaces, a VLAN named **default** is created, and all interfaces are placed into this VLAN. You do not need to perform any other configuration on the switch to set up bridging and VLANs. To use the switch, simply plug the Avaya IP phones into the PoE-enabled ports **ge-0/0/1** through **ge-0/0/7**, and plug in the PCs, file servers, and printers to the non-PoE ports, **ge-0/0/8** through **ge-0/0/12** and **ge-0/0/17** through **ge-0/0/20**.

Step-by-Step Procedure

To configure bridging and VLANs:

1. Make sure the switch is powered on.

2. Connect the wireless access point to switch port **ge-0/0/0**.

3. Connect the seven Avaya phones to switch ports **ge-0/0/1** through **ge-0/0/7**.

4. Connect the five PCs to ports **ge-0/0/8** through **ge-0/0/12**.

5. Connect the two file servers to ports **ge-0/0/17** and **ge-0/0/18**.

6. Connect the two printers to ports **ge-0/0/19** and **ge-0/0/20**.

Results

Check the results of the configuration:

```
user@switch> show configuration
## Last commit: 2008-03-06 00:11:22 UTC by triumph
version 9.0;
system {
    root-authentication {
        encrypted-password "$1$urmA7AFM$x5SaGEUOdSI3u1K/iITGh1"; ## SECRET-DATA
    }
    syslog {
        user * {
            any emergency;
        }
        file messages {
            any notice;
            authorization info;
        }
        file interactive-commands {
            interactive-commands any;
        }
    }
    commit {
        factory-settings {
            reset-chassis-lcd-menu;
            reset-virtual-chassis-configuration;
        }
    }
```

```
}
interfaces {
    ge-0/0/0 {
        unit 0 {
            family ethernet-switching;
        }
    }
    ge-0/0/1 {
        unit 0 {
            family ethernet-switching;
        }
    }
    ge-0/0/2 {
        unit 0 {
            family ethernet-switching;
        }
    }
    ge-0/0/3 {
        unit 0 {
            family ethernet-switching;
        }
    }
    ge-0/0/4 {
        unit 0 {
            family ethernet-switching;
        }
    }
    ge-0/0/5 {
        unit 0 {
            family ethernet-switching;
        }
    }
    ge-0/0/6 {
        unit 0 {
            family ethernet-switching;
        }
    }
    ge-0/0/7 {
        unit 0 {
            family ethernet-switching;
        }
    }
    ge-0/0/8 {
        unit 0 {
```

```
        family ethernet-switching;
    }
}
ge-0/0/9 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/10 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/11 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/12 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/13 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/14 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/15 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/16 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/17 {
```

```
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/18 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/19 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/20 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/21 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/22 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/23 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/1/0 {
    unit 0 {
        family ethernet-switching;
    }
}
xe-0/1/0 {
    unit 0 {
        family ethernet-switching;
    }
}
```

```
    ge-0/1/1 {
        unit 0 {
            family ethernet-switching;
        }
    }
    xe-0/1/1 {
        unit 0 {
            family ethernet-switching;
        }
    }
    ge-0/1/2 {
        unit 0 {
            family ethernet-switching;
        }
    }
    ge-0/1/3 {
        unit 0 {
            family ethernet-switching;
        }
    }
}
protocols {
    lldp {
        interface all;
    }
    rstp;
}
    poe {
        interface all;
    }
```

Verification

IN THIS SECTION

To verify that switching is operational and that a VLAN has been created, perform these tasks:

Verifying That the VLAN Has Been Created

Purpose

Verify that the VLAN named **default** has been created on the switch.

Action

List all VLANs configured on the switch:

```
user@switch> show vlans
```

```
Name            Tag     Interfaces
default
                        ge-0/0/0.0*, ge-0/0/1.0, ge-0/0/2.0, ge-0/0/3.0,
                        ge-0/0/4.0,  ge-0/0/5.0, ge-0/0/6.0, ge-0/0/7.0,
                        ge-0/0/8.0*, ge-0/0/9.0, ge-0/0/10.0, ge-0/0/11.0*,
                        ge-0/0/12.0, ge-0/0/13.0, ge-0/0/14.0, ge-0/0/15.0,
                        ge-0/0/16.0, ge-0/0/17.0, ge-0/0/18.0, ge-0/0/19.0*,
                        ge-0/0/20.0, ge-0/0/21.0, ge-0/0/22.0, ge-0/0/23.0,
                        ge-0/1/0.0*, ge-0/1/1.0*, ge-0/1/2.0*, ge-0/1/3.0*
mgmt
                        me0.0*
```

Meaning

The **show vlans** command lists the VLANs configured on the switch. This output shows that the VLAN **default** has been created.

Verifying That Interfaces Are Associated with the Proper VLANs

Purpose

Verify that Ethernet switching is enabled on switch interfaces and that all interfaces are included in the VLAN.

Action

List all interfaces on which switching is enabled:

```
user@switch> show ethernet-switching interfaces
```

```
Interface   State    VLAN members           Blocking
ge-0/0/0.0  up       default                unblocked
ge-0/0/1.0  down     default                blocked - blocked by STP/RTG
ge-0/0/2.0  down     default                blocked - blocked by STP/RTG
```

```
ge-0/0/3.0  down     default              blocked - blocked by STP/RTG
ge-0/0/4.0  down     default              blocked - blocked by STP/RTG
ge-0/0/5.0  down     default              blocked - blocked by STP/RTG
ge-0/0/6.0  down     default              blocked - blocked by STP/RTG
ge-0/0/7.0  down     default              blocked - blocked by STP/RTG
ge-0/0/8.0  up       default              unblocked
ge-0/0/9.0  down     default              blocked - blocked by STP/RTG
ge-0/0/10.0 down     default              blocked - blocked by STP/RTG
ge-0/0/11.0 up       default              unblocked
ge-0/0/12.0 down     default              blocked - blocked by STP/RTG
ge-0/0/13.0 down     default              blocked - blocked by STP/RTG
ge-0/0/14.0 down     default              blocked - blocked by STP/RTG
ge-0/0/15.0 down     default              blocked - blocked by STP/RTG
ge-0/0/16.0 down     default              blocked - blocked by STP/RTG
ge-0/0/17.0 down     default              blocked - blocked by STP/RTG
ge-0/0/18.0 down     default              blocked - blocked by STP/RTG
ge-0/0/19.0 up       default              unblocked
ge-0/0/20.0 down     default              blocked - blocked by STP/RTG
ge-0/0/21.0 down     default              blocked - blocked by STP/RTG
ge-0/0/22.0 down     default              blocked - blocked by STP/RTG
ge-0/0/23.0 down     default              blocked - blocked by STP/RTG
ge-0/1/0.0  up       default              unblocked
ge-0/1/1.0  up       default              unblocked
ge-0/1/2.0  up       default              unblocked
ge-0/1/3.0  up       default              unblocked
me0.0       up       mgmt                 unblocked
```

Meaning

The **show ethernet-switching interfaces** command lists all interfaces on which switching is enabled (in the **Interfaces** column), along with the VLANs that are active on the interfaces (in the **VLAN members** column). The output in this example shows all the connected interfaces, **ge-0/0/0** through **ge-0/0/12** and **ge-0/0/17** through **ge-0/0/20** and that they are all part of VLAN **default**. Notice that the interfaces listed are the logical interfaces, not the physical interfaces. For example, the output shows **ge-0/0/0.0** instead of **ge-0/0/0**. This is because Junos OS creates VLANs on logical interfaces, not directly on physical interfaces.

Example: Setting Up Bridging with Multiple VLANs

IN THIS SECTION

The QFX Series products use bridging and virtual LANs (VLANs) to connect network devices in a LAN—storage devices, file servers, and other network components—and to segment the LAN into smaller bridging domains.

To segment traffic on a LAN into separate broadcast domains, you create separate virtual LANs (VLANs) on a switch. Each VLAN is a collection of network nodes. When you use VLANs, frames whose origin and destination are in the same VLAN are forwarded only within the local VLAN, and only frames not destined for the local VLAN are forwarded to other broadcast domains. VLANs thus limit the amount of traffic flowing across the entire LAN, reducing the possible number of collisions and packet retransmissions within the LAN.

> **NOTE:** This task uses Junos OS for QFX3500 and QFX3600 switches does not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see "Example: Setting Up Bridging with Multiple VLANs on Switches" on page 243.

This example describes how to configure bridging for the QFX Series and how to create two VLANs to segment the LAN:

Requirements

This example uses the following hardware and software components:

- A configured and provisioned QFX3500 switch
- Junos OS Release 11.1 or later for the QFX Series

Overview and Topology

Switches connect all devices in an office or data center into a single LAN to provide sharing of common resources such as file servers. The default configuration creates a single VLAN, and all traffic on the switch is part of that broadcast domain. Creating separate network segments reduces the span of the broadcast domain and enables you to group related users and network resources without being limited by physical cabling or by the location of a network device in the building or on the LAN.

This example shows a simple configuration to illustrate the basic steps for creating two VLANs on a single switch. One VLAN, called **sales**, is for the sales and marketing group, and a second, called **support**, is for the customer support team. The sales and support groups each have their own dedicated file servers and other resources. For the switch ports to be segmented across the two VLANs, each VLAN must have its own broadcast domain, identified by a unique name and tag (VLAN ID). In addition, each VLAN must be on its own distinct IP subnet.

The topology used in this example consists of a single QFX3500 switch, with a total of 48 10-Gbps Ethernet ports. (For the purposes of this example, the QSFP+ ports Q0-Q3, which are ports xe-0/1/0 through xe-0/1/15, are excluded.)

Table 57: Components of the Multiple VLAN Topology

Property	Settings
Switch hardware	QFX3500 switch configured with 48 10-Gbps Ethernet ports (**xe-0/0/0** through **xe-0/0/47**)
VLAN names and tag IDs	**sales**, tag **100** **support**, tag **200**
VLAN subnets	**sales**: **192.0.2.0/25** (addresses **192.0.2.1** through **192.0.2.126**) **support**: **192.0.2.128/25** (addresses **192.0.2.129** through **192.0.2.254**)
Interfaces in VLAN **sales**	File servers: **xe-0/0/20** and **xe-0/0/21**
Interfaces in VLAN **support**	File servers: **xe-0/0/46** and **xe-0/0/47**
Unused interfaces	**xe-0/0/2** and **xe-0/0/25**

This configuration example creates two IP subnets, one for the sales VLAN and the second for the support VLAN. The switch bridges traffic within a VLAN. For traffic passing between two VLANs, the switch routes the traffic using a Layer 3 routing interface on which you have configured the address of the IP subnet.

To keep the example simple, the configuration steps show only a few devices in each of the VLANs. Use the same configuration procedure to add more LAN devices.

Configuration

CLI Quick Configuration

To quickly configure Layer 2 switching for the two VLANs (**sales** and **support**) and to quickly configure Layer 3 routing of traffic between the two VLANs, copy the following commands and paste them into the switch terminal window:

```
[edit]
set interfaces xe-0/0/0 unit 0 family ethernet-switching vlan members sales
set interfaces xe-0/0/3 unit 0 family ethernet-switching vlan members sales
set interfaces xe-0/0/22 unit 0 family ethernet-switching vlan members sales
set interfaces xe-0/0/20 unit 0 description “Sales file server port”
set interfaces xe-0/0/20 unit 0 family ethernet-switching vlan members sales
set interfaces xe-0/0/24 unit 0 family ethernet-switching vlan members support
set interfaces xe-0/0/26 unit 0 family ethernet-switching vlan members support
set interfaces xe-0/0/44 unit 0 family ethernet-switching vlan members support
set interfaces xe-0/0/46 unit 0 description “Support file server port”
set interfaces xe-0/0/46 unit 0 family ethernet-switching vlan members support
set interfaces vlan unit 0 family inet address 192.0.2.0/25
set interfaces vlan unit 1 family inet address 192.0.2.128/25
set vlans sales l3–interface vlan.0
set vlans sales vlan-id 100
set vlans support vlan-id 200
set vlans support l3-interface vlan.1
```

Step-by-Step Procedure

Configure the switch interfaces and the VLANs to which they belong. By default, all interfaces are in access mode, so you do not have to configure the port mode.

1. Configure the interface for the file server in the **sales** VLAN:

```
[edit interfaces xe-0/0/20 unit 0]
user@switch# set description “Sales file server port”
user@switch# set family ethernet-switching vlan members sales
```

2. Configure the interface for the file server in the **support** VLAN:

```
[edit interfaces xe-0/0/46 unit 0]
user@switch# set description “Support file server port”
user@switch# set family ethernet-switching vlan members support
```

3. Create the subnet for the **sales** broadcast domain:

```
[edit interfaces]
user@switch# set vlan unit 0 family inet address 192.0.2.1/25
```

4. Create the subnet for the **support** broadcast domain:

```
[edit interfaces]
user@switch# set vlan unit 1 family inet address 192.0.2.129/25
```

5. Configure the VLAN tag IDs for the **sales** and **support** VLANs:

```
[edit vlans]
user@switch# set sales vlan-id 100
user@switch# set support vlan-id 200
```

6. To route traffic between the **sales** and **support** VLANs, define the interfaces that are members of each VLAN and associate a Layer 3 interface:

```
[edit vlans]
user@switch# set sales l3-interface vlan.0
user@switch# set support l3-interface vlan.1
```

Results

Display the results of the configuration:

```
user@switch> show configuration
interfaces {
```

```
    xe-0/0/20 {
        unit 0 {
            description “Sales file server port”;
            family ethernet-switching {
                vlan members sales;
            }
        }
    }
    xe-0/0/46 {
        unit 0 {
            description “Support file server port”;
            family ethernet-switching {
                vlan members support;
            }
        }
        vlans {
            unit 0 {
                family inet address 192.0.2.1/25;
            }
            unit 1 {
                family inet address 192.0.2.129/25;
            }
        }
    }
}
vlans {
    sales {
        vlan-id 100;
        interface xe-0/0/0.0:
        interface xe-0/0/3/0;
        interface xe-0/0/20.0;
        interface xe-0/0/22.0;
        l3-interface vlan 0;
    }
    support {
        vlan-id 200;
        interface xe-0/0/24.0:
        interface xe-0/0/26.0;
        interface xe-0/0/44.0;
        interface xe-0/0/46.0;
        l3-interface vlan 1;
    }
}
```

> **TIP:** To quickly configure the sales and support VLAN interfaces, issue the **load merge terminal** command. Then copy the hierarchy and paste it into the switch terminal window.

Verification

IN THIS SECTION

Verify that the **sales** and **support** VLANs have been created and are operating properly, perform these tasks:

Verifying That the VLANs Have Been Created and Associated with the Correct Interfaces

Purpose

Verify that the **sales** and **support** VLANs have been created on the switch and that all connected interfaces on the switch are members of the correct VLAN.

Action

To list all VLANs configured on the switch, use the **show vlans** command:

```
user@switch> show vlans
```

```
Name            Tag     Interfaces
default
                        xe-0/0/1.0, xe-0/0/2.0, xe-0/0/4.0, xe-0/0/5.0,
                        xe-0/0/6.0,  xe-0/0/7.0, xe-0/0/8.0, xe-0/0/9.0,
                        xe-0/0/10.0*, xe-0/0/11.0, xe-0/0/12.0, xe-0/0/13.0*,
                        xe-0/0/14.0, xe-0/0/15.0, xe-0/0/16.0, xe-0/0/17.0,
                        xe-0/0/18.0, xe-0/0/19.0, xe-0/0/21.0, xe-0/0/23.0*,
                        xe-0/0/25.0, xe-0/0/27.0, xe-0/0/28.0, xe-0/0/29.0,
                        xe-0/0/30.0, xe-0/0/31.0, xe-0/0/32.0, xe-0/0/33.0,
                        xe-0/0/34.0, xe-0/0/35.0, xe-0/0/36.0, xe-0/0/37.0,
                        xe-0/0/38.0, xe-0/0/39.0, xe-0/0/40.0, xe-0/0/41.0,
                        xe-0/0/42.0, xe-0/0/43.0, xe-0/0/45.0, xe-0/0/47.0,
```

```
                        xe-0/1/0.0*, xe-0/1/1.0*, xe-0/1/2.0*, xe-0/1/3.0*

sales       100
                        xe-0/0/0.0*, xe-0/0/3.0, xe-0/0/20.0, xe-0/0/22.0

support     200
                        xe-0/0/0.24, xe-0/0/26.0, xe-0/0/44.0, xe-0/0/46.0*
mgmt
                        me0.0*
```

Meaning

The **show vlans** command lists all VLANs configured on the switch and which interfaces are members of each VLAN. This command output shows that the **sales** and **support** VLANs have been created. The **sales** VLAN has a tag ID of 100 and is associated with interfaces **xe-0/0/0.0**, **xe-0/0/3.0**, **xe-0/0/20.0**, and **xe-0/0/22.0**. VLAN **support** has a tag ID of 200 and is associated with interfaces **xe-0/0/24.0**, **xe-0/0/26.0**, **xe-0/0/44.0**, and **xe-0/0/46.0**.

Verifying That Traffic Is Being Routed Between the Two VLANs

Purpose

Verify routing between the two VLANs.

Action

List the Layer 3 routes in the switch Address Resolution Protocol (ARP) table:

```
user@switch> show arp
```

```
MAC Address       Address         Name                    Flags

00:00:0c:06:2c:0d        192.0.2.3       vlan.0                  None
00:13:e2:50:62:e0        192.0.2.11      vlan.1                  None
```

Meaning

Sending IP packets on a multiaccess network requires mapping from an IP address to a MAC address (the physical or hardware address). The ARP table displays the mapping between the IP address and MAC address for both **vlan.0** (associated with **sales**) and **vlan.1** (associated with **support**). These VLANs can route traffic to each other.

Verifying That Traffic Is Being Switched Between the Two VLANs

Purpose

Verify that learned entries are being added to the Ethernet switching table.

Action

List the contents of the Ethernet switching table:

```
user@switch> show ethernet-switching table
```

```
Ethernet-switching table: 8 entries, 5 learned
  VLAN              MAC address       Type         Age Interfaces
  default           *                 Flood          - All-members
  default           00:00:05:00:00:01 Learn          - xe-0/0/10.0
  default           00:00:5e:00:01:09 Learn          - xe-0/0/13.0
  default           00:19:e2:50:63:e0 Learn          - xe-0/0/23.0
  sales             *                 Flood          - All-members
  sales             00:00:5e:00:07:09 Learn          - xe-0/0/0.0
  support           *                 Flood          – All–members
  support           00:00:5e:00:01:01 Learn          – xe-0/0/46.0
```

Meaning

The output shows that learned entries for the **sales** and **support** VLANs have been added to the Ethernet switching table, and are associated with interfaces **xe-0/0/0.0** and **xe-0/0/46.0**. Even though the VLANs were associated with more than one interface in the configuration, these interfaces are the only ones that are currently operating.

Example: Setting Up Bridging with Multiple VLANs on Switches

IN THIS SECTION

The QFX Series products use bridging and virtual LANs (VLANs) to connect network devices in a LAN—storage devices, file servers, and other network components—and to segment the LAN into smaller bridging domains.

To segment traffic on a LAN into separate broadcast domains, you create separate virtual LANs (VLANs) on a switch. Each VLAN is a collection of network nodes. When you use VLANs, frames whose origin and destination are in the same VLAN are forwarded only within the local VLAN, and only frames not destined for the local VLAN are forwarded to other broadcast domains. VLANs thus limit the amount of traffic flowing across the entire LAN, reducing the possible number of collisions and packet retransmissions within the LAN.

This example describes how to configure bridging for the QFX Series and how to create two VLANs to segment the LAN:

> **NOTE:** This task supports the Enhanced Layer 2 Software (ELS) configuration style. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50. If your switch runs software that does not supports ELS, see "Example: Setting Up Bridging with Multiple VLANs" on page 236.

Requirements

This example uses the following hardware and software components:

- A configured and provisioned QFX3500 switch
- Junos OS Release 13.2X50-D15 or later for the QFX Series

Overview and Topology

Switches connect all devices in an office or data center into a single LAN to provide sharing of common resources such as file servers. The default configuration creates a single VLAN, and all traffic on the switch is part of that broadcast domain. Creating separate network segments reduces the span of the broadcast domain and enables you to group related users and network resources without being limited by physical cabling or by the location of a network device in the building or on the LAN.

This example shows a simple configuration to illustrate the basic steps for creating two VLANs on a single switch. One VLAN, called **sales**, is for the sales and marketing group, and a second, called **support**, is for the customer support team. The sales and support groups each have their own dedicated file servers and other resources. For the switch ports to be segmented across the two VLANs, each VLAN must have its own broadcast domain, identified by a unique name and tag (VLAN ID). In addition, each VLAN must be on its own distinct IP subnet.

The topology used in this example consists of a single QFX3500 switch, with a total of 48 10-Gbps Ethernet ports. (For the purposes of this example, the QSFP+ ports Q0-Q3, which are ports xe-0/1/0 through xe-0/1/15, are excluded.)

Table 58: Components of the Multiple VLAN Topology

Property	Settings
Switch hardware	QFX3500 switch configured with 48 10-Gbps Ethernet ports (**xe-0/0/0** through **xe-0/0/47**)
VLAN names and tag IDs	**sales**, tag **100** **support**, tag **200**
VLAN subnets	**sales**: **192.0.2.0/25** (addresses **192.0.2.1** through **192.0.2.126**) **support**: **192.0.2.128/25** (addresses **192.0.2.129** through **192.0.2.254**)
Interfaces in VLAN **sales**	File servers: **xe-0/0/20** and **xe-0/0/21**
Interfaces in VLAN **support**	File servers: **xe-0/0/46** and **xe-0/0/47**
Unused interfaces	**xe-0/0/2** and **xe-0/0/25**

This configuration example creates two IP subnets, one for the sales VLAN and the second for the support VLAN. The switch bridges traffic within a VLAN. For traffic passing between two VLANs, the switch routes the traffic using a Layer 3 routing interface on which you have configured the address of the IP subnet.

To keep the example simple, the configuration steps show only a few devices in each of the VLANs. Use the same configuration procedure to add more LAN devices.

Configuration

CLI Quick Configuration

To quickly configure Layer 2 switching for the two VLANs (**sales** and **support**) and to quickly configure Layer 3 routing of traffic between the two VLANs, copy the following commands and paste them into the switch terminal window:

```
[edit]
```

set interfaces xe-0/0/0 unit 0 family ethernet-switching vlan members sales

set interfaces xe-0/0/3 unit 0 family ethernet-switching vlan members sales

set interfaces xe-0/0/22 unit 0 family ethernet-switching vlan members sales

set interfaces xe-0/0/20 unit 0 description “Sales file server port”

set interfaces xe-0/0/20 unit 0 family ethernet-switching vlan members sales

set interfaces xe-0/0/24 unit 0 family ethernet-switching vlan members support

set interfaces xe-0/0/26 unit 0 family ethernet-switching vlan members support

set interfaces xe-0/0/44 unit 0 family ethernet-switching vlan members support

set interfaces xe-0/0/46 unit 0 description “Support file server port”

set interfaces xe-0/0/46 unit 0 family ethernet-switching vlan members support

set interfaces vlan unit 0 family inet address 192.0.2.0/25

set interfaces vlan unit 1 family inet address 192.0.2.128/25

set vlans sales l3–interface irb.0

set vlans sales vlan-id 100

set vlans support vlan-id 200

set vlans support l3-interface irb.1

Step-by-Step Procedure

Configure the switch interfaces and the VLANs to which they belong. By default, all interfaces are in access mode, so you do not have to configure the port mode.

1. Configure the interface for the file server in the **sales** VLAN:

```
[edit interfaces xe-0/0/20 unit 0]
user@switch# set description “Sales file server port”
user@switch# set family ethernet-switching vlan members sales
```

2. Configure the interface for the file server in the **support** VLAN:

```
[edit interfaces xe-0/0/46 unit 0]
user@switch# set description “Support file server port”
user@switch# set family ethernet-switching vlan members support
```

3. Create the subnet for the **sales** broadcast domain:

```
[edit interfaces]
user@switch# set vlan unit 0 family inet address 192.0.2.1/25
```

4. Create the subnet for the **support** broadcast domain:

```
[edit interfaces]
user@switch# set vlan unit 1 family inet address 192.0.2.129/25
```

5. Configure the VLAN tag IDs for the **sales** and **support** VLANs:

```
[edit vlans]
user@switch# set sales vlan-id 100
user@switch# set support vlan-id 200
```

6. To route traffic between the **sales** and **support** VLANs, define the interfaces that are members of each VLAN and associate a Layer 3 interface:

```
[edit vlans]
user@switch# set sales l3-interface irb.0
user@switch# set support l3-interface irb.1
```

Configuration Results

Display the results of the configuration:

```
user@switch> show configuration
interfaces {
    xe-0/0/20 {
        unit 0 {
            description “Sales file server port”;
            family ethernet-switching {
                vlan members sales;
            }
        }
    }
    xe-0/0/46 {
        unit 0 {
            description “Support file server port”;
            family ethernet-switching {
                vlan members support;
            }
        }
        vlans {
            unit 0 {
                family inet address 192.0.2.1/25;
            }
            unit 1 {
```

```
            family inet address 192.0.2.129/25;
          }
        }
      }
    }
    vlans {
      sales {
        vlan-id 100;
        interface xe-0/0/0.0:
        interface xe-0/0/3/0;
        interface xe-0/0/20.0;
        interface xe-0/0/22.0;
        l3-interface irb0;
      }
      support {
        vlan-id 200;
        interface xe-0/0/24.0:
        interface xe-0/0/26.0;
        interface xe-0/0/44.0;
        interface xe-0/0/46.0;
        l3-interface irb1;
      }
    }
```

TIP: To quickly configure the sales and support VLAN interfaces, issue the **load merge terminal** command. Then copy the hierarchy and paste it into the switch terminal window.

Verification

IN THIS SECTION

Verify that the **sales** and **support** VLANs have been created and are operating properly, perform these tasks:

Verifying That the VLANs Have Been Created and Associated with the Correct Interfaces

Purpose

Verify that the **sales** and **support** VLANs have been created on the switch and that all connected interfaces on the switch are members of the correct VLAN.

Action

To list all VLANs configured on the switch, use the **show vlans** command:

```
user@switch> show vlans
```

```
Name            Tag     Interfaces
default
                        xe-0/0/1.0, xe-0/0/2.0, xe-0/0/4.0, xe-0/0/5.0,
                        xe-0/0/6.0,  xe-0/0/7.0, xe-0/0/8.0, xe-0/0/9.0,
                        xe-0/0/10.0*, xe-0/0/11.0, xe-0/0/12.0, xe-0/0/13.0*,
                        xe-0/0/14.0, xe-0/0/15.0, xe-0/0/16.0, xe-0/0/17.0,
                        xe-0/0/18.0, xe-0/0/19.0, xe-0/0/21.0, xe-0/0/23.0*,
                        xe-0/0/25.0, xe-0/0/27.0, xe-0/0/28.0, xe-0/0/29.0,
                        xe-0/0/30.0, xe-0/0/31.0, xe-0/0/32.0, xe-0/0/33.0,
                        xe-0/0/34.0, xe-0/0/35.0, xe-0/0/36.0, xe-0/0/37.0,
                        xe-0/0/38.0, xe-0/0/39.0, xe-0/0/40.0, xe-0/0/41.0,
                        xe-0/0/42.0, xe-0/0/43.0, xe-0/0/45.0, xe-0/0/47.0,
                        xe-0/1/0.0*, xe-0/1/1.0*, xe-0/1/2.0*, xe-0/1/3.0*

sales       100
                        xe-0/0/0.0*, xe-0/0/3.0, xe-0/0/20.0, xe-0/0/22.0

support     200
                        xe-0/0/0.24, xe-0/0/26.0, xe-0/0/44.0, xe-0/0/46.0*
mgmt
                        me0.0*
```

Meaning

The **show vlans** command lists all VLANs configured on the switch and which interfaces are members of each VLAN. This command output shows that the **sales** and **support** VLANs have been created. The **sales** VLAN has a tag ID of 100 and is associated with interfaces **xe-0/0/0.0**, **xe-0/0/3.0**, **xe-0/0/20.0**, and **xe-0/0/22.0**. VLAN **support** has a tag ID of 200 and is associated with interfaces **xe-0/0/24.0**, **xe-0/0/26.0**, **xe-0/0/44.0**, and **xe-0/0/46.0**.

Verifying That Traffic Is Being Routed Between the Two VLANs

Purpose

Verify routing between the two VLANs.

Action

List the Layer 3 routes in the switch Address Resolution Protocol (ARP) table:

```
user@switch> show arp
```

```
MAC Address       Address         Name                    Flags

00:00:0c:06:2c:0d       192.0.2.3       vlan.0                  None
00:13:e2:50:62:e0       192.0.2.11      vlan.1                  None
```

Meaning

Sending IP packets on a multiaccess network requires mapping from an IP address to a MAC address (the physical or hardware address). The ARP table displays the mapping between the IP address and MAC address for both **vlan.0** (associated with **sales**) and **vlan.1** (associated with **support**). These VLANs can route traffic to each other.

Verifying That Traffic Is Being Switched Between the Two VLANs

Purpose

Verify that learned entries are being added to the Ethernet switching table.

Action

List the contents of the Ethernet switching table:

```
user@switch> show ethernet-switching table
```

```
Ethernet-switching table: 8 entries, 5 learned
  VLAN              MAC address       Type         Age Interfaces
  default           *                 Flood          - All-members
  default           00:00:05:00:00:01 Learn          - xe-0/0/10.0
  default           00:00:5e:00:01:09 Learn          - xe-0/0/13.0
  default           00:19:e2:50:63:e0 Learn          - xe-0/0/23.0
  sales             *                 Flood          - All-members
  sales             00:00:5e:00:07:09 Learn          - xe-0/0/0.0
  support           *                 Flood          – All–members
  support           00:00:5e:00:01:01 Learn          – xe-0/0/46.0
```

Meaning

The output shows that learned entries for the **sales** and **support** VLANs have been added to the Ethernet switching table, and are associated with interfaces **xe-0/0/0.0** and **xe-0/0/46.0**. Even though the VLANs were associated with more than one interface in the configuration, these interfaces are the only ones that are currently operating.

Example: Connecting Access Switches with ELS Support to a Distribution Switch with ELS Support

IN THIS SECTION

> **NOTE:** This example uses Junos OS for EX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50.

In large local area networks (LANs), you commonly need to aggregate traffic from a number of access switches into a distribution switch.

This example describes how to connect access switches to a distribution switch:

Requirements

This example uses the following hardware and software components:

- Three EX Series access switches.
- One EX Series distribution switch.

> **NOTE:** In an access switch-distribution switch topology, you can connect EX Series switches that run a version of Junos OS that supports ELS with EX Series switches that do not run a version of Junos OS that supports ELS. However, this example uses switches running ELS only to show how to configure this topology using the ELS CLI.

- Junos OS Release 12.3R2 or later that supports ELS for EX Series switches.

Before you connect an access switch to a distribution switch, be sure you have:

- Installed the switches. See the installation instructions for your switch.
- Performed the initial software configuration on both switches. For information about the initial software configuration for all EX Series switches except the EX9200 Series switches, see *Connecting and Configuring an EX Series Switch (CLI Procedure)*. For information about the initial software configuration for the EX9200 Series switches, see *Connecting and Configuring an EX9200 Switch (CLI Procedure)*.

Overview and Topology

In a large office that is spread across several floors or buildings, or in a data center, you commonly aggregate traffic from a number of access switches into a distribution switch. This configuration example shows a simple topology to illustrate how to connect three access switches to a distribution switch.

In the topology, the LAN is segmented into two VLANs, one for the sales department and the second for the support team. One 1-Gigabit Ethernet port on one of the access switch's uplink modules connects to the distribution switch, to one 1-Gigabit Ethernet port on the distribution switch.

Figure 1 on page 253 shows an EX9200 distribution switch that is connected to three EX4300 access switches.

Figure 1: Sample Access Switch-Distribution Switch Topology

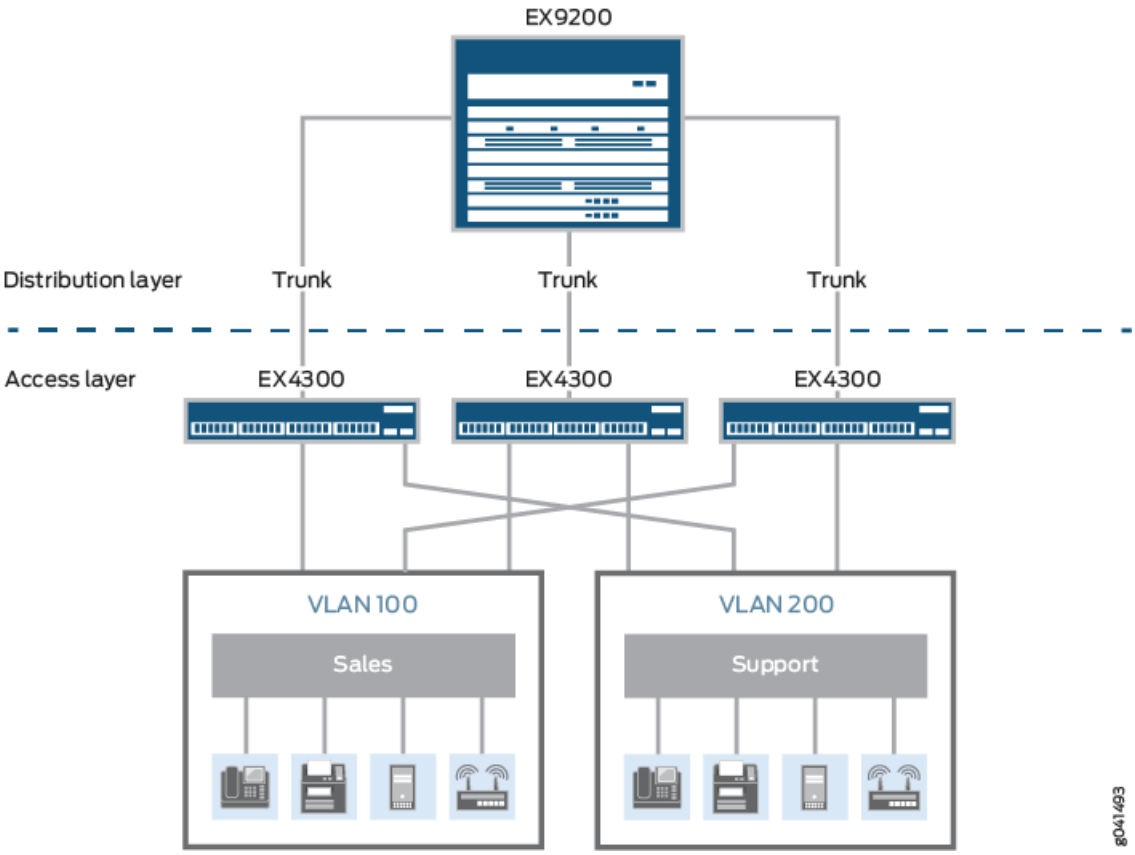


Table 59 on page 253 describes the components of the example topology. The example shows how to configure one of the three access switches. The other access switches could be configured in the same manner.

Table 59: Components of the Topology for Connecting an Access Switch to a Distribution Switch

Property	Settings
Access switch hardware	Three EX4300 switches, each with an uplink module with 1-Gigabit Ethernet ports..
Distribution switch hardware	One EX9208 with up to three EX9200-40T line cards installed, which at full duplex, can provide up to 240 1-Gigabit ports.
VLAN names and tag IDs	**sales**, tag **100** **support**, tag **200**
VLAN subnets	**sales**: 192.0.2.0/25 (addresses 192.0.2.1 through 192.0.2.126) **support**: 192.0.2.128/25 (addresses 192.0.2.129 through 192.0.2.254)

Table 59: Components of the Topology for Connecting an Access Switch to a Distribution Switch *(continued)*

Property	Settings
Trunk port interfaces	On the access switch: ge-0/2/0 On the distribution switch: ge-0/0/0
Access port interfaces in VLAN **sales** (on access switch)	Avaya IP telephones: ge-0/0/3 through ge-0/0/19 Wireless access points: ge-0/0/0 and ge-0/0/1 Printers: ge-0/0/22 and ge-0/0/23 File servers: ge-0/0/20 and ge-0/0/21
Access port interfaces in VLAN **support** (on access switch)	Avaya IP telephones: ge-0/0/25 through ge-0/0/43 Wireless access points: ge-0/0/24 Printers: ge-0/0/44 and ge-0/0/45 File servers: ge-0/0/46 and ge-0/0/47

Configuring the Access Switch

To configure the access switch:

CLI Quick Configuration

To quickly configure the access switch, copy the following commands and paste them into the switch terminal window:

```
[edit]
set interfaces ge-0/0/0 unit 0 description "Sales wireless access point port"
set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members sales
set interfaces ge-0/0/3 unit 0 description "Sales phone port"
set interfaces ge-0/0/3 unit 0 family ethernet-switching vlan members sales
set interfaces ge-0/0/22 unit 0 description "Sales printer port"
set interfaces ge-0/0/22 unit 0 family ethernet-switching vlan members sales
set interfaces ge-0/0/20 unit 0 description "Sales file server port"
set interfaces ge-0/0/20 unit 0 family ethernet-switching vlan members sales
set interfaces ge-0/0/24 unit 0 description "Support wireless access point port"
set interfaces ge-0/0/24 unit 0 family ethernet-switching vlan members support
```

```
set interfaces ge-0/0/26 unit 0 description "Support phone port"
set interfaces ge-0/0/26 unit 0 family ethernet-switching vlan members support
set interfaces ge-0/0/44 unit 0 description "Support printer port"
set interfaces ge-0/0/44 unit 0 family ethernet-switching vlan members support
set interfaces ge-0/0/46 unit 0 description "Support file server port"
set interfaces ge-0/0/46 unit 0 family ethernet-switching vlan members support
set interfaces ge-0/2/0 unit 0 description "Uplink module port connection to distribution switch"
set interfaces ge-0/2/0 unit 0 family ethernet-switching interface-mode trunk
set interfaces ge-0/2/0 native-vlan-id 1
set interfaces ge-0/2/0 unit 0 family ethernet-switching vlan members [sales support]
set interfaces ge-0/2/0 unit 0 family ethernet-switching vlan members 1
set interfaces irb unit 0 family inet address 192.0.2.1/25
set interfaces irb unit 1 family inet address 192.0.2.129/25
set vlans sales description “Sales VLAN”
set vlans sales l3-interface irb.0
set vlans sales vlan-id 100
set vlans support description “Support VLAN”
set vlans support vlan-id 200
set vlans support l3-interface irb.1
```

Step-by-Step Procedure

To configure the access switch:

1. Configure the 1-Gigabit Ethernet interface on the uplink module to be the trunk port that connects to the distribution switch:

   ```
   [edit interfaces]
   user@access-switch# set ge-0/2/0 unit 0 description "Uplink module port connection to distribution switch"
   user@access-switch# set ge-0/2/0 unit 0 family ethernet-switching interface-mode trunk
   ```

2. Specify the VLANs to be aggregated on the trunk port:

   ```
   [edit interfaces]
   user@access-switch# set ge-0/2/0 unit 0 family ethernet-switching vlan members [ sales support ]
   ```

3. To handle untagged packets that are received on the trunk port, create a native VLAN by configuring a VLAN ID and specifying that the trunk port is a member of the native VLAN:

   ```
   [edit interfaces]
   user@access-switch# set ge-0/2/0 native-vlan-id 1
   user@access-switch# set ge-0/2/0 unit 0 family ethernet-switching vlan members 1
   ```

4. Configure the sales VLAN:

   ```
   [edit vlans]
   user@access-switch# set sales description "Sales VLAN"
   user@access-switch# set sales vlan-id 100
   user@access-switch# set sales l3-interface irb.0
   ```

5. Configure the support VLAN:

   ```
   [edit vlans]
   user@access-switch# set support description "Support VLAN"
   user@access-switch# set support vlan-id 200
   user@access-switch# set support l3-interface irb.1
   ```

6. Create the subnet for the sales VLAN:

   ```
   [edit interfaces]
   user@access-switch# set irb unit 0 family inet address 192.0.2.1/25
   ```

7. Create the subnet for the support VLAN:

   ```
   [edit interfaces]
   ```

```
user@access-switch# set irb unit 1 family inet address 192.0.2.129/25
```

8. Configure the interfaces in the sales VLAN:

```
[edit interfaces]
user@access-switch# set ge-0/0/0 unit 0 description "Sales wireless access point port"
user@access-switch# set ge-0/0/0 unit 0 family ethernet-switching vlan members sales
user@access-switch# set ge-0/0/3 unit 0 description "Sales phone port"
user@access-switch# set ge-0/0/3 unit 0 family ethernet-switching vlan members sales
user@access-switch# set ge-0/0/20 unit 0 description "Sales file server port"
user@access—switch# set ge-0/0/20 unit 0 family ethernet-switching vlan members sales
user@access-switch# set ge-0/0/22 unit 0 description "Sales printer port"
user@access-switch# set ge-0/0/22 unit 0 family ethernet-switching vlan members sales
```

9. Configure the interfaces in the support VLAN:

```
[edit interfaces]
user@access-switch# set ge-0/0/24 unit 0 description "Support wireless access point port"
user@access-switch# set ge-0/0/24 unit 0 family ethernet-switching vlan members support
user@access-switch# set ge-0/0/26 unit 0 description "Support phone port"
user@access-switch# set ge-0/0/26 unit 0 family ethernet-switching vlan members support
user@access-switch# set ge-0/0/44 unit 0 description "Support printer port"
user@access-switch# set ge-0/0/44 unit 0 family ethernet-switching vlan members support
user@access-switch# set ge-0/0/46 unit 0 description "Support file server port"
user@access-switch# set ge-0/0/46 unit 0 family ethernet-switching vlan members support
```

Results

Display the results of the configuration:

```
user@access-switch> show configuration
```

```
interfaces {
    ge-0/0/0 {
        unit 0 {
            description "Sales wireless access point port";
            family ethernet-switching {
                vlan {
                    members sales;
                }
            }
```

```
    }
}
ge-0/0/3 {
    unit 0 {
        description "Sales phone port";
        family ethernet-switching {
            vlan {
                members sales;
            }
        }
    }
}
ge-0/0/20 {
    unit 0 {
        description "Sales file server port";
        family ethernet-switching {
            vlan {
                members sales;
            }
        }
    }
}
ge-0/0/22 {
    unit 0 {
        description "Sales printer port";
        family ethernet-switching {
            vlan {
                members sales;
            }
        }
    }
}
ge-0/0/24 {
    unit 0 {
        description "Support wireless access point port";
        family ethernet-switching {
            vlan {
                members support;
            }
        }
    }
}
ge-0/0/26 {
    unit 0 {
```

```
        description "Support phone port";
        family ethernet-switching {
            vlan {
                members support;
            }
        }
    }
}
ge-0/0/44 {
    unit 0 {
        description "Support printer port";
        family ethernet-switching {
            vlan {
                members support;
            }
        }
    }
}
ge-0/0/46 {
    unit 0 {
        description "Support file server port";
        family ethernet-switching {
            vlan {
                members support;
            }
        }
    }
}
ge-0/2/0 {
    native-vlan-id 1;
    unit 0 {
        description "Uplinking module connection to distribution switch";
        family ethernet-switching {
            interface-mode trunk;
            vlan {
                members [ 1 sales support ];
            }
        }
    }
}
irb {
    unit 0 {
        family inet {
            address 192.0.2.1/25;
```

```
            }
        }
        unit 1 {
            family inet {
                address 192.0.2.129/25;
            }
        }
    }
}
vlans {
     sales {
        description "Sales VLAN";
        vlan-id 100;
        l3-interface irb.0;
    }
    support {
        description "Support VLAN";
        vlan-id 200;
        l3-interface irb.1;
    }
}
```

TIP: To quickly configure the access switch, issue the **load merge terminal** command, then copy the hierarchy and paste it into the switch terminal window.

Configuring the Distribution Switch

To configure the distribution switch:

CLI Quick Configuration

To quickly configure the distribution switch, copy the following commands and paste them into the switch terminal window:

set interfaces ge-0/0/0 unit 0 description "Connection to access switch"

set interfaces ge-0/0/0 unit 0 family ethernet-switching interface-mode trunk

set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members [sales support]

set interfaces ge-0/0/0 native-vlan-id 1

set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members 1

set interfaces irb unit 0 family inet address 192.0.2.2/25

set interfaces irb unit 1 family inet address 192.0.2.130/25

set vlans sales description "Sales VLAN"

set vlans sales vlan-id 100

set vlans sales l3-interface irb.0

set vlans support description "Support VLAN"

set vlans support vlan-id 200

set vlans support l3-interface irb.1

Step-by-Step Procedure

To configure the distribution switch:

1. Configure the interface on the switch to be the trunk port that connects to the access switch:

   ```
   [edit interfaces]
   user@distribution-switch# set ge-0/0/0 unit 0 description "Connection to access switch"
   user@distribution-switch# set ge-0/0/0 unit 0 family ethernet-switching interface-mode
   trunk
   ```

2. Specify the VLANs to be aggregated on the trunk port:

   ```
   [edit interfaces]
   user@distribution-switch# set ge-0/0/0 unit 0 family ethernet-switching vlan members [ sales
   support ]
   ```

3. To handle untagged packets that are received on the trunk port, create a native VLAN by configuring a VLAN ID and specifying that the trunk port is a member of the native VLAN:

   ```
   [edit interfaces]
   user@distribution-switch# set ge-0/0/0 native-vlan-id 1
   user@distribution-switch# set ge-0/0/0 unit 0 family ethernet-switching vlan members 1
   ```

4. Configure the sales VLAN:

   ```
   [edit vlans]
   user@distribution-switch# set sales description "Sales VLAN"
   user@distribution-switch# set sales vlan-id 100
   ```

```
user@distribution-switch# set sales l3-interface irb.0
```

The VLAN configuration for the distribution switch includes the **set l3-interface irb.0** command to route traffic between the sales and support VLANs. The VLAN configuration for the access switch does not include this statement because the access switch is not monitoring IP addresses. Instead, the access switch is passing the IP addresses to the distribution switch for interpretation.

5. Configure the support VLAN:

```
[edit vlans]
user@distribution-switch# set support description "Support VLAN"
user@distribution-switch# set support vlan-id 200
user@distribution-switch# set support l3-interface irb.1
```

The VLAN configuration for the distribution switch includes the **set l3-interface irb.1** command to route traffic between the sales and support VLANs. The VLAN configuration for the access switch does not include this statement because the access switch is not monitoring IP addresses. Instead, the access switch is passing the IP addresses to the distribution switch for interpretation.

6. Create the subnet for the sales VLAN:

```
[edit interfaces]
user@distribution-switch# set irb unit 0 family inet address 192.0.2.2/25
```

7. Create the subnet for the support VLAN:

```
[edit interfaces]
user@distribution-switch# set irb unit 1 family inet address 192.0.2.130/25
```

Results

Display the results of the configuration:

```
user@distribution-switch> show configuration
```

```
interfaces {
    ge-0/0/0 {
        native-vlan-id 1;
        unit 0 {
            description "Connection to access switch";
            family ethernet-switching {
                interface-mode trunk;
                vlan {
                    members [ 1 sales support ];
                }
```

```
            }
        }
    }
    irb {
        unit 0 {
            family inet {
                address 192.0.2.2/25;
            }
        }
        unit 1 {
            family inet {
                address 192.0.2.130/25;
            }
        }
    }
}
vlans {
    sales {
        description "Sales VLAN";
        vlan-id 100;
        l3-interface irb.0;
    }
    support {
        description "Support VLAN";
        vlan-id 200;
        l3-interface irb.1;
    }
}
```

TIP: To quickly configure the distribution switch, issue the **load merge terminal** command, then copy the hierarchy and paste it into the switch terminal window.

Verification

IN THIS SECTION

To confirm that the configuration is working properly, perform these tasks:

Verifying the VLAN Members and Interfaces on the Access Switch

Purpose

Verify that the **sales** and **support** VLANs have been created on the switch.

Action

List all VLANs configured on the switch:

```
user@access-switch> show vlans
```

```
Routing instance        VLAN name             Tag         Interfaces
default-switch          sales                    100
                                                            ge-0/0/20.0
                                                            ge-0/0/22.0
                                                            ge-0/0/3.0*
                                                            ge-0/0/0.0*
                                                            ge-0/2/0.0*
default-switch          support                  200
                                                            ge-0/0/24.0
                                                            ge-0/0/26.0
                                                            ge-0/0/44.0*
                                                            ge-0/0/46.0*
                                                            ge-0/2/0.0*
```

Meaning

The output shows the **sales** and **support** VLANs and the interfaces that are configured as members of the respective VLANs.

Verifying the VLAN Members and Interfaces on the Distribution Switch

Purpose

Verify that the **sales** and **support** VLANs have been created on the switch.

Action

List all VLANs configured on the switch:

```
user@distribution-switch> show vlans
```

```
Routing instance        VLAN name             Tag         Interfaces
default-switch          sales                    100
```

```
                                                    ge-0/0/0.0*
default-switch          support                 200
                                                    ge-0/0/0.0*
```

Meaning

The output shows the **sales** and **support** VLANs and the interface (ge-0/0/0.0) that is configured as a member of both VLANs. Interface ge-0/0/0.0 is also the trunk interface connected to the access switch.

Example: Setting Up Bridging with Multiple VLANs for EX Series Switches

IN THIS SECTION

To segment traffic on a LAN into separate broadcast domains, you create separate virtual LANs (VLANs) on an EX Series switch. Each VLAN is a collection of network nodes. When you use VLANs, frames whose origin and destination are in the same VLAN are forwarded only within the local VLAN, and only frames not destined for the local VLAN are forwarded to other broadcast domains. VLANs thus limit the amount of traffic flowing across the entire LAN, reducing the possible number of collisions and packet retransmissions within the LAN.

This example describes how to configure bridging for an EX Series switch and how to create two VLANs to segment the LAN:

Requirements

This example uses the following hardware and software components:

- One EX4200-48P Virtual Chassis switch
- Junos OS Release 9.0 or later for EX Series switches

Before you set up bridging and VLANs, be sure you have:

- Performed the initial switch configuration. See *Connecting and Configuring an EX Series Switch (J-Web Procedure)*.

Overview and Topology

EX Series switches connect all devices in an office or data center into a single LAN to provide sharing of common resources such as printers and file servers and to enable wireless devices to connect to the LAN through wireless access points. The default configuration creates a single VLAN, and all traffic on the switch is part of that broadcast domain. Creating separate network segments reduces the span of the broadcast domain and allows you to group related users and network resources without being limited by physical cabling or by the location of a network device in the building or on the LAN.

This example shows a simple configuration to illustrate the basic steps for creating two VLANs on a single switch. One VLAN, called **sales**, is for the sales and marketing group, and a second, called **support**, is for the customer support team. The sales and support groups each have their own dedicated file servers, printers, and wireless access points. For the switch ports to be segmented across the two VLANs, each VLAN must have its own broadcast domain, identified by a unique name and tag (VLAN ID). In addition, each VLAN must be on its own distinct IP subnet.

The topology for this example consists of one EX4200-48P switch, which has a total of 48 Gigabit Ethernet ports, all of which support Power over Ethernet (PoE). Most of the switch ports connect to Avaya IP telephones. The remainder of the ports connect to wireless access points, file servers, and printers. Table 60 on page 266 explains the components of the example topology.

Table 60: Components of the Multiple VLAN Topology

Property	Settings
Switch hardware	EX4200-48P, 48 Gigabit Ethernet ports, all PoE-enabled (**ge-0/0/0** through **ge-0/0/47**)
VLAN names and tag IDs	**sales**, tag **100** **support**, tag **200**
VLAN subnets	**sales**: **192.0.2.0/25** (addresses **192.0.2.1** through **192.0.2.126**) **support**: **192.0.2.128/25** (addresses **192.0.2.129** through **192.0.2.254**)
Interfaces in VLAN **sales**	Avaya IP telephones: **ge-0/0/3** through **ge-0/0/19** Wireless access points: **ge-0/0/0** and **ge-0/0/1** Printers: **ge-0/0/22** and **ge-0/0/23** File servers: **ge-0/0/20** and **ge-0/0/21**

Table 60: Components of the Multiple VLAN Topology *(continued)*

Property	Settings
Interfaces in VLAN **support**	Avaya IP telephones: **ge-0/0/25** through **ge-0/0/43** Wireless access points: **ge-0/0/24** Printers: **ge-0/0/44** and **ge-0/0/45** File servers: **ge-0/0/46** and **ge-0/0/47**
Unused interfaces	**ge-0/0/2** and **ge-0/0/25**

This configuration example creates two IP subnets, one for the sales VLAN and the second for the support VLAN. The switch bridges traffic within a VLAN. For traffic passing between two VLANs, the switch routes the traffic using a Layer 3 routing interface on which you have configured the address of the IP subnet.

To keep the example simple, the configuration steps show only a few devices in each of the VLANs. Use the same configuration procedure to add more LAN devices.

Configuration

Configure Layer 2 switching for two VLANs:

CLI Quick Configuration

To quickly configure Layer 2 switching for the two VLANs (**sales** and **support**) and to quickly configure Layer 3 routing of traffic between the two VLANs, copy the following commands and paste them into the switch terminal window:

```
[edit]
```

set interfaces ge-0/0/0 unit 0 description “Sales wireless access point port”

set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members sales

set interfaces ge-0/0/3 unit 0 description “Sales phone port”

set interfaces ge-0/0/3 unit 0 family ethernet-switching vlan members sales

set interfaces ge-0/0/22 unit 0 description “Sales printer port”

set interfaces ge-0/0/22 unit 0 family ethernet-switching vlan members sales

set interfaces ge-0/0/20 unit 0 description “Sales file server port”

set interfaces ge-0/0/20 unit 0 family ethernet-switching vlan members sales

set interfaces ge-0/0/24 unit 0 description “Support wireless access point port”

set interfaces ge-0/0/24 unit 0 family ethernet-switching vlan members support

set interfaces ge-0/0/26 unit 0 description “Support phone port”

set interfaces ge-0/0/26 unit 0 family ethernet-switching vlan members support

set interfaces ge-0/0/44 unit 0 description “Support printer port”

set interfaces ge-0/0/44 unit 0 family ethernet-switching vlan members support

set interfaces ge-0/0/46 unit 0 description “Support file server port”

set interfaces ge-0/0/46 unit 0 family ethernet-switching vlan members support

set interfaces vlan unit 0 family inet address 192.0.2.0/25

set interfaces vlan unit 1 family inet address 192.0.2.128/25

set vlans sales l3–interface vlan.0

set vlans sales vlan-id 100

set vlans support vlan-id 200

set vlans support l3-interface vlan.1

Step-by-Step Procedure

Configure the switch interfaces and the VLANs to which they belong. By default, all interfaces are in access mode, so you do not have to configure the port mode.

1. Configure the interface for the wireless access point in the sales VLAN:

```
[edit interfaces ge-0/0/0 unit 0]
user@switch# set description “Sales wireless access point port”
user@switch# set family ethernet-switching vlan members sales
```

2. Configure the interface for the Avaya IP phone in the sales VLAN:

```
[edit interfaces ge-0/0/3 unit 0]
user@switch# set description “Sales phone port”
user@switch# set family ethernet-switching vlan members sales
```

3. Configure the interface for the printer in the sales VLAN:

```
[edit interfaces ge-0/0/22 unit 0]
user@switch# set description “Sales printer port”
user@switch# set family ethernet-switching vlan members sales
```

4. Configure the interface for the file server in the sales VLAN:

```
[edit interfaces ge-0/0/20 unit 0]
user@switch# set description “Sales file server port”
user@switch# set family ethernet-switching vlan members sales
```

5. Configure the interface for the wireless access point in the support VLAN:

```
[edit interfaces ge-0/0/24 unit 0]
user@switch# set description “Support wireless access point port”
user@switch# set family ethernet-switching vlan members support
```

6. Configure the interface for the Avaya IP phone in the support VLAN:

```
[edit interfaces ge-0/0/26 unit 0]
user@switch# set description “Support phone port”
user@switch# set family ethernet-switching vlan members support
```

7. Configure the interface for the printer in the support VLAN:

```
[edit interfaces ge-0/0/44 unit 0]
user@switch# set description “Support printer port”
user@switch# set family ethernet-switching vlan members support
```

8. Configure the interface for the file server in the support VLAN:

```
[edit interfaces ge-0/0/46 unit 0]
user@switch# set description “Support file server port”
user@switch# set family ethernet-switching vlan members support
```

9. Create the subnet for the sales broadcast domain:

```
[edit interfaces]
user@switch# set vlan unit 0 family inet address 192.0.2.1/25
```

10. Create the subnet for the support broadcast domain:

```
[edit interfaces]
user@switch# set vlan unit 1 family inet address 192.0.2.129/25
```

11. Configure the VLAN tag IDs for the sales and support VLANs:

```
[edit vlans]
user@switch# set sales vlan-id 100
```

```
user@switch# set support vlan-id 200
```

12. To route traffic between the sales and support VLANs, define the interfaces that are members of each VLAN and associate a Layer 3 interface:

```
[edit vlans]
user@switch# set sales l3-interface
user@switch# set support l3-interface vlan.1
```

Results

Display the results of the configuration:

```
user@switch> show configuration
interfaces {
    ge-0/0/0 {
        unit 0 {
            description “Sales wireless access point port”;
            family ethernet-switching {
                vlan members sales;
            }
        }
    }
    ge-0/0/3 {
        unit 0 {
            description “Sales phone port”;
            family ethernet-switching {
                vlan members sales;
            }
        }
    }
    ge-0/0/22 {
        unit 0 {
            description “Sales printer port”;
            family ethernet-switching {
                vlan members sales;
            }
        }
    }
    ge-0/0/20 {
        unit 0 {
            description “Sales file server port”;
            family ethernet-switching {
                vlan members sales;
```

```
        }
      }
    }
    ge-0/0/24 {
      unit 0 {
        description “Support wireless access point port”;
        family ethernet-switching {
          vlan members support;
        }
      }
    }
    ge-0/0/26 {
      unit 0 {
        description “Support phone port”;
        family ethernet-switching {
          vlan members support;
        }
      }
    }
    ge-0/0/44 {
      unit 0 {
        description “Support printer port”;
        family ethernet-switching {
          vlan members support;
        }
      }
    }
    ge-0/0/46 {
      unit 0 {
        description “Support file server port”;
        family ethernet-switching {
          vlan members support;
        }
      }
      vlans {
        unit 0 {
          family inet address 192.0.2.0/25;
        }
        unit 1 {
          family inet address 192.0.2.128/25;
        }
      }
    }
  }
```

```
vlans {
    sales {
        vlan-id 100;
        interface ge-0/0/0.0:
        interface ge-0/0/3/0;
        interface ge-0/0/20.0;
        interface ge-0/0/22.0;
        l3-interface vlan 0;
    }
    support {
        vlan-id 200;
        interface ge-0/0/24.0:
        interface ge-0/0/26.0;
        interface ge-0/0/44.0;
        interface ge-0/0/46.0;
        l3-interface vlan 1;
    }
}
```

TIP: To quickly configure the sales and support VLAN interfaces, issue the **load merge terminal** command, then copy the hierarchy and paste it into the switch terminal window.

Verification

IN THIS SECTION

To verify that the “sales” and “support” VLANs have been created and are operating properly, perform these tasks:

Verifying That the VLANs Have Been Created and Associated to the Correct Interfaces

Purpose

Verify that the VLANs **sales** and **support** have been created on the switch and that all connected interfaces on the switch are members of the correct VLAN.

Action

List all VLANs configured on the switch:

Use the operational mode commands:

```
user@switch>  show vlans
```

```
Name            Tag     Interfaces
default
                        ge-0/0/1.0, ge-0/0/2.0, ge-0/0/4.0, ge-0/0/5.0,
                        ge-0/0/6.0,  ge-0/0/7.0, ge-0/0/8.0, ge-0/0/9.0,
                        ge-0/0/10.0*, ge-0/0/11.0, ge-0/0/12.0, ge-0/0/13.0*,
                        ge-0/0/14.0, ge-0/0/15.0, ge-0/0/16.0, ge-0/0/17.0,
                        ge-0/0/18.0, ge-0/0/19.0, ge-0/0/21.0, ge-0/0/23.0*,
                        ge-0/0/25.0, ge-0/0/27.0, ge-0/0/28.0, ge-0/0/29.0,
                        ge-0/0/30.0, ge-0/0/31.0, ge-0/0/32.0, ge-0/0/33.0,
                        ge-0/0/34.0, ge-0/0/35.0, ge-0/0/36.0, ge-0/0/37.0,
                        ge-0/0/38.0, ge-0/0/39.0, ge-0/0/40.0, ge-0/0/41.0,
                        ge-0/0/42.0, ge-0/0/43.0, ge-0/0/45.0, ge-0/0/47.0,
                        ge-0/1/0.0*, ge-0/1/1.0*, ge-0/1/2.0*, ge-0/1/3.0*

sales       100
                        ge-0/0/0.0*, ge-0/0/3.0, ge-0/0/20.0, ge-0/0/22.0

support     200
                        ge-0/0/24.0, ge-0/0/26.0, ge-0/0/44.0, ge-0/0/46.0*
mgmt
                        me0.0*
```

Meaning

The **show vlans** command lists all VLANs configured on the switch and which interfaces are members of each VLAN. This command output shows that the **sales** and **support** VLANs have been created. The **sales** VLAN has a tag ID of 100 and is associated with interfaces **ge-0/0/0.0**, **ge-0/0/3.0**, **ge-0/0/20.0**, and **ge-0/0/22.0**. VLAN **support** has a tag ID of 200 and is associated with interfaces **ge-0/0/24.0**, **ge-0/0/26.0**, **ge-0/0/44.0**, and **ge-0/0/46.0**.

Verifying That Traffic Is Being Routed Between the Two VLANs

Purpose

Verify routing between the two VLANs.

Action

List the Layer 3 routes in the switch's Address Resolution Protocol (ARP) table:

```
user@switch> show arp
```

```
MAC Address       Address         Name                    Flags

00:00:0c:06:2c:0d        192.0.2.3       vlan.0                  None
00:13:e2:50:62:e0        192.0.2.11      vlan.1                  None
```

Meaning

Sending IP packets on a multiaccess network requires mapping from an IP address to a MAC address (the physical or hardware address). The ARP table displays the mapping between the IP address and MAC address for both **vlan.0** (associated with **sales**) and **vlan.1** (associated with **support**). These VLANs can route traffic to each other.

Verifying That Traffic Is Being Switched Between the Two VLANs

Purpose

Verify that learned entries are being added to the Ethernet switching table.

Action

List the contents of the Ethernet switching table:

```
user@switch> show ethernet-switching table
```

```
Ethernet-switching table: 8 entries, 5 learned
  VLAN              MAC address       Type         Age Interfaces
  default           *                 Flood          - All-members
  default           00:00:05:00:00:01 Learn          - ge-0/0/10.0
  default           00:00:5e:00:01:09 Learn          - ge-0/0/13.0
  default           00:19:e2:50:63:e0 Learn          - ge-0/0/23.0
  sales             *                 Flood          - All-members
  sales             00:00:5e:00:07:09 Learn          - ge-0/0/0.0
  support           *                 Flood          – All–members
  support           00:00:5e:00:01:01 Learn          – ge-0/0/46.0
```

Meaning

The output shows that learned entries for the **sales** and **support** VLANs have been added to the Ethernet switching table, and are associated with interfaces **ge-0/0/0.0** and **ge-0/0/46.0**. Even though the VLANs were associated with more than one interface in the configuration, these interfaces are the only ones that are currently operating.

SEE ALSO

Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch | **226**

Example: Connecting an EX Series Access Switch to a Distribution Switch

Understanding Bridging and VLANs on Switches | **168**

Example: Connecting an Access Switch to a Distribution Switch

IN THIS SECTION

In large local area networks (LANs), you commonly need to aggregate traffic from a number of access switches into a distribution switch.

This example describes how to connect an access switch to a distribution switch:

Requirements

This example uses the following hardware and software components:

- For the distribution switch, one EX 4200-24F switch. This model is designed to be used as a distribution switch for aggregation or collapsed core network topologies and in space-constrained data centers. It has twenty-four 1-Gigabit Ethernet fiber SFP ports and an EX-UM-2XFP uplink module with two 10-Gigabit Ethernet XFP ports.

- For the access switch, one EX 3200-24P, which has twenty-four 1-Gigabit Ethernet ports, all of which support Power over Ethernet (PoE), and an uplink module with four 1-Gigabit Ethernet ports.
- Junos OS Release 11.1 or later for the QFX Series

Overview and Topology

In a large office that is spread across several floors or buildings, or in a data center, you commonly aggregate traffic from a number of access switches into a distribution switch. This configuration example shows a simple topology to illustrate how to connect a single access switch to a distribution switch.

In the topology, the LAN is segmented into two VLANs, one for the sales department and the second for the support team. One 1-Gigabit Ethernet port on the access switch's uplink module connects to the distribution switch, to one 1-Gigabit Ethernet port on the distribution switch.

Table 61 on page 276 explains the components of the example topology. The example shows how to configure one of the three access switches. The other access switches could be configured in the same manner.

Table 61: Components of the Topology for Connecting an Access Switch to a Distribution Switch

Property	Settings
Access switch hardware	EX 3200-24P, 24 1-Gigabit Ethernet ports, all PoE-enabled (**ge-0/0/0** through **ge-0/0/23**); one 4-port 1–Gigabit Ethernet uplink module (EX-UM-4SFP)
Distribution switch hardware	EX 4200-24F, 24 1-Gigabit Ethernet fiber SPF ports (**ge-0/0/0** through **ge-0/0/23**); one 2–port 10–Gigabit Ethernet XFP uplink module (EX-UM-4SFP)
VLAN names and tag IDs	**sales**, tag **100support**, tag **200**
VLAN subnets	**sales**: **192.0.2.0/25** (addresses **192.0.2.1** through **192.0.2.126**)**support**: **192.0.2.128/25** (addresses **192.0.2.129** through **192.0.2.254**)
Trunk port interfaces	On the access switch: **ge-0/1/0**On the distribution switch: **ge-0/0/0**
Access port interfaces in VLAN **sales** (on access switch)	Avaya IP telephones: **ge-0/0/3** through **ge-0/0/19**Wireless access points: **ge-0/0/0** and **ge-0/0/1**Printers: **ge-0/0/22** and **ge-0/0/23**File servers: **ge-0/0/20** and **ge-0/0/21**
Access port interfaces in VLAN **support** (on access switch)	Avaya IP telephones: **ge-0/0/25** through **ge-0/0/43**Wireless access points: **ge-0/0/24**Printers: **ge-0/0/44** and **ge-0/0/45**File servers: **ge-0/0/46** and **ge-0/0/47**

Table 61: Components of the Topology for Connecting an Access Switch to a Distribution Switch *(continued)*

Unused interfaces on access switch	**ge-0/0/2** and **ge-0/0/25**

Configuring the Access Switch

To configure the access switch:

CLI Quick Configuration

To quickly configure the access switch, copy the following commands and paste them into the switch terminal window:

```
[edit]
```

set interfaces ge-0/0/0 unit 0 description “Sales Wireless access point port”

set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members sales

set interfaces ge-0/0/3 unit 0 description “Sales phone port”

set interfaces ge-0/0/3 unit 0 family ethernet-switching vlan members sales

set interfaces ge-0/0/22 unit 0 description “Sales printer port”

set interfaces ge-0/0/22 unit 0 family ethernet-switching vlan members sales

set interfaces ge-0/0/20 unit 0 description “Sales file server port”

set interfaces ge-0/0/20 unit 0 family ethernet-switching vlan members sales

set interfaces ge-0/0/24 unit 0 description “Support wireless access point port”

set interfaces ge-0/0/24 unit 0 family ethernet-switching vlan members support

set interfaces ge-0/0/26 unit 0 description “Support phone port”

set interfaces ge-0/0/26 unit 0 family ethernet-switching vlan members support

set interfaces ge-0/0/44 unit 0 description “Support printer port”

set interfaces ge-0/0/44 unit 0 family ethernet-switching vlan members support

set interfaces ge-0/0/46 unit 0 description “Support file server port”

set interfaces ge-0/0/46 unit 0 family ethernet-switching vlan members support

set interfaces ge-0/1/0 unit 0 description “Uplink module port connection to distribution switch”

set interfaces ge-0/1/0 unit 0 family ethernet-switching port-mode trunk

set interfaces ge-0/1/0 unit 0 family ethernet-switching native-vlan-id 1

set interfaces ge-0/1/0 unit 0 family ethernet switching vlan members [sales support]

set interfaces vlan unit 0 family inet address 192.0.2.1/25

set interfaces vlan unit 1 family inet address 192.0.2.129/25

set vlans sales interface ge-0/0/0.0

set vlans sales interface ge-0/0/3.0

set vlans sales interface ge-0/0/22.0

set vlans sales interface ge-0/0/20.0

set vlans sales l3-interface vlan.0

set vlans sales vlan-id 100

set vlans sales vlan-description "Sales VLAN"

set vlans support interface ge-0/0/24.0

set vlans support interface ge-0/0/26.0

set vlans support interface ge-0/0/44.0

set vlans support interface ge-0/0/46.0

set vlans support vlan-id 200

set vlans support l3–interface vlan.1

set vlans support vlan-description "Support VLAN"

Step-by-Step Procedure

To configure the access switch:

1. Configure the 1-Gigabit Ethernet interface on the uplink module to be the trunk port that connects to the distribution switch:

   `[edit interfaces ge-0/1/0 unit 0]user@access-switch#` **set**description **"Uplink module port connection to distribution switch"**`user@access-switch#` **set ethernet-switching** port-mode **trunk**

2. Specify the VLANs to be aggregated on the trunk port:

```
[edit interfaces ge-0/1/0 unit 0]user@access-switch# set ethernet-switching
vlanmembers [ sales support ]
```

3. Configure the VLAN ID to use for packets that are received with no dot1q tag (untagged packets):

```
[edit interfaces ge-0/1/0 unit 0]user@access-switch# set ethernet-switching
native-vlan-id 1
```

4. Configure the sales VLAN:

```
[edit vlans sales]user@access-switch# set vlan-description “Sales
VLAN”user@access-switch# set vlan-id 100user@access-switch# set l3-interface (VLAN)
vlan.0
```

5. Configure the support VLAN:

```
[edit vlans support]user@access-switch# set vlan-description “Support
VLAN”user@access-switch# set vlan-id 200user@access-switch# set l3-interface (VLAN)
vlan.1
```

6. Create the subnet for the sales broadcast domain:

```
[edit interfaces]user@access-switch# set vlan unit 0 family inet address 192.0.2.1/25
```

7. Create the subnet for the support broadcast domain:

```
[edit interfaces]user@access-switch# set vlan unit 1 family inet address 192.0.2.129/25
```

8. Configure the interfaces in the sales VLAN:

```
[edit interfaces]user@access-switch# set ge-0/0/0 unit 0 description “Sales wireless
access point port”user@access-switch# set ge-0/0/0 unit 0 family ethernet-switching vlan
members salesuser@access-switch# set ge-0/0/3 unit 0 description “Sales phone
port”user@access-switch# set ge-0/0/3 unit 0 family ethernet-switching vlan members
salesuser@access-switch# set ge-0/0/20 unit 0 description “Sales file server
port”user@access—switch# set ge-0/0/20 unit 0 family ethernet-switching vlan members
salesuser@access-switch# set ge-0/0/22 unit 0 description “Sales printer
port”user@access-switch# set ge-0/0/22 unit 0 family ethernet-switching vlan members sales
```

9. Configure the interfaces in the support VLAN:

```
[edit interfaces]user@access-switch# set ge-0/0/24 unit 0 description “Support wireless
access point port”user@access-switch# set ge-0/0/24 unit 0 family ethernet-switching vlan
```

members supportuser@access-switch# **set ge-0/0/26 unit 0 description “Support phone port”**user@access-switch# **set ge-0/0/26 unit 0 family ethernet-switching vlan members support**user@access-switch# **set ge-0/0/44 unit 0 description “Support printer port”**user@access-switch# **set ge-0/0/44 unit 0 family ethernet-switching vlan members support**user@access-switch# **set ge-0/0/46 unit 0 description “Support file server port”**user@access-switch# **set ge-0/0/46 unit 0 family ethernet-switching vlan members support**

10. Configure descriptions and VLAN tag IDs for the sales and support VLANs:

    [edit vlans]user@access-switch# **set sales vlan-description “Sales VLAN”**user@access-switch# **set sales vlan-id 100**user@access-switch# **set support vlan-description “Support VLAN”**user@access-switch# **set support vlan-id 200**

11. To route traffic between the sales and support VLANs and associate a Layer 3 interface with each VLAN:

    [edit vlans]user@access-switch# **set sales l3-interface vlan.0**user@access-switch# **set support l3-interface vlan.1**

Results

Display the results of the configuration:

```
user@access-switch> show
interfaces {
    ge-0/0/0 {
        unit 0 {
            description “Sales wireless access point port”;
            family ethernet-switching {
                vlan members sales;
            }
        }
    }
    ge-0/0/3 {
        unit 0 {
            description “Sales phone port”;
                family ethernet-switching {
                vlan members sales;
            }
        }
    }
    ge-0/0/20 {
        unit 0 {
```

```
        description “Sales file server port”;
            family ethernet-switching {
            vlan members sales;
        }
    }
}
ge-0/0/22 {
    unit 0 {
        description “Sales printer port”;
            family ethernet-switching {
            vlan members sales;
        }
    }
}
ge-0/0/24 {
    unit 0 {
        description “Support wireless access point port”;
            family ethernet-switching {
            vlan members support;
        }
    }
}
ge-0/0/26 {
    unit 0 {
        description “Support phone port”;
            family ethernet-switching {
            vlan members support;
        }
    }
}
ge-0/0/44 {
    unit 0 {
        description “Support printer port”;
            family ethernet-switching {
            vlan members sales;
        }
    }
}
ge-0/0/46 {
    unit 0 {
        description “Support file server port”;
            family ethernet-switching {
            vlan members support;
        }
```

```
        }
    }
    ge-0/1/0 {
        unit 0 {
            description “Uplink module port connection to distribution switch”;
                family ethernet-switching {
                port-mode trunk;
                    vlan members [ sales support ];
                native-vlan-id 1;
            }
        }
    }
    vlan {
        unit 0 {
            family inet address 192.0.2.1/25;
        }
        unit 1 {
            family inet address 192.0.2.129/25;
        }
    }
}
vlans {
    sales {
        vlan-id 100;
        vlan-description “Sales VLAN”;
        l3-interface vlan.0;
    }
    support {
        vlan-id 200;
        vlan-description “Support VLAN”;
        l3-interface vlan.1;
    }
}
```

TIP: To quickly configure the distribution switch, issue the **load merge terminal** command, then copy the hierarchy and paste it into the switch terminal window.

Configuring the Distribution Switch

To configure the distribution switch:

CLI Quick Configuration

To quickly configure the distribution switch, copy the following commands and paste them into the switch terminal window:

set interfaces ge-0/0/0 description “Connection to access switch”

set interfaces ge-0/0/0 ethernet-switching port-mode trunk

set interfaces ge-0/0/0 ethernet-switching vlan members [sales support]

set interfaces ge-0/0/0 ethernet-switching native-vlan-id 1

set interfaces vlan unit 0 family inet address 192.0.2.2/25

set interfaces vlan unit 1 family inet address 192.0.2.130/25

set vlans sales vlan-description “Sales VLAN”

set vlans sales vlan-id 100

set vlans sales l3-interface vlan.0

set vlans support vlan-description “Support VLAN”

set vlans support vlan-id 200

set vlans support l3-interface vlan.1

Step-by-Step Procedure

To configure the distribution switch:

1. Configure the interface on the switch to be the trunk port that connects to the access switch:

   `[edit interfaces ge-0/0/0 unit 0]user@distribution-switch#` **set description “Connection to access switch”**`user@distribution-switch#` **set ethernet-switching port-mode trunk**

2. Specify the VLANs to be aggregated on the trunk port:

   `[edit interfaces ge-0/0/0 unit 0]user@distribution-switch#` **set ethernet-switching vlanmembers [sales support]**

3. Configure the VLAN ID to use for packets that are received with no dot1q tag (untagged packets):

   `[edit interfaces]user@distribution-switch#` **set ge-0/0/0 ethernet-switching native-vlan-id** `1`

4. Configure the sales VLAN:

   ```
   [edit vlans sales]user@distribution-switch# set vlan-description “Sales VLAN”user@distribution-switch# set vlan-id 100user@distribution-switch# set l3-interface (VLAN) vlan.0
   ```

5. Configure the support VLAN:

   ```
   [edit vlans support]user@distribution-switch# set vlan-description “Support VLAN”user@distribution-switch# set vlan-id 200user@distribution-switch# set l3-interface (VLAN) vlan.1
   ```

6. Create the subnet for the sales broadcast domain:

   ```
   [edit interfaces]user@distribution-switch# set vlan unit 0 family inet address 192.0.2.2/25
   ```

7. Create the subnet for the support broadcast domain:

   ```
   [edit interfaces] user@distribution-switch# set vlan unit 1 family inet address 192.0.2.130/25
   ```

Results

Display the results of the configuration:

```
user@distribution-switch> show
interfaces {
    ge-0/0/0 {
        description “Connection to access switch”;
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
                vlan members [ sales support ];
                native-vlan-id 1;
            }
        }
    }
    vlan {
        unit 0 {
            family inet address 192.0.2.2/25;
        }
        unit 1 {
            family inet address 192.0.2.130/25;
```

```
        }
    }
}
vlans {
    sales {
        vlan-id 100;
        vlan-description “Sales VLAN”;
        l3-interface vlan.0;
    }
    support {
        vlan-id 200;
        vlan-description “Support VLAN”;
        l3-interface vlan.1;
    }
}
```

TIP: To quickly configure the distribution switch, issue the **load merge terminal** command, then copy the hierarchy and paste it into the switch terminal window.

Verification

IN THIS SECTION

To confirm that the configuration is working properly, perform these tasks:

Verifying the VLAN Members and Interfaces on the Access Switch

Purpose

Verify that the **sales** and **support** have been created on the switch.

Action

List all VLANs configured on the switch:

```
user@switch> show vlans
```

```
Name            Tag     Interfaces
default
                        ge-0/0/1.0, ge-0/0/2.0, ge-0/0/4.0, ge-0/0/5.0,
                        ge-0/0/6.0, ge-0/0/7.0, ge-0/0/8.0*, ge-0/0/9.0,

                        ge-0/0/10.0, ge-0/0/11.0*, ge-0/0/12.0, ge-0/0/13.0,
                        ge-0/0/14.0, ge-0/0/15.0, ge-0/0/16.0, ge-0/0/17.0,
                        ge-0/0/18.0, ge-0/0/19.0*,ge-0/0/21.0, ge-0/0/23.0,
                        ge-0/0/25.0, ge-0/0/27.0*,ge-0/0/28.0, ge-0/0/29.0,
                        ge-0/0/30.0, ge-0/0/31.0*,ge-0/0/32.0, ge-0/0/33.0,
                        ge-0/0/34.0, ge-0/0/35.0*,ge-0/0/36.0, ge-0/0/37.0,
                        ge-0/0/38.0, ge-0/0/39.0*,ge-0/0/40.0, ge-0/0/41.0,
                        ge-0/0/42.0, ge-0/0/43.0*,ge-0/0/45.0, ge-0/0/47.0,
                        ge-0/1/1.0*, ge-0/1/2.0*, ge-0/1/3.0*

sales           100
                        ge-0/0/0.0*, ge-0/0/3.0, ge-0/0/20.0, ge-0/0/22.0,
                        ge-0/1/0.0*,

support         200
                        ge-0/0/24.0*, ge-0/0/26.0, ge-0/0/44.0, ge-0/0/46.0,

mgmt
                        me0.0*
```

Meaning

The output shows the **sales** and **support** VLANs and the interfaces associated with them.

Verifying the VLAN Members and Interfaces on the Distribution Switch

Purpose

Verify that the **sales** and **support** have been created on the switch.

Action

List all VLANs configured on the switch:

```
user@switch> show vlans
```

```
Name            Tag     Interfaces
default
                        ge-0/0/1.0, ge-0/0/2.0, ge-0/0/3.0, ge-0/0/4.0,
```

```
                      ge-0/0/5.0, ge-0/0/6.0, ge-0/0/7.0*, ge-0/0/8.0,

                      ge-0/0/9.0, ge-0/0/10.0*, ge-0/0/11.0, ge-0/0/12.0,
                      ge-0/0/13.0, ge-0/0/14.0, ge-0/0/15.0, ge-0/0/16.0,
                      ge-0/0/17.0, ge-0/0/18.0*, ge-0/0/19.0, ge-0/0/20.0,
                      ge-0/0/21.0, ge-0/0/22.0*, ge-0/0/23.0, ge-0/1/1.0*,
                      ge-0/1/2.0*, ge-0/1/3.0*

sales         100
                      ge-0/0/0.0*

support       200
                      ge-0/0/0.0*

mgmt
                      me0.0*
```

Meaning

The output shows the **sales** and **support** VLANs associated to interface **ge-0/0/0.0**. Interface **ge-0/0/0.0** is the trunk interface connected to the access switch.

Configuring a Logical Interface for Access Mode

Enterprise network administrators can configure a single logical interface to accept untagged packets and forward the packets within a specified VLAN. A logical interface configured to accept untagged packets is called an *access interface* or *access port*.

```
interface-mode access;
```

You can include this statement at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number* family ethernet-switching]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family ethernet-switching]**

When an untagged or tagged packet is received on an access interface, the packet is accepted, the VLAN ID is added to the packet, and the packet is forwarded within the VLAN that is configured with the matching VLAN ID.

The following example configures a logical interface as an access port with a VLAN ID of 20 on routers and switches that support the enhanced Layer 2 software:

```
[edit interfaces ge-1/2/0]
unit 1 {
    family ethernet-switching {
        interface-mode access;
        vlan members 20;
    }
}
```

SEE ALSO

Ethernet Interfaces User Guide for Routing Devices

Configuring the Native VLAN Identifier

> **NOTE:** This task uses Junos OS for EX Series switches and Junos OS for QFX3500 and QFX3600 switches that does not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see "Configuring the Native VLAN Identifier on Switches With ELS Support" on page 290. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50.

EX Series switches support receiving and forwarding routed or bridged Ethernet frames with 802.1Q VLAN tags. The logical interface on which untagged packets are to be received must be configured with the same native VLAN ID as that configured on the physical interface.

To configure the native VLAN ID using the CLI:

1. Configure the port mode so that the interface is in multiple VLANs and can multiplex traffic between different VLANs. Trunk interfaces typically connect to other switches and to routers on the LAN. Configure the port mode as **trunk**:

   ```
   [edit interfaces ge-0/0/3 unit 0 family ethernet-switching]
   user@switch# set port-mode trunk
   ```

2. Configure the native VLAN ID:

   ```
   [edit interfaces ge-0/0/3 unit 0 family ethernet-switching]
   user@switch# set native-vlan-id 3
   ```

Configuring the Native VLAN Identifier on Switches With ELS Support

> **NOTE:** This task uses Junos OS for EX Series switches and Junos OS for QFX3500 and QFX3600 switches with support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does not support ELS, see "Configuring the Native VLAN Identifier" on page 52. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50.

Switches can receive and forward routed or bridged Ethernet frames with 802.1Q VLAN tags. Typically, trunk ports, which connect switches to each other, accept untagged control packets but do not accept untagged data packets. You can enable a trunk port to accept untagged data packets by configuring a native VLAN ID on the interface on which you want the untagged data packets to be received. The logical interface on which untagged packets are to be received must be configured with the same VLAN ID as the native VLAN ID configured on the physical interface.

To configure the native VLAN ID by using the command-line interface (CLI):

1. On the interface on which you want untagged data packets to be received, set the interface mode to **trunk**, which specifies that the interface is in multiple VLANs and can multiplex traffic between different VLANs.:

   ```
   [edit interfaces]
   user@switch# set interface-name unit logical-unit-number family
   ethernet-switching interface-mode trunk
   ```

2. Configure the native VLAN ID:

   ```
   [edit interfaces]
   user@switch# set interface-name native-vlan-id vlan-id
   ```

3. Specify that the logical interface that will receive the untagged data packets is a member of the native VLAN:

   ```
   [edit interfaces]
   user@switch# set interface-name unit logical-unit-number family
   ethernet-switching vlan members vlan-id
   ```

Configuring VLAN Encapsulation

To configure encapsulation on an interface, enter the **encapsulation** statement at the **[edit interfaces *interface-name*]** hierarchy level:

```
[edit interfaces interface-name]
encapsulation type;
```

The following list contains important notes regarding encapsulation:

- Ethernet interfaces in VLAN mode can have multiple logical interfaces. In CCC and VPLS modes, VLAN IDs from 1 through 511 are reserved for normal VLANs, and VLAN IDs 512 through 4094 are reserved for CCC or VPLS VLANs. For 4-port Fast Ethernet interfaces, you can use VLAN IDs 512 through 1024 for CCC or VPLS VLANs.
- For encapsulation type **flexible-ethernet-services**, all VLAN IDs are valid.
- For some encapsulation types, including flexible Ethernet services, Ethernet VLAN CCC, and VLAN VPLS, you can also configure the encapsulation type that is used inside the VLAN circuit itself. To do this, include the **encapsulation** statement:

  ```
  encapsulation (vlan-ccc | vlan-tcc | vlan-vpls);
  ```

  You can include this statement at the following hierarchy levels:

  - **[edit interfaces *interface-name* unit *logical-unit-number*]**
  - **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]**
- You cannot configure a logical interface with VLAN CCC or VLAN VPLS encapsulation unless you also configure the physical device with the same encapsulation or with flexible Ethernet services encapsulation. In general, the logical interface must have a VLAN ID of 512 or higher; if the VLAN ID is 511 or lower, it will be subject to the normal destination filter lookups in addition to source address filtering. However if you configure flexible Ethernet services encapsulation, this VLAN ID restriction is removed.

In general, you configure an interface's encapsulation at the **[edit interfaces *interface-name*]** hierarchy level.

Example: Configuring VLAN Encapsulation on a Gigabit Ethernet Interface

Configure VLAN CCC encapsulation on a Gigabit Ethernet interface:

```
interfaces ge-2/1/0 {
    vlan-tagging;
    encapsulation vlan-ccc;
```

```
    unit 0 {
        encapsulation vlan-ccc;
        vlan-id 600;
    }
}
```

Example: Configuring VLAN Encapsulation on an Aggregated Ethernet Interface

Configure VLAN CCC encapsulation on an aggregated Gigabit Ethernet interface:

```
interfaces ae0 {
    vlan-tagging;
    encapsulation vlan-vpls;
    unit 0 {
        vlan-id 100;
    }
}
```

SEE ALSO

Ethernet Interfaces User Guide for Routing Devices

10

CHAPTER

Configuring 802.1Q VLANs

802.1Q VLANs Overview

For Ethernet, Fast Ethernet, Tri-Rate Ethernet copper, Gigabit Ethernet, 10-Gigabit Ethernet, and aggregated Ethernet interfaces supporting VPLS, the Junos OS supports a subset of the IEEE 802.1Q standard for channelizing an Ethernet interface into multiple logical interfaces, allowing many hosts to be connected to the same Gigabit Ethernet switch, but preventing them from being in the same routing or bridging domain.

RELATED DOCUMENTATION

802.1Q VLAN IDs and Ethernet Interface Types

A VLAN (virtual LAN) abstracts the idea of the local area network (LAN) by providing data link connectivity for a subnet. VLANs make it easy for network administrators to partition a single switched network to match the functional and security requirements of their systems without having to run new cables or make major changes in their current network infrastructure. Each VLAN can be uniquely identified by VLAN ID, which is transmitted & received as IEEE 802.1Q tag in an Ethernet frame.

You can partition the router into up to 4095 different VLANs—depending on the router model and the physical interface types—by associating logical interfaces with specific VLAN IDs.

VLAN ID 0 is reserved for tagging the priority of frames. VLAN IDs 1 through 511 are reserved for normal VLANs. VLAN IDs 512 and above are reserved for VLAN circuit cross-connect (CCCs).

For Gigabit Ethernet IQ interfaces and Gigabit Ethernet PICs with SFPs (except the 10-port Gigabit Ethernet PIC and the built-in Gigabit Ethernet port on the M7i router), you can configure flexible Ethernet services encapsulation on the physical interface. With flexible Ethernet services encapsulation, VLAN IDs from 1 through 511 are no longer reserved for normal VLANs.

The maximum number of user-configurable VLANs is 15 on each port of the Dense-FE PIC (8-port/12-port/48-port).

Table 62 on page 296 lists VLAN ID range by interface type.

Table 62: VLAN ID Range by Interface Type

Interface Type	VLAN ID Range
Aggregated Ethernet for Fast Ethernet	1 through 1023
Aggregate Ethernet for Gigabit Ethernet	1 through 4094
4-port, 8-port, and 12-port Fast Ethernet	1 through 1023
48-port Fast Ethernet	1 through 4094
Tri-Rate Ethernet copper	1 through 4094
Gigabit Ethernet	1 through 4094
Gigabit Ethernet IQ	1 through 4094
10-Gigabit Ethernet	1 through 4094
100-Gigabit Ethernet	1 through 4094

Table 62: VLAN ID Range by Interface Type *(continued)*

Interface Type	VLAN ID Range
Management and internal Ethernet interfaces	1 through 1023

NOTE: For Gigabit Ethernet IQ and Gigabit Ethernet PICs with SFPs (except the built-in Gigabit Ethernet port on the M7i router), VLAN IDs on a single interface can differ from each other.

Because IS-IS has an 8-bit limit for broadcast multiaccess media, you cannot set up more than 255 adjacencies over Gigabit Ethernet using VLAN tagging. For more information, see the *Junos OS Routing Protocols Library*.

RELATED DOCUMENTATION

802.1Q VLANs Overview | **295**

Ethernet Interfaces User Guide for Routing Devices

Configuring Dynamic 802.1Q VLANs

You can configure the router to dynamically create VLANs when a client accesses an interface and requests a VLAN ID that does not yet exist. When a client accesses a VLAN interface, the router instantiates a VLAN dynamic profile that you have associated with the interface. Using the settings in the dynamic profile, the router extracts information about the client from the incoming packet (for example, the interface and unit values), saves this information in the routing table, and creates a VLAN or stacked VLAN ID for the client from a range of VLAN IDs that you configure for the interface.

Dynamically configuring VLANs or stacked VLANs requires the following general steps:

1. Configure a dynamic profile for dynamic VLAN or dynamic stacked VLAN creation.
2. Associate the VLAN or stacked VLAN dynamic profile with the interface.
3. Specify the Ethernet packet type that the VLAN dynamic profile accepts.
4. Define VLAN ranges for use by the dynamic profile when creating VLAN IDs.

For procedures on how to configure dynamic VLANs and dynamic stacked VLANs for client access, see the *Junos OS Broadband Subscriber Management and Services Library*.

RELATED DOCUMENTATION

802.1Q VLANs Overview | 295

Ethernet Interfaces User Guide for Routing Devices

Enabling VLAN Tagging

You can configure the router to receive and forward single-tag frames, dual-tag frames, or a mixture of single-tag and dual-tag frames.

NOTE: If you configure VLAN tagging on Gigabit Ethernet IQ, IQ2, and IQ2-E interfaces on M320, M120, and T Series routers, Junos OS creates an internal logical interface that reserves 50 Kbps of bandwidth from Gigabit Ethernet IQ interfaces and 2 Mbps of bandwidth from Gigabit Ethernet IQ2 and IQ2-E interfaces. As a result, the effective available bandwidth for these interface types is now 999.5 Mbps and 998 Mbps, respectively.

1. To configure the router to receive and forward single-tag frames with 802.1Q VLAN tags, include the **vlan-tagging** statement at the **[edit interfaces *interface-name*]** hierarchy level:

   ```
   [edit interfaces interface-name]
   user@host# vlan-tagging;
   ```

2. To configure the router to receive and forward dual-tag frames with 802.1Q VLAN tags, include the **stacked-vlan-tagging** statement at the **[edit interfaces *interface-name*]** hierarchy level:

   ```
   [edit interfaces interface-name]
   user@host# stacked-vlan-tagging;
   ```

3. Mixed tagging is supported for Gigabit Ethernet interfaces on Gigabit Ethernet IQ2 and IQ2-E, and IQ or IQE PICs on M Series and T Series routers, for all router Gigabit and 10-Gigabit Ethernet interfaces on MX Series routers, and for aggregated Ethernet interfaces with member links in IQ2 and IQ2-E PICs or in MX Series DPCs. Mixed tagging enables to configure two logical interfaces on the same Ethernet port, one with single-tag framing and one with dual-tag framing.

NOTE: Mixed tagging is not supported on Fast Ethernet interfaces.

To configure mixed tagging:

a. Configure the **flexible-vlan-tagging** statement at the **[edit interfaces ge-*fpc*/*pic*/*port*]** hierarchy level.

```
[edit interfaces ge-fpc/pic/port]
user@host#flexible-vlan-tagging;
```

b. Configure the **vlan-tags** statement with **inner** and **outer** options or the **vlan-id** statement at the **[edit interfaces ge-*fpc*/*pic*/*port* unit *logical-unit-number*]** hierarchy level:

```
[edit interfaces ge-fpc/pic/port unit logival-unit-number]
user@host# vlan-id number;
family family {
   address address;
}
user@host# vlan-tags inner tpid.vlan-id outer tpid.vlan-id;
family family {
   address address;
}
```

NOTE: If you configure the physical interface MTU for mixed tagging, then you must increase the MTU to 4 bytes more than the MTU value you would configure for a standard VLAN-tagged interface.

For example, if the MTU value is configured to be 1018 on a VLAN-tagged interface, then the MTU value on a flexible VLAN tagged interface must be 1022—4 bytes more. The additional 4 bytes accommodates the future addition of a stacked VLAN tag configuration on the same physical interface.

If the same physical interface MTU value is configured on both the VLAN and flexible VLAN-tag routers, the L2 circuit configuration does not come up and a MTU mismatch is logged. However, normal traffic flow is unaffected.

For encapsulation type **flexible-ethernet-services**, all VLAN IDs are valid.

4. For 1-, 4-, and 8-port Gigabit Ethernet IQ2 and IQ2-E PICs, for 1-port 10-Gigabit Ethernet IQ2 and IQ2-E PICs, for all MX Series router Gigabit Ethernet, Tri-Rate Ethernet copper, and 10-Gigabit Ethernet interfaces configured for 802.1Q flexible VLAN tagging, and for aggregated Ethernet interfaces on IQ2 and IQ2-E PICs or MX Series DPCs, you can configure mixed tagging support for untagged packets on a port. Untagged packets are accepted on the same mixed VLAN-tagged port. To accept untagged packets, include the **native-vlan-id** statement and the **flexible-vlan-tagging** statement at the **[edit interfaces *interface-name*]** hierarchy level:

   ```
   [edit interfaces ge-fpc/pic/port]
   flexible-vlan-tagging;
   native-vlan-id number;
   ```

   The logical interface on which untagged packets are to be received must be configured with the same native VLAN ID as that configured on the physical interface. To configure the logical interface, include the **vlan-id** statement (matching the **native-vlan-id** statement on the physical interface) at the **[edit interfaces *interface-name* unit *logical-unit-number*]** hierarchy level.

RELATED DOCUMENTATION

- 802.1Q VLANs Overview | **295**
- Configuring VLAN and Extended VLAN Encapsulation | **315**
- Stacking a VLAN Tag | **393**
- *Ethernet Interfaces User Guide for Routing Devices*
- Sending Untagged Traffic Without VLAN ID to Remote End | **302**

Configuring Tagged Interface with multiple tagged vlans and native vlan

You can configure the router to receive and forward single-tag frames, dual-tag frames, or a mixture of single-tag and dual-tag frames.

1. To configure the router to receive and forward single-tag frames with 802.1Q VLAN tags, include the **vlan-tagging** statement at the **[edit interfaces *interface-name*]** hierarchy level:

```
[edit interfaces interface-name]
user@host# vlan-tagging;
```

2. Configure the **flexible-vlan-tagging** statement at the **[edit interfaces ge-*fpc/pic/port*]** hierarchy level.

```
[edit interfaces ge-fpc/pic/port]
user@host#flexible-vlan-tagging;
```

For encapsulation type **flexible-ethernet-services**, all VLAN IDs are valid.

3. To accept untagged packets, include the **native-vlan-id** statement and the **flexible-vlan-tagging** statement at the **[edit interfaces *interface-name*]** hierarchy level:

```
[edit interfaces ge-fpc/pic/port]
flexible-vlan-tagging;
native-vlan-id number;
```

The range for **native-vlan-id** is 0 to 4094.

The logical interface on which untagged packets are to be received must be configured with the same native VLAN ID as that configured on the physical interface. To configure the logical interface, include the **vlan-id** statement (matching the **native-vlan-id** statement on the physical interface) at the **[edit interfaces *interface-name* unit *logical-unit-number*]** hierarchy level.

4. Configure the **vlan-id range** providing the range at the **[edit interfaces ge-*fpc/pic/port* unit *logical-unit-number*]** hierarchy level:

```
[edit interfaces ge-fpc/pic/port unit logical-unit-number]
user@host# vlan-id range number;
```

5. Configure the **vlan-tags** statement with **inner** and **outer** options or the **vlan-id range** statement at the **[edit interfaces ge-*fpc/pic/port* unit *logical-unit-number*]** hierarchy level:

```
[edit interfaces ge-fpc/pic/port unit logical-unit-number]
user@host# vlan-id range number;
user@host# vlan-tags outer tpid.vlan-id inner tpid.vlan-id;
```

The range for **inner** and **outer** option 32 to 4094.

To verify the configuration execute the **show** command.

```
user@host> show configuration
```

```
set interfaces ge-1/0/3 flexible-vlan-tagging
set interfaces ge-1/0/3 native-vlan-id 1010
set interfaces ge-1/0/3 unit 1 vlan-id-range 100-200
set interfaces ge-1/0/3 unit 2 vlan-tags outer 300
set interfaces ge-1/0/3 unit 2 vlan-tags inner 123
set interfaces ge-1/0/3 unit 3 vlan-tags outer 400
set interfaces ge-1/0/3 unit 3 vlan-tags inner 323
```

RELATED DOCUMENTATION

- 802.1Q VLANs Overview | **295**
- Configuring VLAN and Extended VLAN Encapsulation | **315**
- Stacking a VLAN Tag | **393**
- *Ethernet Interfaces User Guide for Routing Devices*
- Sending Untagged Traffic Without VLAN ID to Remote End | **302**

Sending Untagged Traffic Without VLAN ID to Remote End

Send traffic without the native VLAN ID (**native-vlan-id**) to the remote end of the network if untagged traffic is received.

If this option is not configured, then **native-vlan-id** is added to untagged traffic. But if this option is configured, then **native-vlan-id** is not added to untagged traffic.

NOTE:

- This feature works only on MX series routers with MPCs/MICs. Configuring this option with DPC results in no behavior change. But, if this option is configured with Aggregated Ethernet (AE) in which the sub interfaces reside across MPCs/MICs and DPC, MPCs/MICs and DPC will show a different behavior.
- In the egress direction, this feature is disrupted by VLAN normalization. Because of normalization, the egress interface cannot distinguish between untagged traffic and tagged traffic. And untagged traffic is sent out with **native-vlan-id**. Consider this while configuring both VLAN normalization and new **native-vlan-id** option.

  There will be a problem with ingress firewall filter if filter term includes **native-vlan-id**. With **no-native-vlan-insert** option configured, **native-vlan-id** will not be inserted to untagged traffic. So, firewall filter term will not match with untagged traffic. But if incoming traffic have VLAN ID which is equal to **native-vlan-id**, then firewall filter term will match and firewall will work.
- When this feature is used with AE, all sub-interfaces of AE should be in same type of FPC.

RELATED DOCUMENTATION

Configuring Interface Encapsulation on Physical Interfaces

802.1Q VLANs Overview | **295**

Configuring VPLS Interface Encapsulation

native-vlan-id | **1276**

no-native-vlan-insert | **1288**

Enabling VLAN Tagging | **298**

Configuring Flexible VLAN Tagging on PTX Series Packet Transport Routers

This topic describes how to configure flexible VLAN tagging on PTX Series Packet Transport Routers. In addition to VLAN tagging and stacked VLAN tagging, you can configure a port for flexible tagging. With flexible VLAN tagging, you can configure two logical interfaces on the same Ethernet port, one with single-tag framing and one with dual-tag framing.

To configure mixed tagging, include the **flexible-vlan-tagging** statement at the **[edit interfaces et-*fpc*/*pic*/*port*]** hierarchy level. You must also include the **vlan-tags** statement with **inner** and **outer** options or the **vlan-id** statement at the **[edit interfaces et-*fpc*/*pic*/*port* unit *logical-unit-number*]** hierarchy level:

```
[edit interfaces et-fpc/pic/port]
flexible-vlan-tagging;
unit logical-unit-number {
    vlan-id number;
}
unit logical-unit-number {
    vlan-tags inner tpid.vlan-id outer tpid.vlan-id;
}
```

RELATED DOCUMENTATION

Enabling VLAN Tagging | 298

Configuring an MPLS-Based VLAN CCC with Pop, Push, and Swap and Control Passthrough

For providing Layer 2 VPN services across your network, you might want to configure the ability to push, pop or swap 802.1Q tags on frames entering and leaving edge routers, allowing you to use a single VLAN-circuit cross-connect (CCC) [VLAN-CCC] logical interface to handle both dual-tag and single-tag packets. This feature thus provides interoperability between Layer 2 services with a distinct VLAN at the local or remote end or in instances where a Layer 2 service comes with a certain VLAN, but the remote peer has a different VLAN or no VLAN.

This feature includes the ability to enable passthrough of certain Ethertype/DMAC-matched frames over the Layer 2 circuit after successful VLAN tag operations on the VLAN CCC logical interface.

If you configure this feature, VLAN tags are applied when traffic is sent to and from the Layer 2 circuit interface. The pop, push, and swap operations are performed only on the outer tag. The pop VLAN tag removes the VLAN tag from the top of the VLAN tag stack. The push VLAN tag adds a new outer VLAN tag, and the swap VLAN tag replaces the existing outer VLAN tag with the new VLAN tag.

You can configure inet, inet6, or VLAN-CCC connections on a single Ethernet network interface or an aggregated Ethernet interface, enabling you to set different forwarding rules for tagged and untagged traffic on the same interface. For example, you can forward tagged packets over the Layer 2 circuit and route untagged traffic in native VLAN mode.

NOTE: Limitations for this feature on PTX routers are:

- VLAN operations on STP and CDP packets are not supported.
- You can't configure the VLAN-CCC logical interface with the native VLAN ID.
- LACP point-to-point connections between PE routers do not work if you configure **l2circuit-control-passthrough**. (Static LAG works, however.)

To configure a PE router with a VLAN CCC, an MPLS-based Layer 2 circuit, VLAN pop, push, and swap operations, and enabling passthrough of certain Ethertype/DMAC-matched frames:

NOTE: The following procedure uses actual interface names for the router's network interfaces instead of the variable *interface-name* so that you can quickly see their configuration differences. Remember that you can also configure the feature on aggregated Ethernet interfaces.

1. Configure OSPF on the loopback (or router address) and core interface:

   NOTE: The routing protocol can be OSPF or IS-IS.

   ```
   [edit]
   user@host# set protocols ospf area 0.0.0.0 interface lo0.0 passive
   user@host# set protocols ospf area 0.0.0.0 interface et-0/0/0:0
   ```

2. Enable traffic engineering for the routing protocol:

   ```
   [edit]
   user@host# set protocols ospf traffic-engineering
   ```

3. Configure an IP address for the loopback interface and for the core interface:

   ```
   [edit]
   user@host# set interfaces lo0 unit logical-unit-number family inet address address
   user@host# set interfaces et-0/0/0:0 unit 0 family inet address address
   ```

4. Configure the customer edge interface as a Layer 2 circuit from the local PE router to the other PE router:

   TIP: Use the router address of the other router as the neighbor address. It is the virtual circuit identifier together with the neighbor address that provides the unique address for the circuit.

   ```
   [edit]
   user@host# set protocols l2circuit neighbor address interface et-0/0/1:1.0 virtual-circuit-id identifier
   ```

5. Configure MPLS on the core interfaces:

   ```
   [edit]
   user@host# set protocols mpls interface all
   ```

6. Configure LDP on the loopback interface and the core interfaces:

   ```
   [edit]
   user@host# set ldp interface lo0.0
   user@host# set ldp interface et-0/0/0.0
   user@host# set ldp interface all
   ```

7. Configure **family mpls** on the logical unit of the core interface:

```
[edit]
user@host#  set interfaces et-0/0/0:0 unit 0 family mpls
```

> **NOTE:** You can enable **family mpls** on either individual interfaces, aggregated Ethernet interfaces, or tagged VLAN interfaces.

8. Specify the router ID:

```
[edit]
user@host#  set routing-options router-id address
```

9. Enable VLAN tagging on the customer edge interface of the local PE router:

```
[edit]
user@host#  set interfaces et-0/0/1:1 vlan-tagging
```

10. Configure the customer edge interface to use flexible Ethernet services encapsulation:

```
[edit]
user@host#  set interfaces et-0/0/1:1 encapsulation flexible-ethernet-services
```

11. Configure the logical unit of the customer edge interface with a VLAN ID:

```
[edit]
user@host#  set interfaces et-0/0/1:1 unit 0 vlan-id vlan-id
```

12. Configure the logical unit on the customer edge interface to use VLAN CCC encapsulation:

```
[edit]
user@host#  set interfaces et-0/0/1:1 unit 0 encapsulation vlan-ccc
```

13. Configure the logical unit on the customer edge interface to pop the tag off the input VLAN and then push the tag to the output VLAN:

```
[edit]
user@host#  set interfaces et-0/0/1:1 unit 0 input-vlan-map push
[edit]
user@host#  set interfaces et-0/0/1:1 unit 0 output-vlan-map pop
```

14. (Optional) Configure Layer 2 circuit traceoptions:

```
[edit]
user@host# set protocols l2circuit traceoptions file l2ckt.log
user@host# set protocols l2circuit traceoptions flag connections detail
```

15. (Optional) Configure the VLAN CCC logical interface so that encapsulation mismatches and MTU mismatches between this interface and the interface on the other PE router are ignored:

```
[edit]
user@host# set protocolsl2circuit neighbor address interface et-0/0/1:1.0
ignore-encapsulation-mismatch
user@host# set protocolsl2circuit neighbor address interface et-0/0/1:1.0 ignore-mtu-mismatch
```

16. To enable passthrough of certain Ethertype/DMAC-matched frames, configure Layer 2 circuit control passthrough:

```
[edit]
user@host# set forwarding-options l2circuit-control-passthrough
```

RELATED DOCUMENTATION

CCC Overview

Binding VLAN IDs to Logical Interfaces

This topic describes how to configure logical interfaces to receive and forward VLAN-tagged frames:

To configure a logical interface to receive and forward VLAN-tagged frames, you must bind a VLAN ID, a range of VLAN IDs, or a list of VLAN IDs to the logical interface. Table 63 on page 309 lists the configuration statements you use to bind VLAN IDs to logical interfaces, organized by scope of the VLAN IDs used to match incoming packets. You can configure these statements at the **[edit interfaces *interface-name* unit *logical-unit-number*]** hierarchy level or at the **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]** hierarchy level.

Table 63: Configuration Statements Used to Bind VLAN IDs to Logical Interfaces

Scope of VLAN ID Matching	Type of VLAN Framing Supported on the Logical Interface	
	Single-Tag Framing	Dual-Tag Framing
VLAN ID	**vlan-id *vlan-id*;**	**vlan-tags outer *tpid*.<*vlan-id*> inner *tpidvlan-id*;**
VLAN ID Range	**vlan-id-range *vlan-id*–*vlan-id*;**	**vlan-tags outer *tpid.vlan-id* inner-range *tpid.vlan-id*–*vlan-id*;**
VLAN ID List	**vlan-id-list [*vlan-id vlan-id*–*vlan-id*];**	**vlan-tags outer <*tpid.*>*vlan-id* inner-list [*vlan-id vlan-id*–*vlan-id*];**

NOTE: The **inner-list** option of the **vlan-tags** statement does not support Tag Protocol ID (TPID) values.

1. A logical interface that you have associated (bound) to a particular VLAN ID will receive and forward incoming frames that contain a matching VLAN ID. To bind a VLAN ID to a single-tag logical interface, include the **vlan-id** statement at the **[edit interfaces *interface-name* unit *logical-unit-number*]** hierarchy level or at the **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]** hierarchy level.

   ```
   [edit interfaces interface-name unit logical-unit-number]
   user@host# vlan-id vlan-id;
   ```

   To configure an Ethernet interface to support single-tag logical interfaces, include the **vlan-tagging** statement at the **[edit interfaces *ethernet-interface-name*]** hierarchy level. To support mixed tagging, include the **flexible-vlan-tagging** statement instead.

2. To bind a VLAN ID to a dual-tag logical interface, include the **vlan-tags** statement at the **[edit interfaces *ethernet-interface-name* unit *logical-unit-number*]** hierarchy level or at the **[edit logical-systems *logical-system-name* interfaces *ethernet-interface-name* unit *logical-unit-number*]** hierarchy level:

   ```
   [edit interfaces ethernet-interface-name unit logical-unit-number]
   vlan-tags inner <tpid.>vlan-id outer <tpid.>vlan-id;
   ```

   To configure an Ethernet interface to support dual-tag logical interfaces, include the **stacked-vlan-tagging** statement at the **[edit interfaces *ethernet-interface-name*]** hierarchy level. To support mixed tagging, include the **flexible-vlan-tagging** statement instead.

3. A VLAN range can be used by service providers to interconnect multiple VLANs belonging to a particular customer over multiple sites. Using a VLAN ID range conserves switch resources and simplifies configuration. To bind a range of VLAN IDs to a single-tag logical interface, include the **vlan-id-range** statement at the **[edit interfaces *ethernet-interface-name* unit *logical-unit-number*]** hierarchy level or at the **[edit logical-systems *logical-system-name* interfaces *ethernet-interface-name* unit *logical-unit-number*]** hierarchy level.

   ```
   [edit interfaces ethernet-interface-name unit logical-unit-number]
   vlan-id-range vlan-id-vlan-id;
   ```

4. To bind a range of VLAN IDs to a dual-tag logical interface, include the **vlan-tags** statement. Use the **inner–list** option to specify the VLAN IDs as an inclusive range by separating the starting VLAN ID and ending VLAN ID with a hyphen. You can include the statement at the **[edit interfaces *ethernet-interface-name* unit *logical-unit-number*]** hierarchy level or at the **[edit logical-systems *logical-system-name* interfaces *ethernet-interface-name* unit *logical-unit-number*]** hierarchy level.

   ```
   [edit interfaces ethernet-interface-name unit logical-unit-number]
   vlan-tags inner–list [ vlan-id vlan-id–vlan-id ] outer <tpid.>vlan-id;
   ```

   To configure an Ethernet interface to support dual-tag logical interfaces, include the **stacked-vlan-tagging** statement at the **[edit interfaces *ethernet-interface-name*]** hierarchy level. To support mixed tagging, include the **flexible-vlan-tagging** statement instead.

In Junos OS Release 9.5 and later, on MX Series routers and in Junos OS Release 12.2R2 and later on EX Series switches, you can bind a list of VLAN IDs to a single logical interface, eliminating the need to configure a separate logical interface for every VLAN or VLAN range. A logical interface that accepts packets tagged with any VLAN ID specified in a VLAN ID list is called a *VLAN-bundled* logical interface.

You can use VLAN-bundled logical interfaces to configure circuit cross-connects between Layer 2 VPN routing instances or Layer 2 circuits. Using VLAN-bundled logical interfaces simplifies configuration and reduces use of system resources such as logical interfaces, next hops, and circuits.

As an alternative to configuring multiple logical interfaces (one for each VLAN ID and one for each range of VLAN IDs), you can configure a single VLAN-bundled logical interface based on a list of VLAN IDs.

> **NOTE:** The **vlan-id** option is not supported to achieve VLAN normalization on VPLS instances that are configured with **vlan-id-list**. However, you can use the **vlan-maps** option to achieve VLAN normalization.

1. To bind a list of VLAN IDs to a single-tag logical interface, include the **vlan-id-list** statement at the **[edit interfaces *ethernet-interface-name* unit *logical-unit-number*]** hierarchy level or at the **[edit logical-systems *logical-system-name* interfaces *ethernet-interface-name* unit *logical-unit-number*]** hierarchy level. Specify the VLAN IDs in the list individually by using a space to separate each ID, as an inclusive list by separating the starting VLAN ID and ending VLAN ID with a hyphen, or as a combination of both.

   ```
   [edit interfaces ethernet-interface-name unit logical-unit-number]
   user@host# vlan-id-list [ vlan-id vlan-id–vlan-id ];
   ```

   To configure an Ethernet interface to support single-tag logical interfaces, include the **vlan-tagging** statement at the **[edit interfaces *ethernet-interface-name*]** hierarchy level. To support mixed tagging, include the **flexible-vlan-tagging** statement instead.

2. To bind a list of VLAN IDs to a dual-tag logical interface, include the **vlan-tags** statement at the **[edit interfaces *ethernet-interface-name* unit *logical-unit-number*]** hierarchy level or at the **[edit logical-systems *logical-system-name* interfaces *ethernet-interface-name* unit *logical-unit-number*]** hierarchy level. Use the **inner-list** option to specify the VLAN IDs individually by using a space to separate each ID, as an inclusive list by separating the starting VLAN ID and ending VLAN ID with a hyphen, or as a combination of both.

   ```
   [edit interfaces ethernet-interface-name unit logical-unit-number]
   user@host# vlan-tags inner-list [vlan-id vlan-id–vlan-id ] outer <tpid>vlan-id;
   ```

> **NOTE:** The **inner-list** option of the **vlan-tags** statement does not support Tag Protocol ID (TPID) values.

To configure an Ethernet interface to support dual-tag logical interfaces, include the **stacked-vlan-tagging** statement at the **[edit interfaces *ethernet-interface-name*]** hierarchy level. To support mixed tagging, include the **flexible-vlan-tagging** statement instead.

The following sample configuration configures two different lists of VLAN IDs on two different logical ports.

```
[edit interfaces]
ge-1/1/0 {
    vlan-tagging; # Only for single-tagging
    encapsulation flexible-ethernet-services;
    unit 10 {
        encapsulation vlan-ccc;
        vlan-id-list [20 30–40 45];
    }
}
ge-1/1/1 {
    flexible-vlan-tagging; # Only for mixed tagging
    encapsulation flexible-ethernet-services;
    unit 10 {
        encapsulation vlan-ccc;
        vlan-id-list [1 10 20 30–40];
    }
    unit 20 {
        encapsulation vlan-ccc;
        vlan-tags outer 200 inner-list [50–60 80 90–100];
    }
}
```

In the example configuration above, **ge-1/1/0** supports single-tag logical interfaces, and **ge-1/1/1** supports mixed tagging. The single-tag logical interfaces **ge-1/1/0.10** and **ge-1/1/1.20** each bundle lists of VLAN IDs. The dual-tag logical interface **ge-1/1/1.20** bundles lists of inner VLAN IDs.

TIP: You can group a range of identical interfaces into an interface range and then apply a common configuration to that interface range. For example, in the above example configuration, both interfaces ge-1/1/0 and ge-1/1/1 have the same physical encapsulation type of **flexible-ethernet-services**. Thus you can define an interface range with the interfaces ge-1/1/0 and ge-1/1/1 as its members and apply the encapsulation type flexible-ethernet-services to that defined interface range.

RELATED DOCUMENTATION

802.1Q VLANs Overview | **295**

Configuring Interface Ranges

Ethernet Interfaces User Guide for Routing Devices

Associating VLAN IDs to VLAN Demux Interfaces

IN THIS SECTION

The following sections describe how to configure VLAN demux interfaces to receive and forward VLAN-tagged frames:

Associating VLAN IDs to VLAN Demux Interfaces Overview

To configure a VLAN demux interface to receive and forward VLAN-tagged frames, you must associate a VLAN ID or dual tagged (stacked) VLAN ID to the interface. Table 64 on page 314 shows the configuration statements you use to associate VLAN IDs to VLAN demux interfaces, depending on the VLAN tag framing you use:

Table 64: Configuration Statements Used to Associate VLAN IDs to VLAN Demux Interfaces

	Single-Tag Framing	Dual-Tag Framing
Statement Format	vlan-id *vlan-id*;	vlan-tags outer *tpid*.<*vlan-id*> inner *tpidvlan-id*;

You can include all of the statements at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number*]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]**
- **[edit interfaces *demux0* unit *logical-unit-number*]**

Associating a VLAN ID to a VLAN Demux Interface

A VLAN demux interface that you have associated to a particular VLAN ID receives and forwards incoming frames that contain a matching VLAN ID. You can associate a VLAN ID to a single-tag logical interface or to a dual-tagged (stacked) logical interface.

1. Associating a VLAN ID to a Single-Tag VLAN Demux Interface | 314
2. Associating a VLAN ID to a Dual-Tag VLAN Demux Interface | 315

Associating a VLAN ID to a Single-Tag VLAN Demux Interface

To associate a VLAN ID to a single-tag VLAN demux interface, include the **vlan-id** statement at the **[edit interfaces *demux0* unit *logical-unit-number*]** hierarchy level:

```
vlan-id vlan-id;
```

To configure an interface to support single-tag logical interfaces, you must also include the **vlan-tagging** statement at the **[edit interfaces *interface-name*]** hierarchy level. To support mixed tagging, include the **flexible-vlan-tagging** statement instead.

SEE ALSO

Configuring a VLAN Demultiplexing Interface

Associating a VLAN ID to a Dual-Tag VLAN Demux Interface

To associate a VLAN ID to a dual-tag VLAN demux interface, include the **vlan-tags** statement at the **[edit interfaces *demux0* unit *logical-unit-number*]** hierarchy level:

```
vlan-tags inner <tpid.>vlan-id outer <tpid.>vlan-id;
```

To configure an interface to support dual-tag logical interfaces, include the **stacked-vlan-tagging** statement at the **[edit interfaces *interface-name*]** hierarchy level. To support mixed tagging, include the **flexible-vlan-tagging** statement instead.

SEE ALSO

802.1Q VLANs Overview | **295**

Configuring a VLAN Demultiplexing Interface

Ethernet Interfaces User Guide for Routing Devices

Configuring VLAN and Extended VLAN Encapsulation

To configure encapsulation on an interface, enter the **encapsulation** statement at the **[edit interfaces *interface-name*]** hierarchy level:

```
[edit interfaces interface-name]
user@host# encapsulation type
```

The following list contains important notes regarding VLAN encapsulation:

- Starting with Junos OS Release 8.1, , Gigabit Ethernet IQ, Gigabit Ethernet PICs with small form-factor pluggable optics (SFPs), and MX Series router Gigabit Ethernet, Tri-Rate Ethernet copper, and 10-Gigabit Ethernet interfaces with VLAN tagging enabled can use **flexible-ethernet-services**, **vlan-ccc** , or **vlan-vpls** encapsulation.
- Starting with Junos OS Release 9.5, aggregated Ethernet interfaces configured for VPLS can use **flexible-ethernet-services**, **vlan-ccc**, or **vlan-vpls**.
- Ethernet interfaces in VLAN mode can have multiple logical interfaces. In CCC and VPLS modes, VLAN IDs from 1 through 511 are reserved for normal VLANs, and VLAN IDs 512 through 4094 are reserved for CCC or VPLS VLANs. For 4-port Fast Ethernet interfaces, you can use VLAN IDs 512 through 1024 for CCC or VPLS VLANs. For encapsulation type **flexible-ethernet-services**, all VLAN IDs are valid.

- For flexible Ethernet services, Ethernet VLAN CCC and VLAN VPLS, you can also configure the encapsulation type that is used inside the VLAN circuit itself. To do this, include the **encapsulation** statement at the **[edit interfaces *interface-name* unit *logical-unit-number*]** hierarchy level or at the **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]** hierarchy level.
- You cannot configure a logical interface with VLAN CCC or VLAN VPLS encapsulation unless you also configure the physical device with the same encapsulation or with flexible Ethernet services encapsulation. In general, the logical interface must have a VLAN ID of 512 or higher; if the VLAN ID is 511 or lower, it will be subject to the normal destination filter lookups in addition to source address filtering. However if you configure flexible Ethernet services encapsulation, this VLAN ID restriction is removed.
- Gigabit Ethernet, 4-port Fast Ethernet, MX Series router Gigabit Ethernet, Tri-Rate Ethernet copper, 10-Gigabit Ethernet, and aggregated Ethernet interfaces with VLAN tagging enabled can use **extended-vlan-ccc** or **extended-vlan-vpls**, which allow 802.1Q tagging.
- For extended VLAN CCC and extended VLAN VPLS encapsulation, all VLAN IDs 1 and higher are valid. VLAN ID 0 is reserved for tagging the priority of frames.
- For extended VLAN CCC, the VLAN IDs on ingress and egress interfaces must be the same. For back-to-back connections, all VLAN IDs must be the same.

Release History Table

Release	Description
9.5	Starting with Junos OS Release 9.5, aggregated Ethernet interfaces configured for VPLS can use **flexible-ethernet-services**, **vlan-ccc**, or **vlan-vpls**.
8.1	Starting with Junos OS Release 8.1, , Gigabit Ethernet IQ, Gigabit Ethernet PICs with small form-factor pluggable optics (SFPs), and MX Series router Gigabit Ethernet, Tri-Rate Ethernet copper, and 10-Gigabit Ethernet interfaces with VLAN tagging enabled can use **flexible-ethernet-services**, **vlan-ccc** , or **vlan-vpls** encapsulation.

RELATED DOCUMENTATION

Configuring Interface Encapsulation on Physical Interfaces

802.1Q VLANs Overview | **295**

Configuring VPLS Interface Encapsulation

Ethernet Interfaces User Guide for Routing Devices

Configuring a Layer 2 VPN Routing Instance on a VLAN-Bundled Logical Interface

IN THIS SECTION

This topic describes how to configure a Layer 2 VPN routing instance on a logical interface bound to a list of VLAN IDs.

Configuring a VLAN-Bundled Logical Interface to Support a Layer 2 VPN Routing Instance

To configure a VLAN-bundled logical interface, specify the list of VLAN IDs by including the **vlan-id-list** statement or the **vlan-tags** statement on a provider edge (PE) router:

```
interfaces {
    ethernet-interface-name {
        vlan-tagging; # Support single- or dual-tag logical interfaces
        flexible-vlan-tagging; # Support mixed tagging
        encapsulation (extended-vlan-ccc | flexible-ethernet-services);
        unit logical-unit-number {
            vlan-id-list [vlan-id vlan-id–vlan-id]; # For single-tag
            vlan-tags outer <tpid.>vlan-id inner-list [vlan-id vlan-id–vlan-id]; # For dual-tag
        }
        . . .
    }
}
```

You can include the statements at the following hierarchy levels:

- **[edit]**
- **[edit logical-systems *logical-system-name*]**

SEE ALSO

802.1Q VLANs Overview | **295**

Ethernet Interfaces User Guide for Routing Devices

Specifying the Interface Over Which VPN Traffic Travels to the CE Router

To configure a Layer 2 VPN routing instance on a PE router, include the **instance-type** statement and specify the value **l2vpn**. To specify an interface connected to the router, include the **interface** statement and specify the VLAN-bundled logical interface:

```
instance-type l2vpn;
interface logical-interface-name;
```

You can include the statements at the following hierarchy levels:

- **[edit routing-instances *routing-instance-name*]**
- **[edit logical-systems *logical-system-name* routing-instances *routing-instance-name*]**

SEE ALSO

802.1Q VLANs Overview | **295**

Ethernet Interfaces User Guide for Routing Devices

Specifying the Interface to Handle Traffic for a CCC

To configure the VLAN-bundled logical interface as the interface to handle traffic for a circuit connected to the Layer 2 VPN routing instance, include the following statements:

```
protocols {
    l2vpn {
        (control-word | no-control-word);
        encapsulation-type (ethernet | ethernet-vlan);
        site site-name {
            site-identifier identifier;
            interface logical-interface-name { # VLAN-bundled logical interface
```

```
            . . . interface-options . . .
          }
        }
      }
    }
```

You can include the statements at the same hierarchy level at which you include the **instance-type l2vpn** and **interface *logical-interface-name*** statements:

- **[edit routing-instances *routing-instance-name*]**
- **[edit logical-systems *logical-system-name* routing-instances *routing-instance-name*]**

To enable a Layer 2 VPN routing instance on a PE router, include the **l2vpn** statement. For more information, see the *Junos OS VPNs Library for Routing Devices*.

The **encapsulation-type** statement specifies the Layer 2 protocol used for traffic from the customer edge (CE) router. If the Layer 2 VPN routing instance is being connected to a single-tag Layer 2 circuit, specify **ethernet** as the encapsulation type. If the Layer 2 VPN routing instance is being connected to a dual-tag Layer 2 circuit, specify **ethernet-vlan** as the encapsulation type.

To specify the interface to handle traffic for a circuit connected to the Layer 2 VPN routing instance, include the **interface** statement and specify the VLAN-bundled logical interface.

SEE ALSO

802.1Q VLANs Overview | **295**

Example: Configuring a Layer 2 VPN Routing Instance on a VLAN-Bundled Logical Interface | **320**

Ethernet Interfaces User Guide for Routing Devices

Example: Configuring a Layer 2 VPN Routing Instance on a VLAN-Bundled Logical Interface

The following configuration shows that the single-tag logical interface **ge-1/0/5.0** bundles a list of VLAN IDs, and the logical interface **ge-1/1/1.0** supports IPv4 traffic using IP address 10.30.1.130 and can participate in an MPLS path.

```
[edit interfaces]
ge-1/0/5 {
    vlan-tagging;
    encapsulation extended-vlan-ccc;
    unit 0 { # VLAN-bundled logical interface
        vlan-id-list [513 516 520-525];
    }
}
    ge-1/1/1 {
        unit 0 {
            family inet {
                address 10.30.1.1/30;
            }
            family mpls;
        }
    }
```

The following configuration shows the type of traffic supported on the Layer 2 VPN routing instance:

```
[edit protocols]
rsvp {
    interface all;
    interface lo0.0;
}
mpls {
    label-switched-path lsp {
        to 10.255.69.128;
    }
    interface all;
}
bgp {
    group g1 {
        type internal;
        local-address 10.255.69.96;
```

```
        family l2vpn {
            signaling;
        }
        neighbor 10.255.69.128;
    }
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface lo0.0;
        interface ge-1/1/1.0;
    }
}
```

The following configuration shows that the VLAN-bundled logical interface is the interface over which VPN traffic travels to the CE router and handles traffic for a CCC to which the VPN connects.

```
[edit routing-instances]
red {
    instance-type l2vpn;
    interface ge-1/0/5.0; # VLAN-bundled logical interface
    route-distinguisher 10.255.69.96:100;
    vrf-target target:1:1;
    protocols {
        l2vpn {
            encapsulation-type ethernet; # For single-tag VLAN logical interface
            site CE_ultima {
                site-identifier 1;
                interface ge-1/0/5.0;
            }
        }
    }
}
```

NOTE: Because the VLAN-bundled logical interface supports single-tag frames, Ethernet is the Layer 2 protocol used to encapsulate incoming traffic. Although the connection spans multiple VLANs, the VLANs are bundled and therefore can be encapsulated as a single VLAN.

However, with Ethernet encapsulation, the circuit signal processing does not check that the VLAN ID list is the same at both ends of the CCC connection.

RELATED DOCUMENTATION

802.1Q VLANs Overview | 295

Ethernet Interfaces User Guide for Routing Devices

Specifying the Interface Over Which VPN Traffic Travels to the CE Router

To configure a Layer 2 VPN routing instance on a PE router, include the **instance-type** statement and specify the value **l2vpn**. To specify an interface connected to the router, include the **interface** statement and specify the VLAN-bundled logical interface:

```
instance-type l2vpn;
interface logical-interface-name;
```

You can include the statements at the following hierarchy levels:

- **[edit routing-instances *routing-instance-name*]**
- **[edit logical-systems *logical-system-name* routing-instances *routing-instance-name*]**

RELATED DOCUMENTATION

802.1Q VLANs Overview | 295

Ethernet Interfaces User Guide for Routing Devices

Configuring Access Mode on a Logical Interface

Enterprise network administrators can configure a single logical interface to accept untagged packets and forward the packets within a specified bridge domain. A logical interface configured to accept untagged packets is called an *access interface* or *access port*. Access interface configuration is supported on MX Series routers only.

To configure access mode on a logical interface, use the **interface-mode access** statement at the **[edit interfaces *interface-name* unit *logical-unit-number* family bridge]** hierarchy level or at the **[edit**

logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family bridge] hierarchy level.

When an untagged packet is received on an access interface, the packet is accepted, the configured VLAN ID is added to the packet, and the packet is forwarded within the bridge domain that is configured with the matching VLAN ID.

The following example configures a logical interface as an access port with a VLAN ID of 20:

```
[edit interfaces ge-1/2/0]
unit 0 {
    family bridge {
        interface-mode access;
        vlan-id 20;
    }
}
```

RELATED DOCUMENTATION

802.1Q VLANs Overview | **295**

Ethernet Interfaces User Guide for Routing Devices

Configuring a Logical Interface for Trunk Mode

As an alternative to configuring a logical interface for each VLAN, enterprise network administrators can configure a single logical interface to accept untagged packets or packets tagged with any VLAN ID specified in a list of VLAN IDs. Using a VLAN ID list conserves switch resources and simplifies configuration. A logical interface configured to accept packets tagged with any VLAN ID specified in a list is called a *trunk interface* or *trunk port*. Trunk interface configuration is supported on MX Series routers only. Trunk interfaces support integrated routing and bridging (IRB).

To configure a logical interface to accept any packet tagged with a VLAN ID that matches the list of VLAN IDs, include the **interface-mode** statement and specify the **trunk** option:

```
interface-mode trunk;
```

You can include this statement at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number* family bridge]**

- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family bridge]**

RELATED DOCUMENTATION

> 802.1Q VLANs Overview | **295**
>
> *Ethernet Interfaces User Guide for Routing Devices*

Configuring the VLAN ID List for a Trunk Interface

To configure the list of VLAN IDs to be accepted by the trunk port, include the **vlan-id-list** statement and specify the list of VLAN IDs. You can specify individual VLAN IDs with a space separating the ID numbers, specify a range of VLAN IDs with a dash separating the ID numbers, or specify a combination of individual VLAN IDs and a range of VLAN IDs.

```
vlan-id-list [number number-number];
```

You can include this statement at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number* family bridge interface-mode trunk]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family bridge interface-mode trunk]**

When a packet is received that is tagged with a VLAN ID specified in the trunk interface list of VLAN IDs, the packet is accepted and forwarded within the bridge domain that is configured with the matching VLAN ID.

When a packet is received that is tagged with a VLAN ID not specified in the trunk interface list of VLAN IDs, the native VLAN ID is pushed in front of the existing VLAN tag or tags and the packet is forwarded within the bridge domain that is configured with the matching VLAN ID.

When an untagged packet is received on a trunk interface, the native VLAN ID is added to the packet and the packet is forwarded within the bridge domain that is configured with the matching VLAN ID.

A bridge domain configured with a matching VLAN ID must be configured before the trunk interface is configured. To learn more about configuring bridge domains, see the *Junos Routing Protocols Configuration Guide*.

RELATED DOCUMENTATION

Configuring a Trunk Interface on a Bridge Network

On MX Series routers, you can configure a trunk interface on a bridge network.

The following output sample shows trunk port configuration on a bridge network:

`user@host#` **run show interfaces**

```
ge-0/0/0 {
    flexible-vlan-tagging;
    encapsulation flexible-ethernet-services;
    unit 0 {
        encapsulation vlan-bridge;
        vlan-id 1;
    }
}
ge-2/0/0 {
    unit 0 {
        family bridge {
            interface-mode trunk;
            vlan-id-list 1-200;
        }
    }
}
ge-2/0/1 {
    flexible-vlan-tagging;
    encapsulation flexible-ethernet-services;
    unit 0 {
        encapsulation vlan-bridge;
        vlan-id 1;
    }
}
```

If you want **igmp-snooping** to be functional for a bridge domain, then you should not configure **interface-mode** and **irb** for that bridge domain. Such a configuration commit succeeds, but IGMP snooping

is not functional, and a message informing the same is displayed as shown after the sample configuration below:

```
user@host# run show configuration
```

```
interfaces {
    ge-5/1/1 {
        flexible-vlan-tagging;
        native-vlan-id 1;
        unit 0 {
            family bridge {
                interface-mode trunk;
                vlan-id-list 401;
            }
        }
    }
    irb {
        unit 401 {
            family inet {
                address 192.168.2.2/27;
            }
        }
    }
}
protocols {
    igmp {
        interface all;
    }
}
bridge-domains {
    VLAN-401 {
        vlan-id 401;
        routing-interface irb.401;
        protocols {
            igmp-snooping;
        }
    }
}
```

```
user@host# commit
```

```
[edit bridge-domains]
  'VLAN-401'
```

```
    IGMP Snooping not supported with IRB and trunk mode interface ge-5/1/1.0
commit complete
```

To achieve IGMP snooping for a bridge domain, you should use such a configuration as shown in the following example:

user@host# **run show configuration**

```
interfaces {
    ge-0/0/1 {
        flexible-vlan-tagging;
        native-vlan-id 1;
        encapsulation flexible-ethernet-services;
        unit 0 {
            encapsulation vlan-bridge;
            vlan-id 401;
        }
    }
    irb {
        unit 401 {
            family inet {
                address 192.168.2.2/27;
            }
        }
    }
}
protocols {
    igmp {
        interface all;
    }
}
bridge-domains {
    VLAN-401 {
        vlan-id 401;
        interface ge-0/0/1.0;
        routing-interface irb.401;
        protocols {
            igmp-snooping;
        }
    }
}
```

user@host# **commit**

```
commit complete
```

RELATED DOCUMENTATION

- 802.1Q VLANs Overview | **295**
- interface-mode | **1215**
- *Ethernet Interfaces User Guide for Routing Devices*

Configuring a VLAN-Bundled Logical Interface to Support a Layer 2 VPN Routing Instance

To configure a VLAN-bundled logical interface, specify the list of VLAN IDs by including the **vlan-id-list** statement or the **vlan-tags** statement on a provider edge (PE) router:

```
interfaces {
    ethernet-interface-name {
        vlan-tagging; # Support single- or dual-tag logical interfaces
        flexible-vlan-tagging; # Support mixed tagging
        encapsulation (extended-vlan-ccc | flexible-ethernet-services);
        unit logical-unit-number {
            vlan-id-list [vlan-id vlan-id–vlan-id]; # For single-tag
            vlan-tags outer <tpid.>vlan-id inner-list [vlan-id vlan-id–vlan-id]; # For dual-tag
        }
        . . .
    }
}
```

You can include the statements at the following hierarchy levels:

- **[edit]**
- **[edit logical-systems *logical-system-name*]**

RELATED DOCUMENTATION

Configuring a VLAN-Bundled Logical Interface to Support a Layer 2 VPN Routing Instance

To configure a VLAN-bundled logical interface, specify the list of VLAN IDs by including the **vlan-id-list** statement or the **vlan-tags** statement:

```
interfaces {
    ethernet-interface-name {
        vlan-tagging; # Support single- or dual-tag logical interfaces
        flexible-vlan-tagging; # Support mixed tagging
        encapsulation (extended-vlan-ccc | flexible-ethernet-services);
        unit logical-unit-number {
            encapsulation vlan–ccc; # Required for single-tag
            vlan-id-list [vlan-id vlan-id–vlan-id]; # For single-tag
            vlan-tags outer tpid.vlan-id inner-list [vlan-id vlan-id–vlan-id]; # For dual-tag
        }
        . . .
    }
}
```

You can include the statements at the following hierarchy levels:

- **[edit]**
- **[edit logical-systems *logical-system-name*]**

For a single-tag logical interface, include the **encapsulation** statement and specify **vlan-ccc** so that CCC circuit encapsulation is used inside the Layer 2 circuit.

NOTE: In the case of a dual-tag logical interface, the Junos OS automatically uses the **vlan-ccc** encapsulation type.

RELATED DOCUMENTATION

802.1Q VLANs Overview | 295
Specifying the Interface to Handle Traffic for a CCC Connected to the Layer 2 Circuit | 331
Ethernet Interfaces User Guide for Routing Devices

Configuring a Layer 2 Circuit on a VLAN-Bundled Logical Interface

IN THIS SECTION

This topic describes how to configure a Layer 2 circuit on a logical interface bound to a list of VLAN IDs.

Configuring a VLAN-Bundled Logical Interface to Support a Layer 2 VPN Routing Instance

To configure a VLAN-bundled logical interface, specify the list of VLAN IDs by including the **vlan-id-list** statement or the **vlan-tags** statement:

```
interfaces {
   ethernet-interface-name {
      vlan-tagging; # Support single- or dual-tag logical interfaces
      flexible-vlan-tagging; # Support mixed tagging
      encapsulation (extended-vlan-ccc | flexible-ethernet-services);
      unit logical-unit-number {
         encapsulation vlan–ccc; # Required for single-tag
         vlan-id-list [vlan-id vlan-id–vlan-id]; # For single-tag
         vlan-tags outer tpid.vlan-id inner-list [vlan-id vlan-id–vlan-id]; # For dual-tag
      }
      . . .
   }
```

```
}
```

You can include the statements at the following hierarchy levels:

- **[edit]**
- **[edit logical-systems *logical-system-name*]**

For a single-tag logical interface, include the **encapsulation** statement and specify **vlan-ccc** so that CCC circuit encapsulation is used inside the Layer 2 circuit.

> **NOTE:** In the case of a dual-tag logical interface, the Junos OS automatically uses the **vlan-ccc** encapsulation type.

SEE ALSO

802.1Q VLANs Overview | **295**

Specifying the Interface to Handle Traffic for a CCC Connected to the Layer 2 Circuit | **331**

Ethernet Interfaces User Guide for Routing Devices

Specifying the Interface to Handle Traffic for a CCC Connected to the Layer 2 Circuit

To configure the VLAN-bundled logical interface as the interface to handle traffic for a circuit connected to the Layer 2 circuit, include the following statements:

```
l2circuit {
    neighbor address {
        interface logical-interface-name {
            virtual-circuit-id number;
            no-control-word;
        }
    }
}
```

You can include the statements at the following hierarchy levels:

- **[edit protocols]**
- **[edit logical-systems *logical-system-name* protocols]**

To enable a Layer 2 circuit, include the **l2circuit** statement.

To configure the router as a neighbor for a Layer 2 circuit, specify the neighbor address using the **neighbor** statement.

To specify the interface to handle traffic for a circuit connected to the Layer 2 circuit, include the **interface** statement and specify the VLAN-bundled logical interface.

SEE ALSO

802.1Q VLANs Overview | **295**

Example: Configuring a Layer 2 Circuit on a VLAN-Bundled Logical Interface | **332**

Configuring a VLAN-Bundled Logical Interface to Support a Layer 2 VPN Routing Instance | **329**

Ethernet Interfaces User Guide for Routing Devices

Example: Configuring a Layer 2 Circuit on a VLAN-Bundled Logical Interface

The following configuration shows that the single-tag logical interface **ge-1/0/5.0** bundles a list of VLAN IDs, and the logical interface **ge-1/1/1.0** supports IPv4 traffic using IP address 10.30.1.1/30 and can participate in an MPLS path.

```
[edit interfaces]
ge-1/0/5 {
    vlan-tagging;
    encapsulation extended-vlan-ccc;
    unit 0 { # VLAN-bundled logical interface
        vlan-id-list [513 516 520-525];
    }
}
ge-1/1/1 {
    unit 0 {
        family inet {
```

```
            address 10.30.1.1/30;
        }
        family mpls;
    }
}
```

The following configuration shows the type of traffic supported on the Layer 2 VPN routing instance, and shows that the VLAN-bundled logical interface handles traffic for a CCC to which the Layer 2 circuit connects:

```
[edit protocols]
rsvp {
    interface all;
    interface lo0.0;
}
mpls {
    label-switched-path lsp {
        to 10.255.69.128;
    }
    interface all;
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface lo0.0;
        interface ge-1/1/1.0;
    }
}
ldp {
    interface ge-1/1/1.0;
    interface ge-1/0/5.0; # VLAN-bundled logical interface
    interface lo0.0;
}
l2circuit {
    neighbor 10.255.69.128 {
        interface ge-1/0/5.0 { # VLAN-bundled logical interface
            virtual-circuit-id 3;
            no-control-word;
        }
    }
}
```

RELATED DOCUMENTATION

802.1Q VLANs Overview | 295

Ethernet Interfaces User Guide for Routing Devices

Guidelines for Configuring VLAN ID List-Bundled Logical Interfaces That Connect CCCs

IN THIS SECTION

For MX Series routers, you can bind a list of VLAN IDs to a logical interface, configure a Layer 2 VPN routing instance or Layer 2 circuit on the logical interface, and then use the logical interface to configure a circuit cross-connect (CCC) to another Layer 2 VPN routing instance or Layer 2 circuit.

A CCC allows you to configure transparent connections between two circuits so that packets from the source circuit are delivered to the destination circuit with, at most, the Layer 2 address being changed. You configure a CCC by connecting circuit interfaces of the same type. For more information, see *Circuit and Translational Cross-Connects Overview*.

NOTE: The Junos OS supports binding of Ethernet logical interfaces to lists of VLAN IDs on MX Series routers only. For all other routers, you can bind an Ethernet logical interface to only a single VLAN ID or to a single range of VLAN IDs.

The following configuration guidelines apply to bundling lists of VLAN IDs to Ethernet logical interfaces used to configure CCCs:

Guidelines for Configuring Physical Link-Layer Encapsulation to Support CCCs

To enable a physical interface to support VLAN-bundled logical interfaces that you will use to configure a CCC, you must specify one of the following physical link-layer encapsulation types as the value of the **encapsulation** statement at the **[edit interfaces *interface-name*]** hierarchy level:

```
[edit interfaces interface-name]
encapsulation (extended-vlan-ccc | flexible-ethernet-services);
```

- **extended-vlan-ccc**—For Ethernet interfaces with standard TPID tagging.
- **flexible-ethernet-services**—For supported Gigabit Ethernet interfaces for which you want to configure multiple per-unit Ethernet encapsulations.

For more information about configuring the encapsulation on a physical interface, see *Configuring Interface Encapsulation on Physical Interfaces*.

Guidelines for Configuring Logical Link-Layer Encapsulation to Support CCCs

For VLAN-bundled logical interfaces that you use to configure a CCC, specific logical link-layer encapsulation types are used inside the circuits themselves.

Table 65 on page 335 describes the logical link-layer encapsulation types used within circuits connected using VLAN-bundled logical interfaces of the same type.

Table 65: Encapsulation Inside Circuits CCC-Connected by VLAN-Bundled Logical Interfaces

Encapsulation Inside the Circuit	Layer 2 Circuit Joined by Configuring an Interface-to-Interface CCC Connection	
	Layer 2 VPN Routing Instance	Layer 2 Circuit
Syntax	encapsulation-type (ethernet | ethernet-vlan);	encapsulation vlan-ccc;
Hierarchy Level	[edit routing-instances *routing-instance-name* protocols l2vpn], [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn]	[edit interfaces *ethernet-interface-name* unit *logical-unit-number*], [edit logical-systems *logical-system-name* interfaces *ethernet-interface-name* unit *logical-unit-number*]

Table 65: Encapsulation Inside Circuits CCC-Connected by VLAN-Bundled Logical Interfaces *(continued)*

Encapsulation Inside the Circuit	Layer 2 Circuit Joined by Configuring an Interface-to-Interface CCC Connection	
	Layer 2 VPN Routing Instance	**Layer 2 Circuit**
Usage Guidelines	See the *Junos OS VPNs Library for Routing Devices*.	See *Configuring Interface Encapsulation on Logical Interfaces*, *Circuit and Translational Cross-Connects Overview*, and *Defining the Encapsulation for Switching Cross-Connects*.
For a Single-Tag Logical Interface	The MX Series router automatically uses **ethernet** as the Layer 2 protocol used to encapsulate incoming traffic. Although the connection spans multiple VLANs, the VLANs are bundled and therefore can be encapsulated as a single VLAN. **NOTE:** With **ethernet** encapsulation, the circuit signal processing does not check that the VLAN ID list is the same at both ends of the CCC connection.	Configure the MX Series router to use **vlan-ccc** as the logical link-layer encapsulation type.
For a Dual-Tag Logical Interface	Configure the MX Series router to use **ethernet-vlan** as the Layer 2 protocol to encapsulate incoming traffic. With **ethernet-vlan** encapsulation, circuit signal processing checks that the VLAN ID list is the same at both ends of the CCC connection. If a VLAN ID list mismatch is detected, you can view the error condition in the **show interfaces** command output.	The MX Series router automatically uses **vlan-ccc** as the logical link-layer encapsulation type, regardless of the value configured.

RELATED DOCUMENTATION

802.1Q VLANs Overview | **295**

Binding VLAN IDs to Logical Interfaces | **309**

Defining the Encapsulation for Switching Cross-Connects

Specifying the Interface to Handle Traffic for a CCC

To configure the VLAN-bundled logical interface as the interface to handle traffic for a circuit connected to the Layer 2 VPN routing instance, include the following statements:

```
protocols {
    l2vpn {
        (control-word | no-control-word);
        encapsulation-type (ethernet | ethernet-vlan);
        site site-name {
            site-identifier identifier;
            interface logical-interface-name { # VLAN-bundled logical interface
                . . . interface-options . . .
            }
        }
    }
}
```

You can include the statements at the same hierarchy level at which you include the **instance-type l2vpn** and **interface *logical-interface-name*** statements:

- **[edit routing-instances *routing-instance-name*]**
- **[edit logical-systems *logical-system-name* routing-instances *routing-instance-name*]**

To enable a Layer 2 VPN routing instance on a PE router, include the **l2vpn** statement. For more information, see the *Junos OS VPNs Library for Routing Devices*.

The **encapsulation-type** statement specifies the Layer 2 protocol used for traffic from the customer edge (CE) router. If the Layer 2 VPN routing instance is being connected to a single-tag Layer 2 circuit, specify **ethernet** as the encapsulation type. If the Layer 2 VPN routing instance is being connected to a dual-tag Layer 2 circuit, specify **ethernet-vlan** as the encapsulation type.

To specify the interface to handle traffic for a circuit connected to the Layer 2 VPN routing instance, include the **interface** statement and specify the VLAN-bundled logical interface.

RELATED DOCUMENTATION

802.1Q VLANs Overview | **295**

Example: Configuring a Layer 2 VPN Routing Instance on a VLAN-Bundled Logical Interface | **320**

Specifying the Interface to Handle Traffic for a CCC Connected to the Layer 2 Circuit

To configure the VLAN-bundled logical interface as the interface to handle traffic for a circuit connected to the Layer 2 circuit, include the following statements:

```
l2circuit {
    neighbor address {
        interface logical-interface-name {
            virtual-circuit-id number;
            no-control-word;
        }
    }
}
```

You can include the statements at the following hierarchy levels:

- **[edit protocols]**
- **[edit logical-systems *logical-system-name* protocols]**

To enable a Layer 2 circuit, include the **l2circuit** statement.

To configure the router as a neighbor for a Layer 2 circuit, specify the neighbor address using the **neighbor** statement.

To specify the interface to handle traffic for a circuit connected to the Layer 2 circuit, include the **interface** statement and specify the VLAN-bundled logical interface.

RELATED DOCUMENTATION

802.1Q VLANs Overview | **295**

Example: Configuring a Layer 2 Circuit on a VLAN-Bundled Logical Interface | **332**

Configuring a VLAN-Bundled Logical Interface to Support a Layer 2 VPN Routing Instance | **329**

11

CHAPTER

Configuring Static ARP Table Entries

Static ARP Table Entries Overview

For Fast Ethernet, Gigabit Ethernet, Tri-Rate Ethernet copper, and 10-Gigabit Ethernet interfaces, you can configure static ARP table entries, defining mappings between IP and MAC addresses.

RELATED DOCUMENTATION

Ethernet Interfaces User Guide for Routing Devices

Configuring Static ARP Table Entries For Mapping IP Addresses to MAC Addresses

By default, the device responds to an Address Resolution Protocol (ARP) request only if the destination address of the ARP request is on the local network of the incoming interface. For Fast Ethernet or Gigabit Ethernet interfaces, you can configure static ARP entries that associate the IP addresses of nodes on the same Ethernet subnet with their media access control (MAC) addresses. These static ARP entries enable the device to respond to ARP requests even if the destination address of the ARP request is not local to the incoming Ethernet interface.

Also, unlike dynamically learned ARP entries, static ARP entries do not age out. You can also configure static ARP entries in a troubleshooting situation or if your device is unable to learn a MAC address dynamically.

NOTE: By default, an ARP policer is installed that is shared among all the Ethernet interfaces on which you have configured the **family inet** statement. By including the **arp** statement at the **[edit interfaces *interface-name* unit *logical-unit-number* family inet policer]** hierarchy level, you can apply a specific ARP-packet policer to an interface. This feature is not available on EX Series switches.

To configure static ARP entries:

1. In the configuration mode, at the [**edit**] hierarchy level, configure the router interface on which the ARP table entries for the router is configured.

   ```
   [edit]
   ```

```
user@host# edit interfaces interface-name
```

2. Configure the protocol family, the logical unit of the interface, and the interface address of the router interface at the [**edit interfaces *interface-name***] hierarchy level. While configuring the protocol family, specify **inet** as the protocol family.

> **NOTE:** When you need to conserve IP addresses, you can configure an Ethernet interface to be unnumbered by including the **unnumbered-address** statement at the **[edit interfaces *interface-name* unit *logical-unit-number* family inet]** hierarchy level.

```
[edit interfaces interface-name]
user@host# edit unit logical-unit-number family inet address interface-address
```

3. Configure a static ARP entry by specifying the IP address and the MAC address that are to be mapped to each other. The IP address specified must be part of the subnet defined in the enclosing **address** statement. The MAC address must be specified as hexadecimal bytes in the following formats: ***nnnn.nnnn.nnnn*** or ***nn*:*nn*:*nn*:*nn*:*nn*:*nn***format. For instance, you can use either **0011.2233.4455** or **00:11:22:33:44:55**.

```
[edit interfaces interface-name unit logical-unit-number family inet address interface-address
user@host# set arp ip-address mac mac-address
```

4. Configure another static ARP entry by specifying the IP address and the MAC address that are to be mapped to each other. You can also associate a multicast MAC address with a unicast IP address by including the **multicast-mac** option with the **arp** statement. You can optionally configure the router to respond to ARP requests for the specified IP address by using the **publish** option with the **arp** statement.

> **NOTE:** For unicast MAC addresses only, if you include the **publish** option, the router or switch replies to proxy ARP requests.

```
[edit interfaces interface-name unit logical-unit-number family inet address interface-address
user@host# set arp ip-address multicast-mac mac-address publish
```

NOTE: The Junos OS supports the IPv6 static neighbor discovery cache entries, similar to the static ARP entries in IPv4.

RELATED DOCUMENTATION

- *arp*
- *Management Ethernet Interface Overview*
- *Applying Policers*
- *Configuring an Unnumbered Interface*
- *Ethernet Interfaces User Guide for Routing Devices*

12

CHAPTER

Configuring Restricted and Unrestricted Proxy ARP

Restricted and Unrestricted Proxy ARP Overview

By default, the Junos OS responds to an Address Resolution Protocol (ARP) request only if the destination address of the ARP request is local to the incoming interface.

For Ethernet Interfaces, you can configure the router or switches to proxy-reply to the ARP requests using the restricted or unrestricted proxy ARP configuration.

You might want to configure restricted or unrestricted proxy ARP for routers that act as provider edge (PE) devices in Ethernet Layer 2 LAN switching domains.

> **NOTE:** From Junos OS Release 10.0 onward, Junos OS does not respond to proxy ARP requests with the default route 0.0.0.0. This behavior is in compliance with RFC 1027.

Restricted Proxy ARP

Restricted proxy ARP enables the router or switch to respond to the ARP requests in which the physical networks of the source and target are not the same and the router or switch has an active route to the target address in the ARP request. The router does not reply if the target address is on the same subnet and the same interface as the ARP requestor.

Unrestricted Proxy ARP

Unrestricted proxy ARP enables the router or switch to respond to any ARP request, on condition that the router has an active route to the destination address of the ARP request. The route is not limited to the incoming interface of the request, nor is it required to be a direct route.

WARNING: If you configure unrestricted proxy ARP, the proxy router replies to ARP requests for the target IP address on the same interface as the incoming ARP request. This behavior is appropriate for cable modem termination system (CMTS) environments, but might cause Layer 2 reachability problems if you enable unrestricted proxy ARP in other environments.

When an IP client broadcasts the ARP request across the Ethernet wire, the end node with the correct IP address responds to the ARP request and provides the correct MAC address. If the unrestricted proxy ARP feature is enabled, the router response is redundant and might fool the IP client into determining that the destination MAC address within its own subnet is the same as the address of the router.

> **NOTE:** While the destination address can be remote, the source address of the ARP request must be on the same subnet as the interface upon which the ARP request is received. For security reasons, this rule applies to both unrestricted and restricted proxy ARP.

Topology Considerations for Unrestricted Proxy ARP

In most situations, you should not configure the router or switch to perform unrestricted proxy ARP. Do so only for special situations, such as when cable modems are used. Figure 2 on page 345 and Figure 3 on page 346 show examples of situations in which you might want to configure unrestricted proxy ARP.

In Figure 2 on page 345, the edge device is not running any IP protocols. In this case, you configure the core router to perform unrestricted proxy ARP. The edge device is the client of the proxy.

In Figure 3 on page 346, the Broadband Remote Access Server (B-RAS) routers are not running any IP protocols. In this case, you configure unrestricted proxy ARP on the B-RAS interfaces. This allows the core device to behave as though it is directly connected to the end users.

Figure 2: Edge Device Case for Unrestricted Proxy ARP

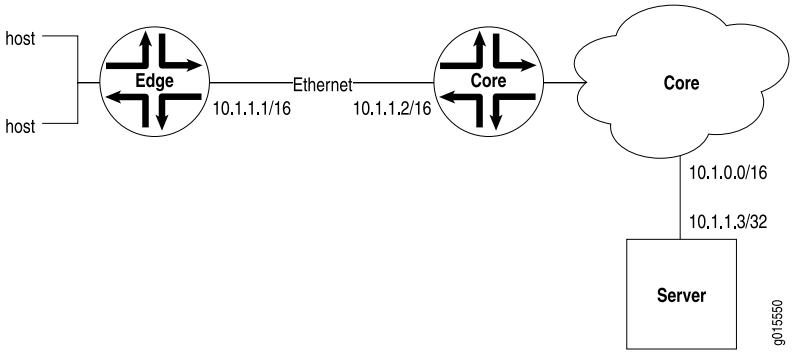
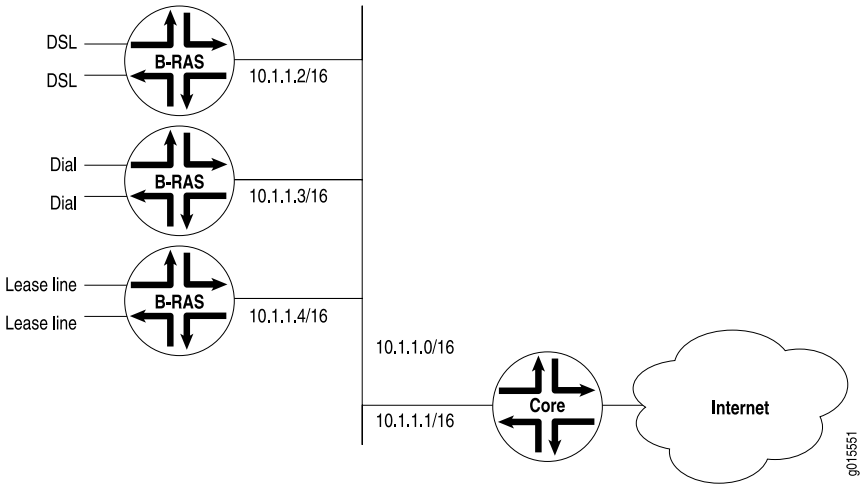


Figure 3: Core Device Case for Unrestricted Proxy ARP



RELATED DOCUMENTATION

Ethernet Interfaces User Guide for Routing Devices

Configuring Restricted and Unrestricted Proxy ARP

To configure restricted or unrestricted proxy ARP, include the **proxy-arp** statement:

```
proxy-arp (restricted |unrestricted);
```

You can include this statement at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number*]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]**

To return to the default—that is, to disable restricted or unrestricted proxy ARP—delete the **proxy-arp** statement from the configuration:

```
[edit]
```

```
user@host# delete interfaces interface-name unit logical-unit-number proxy-arp
```

You can track the number of restricted or unrestricted proxy ARP requests processed by the router or switch by issuing the **show system statistics arp** operational mode command.

> **NOTE:** When proxy ARP is enabled as default or unrestricted, the router or switch responds to any ARP request as long as the device has an active route to the target address of the ARP request. This gratuitous ARP behavior can result in an error when the receiving interface and target response interface are the same and the end device (for example, a client) performs a duplicate address check. To prevent this error, configure the router or switch interface with the **no-gratuitous-arp-reply** statement. See "Configuring Gratuitous ARP" on page 349 for information about how to disable responses to gratuitous ARP requests.

RELATED DOCUMENTATION

- *Ethernet Interfaces User Guide for Routing Devices*

13
CHAPTER

Configuring Gratuitous ARP

Configuring Gratuitous ARP

Gratuitous Address Resolution Protocol (ARP) requests help detect duplicate IP addresses. A gratuitous ARP is a broadcast request for a router's own IP address. If a router or switch sends an ARP request for its own IP address and no ARP replies are received, the router- or switch-assigned IP address is not being used by other nodes. However, if a router or switch sends an ARP request for its own IP address and an ARP reply is received, the router- or switch-assigned IP address is already being used by another node.

Gratuitous ARP replies are reply packets sent to the broadcast MAC address with the target IP address set to be the same as the sender's IP address. When the router or switch receives a gratuitous ARP reply, the router or switch can insert an entry for that reply in the ARP cache. By default, updating the ARP cache on gratuitous ARP replies is disabled on the router or switch.

To enable updating of the ARP cache for gratuitous ARPs:

1. In configuration mode, go to the **[edit interfaces interface-name]** hierarchy level.

   ```
   [edit]
   user@host# edit interfaces interface-name
   ```

2. Include the **gratuitous-arp-reply** statement.

   ```
   [edit interfaces interface-name]
   user@host# set gratuitous-arp-reply
   ```

To restore the default behavior, that is, to disable updating of the ARP cache for gratuitous ARP, delete the **gratuitous-arp-reply** statement from the configuration:

```
[edit interfaces interface-name]
user@host# delete gratuitous-arp-reply;
```

By default, the router or switch responds to gratuitous ARP requests. However, on Ethernet interfaces, you can disable responses to gratuitous ARP requests.

To disable responses to gratuitous ARP requests:

1. In configuration mode, go to the **[edit interfaces interface-name]** hierarchy level.

   ```
   [edit]
   user@host# edit interfaces interface-name
   ```

2. Include the **no-gratuitous-arp-request** statement.

```
[edit interfaces interface-name]
user@host# set no-gratuitous-arp-request
```

To return to the default—that is, to respond to gratuitous ARP requests—delete the **no-gratuitous-arp-request** statement from the configuration:

```
[edit interfaces interface-name]
user@host# delete no-gratuitous-arp-request
```

RELATED DOCUMENTATION

- gratuitous-arp-reply | **1186**
- no-gratuitous-arp-request | **1282**
- *Ethernet Interfaces Overview*
- *Ethernet Interfaces User Guide for Routing Devices*

14

CHAPTER

Adjusting the ARP Aging Timer

Adjusting the ARP Aging Timer

By default, the ARP aging timer is set at 20 minutes. In environments with many directly attached hosts, such as metro Ethernet environments, increasing the amount of time between ARP updates by configuring the ARP aging timer can improve performance in an event where having thousands of clients time out at the same time might impact packet forwarding performance. In environments where there are devices connected with lower ARP aging timers (less than 20 minutes), decreasing the ARP aging timer can improve performance by preventing the flooding of traffic toward next hops with expired ARP entries. In most environments, the default ARP aging timer value does not need to be adjusted.

To configure the system-wide ARP aging timer, include the **aging-timer** statement at the **[edit system arp]** hierarchy level:

```
[edit system arp]
user@host# aging-timer minutes
```

The aging timer range is from 1 through 240 minutes. The timer value you configure takes effect as ARP entries expire. In other words, each subsequent refreshed ARP entry receives the new timer value. The new timer value does not apply to ARP entries that exist at the time you commit the configuration.

For more information about statements you can configure at the **[edit system]** hierarchy level, see the *Junos OS Administration Library*.

RELATED DOCUMENTATION

- *arp*
- *Ethernet Interfaces Overview*
- *Ethernet Interfaces User Guide for Routing Devices*

15
CHAPTER

Configuring Tagged VLANs

Configuring Tagged VLANs

IN THIS SECTION

Creating a Series of Tagged VLANs

When you divide an Ethernet LAN into multiple VLANs, each VLAN is assigned a unique IEEE 802.1Q tag. This tag is associated with each frame in the VLAN, and the network nodes receiving the traffic can use the tag to identify which VLAN a frame is associated with.

Instead of configuring VLANs and 802.1Q tags one at a time for a trunk interface, you can configure a VLAN range to create a series of tagged VLANs.

When an Ethernet LAN is divided into VLANs, each VLAN is identified by a unique 802.1Q tag. The tag is applied to all frames so that the network nodes receiving the frames can detect which VLAN the frames belong to. Trunk ports, which multiplex traffic among a number of VLANs, use the tag to determine the origin of frames and where to forward them.

For example, you could configure the VLAN **employee** and specify a tag range of **10 through 12**. This creates the following VLANs and tags:

- VLAN **employee-10**, tag **10**
- VLAN **employee-11**, tag **11**
- VLAN **employee-12**, tag **12**

Creating tagged VLANs in a series has the following limitations:

- Layer 3 interfaces do not support this feature.
- Because an access interface can only support one VLAN member, access interfaces also do not support this feature.

> **NOTE:** This task uses Junos OS for QFX3500 and QFX3600 switches that does not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does support ELS, see "Creating a Series of Tagged VLANs on Switches with ELS Support" on page 359.

To configure a series of tagged VLANs using the CLI (here, the VLAN is **employee**):

1. Configure the series (here, a VLAN series from 120 through 130):

```
[edit]
user@switch# set vlans employee vlan-range 120-130
```

2. Associate a series of tagged VLANs when you configure an interface in one of two ways:

- Include the name of the series:

```
[edit interfaces]
user@switch# set interfaces xe-0/0/22.0 family ethernet-switching vlanmembers employee
```

- Include the VLAN range:

```
[edit interfaces]
user@switch# set interfaces xe-0/0/22.0 family ethernet-switching vlan members 120–130
```

Associating a series of tagged VLANS to an interface by name or by VLAN range has the same result: VLANs **__employee_120__** through **__employee_130__** are created.

> **NOTE:** When a series of VLANs is created using the **vlan-range** command, the VLAN names are preceded and followed by a double underscore.

Creating a Series of Tagged VLANs on EX Series Switches (CLI Procedure)

To identify which VLAN traffic belongs to, all frames on an Ethernet VLAN are identified by a tag, as defined in the IEEE 802.1Q standard. These frames are *tagged* and are encapsulated with 802.1Q tags. For a simple network that has only a single VLAN, all traffic has the same 802.1Q tag.

Instead of configuring VLANS and 802.1Q tags one at a time for a trunk interface, you can configure a VLAN range to create a series of tagged VLANs.

When an Ethernet LAN is divided into VLANs, each VLAN is identified by a unique 802.1Q tag. The tag is applied to all frames so that the network nodes receiving the frames know which VLAN the frames belong to. Trunk ports, which multiplex traffic among a number of VLANs, use the tag to determine the origin of frames and where to forward them.

For example, you could configure the VLAN **employee** and specify a tag range of **10-12**. This creates the following VLANs and tags:

- VLAN **employee-10**, tag **10**
- VLAN **employee-11**, tag **11**
- VLAN **employee-12**, tag **12**

Creating tagged VLANs in a series has the following limitations:

- Layer 3 interfaces do not support this feature.
- Because an access interface can only support one VLAN member, access interfaces also do not support this feature.
- Voice over IP (VoIP) configurations do not support a range of tagged VLANs.

To configure a series of tagged VLANs using the CLI (here, the VLAN is **employee**):

1. Configure the series (here, a VLAN series from 120 through 130):

```
[edit]
user@switch# set vlans employee vlan-range 120-130
```

2. Associate a series of tagged VLANs when you configure an interface in one of two ways:

- Include the name of the series:

```
[edit interfaces]
user@switch# set interfaces ge-0/0/22.0 family ethernet-switching vlan members employee
```

- Include the VLAN range:

```
[edit interfaces]
user@switch# set interfaces ge-0/0/22.0 family ethernet-switching vlan members 120–130
```

Associating a series of tagged VLANS to an interface by name or by VLAN range have the same result: VLANs **__employee_120__** through **__employee_130__** are created.

> **NOTE:** When a series of VLANs are created using the **vlan-range** command, the VLAN names are prefixed and suffixed with a double underscore.

SEE ALSO

Verifying That a Series of Tagged VLANs Has Been Created on an EX Series Switch | 363

Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch | 226

Example: Setting Up Bridging with Multiple VLANs for EX Series Switches | 265

Example: Connecting an EX Series Access Switch to a Distribution Switch

Understanding Bridging and VLANs on Switches | 168

Creating a Series of Tagged VLANs on Switches with ELS Support

When you divide an Ethernet LAN into multiple VLANs, each VLAN is assigned a unique IEEE 802.1Q tag. This tag is associated with each frame in the VLAN, and the network nodes receiving the traffic can use the tag to identify which VLAN a frame is associated with.

Instead of configuring VLANs and 802.1Q tags one at a time for a trunk interface, you can configure a VLAN range to create a series of tagged VLANs.

When an Ethernet LAN is divided into VLANs, each VLAN is identified by a unique 802.1Q tag. The tag is applied to all frames so that the network nodes receiving the frames can detect which VLAN the frames belong to. Trunk ports, which multiplex traffic among a number of VLANs, use the tag to determine the origin of frames and where to forward them.

For example, you could configure the VLAN **employee** and specify a tag range of **10 through 12**. This creates the following VLANs and tags:

- VLAN **employee-10**, tag **10**
- VLAN **employee-11**, tag **11**
- VLAN **employee-12**, tag **12**

Creating tagged VLANs in a series has the following limitations:

- Layer 3 interfaces do not support this feature.
- Because an access interface can only support one VLAN member, access interfaces also do not support this feature.

> **NOTE:** This task uses Junos OS for Junos OS for QFX3500 and QFX3600 switches with support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does not support ELS, see "Creating a Series of Tagged VLANs" on page 355. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50.

To configure a series of tagged VLANs using the CLI (here, the VLAN is **employee**):

1. Configure the series (here, a VLAN series from 120 through 130):

```
[edit]
user@switch#  set vlans employee vlan-id-list [ 120-130 ]
```

2. Associate a series of tagged VLANs when you configure an interface in one of two ways:

- Include the name of the series:

```
[edit interfaces]
user@switch#  set interfaces xe-0/0/22.0 family ethernet-switching vlanmembers employee
```

- Include the VLAN range:

```
[edit interfaces]
user@switch#  set interfaces xe-0/0/22.0 family ethernet-switching vlan members 120–130
```

Associating a series of tagged VLANS to an interface by name or by VLAN range the same result: VLANs **__employee_120__** through **__employee_130__** are created.

> **NOTE:** When a series of VLANs is created using the **vlan-id-list** command, the VLAN names are preceded and followed by a double underscore.

SEE ALSO

Example: Setting Up Bridging with Multiple VLANs on Switches | 243

Understanding Bridging and VLANs on Switches | 168

Verifying That a Series of Tagged VLANs Has Been Created

Purpose

Verify that a series of tagged VLANs has been created on the switch.

Action

1. Display the VLANs in the ascending order of their VLAN ID:

```
user@switch>  show vlans sort-by tag
```

```
Name              Tag        Interfaces
__employee_120__  120
                             xe-0/0/22.0*
__employee_121__  121
                             xe-0/0/22.0*
__employee_122__  122
                             xe-0/0/22.0*
__employee_123__  123
                             xe-0/0/22.0*
__employee_124__  124
                             xe-0/0/22.0*
__employee_125__  125
                             xe-0/0/22.0*
__employee_126__  126
                             xe-0/0/22.0*
__employee_127__  127
                             xe-0/0/22.0*
__employee_128__  128
                             xe-0/0/22.0*
__employee_129__  129
                             xe-0/0/22.0*
__employee_130__  130
                             xe-0/0/22.0*
```

2. Display the VLANs by the alphabetical order of the VLAN name:

   `user@switch>` **show vlans sort-by name**

```
Name              Tag   Interfaces

__employee_120__ 120
                        xe-0/0/22.0*
__employee_121__ 121
                        xe-0/0/22.0*
__employee_122__ 122
                        xe-0/0/22.0*
__employee_123__ 123
                        xe-0/0/22.0*
__employee_124__ 124
                        xe-0/0/22.0*
__employee_125__ 125
                        xe-0/0/22.0*
```

```
__employee_126__ 126
                        xe-0/0/22.0*
__employee_127__ 127
                        xe-0/0/22.0*
__employee_128__ 128
                        xe-0/0/22.0*
__employee_129__ 129
                        xe-0/0/22.0*
__employee_130__ 130
                        xe-0/0/22.0*
```

3. Display the VLANs by specifying the VLAN range name (here, the VLAN range name is **employee**):

   `user@switch>` **show vlans employee**

```
Name             Tag   Interfaces

__employee_120__ 120
                        xe-0/0/22.0*
__employee_121__ 121
                        xe-0/0/22.0*
__employee_122__ 122
                        xe-0/0/22.0*
__employee_123__ 123
                        xe-0/0/22.0*
__employee_124__ 124
                        xe-0/0/22.0*
__employee_125__ 125
                        xe-0/0/22.0*
__employee_126__ 126
                        xe-0/0/22.0*
__employee_127__ 127
                        xe-0/0/22.0*
__employee_128__ 128
                        xe-0/0/22.0*
__employee_129__ 129
                        xe-0/0/22.0*
__employee_130__ 130
                        xe-0/0/22.0*
```

Meaning

The sample output shows the VLANs configured on the switch. The series of tagged VLANs is displayed: **__employee__120__** through **__employee_130__**. Each of the tagged VLANs is configured on the trunk interface **xe-0/0/22.0**. The asterisk (*) next to the interface name indicates that the interface is **UP**.

When a series of VLANs is created using the **vlan-range** statement, the VLAN names are preceded and followed by a double underscore.

Verifying That a Series of Tagged VLANs Has Been Created on an EX Series Switch

Purpose

Verify that a series of tagged VLANs is created on the switch.

Action

Display the VLANs in the ascending order of their VLAN ID:

```
user@switch>  show vlans sort-by tag
```

```
Name             Tag        Interfaces
__employee_120__  120
                            ge-0/0/22.0*
__employee_121__  121
                            ge-0/0/22.0*
__employee_122__  122
                            ge-0/0/22.0*
__employee_123__  123
                            ge-0/0/22.0*
__employee_124__  124
                            ge-0/0/22.0*
__employee_125__  125
                            ge-0/0/22.0*
__employee_126__  126
                            ge-0/0/22.0*
__employee_127__  127
                            ge-0/0/22.0*
__employee_128__  128
                            ge-0/0/22.0*
__employee_129__  129
                            ge-0/0/22.0*
```

```
__employee_130__  130
                            ge-0/0/22.0*
```

Display the VLANs by the alphabetical order of the VLAN name:

```
user@switch>  show vlans sort-by name
```

```
Name             Tag   Interfaces

__employee_120__ 120
                       ge-0/0/22.0*
__employee_121__ 121
                       ge-0/0/22.0*
__employee_122__ 122
                       ge-0/0/22.0*
__employee_123__ 123
                       ge-0/0/22.0*
__employee_124__ 124
                       ge-0/0/22.0*
__employee_125__ 125
                       ge-0/0/22.0*
__employee_126__ 126
                       ge-0/0/22.0*
__employee_127__ 127
                       ge-0/0/22.0*
__employee_128__ 128
                       ge-0/0/22.0*
__employee_129__ 129
                       ge-0/0/22.0*
__employee_130__ 130
                       ge-0/0/22.0*
```

Display the VLANs by specifying the VLAN-range name (here, the VLAN-range name is **employee**):

```
user@switch>  show vlans employee
```

```
Name             Tag   Interfaces

__employee_120__ 120
                       ge-0/0/22.0*
```

```
__employee_121__ 121
                      ge-0/0/22.0*
__employee_122__ 122
                      ge-0/0/22.0*
__employee_123__ 123
                      ge-0/0/22.0*
__employee_124__ 124
                      ge-0/0/22.0*
__employee_125__ 125
                      ge-0/0/22.0*
__employee_126__ 126
                      ge-0/0/22.0*
__employee_127__ 127
                      ge-0/0/22.0*
__employee_128__ 128
                      ge-0/0/22.0*
__employee_129__ 129
                      ge-0/0/22.0*
__employee_130__ 130
                      ge-0/0/22.0*
```

Meaning

The sample output shows the VLANs configured on the switch. The series of tagged VLANs is displayed: **__employee__120__** through **__employee_130__**. Each of the tagged VLANs is configured on the trunk interface **ge-0/0/22.0**. The asterisk (*) beside the interface name indicates that the interface is **UP**.

When a series of VLANs is created using the **vlan-range** statement, the VLAN names are prefixed and suffixed with a double underscore.

Configuring Double-Tagged VLANs on Layer 3 Logical Interfaces

Junos OS supports a subset of the IEEE 802.1Q standard for channelizing an Ethernet interface into multiple logical interfaces, allowing many hosts to be connected to the same switch but preventing them from being in the same routing or bridging domain. When an Ethernet LAN is divided into VLANs, each VLAN is identified by a unique 802.1Q tag. The tag is applied to all frames so that network nodes receiving the frames can detect which VLAN the frames belong to.

You can configure double VLAN tags (that is, an inner and an outer tag) on a Layer 3 logical interface (sometimes called a “Layer 3 subinterface”).

Support for double-tagging VLANs on Layer 3 logical interfaces includes:

- Configuration of an IPv4, an IPv6, or an **mpls** family on the logical interface
- Configuration over an aggregated Ethernet interface
- Configuration of multiple logical interfaces on a single physical interface

> **NOTE:** This feature does not include support for the following:
>
> - VLAN rewrite (**input-vlan-map** or **output-vlan-map**)
> - TPID configuration (on physical or logical interfaces)
> - **native-inner-vlan-id**; **outer-vlan-id-list**; **inner-vlan-id-list**; or **vlan-id-range**

To configure a double-tagged Layer 3 logical interface:

1. Apply flexible VLAN tagging to the physical interface:

   ```
   [edit]
   user@switch# set interfaces interface-name flexible-vlan-tagging
   ```

2. Configure inner and outer VLAN tags on the logical interface:

   ```
   [edit]
   user@switch# set interfaces interface-name unit logical-unit-number vlan-tags outer vlan-id
   user@switch# set interfaces interface-name unit logical-unit-number vlan-tags inner vlan-id
   ```

3. Set the family type and, if needed, the address on the logical interface:

   ```
   [edit]
   user@switch# set interfaces interface-name unit logical-unit-number family family-type address
   address
   ```

SEE ALSO

Configuring VLANs on Switches with Enhanced Layer 2 Support | 179

Stacking a VLAN Tag

To stack a VLAN tag on all tagged frames entering or exiting the interface, include the **push**, **vlan-id**, and **tag-protocol-id** statements in the input VLAN map or the output VLAN map:

```
input-vlan-map input-vlan-map {
    push;
    vlan-id number;
    tag-protocol-id tpid;
}
output-vlan-map {
    push;
    tag-protocol-id tpid;
}
```

You can include these statements at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number*]**
- **[edit interfaces *interface-name* unit *logical-unit-number*]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]**

The VLAN IDs you define in the input VLAN maps are stacked on top of the VLAN ID bound to the logical interface.

All TPIDs you include in input and output VLAN maps must be among those you specify at the **[edit interfaces *interface-name* ether-options ethernet-switch-profile tag-protocol-id [*tpids*]]** hierarchy level.

Rewriting a VLAN Tag and Adding a New Tag

On Ethernet IQ, IQ2 and IQ2-E interfaces, on MX Series router Gigabit Ethernet, Tri-Rate Ethernet copper, and 10-Gigabit Ethernet interfaces, on aggregated Ethernet interfaces using Gigabit Ethernet IQ2 and IQ2-E or 10-Gigabit Ethernet PICs on MX Series routers, and on Gigabit Ethernet and 10-Gigabit Ethernet interfaces on EX Series switches, to replace the outer VLAN tag of the incoming frame with a user-specified VLAN tag value, include the **swap-push** statement in the input VLAN map or output VLAN map:

```
swap-push
```

A user-specified outer VLAN tag is pushed in front. The outer tag becomes an inner tag in the final frame. The stacked and rewriting Gigabit-Ethernet VLAN Tags are also referred to as Q-in-Q tunneling.

You can include this statement at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number* input-vlan-map]**
- **[edit interfaces *interface-name* unit *logical-unit-number* output-vlan-map]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* input-vlan-map]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* output-vlan-map]**

See "Rewrite Operations and Statement Usage for Input VLAN Maps" on page 376 and "Rewrite Operations and Statement Usage for Output VLAN Maps" on page 377 for information about configuring inner and outer VLAN ID values and inner and outer TPID values required for VLAN maps.

SEE ALSO

input-vlan-map | 1195

output-vlan-map | 1294

swap-push | 1356

unit

Ethernet Interfaces User Guide for Routing Devices

Rewriting the Inner and Outer VLAN Tags

On Ethernet IQ, IQ2 and IQ2-E interfaces, on MX Series router Gigabit Ethernet, Tri-Rate Ethernet copper, and 10-Gigabit Ethernet interfaces, and on aggregated Ethernet interfaces using Gigabit Ethernet IQ2 and IQ2-E or 10-Gigabit Ethernet PICs on MX Series routers, to replace both the inner and the outer VLAN tags of the incoming frame with a user-specified VLAN tag value, include the **swap-swap** statement in the input VLAN map or output VLAN map: The stacked and rewriting Gigabit-Ethernet VLAN Tags are also referred to as Q-in-Q tunneling.

```
swap-swap;
```

You can include this statement at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number* input-vlan-map]**
- **[edit interfaces *interface-name* unit *logical-unit-number* output-vlan-map]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* input-vlan-map]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* output-vlan-map]**

See "Rewrite Operations and Statement Usage for Input VLAN Maps" on page 376 and "Rewrite Operations and Statement Usage for Output VLAN Maps" on page 377 for information about configuring inner and outer VLAN ID values and inner and outer TPID values required for VLAN maps.

SEE ALSO

> *Ethernet Interfaces User Guide for Routing Devices*

Rewriting the VLAN Tag on Tagged Frames

To rewrite the VLAN tag on all tagged frames entering the interface to a specified VLAN ID and TPID, include the **swap**, **tag-protocol-id**, and **vlan-id** statements in the input VLAN map:

```
input-vlan-map {
    swap;
    vlan-id number;
    tag-protocol-id tpid;
}
```

To rewrite the VLAN tag on all tagged frames exiting the interface to a specified VLAN ID and TPID, include the **swap** and **tag-protocol-id** statements in the output VLAN map:

```
output-vlan-map {
    swap;
    vlan-id number;
    tag-protocol-id tpid;
}
```

You can include these statements at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number* input-vlan-map]**

- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* input-vlan-map]**

You cannot include both the **swap** statement and the **vlan-id** statement in the output VLAN map configuration. If you include the **swap** statement in the configuration, the VLAN ID in outgoing frames is rewritten to the VLAN ID bound to the logical interface. For more information about binding a VLAN ID to the logical interface, see "802.1Q VLANs Overview" on page 295.

The swap operation works on the outer tag only, whether or not you include the **stacked-vlan-tagging** statement in the configuration. For more information, see "Examples: Stacking and Rewriting Gigabit Ethernet IQ VLAN Tags" on page 405.

SEE ALSO

> *Ethernet Interfaces User Guide for Routing Devices*

Configuring VLAN Translation with a VLAN ID List

In many cases, the VLAN identifiers on the frames of an interface's packets are not correct. VLAN translation, or VLAN rewrite, allows you to configure bidirectional VLAN identifier translation with a list on frames arriving on and leaving from a logical interface. This lets you use unique VLAN identifiers internally and maintain legacy VLAN identifiers on logical interfaces.

To perform VLAN translation on the packets on a trunk interface, insert the **vlan-rewrite** statement at the **[edit interfaces *interface-name* unit *unit-number*]** hierarchy level. You must also include the **interface-mode trunk** statement within the **[edit interfaces *interface-name* unit *unit-number* family ethernet-switching]** hierarchy because VLAN translation is only supported on trunk interfaces. The reverse translation takes place on traffic exiting the interface. In other words, if VLAN 200 is translated to 500 on traffic entering the interface, VLAN 500 is translated to VLAN 200 on traffic leaving the interface.

NOTE: You can configure either flexible VLAN tagging or trunk mode on interfaces. VLAN translation does not support both.

The following example translates incoming trunk packets from VLAN identifier 200 to 500 and 201 to 501 (other valid VLAN identifiers are not affected):

```
[edit interfaces ge-1/0/1]
unit 0 {
   ... # Other logical interface statements
   family ethernet-switching {
      interface-mode trunk # Translation is only for trunks
      vlan {
         members 500-501;
      }
      vlan-rewrite {
         translate 200 500;
         translate 201 501;
      }
   ... # Other ethernet-switching statements
   }
}
```

NOTE: This example also translates frame VLANs from 500 to 200 and 501 to 201 on egress.

Configuring VLAN Translation on Security Devices

VLAN translation allow service providers to create a Layer 2 Ethernet connection between two customer sites. Providers can segregate different customers' VLAN traffic on a link (for example, if the customers use overlapping VLAN IDs) or bundle different customer VLANs into a single service VLAN. Data centers can use Q-in-Q tunneling to isolate customer traffic within a single site or when customer traffic flows between cloud data centers in different geographic locations.

Before you begin configuring VLAN translation, make sure you have created and configured the necessary customer VLANs on the neighboring switches. See *Configuring VLANs*.

VLAN translation can be done in two ways:

- To configure VLAN translation in VLAN retagging, an enterprise provider style of VLAN translation can be achieved by following CLI configuration:

  [edit]
  `user@host#` **set interfaces *intf-name* unit 0 family ethernet-switching interface-mode trunk**
  `user@host#` **set interfaces *intf-name* unit 0 family ethernet-switching vlan members v1000**
  `user@host#` **set interfaces *intf-name* unit 0 family ethernet-switching vlan-rewrite translate 500 1000**

- To configure VLAN translation in Q-in-Q, a service provider style of VLAN translation can be achieved by following CLI configuration:

  [edit]
  `user@host#` **set interfaces *intf-name* flexible-vlan-tagging**
  `user@host#` **set interfaces *intf-name* encapsulation extended-vlan-bridge**
  `user@host#` **set interfaces *intf-name* unit 100 vlan-id 500**
  `user@host#` **set interfaces *intf-name* unit 100 input-vlan-map swap**
  `user@host#` **set interfaces *intf-name* unit 100 input-vlan-map tag-protocol-id 0x8100**
  `user@host#` **set interfaces *intf-name* unit 100 output-vlan-map swap**
  `user@host#` **set interfaces *intf-name* unit 100 family ethernet-switching vlan members v1000**

SEE ALSO

Understanding Q-in-Q Tunneling and VLAN Translation | 887

Example: Configuring VLAN Retagging for Layer 2 Transparent Mode on a Security Device

IN THIS SECTION

This example shows how to configure VLAN retagging on a Layer 2 trunk interface to selectively screen incoming packets and redirect them to a security device without affecting other VLAN traffic.

Requirements

Before you begin, determine the mapping you want to include for the VLAN retagging. See "Understanding VLAN Retagging on Security Devices" on page 1001.

Overview

In this example, you create a Layer 2 trunk interface called ge-3/0/0 and configure it to receive packets with VLAN identifiers 1 through 10. Packets that arrive on the interface with VLAN identifier 11 are retagged with VLAN identifier 2. Before exiting the trunk interface, VLAN identifier 2 in the retagged packets is replaced with VLAN identifier 11. All VLAN identifiers in the retagged packets change back when you exit the trunk interface.

Configuration

Step-by-Step Procedure

To configure VLAN retagging on a Layer 2 trunk interface:

1. Create a Layer 2 trunk interface.

```
[edit]
user@host#set interfaces ge-3/0/0 unit 0 family ethernet-switching interface-mode trunk vlan members
  1–10
```

2. Configure VLAN retagging.

```
[edit]
user@host#set interfaces ge-3/0/0 unit 0 family ethernet-switching vlan-rewrite translate 11 2
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show interfaces ge-3/0/0** command.

SEE ALSO

Ethernet Switching and Layer 2 Transparent Mode Overview | 41

Example: Configuring Layer 2 Logical Interfaces on Security Devices | 963

Configuring Inner and Outer TPIDs and VLAN IDs

For some rewrite operations, you must configure the inner or outer TPID values and inner or outer VLAN ID values. These values can be applied to either the input VLAN map or the output VLAN map.

On Ethernet IQ, IQ2, and IQ2-E interfaces; on MX Series router Gigabit Ethernet, Tri-Rate Ethernet copper, and 10-Gigabit Ethernet interfaces; and on aggregated Ethernet interfaces using Gigabit Ethernet IQ2 and IQ2-E or 10-Gigabit Ethernet PICs on MX Series routers, to configure the inner TPID, include the **inner-tag-protocol-id** statement:

```
inner-tag-protocol-id tpid;
```

You can include this statement at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number* input-vlan-map]**
- **[edit interfaces *interface-name* unit *logical-unit-number* output-vlan-map]**

- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* input-vlan-map]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* output-vlan-map]**

For the inner VLAN ID, include the **inner-vlan-id** statement. For the outer TPID, include the **tag-protocol-id** statement. For the outer VLAN ID, include the **vlan-id** statement:

```
input-vlan-map {
   (pop | pop-pop | pop-swap | push | push-push | swap | swap-push | swap-swap);
   inner-tag-protocol-id tpid;
   inner-vlan-id number;
   tag-protocol-id tpid;
   vlan-id number;
}
output-vlan-map {
   (pop | pop-pop | pop-swap | push | push-push | swap | swap-push | swap-swap);
   inner-tag-protocol-id tpid;
   inner-vlan-id number;
   tag-protocol-id tpid;
   vlan-id number;
}
```

For aggregated Ethernet interfaces using Gigabit Ethernet IQ interfaces, include the **tag-protocol-id** statement for the outer TPID. For the outer VLAN ID, include the **vlan-id** statement:

```
input-vlan-map {
   (pop | push | swap);
   tag-protocol-id tpid;
   vlan-id number;
}
output-vlan-map {
   (pop | push | swap);
   tag-protocol-id tpid;
   vlan-id number;
}
```

You can include these statements at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number*]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]**

The VLAN IDs you define in the input VLAN maps are stacked on top of the VLAN ID bound to the logical interface. For more information about binding a VLAN ID to the logical interface, see “802.1Q VLANs Overview” on page 295.

All TPIDs you include in input and output VLAN maps must be among those you specify at the **[edit interfaces *interface-name* ether-options ethernet-switch-profile tag-protocol-id [*tpids*]]** hierarchy level.

Table 66 on page 376 and Table 67 on page 377 specify when these statements are required. Table 66 on page 376 indicates valid statement combinations for rewrite operations for the input VLAN map. “No” means the statement must not be included in the input VLAN map for the rewrite operation. “Optional” means the statement may be optionally specified for the rewrite operation in the input VLAN map. “Any” means that you must include the **vlan-id** statement, **tag-protocol-id** statement, **inner-vlan-id** statement, or **inner-tag-protocol-id** statement.

Table 66: Rewrite Operations and Statement Usage for Input VLAN Maps

	Input VLAN Map Statements			
Rewrite Operation	**vlan-id**	**tag-protocol-id**	**inner-vlan-id**	**inner-tag-protocol-id**
push	Optional	Optional	No	No
pop	No	No	No	No
swap	Any	Any	No	No
push-push	Optional	Optional	Optional	optional
swap-push	Optional	Optional	Any	Any
swap-swap	Optional	Optional	Any	Any
pop-swap	No	No	Any	Any
pop-pop	No	No	No	No

Table 67 on page 377 indicates valid statement combinations for rewrite operations for the output VLAN map. “No” means the statement must not be included in the output VLAN map for the rewrite operation. “Optional” means the statement may be optionally specified for the rewrite operation in the output VLAN map.

Table 67: Rewrite Operations and Statement Usage for Output VLAN Maps

	Output VLAN Map Statements			
Rewrite Operation	**vlan-id**	**tag-protocol-id**	**inner-vlan-id**	**inner-tag-protocol-id**
push	No	Optional	No	No
pop	No	No	No	No
swap	No	Optional	No	No
push-push	No	Optional	No	Optional
swap-push	No	Optional	No	Optional
swap-swap	No	Optional	No	Optional
pop-swap	No	No	No	Optional
pop-pop	No	No	No	No

The following examples use Table 66 on page 376 and Table 67 on page 377 and show how the **pop-swap** operation can be configured in an input VLAN map and an output VLAN map:

Input VLAN Map with inner-vlan-id Statement, Output VLAN Map with Optional inner-tag-protocol-id Statement

```
[edit interfaces interface-name unit logical-unit-number]
input-vlan-map {
    pop-swap;
    inner-vlan-id number;
}
output-vlan-map {
    pop-swap;
    inner-tag-protocol-id tpid;
}
```

Input VLAN Map with inner-tag-protocol-id Statement, Output VLAN Map with Optional inner-tag-protocol-id Statement

```
[edit interfaces interface-name unit logical-unit-number]
input-vlan-map {
    pop-swap;
    inner-tag-protocol-id tpid;
}
output-vlan-map {
    pop-swap;
    inner-tag-protocol-id tpid;
}
```

Input VLAN Map with inner-tag-protocol-id and inner-vlan-id Statements

```
[edit interfaces interface-name unit logical-unit-number]
input-vlan-map {
    pop-swap;
    inner-vlan-id number;
    inner-tag-protocol-id tpid;
}
```

RELATED DOCUMENTATION

Using the Enhanced Layer 2 Software CLI | 50

Example: Configuring Redundant Trunk Links for Faster Recovery on Devices with ELS Support | 940

16

CHAPTER

Stacking and Rewriting Gigabit Ethernet VLAN Tags

Stacking and Rewriting Gigabit Ethernet VLAN Tags Overview

Stacking and rewriting VLAN tags, commonly known as Q-in-Q tunneling, allows you to use an additional (outer) VLAN tag to differentiate between customer edge (CE) routers that share one VLAN ID. A frame can be received on an interface, or it can be internal to the system (as a result of the **input-vlan-map** statement).

On IQ2 interfaces, 10-Gigabit Ethernet LAN/WAN PIC, 40-Gigabit Ethernet MIC, 100-Gigabit Ethernet MIC, IQ2-E interfaces, and MX Series interfaces, when a VLAN tag is pushed, the inner VLAN IEEE 802.1p bits are copied to the IEEE bits of the VLAN or VLANs being pushed. If the original packet is untagged, the IEEE bits of the VLAN or VLANs being pushed are set to 0.

> **NOTE:** When swap-by-poppush is configured on the interface, when a VLAN tag is swapped, the inner VLAN IEEE 802.1p bits are copied to the IEEE bits of the VLAN being swapped. If swap-by-poppush is not configured on the interface, the VLAN IEEE 802.1p bits of the of the VLAN being swapped remains same.

You can stack and rewrite VLAN tags on the following interfaces:

- Gigabit Ethernet
- Gigabit Ethernet IQ
- 10-Gigabit Ethernet LAN/WAN PIC
- 40-Gigabit Ethernet MIC
- 100-Gigabit Ethernet MIC
- Gigabit Ethernet IQ2 and IQ2-E
- 10-Gigabit Ethernet IQ2 and IQ2-E interfaces, and MX Series router Gigabit Ethernet Interfaces
- Tri-Rate Ethernet copper, and 10-Gigabit Ethernet interfaces with the VLAN encapsulation type configured to support Layer 2 tunneling protocols such as circuit cross-connect (CCC) or virtual private LAN service (VPLS) (as described in “802.1Q VLANs Overview” on page 295)

RELATED DOCUMENTATION

802.1Q VLANs Overview | 295

Stacking and Rewriting Gigabit Ethernet VLAN Tags | 382

Stacking and Rewriting Gigabit Ethernet VLAN Tags

You can configure rewrite operations to stack (**push**), remove (**pop**), or rewrite (**swap**) tags on single-tagged frames and dual-tagged frames. If a port is not tagged, rewrite operations are not supported on any logical interface on that port. Stacked and rewriting Gigabit-Ethernet VLAN Tags are also referred to as Q-in-Q tunneling.

You can configure the following VLAN rewrite operations:

- **pop**—Remove a VLAN tag from the top of the VLAN tag stack. The outer VLAN tag of the frame is removed.
- **pop-pop**—For Ethernet IQ2, 10-Gigabit Ethernet LAN/WAN PIC, and IQ2-E interfaces, remove both the outer and inner VLAN tags of the frame.
- **pop-swap**—For Ethernet IQ2, 10-Gigabit Ethernet LAN/WAN PIC, and IQ2-E interfaces, remove the outer VLAN tag of the frame, and replace the inner VLAN tag of the frame with a user-specified VLAN tag value. The inner tag becomes the outer tag in the final frame.
- **push**—Add a new VLAN tag to the top of the VLAN stack. An outer VLAN tag is pushed in front of the existing VLAN tag.
- **push-push**—For Ethernet IQ2, 10-Gigabit Ethernet LAN/WAN PIC, and IQ2-E interfaces, push two VLAN tags in front of the frame.
- **swap-push**—For Ethernet IQ2, 10-Gigabit Ethernet LAN/WAN PIC, and IQ2-E interfaces, replace the outer VLAN tag of the frame with a user-specified VLAN tag value. A user-specified outer VLAN tag is pushed in front. The outer tag becomes an inner tag in the final frame.
- **swap-swap**—For Ethernet IQ2, 10-Gigabit Ethernet LAN/WAN PIC, and IQ2-E interfaces, replace both the inner and the outer VLAN tags of the incoming frame with a user-specified VLAN tag value.

You configure VLAN rewrite operations for logical interfaces in the input VLAN map for incoming frames and in the output VLAN map for outgoing frames. To configure the input VLAN map, include the **input-vlan-map** statement:

```
input-vlan-map {
    ...interface-specific configuration...
}
```

To configure the output VLAN map, include the **output-vlan-map** statement:

```
output-vlan-map {
    ...interface-specific configuration...
}
```

You can include both statements at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number*]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]**

The type of VLAN rewrite operation permitted depends upon whether the frame is single-tagged or dual-tagged. Table 68 on page 383 shows supported rewrite operations and whether they can be applied to single-tagged frames or dual-tagged frames. The table also indicates the number of tags being added or removed during the operation.

Table 68: Rewrite Operations on Untagged, Single-Tagged, and Dual-Tagged Frames

Rewrite Operation	Untagged	Single-Tagged	Dual-Tagged	Number of Tags
pop	No	Yes	Yes	– 1
push	Sometimes	Yes	Yes	+1
swap	No	Yes	Yes	0
push-push	Sometimes	Yes	Yes	+2
swap-push	No	Yes	Yes	+1
swap-swap	No	No	Yes	0
pop-pop	No	No	Yes	– 2
pop-swap	No	No	Yes	– 1

The rewrite operations **push** and **push-push** can be valid in certain circumstances on frames that are not tagged. For example, a single-tagged logical interface (interface 1) and a dual-tagged logical interface (interface 2) have the following configurations:

Interface 1

```
[edit interfaces interface-name unit logical-unit-number]
```

```
input-vlan-map {
    pop;
}
output-vlan-map {
    push;
}
```

Interface 2

```
[edit interfaces interface-name unit logical-unit-number]
input-vlan-map {
    pop-pop;
}
output-vlan-map {
    push-push;
}
```

When a frame is received on the interface as a result of the **input-vlan-map** operation, the frame is not tagged. As it goes out of the second interface, the **output-vlan-map** operation **push-push** is applied to it. The resulting frame will be dual-tagged at the logical interface output.

Depending on the VLAN rewrite operation, you configure the rewrite operation for the interface in the input VLAN map, the output VLAN map, or in both the input VLAN map and the output VLAN map. Table 69 on page 384 shows what rewrite operation combinations you can configure. “None” means that no rewrite operation is specified for the VLAN map.

Table 69: Applying Rewrite Operations to VLAN Maps

Input VLAN Map	Output VLAN Map								
	none	push	pop	swap	push-push	swap-push	swap-swap	pop-pop	swap-pop
none	Yes	No	No	Yes	No	No	Yes	No	No
push	No	No	Yes	No	No	No	No	No	No
pop	No	Yes	No	No	No	No	No	No	No
swap	Yes	No	No	Yes	No	No	No	No	No

Table 69: Applying Rewrite Operations to VLAN Maps *(continued)*

Input VLAN Map	Output VLAN Map								
	none	push	pop	swap	push-push	swap-push	swap-swap	pop-pop	swap-pop
push-push	No	No	No	No	No	No	No	Yes	No
swap-push	No	No	No	No	No	No	No	No	Yes
swap-swap	Yes	No	No	No	No	No	Yes	No	No
pop-pop	No	No	No	No	Yes	No	No	No	No
pop-swap	No	No	No	No	No	Yes	No	No	No

You must know whether the VLAN rewrite operation is valid and is applied to the input VLAN map or the output VLAN map. You must also know whether the rewrite operation requires you to include statements to configure the inner and outer TPIDs and inner and outer VLAN IDs in the input VLAN map or output VLAN map. For information about configuring inner and outer TPIDs and inner and outer VLAN IDs, see "Configuring Inner and Outer TPIDs and VLAN IDs" on page 388.

RELATED DOCUMENTATION

Stacking and Rewriting Gigabit Ethernet VLAN Tags Overview | **381**

Understanding swap-by-poppush | **417**

swap-by-poppush | **1355**

Ethernet Interfaces User Guide for Routing Devices

Configuring Frames with Particular TPIDs to Be Processed as Tagged Frames

For Gigabit Ethernet IQ interfaces, aggregated Ethernet with Gigabit Ethernet IQ interfaces, Gigabit Ethernet PICs with SFPs (except the 10-port Gigabit Ethernet PIC and the built-in Gigabit Ethernet port on the M7i router), and MX Series router Gigabit Ethernet, Tri-Rate Ethernet copper, and 10-Gigabit Ethernet interfaces, you can configure frames with particular TPIDs to be processed as tagged frames. To do this, you specify up to eight IEEE 802.1Q TPID values per port; a frame with any of the specified TPIDs

is processed as a tagged frame; however, with IQ2 and IQ2-E interfaces, only the first four IEEE 802.1Q TPID values per port are supported. The stacked and rewriting Gigabit-Ethernet VLAN Tags are also referred to as Q-in-Q tunneling. To configure the TPID values, include the **tag-protocol-id** statement:

```
tag-protocol-id [ tpids ];
```

You can include this statement at the following hierarchy levels:

- **[edit interfaces *interface-name* gigether-options ethernet-switch-profile]**
- **[edit interfaces *interface-name* aggregated-ether-options ethernet-switch-profile]**

All TPIDs you include in input and output VLAN maps must be among those you specify at the **[edit interfaces *interface-name* gigether-options ethernet-switch-profile tag-protocol-id [*tpids*]]** or **[edit interfaces *interface-name* aggregated-ether-options ethernet-switch-profile tag-protocol-id [*tpids*]]** hierarchy level.

RELATED DOCUMENTATION

Stacking and Rewriting Gigabit Ethernet VLAN Tags Overview | **381**

aggregated-ether-options

ethernet-switch-profile | **1150**

gigether-options

tag-protocol-id | **1362**

Ethernet Interfaces User Guide for Routing Devices

Configuring Tag Protocol IDs (TPIDs) on PTX Series Packet Transport Routers

This topic describes how to configure the TPIDs expected to be sent or received on a particular VLAN for PTX Series Packet Transport Routers.

For other types of Juniper Networks Ethernet PICs, you could configure 8 TPIDs per port. However, the PTX Series Packet Transport Routers use MTIP and TL to classify a specific TPID and Ethernet type. For MTIP, you can configure a maximum of 8 TPIDs for each MAC chip.

As a consequence, you can specify the **tag-protocol-id** configuration statement only for the first port (**0**) of a PTX Series Ethernet PIC. If you configure **tag-protocol-id** statements on the other port, the configuration is ignored and a system error is recorded.

For example, the following is a supported configuration:

```
[edit interfaces et-2/0/0]
gigether-options {
   ethernet-switch-profile {
      tag-protocol-id [0x8100 0x9100];
   }
}
```

The **tag-protocol-id** configuration statement supports up to eightTPIDs on port **0** of a given Ethernet PIC. All eight TPIDs are populated to the two MTIPs and TLs associated with the Ethernet PIC.

RELATED DOCUMENTATION

Configuring Frames with Particular TPIDs to Be Processed as Tagged Frames | 385

Configuring Flexible VLAN Tagging on PTX Series Packet Transport Routers | 303

Configuring Stacked VLAN Tagging

To configure stacked VLAN tagging for all logical interfaces on a physical interface:

1. In configuration mode, go to the **[edit interfaces interface-name]** hierarchy level.

   ```
   [edit]
   user@host# edit interfaces interface-name
   ```

2. Include the **stacked-vlan-tagging** statement.

   ```
   [edit interfaces interface-name]
   user@host# set stacked-vlan-tagging
   ```

If you include the **stacked-vlan-tagging** statement in the configuration, you must configure dual VLAN tags for all logical interfaces on the physical interface. For more information, see "Stacking a VLAN Tag" on page 393.

RELATED DOCUMENTATION

stacked-vlan-tagging | 1350

Stacking a VLAN Tag | 393

Ethernet Interfaces User Guide for Routing Devices

Configuring Dual VLAN Tags

To configure dual VLAN tags on a logical interface, include the **vlan-tags** statement:

```
vlan-tags inner <tpid.>vlan-id outer <tpid.>vlan-id;
```

You can include this statement at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number*]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]**

The outer tag VLAN ID range is from 1 through 511 for normal interfaces, and from 512 through 4094 for VLAN CCC or VLAN VPLS interfaces. The inner tag is not restricted.

You must also include the **stacked-vlan-tagging** statement in the configuration. See "Examples: Stacking and Rewriting Gigabit Ethernet IQ VLAN Tags" on page 405.

RELATED DOCUMENTATION

unit

Examples: Stacking and Rewriting Gigabit Ethernet IQ VLAN Tags | 405

Ethernet Interfaces User Guide for Routing Devices

Configuring Inner and Outer TPIDs and VLAN IDs

For some rewrite operations, you must configure the inner or outer tag-protocol identifier (TPID) values and inner or outer virtual local area network identifier (VLAN ID) values. These values can be applied to either the input VLAN map or the output VLAN map. The stacked and rewriting Gigabit-Ethernet VLAN Tags are also referred to as Q-in-Q tunneling.

1. On Ethernet IQ, IQ2, and IQ2-E interfaces; on MX Series router Gigabit Ethernet, Tri-Rate Ethernet copper, and 10-Gigabit Ethernet interfaces; and on aggregated Ethernet interfaces using Gigabit Ethernet

IQ2 and IQ2-E or 10-Gigabit Ethernet PICs on MX Series routers, to configure the inner TPID, include the **inner-tag-protocol-id** statement at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number* input-vlan-map]**
- **[edit interfaces *interface-name* unit *logical-unit-number* output-vlan-map]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* input-vlan-map]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* output-vlan-map]**

```
user@host# set inner-tag-protocol-id tpid;
```

2. For the inner VLAN ID, include the **inner-vlan-id** statement. For the outer TPID, include the **tag-protocol-id** statement. For the outer VLAN ID, include the **vlan-id** statement at the **[edit interfaces *interface-name* unit *logical-unit-number*]** hierarchy level or at the **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]** hierarchy level.

```
input-vlan-map {
    (pop | pop-pop | pop-swap | push | push-push | swap | swap-push | swap-swap);
    inner-tag-protocol-id tpid;
    inner-vlan-id number;
    tag-protocol-id tpid;
    vlan-id number;
}
output-vlan-map {
    (pop | pop-pop | pop-swap | push | push-push | swap | swap-push | swap-swap);
    inner-tag-protocol-id tpid;
    inner-vlan-id number;
    tag-protocol-id tpid;
    vlan-id number;
}
```

3. For aggregated Ethernet interfaces using Gigabit Ethernet IQ interfaces, include the **tag-protocol-id** statement for the outer TPID. For the outer VLAN ID, include the **vlan-id** statement at the **[edit interfaces *interface-name* unit *logical-unit-number*]** hierarchy level or at the **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]** hierarchy level.

```
input-vlan-map {
    (pop | push | swap);
    tag-protocol-id tpid;
    vlan-id number;
```

```
}
output-vlan-map {
    (pop | push | swap);
    tag-protocol-id tpid;
    vlan-id number;
}
```

The VLAN IDs you define in the input VLAN maps are stacked on top of the VLAN ID bound to the logical interface. For more information about binding a VLAN ID to the logical interface, see "802.1Q VLANs Overview" on page 295.

All TPIDs you include in input and output VLAN maps must be among those you specify at the **[edit interfaces *interface-name* gigether-options ethernet-switch-profile tag-protocol-id [*tpids*]]** hierarchy level or **[edit interfaces *interface-name* aggregated-ether-options ethernet-switch-profile tag-protocol-id [*tpids*]]** hierarchy level. For more information, see "Configuring Frames with Particular TPIDs to Be Processed as Tagged Frames" on page 385.

Table 66 on page 376 and Table 67 on page 377 specify when these statements are required. Table 66 on page 376 indicates valid statement combinations for rewrite operations for the input VLAN map. "No" means the statement must not be included in the input VLAN map for the rewrite operation. "Optional" means the statement may be optionally specified for the rewrite operation in the input VLAN map. "Any" means that you must include the **vlan-id** statement, **tag-protocol-id** statement, **inner-vlan-id** statement, or **inner-tag-protocol-id** statement.

Table 70: Rewrite Operations and Statement Usage for Input VLAN Maps

	Input VLAN Map Statements			
Rewrite Operation	**vlan-id**	**tag-protocol-id**	**inner-vlan-id**	**inner-tag-protocol-id**
push	Optional	Optional	No	No
pop	No	No	No	No
swap	Any	Any	No	No
push-push	Optional	Optional	Optional	optional
swap-push	Optional	Optional	Any	Any
swap-swap	Optional	Optional	Any	Any
pop-swap	No	No	Any	Any

Table 70: Rewrite Operations and Statement Usage for Input VLAN Maps *(continued)*

	Input VLAN Map Statements			
pop-pop	No	No	No	No

Table 67 on page 377 indicates valid statement combinations for rewrite operations for the output VLAN map. “No” means the statement must not be included in the output VLAN map for the rewrite operation. “Optional” means the statement may be optionally specified for the rewrite operation in the output VLAN map.

Table 71: Rewrite Operations and Statement Usage for Output VLAN Maps

	Output VLAN Map Statements			
Rewrite Operation	**vlan-id**	**tag-protocol-id**	**inner-vlan-id**	**inner-tag-protocol-id**
push	No	Optional	No	No
pop	No	No	No	No
swap	No	Optional	No	No
push-push	No	Optional	No	Optional
swap-push	No	Optional	No	Optional
swap-swap	No	Optional	No	Optional
pop-swap	No	No	No	Optional
pop-pop	No	No	No	No

Input VLAN Map with inner-vlan-id Statement, Output VLAN Map with Optional inner-tag-protocol-id Statement

```
[edit interfaces interface-name unit logical-unit-number]
input-vlan-map {
    pop-swap;
    inner-vlan-id number;
}
```

```
output-vlan-map {
    pop-swap;
    inner-tag-protocol-id tpid;
}
```

Input VLAN Map with inner-tag-protocol-id Statement, Output VLAN Map with Optional inner-tag-protocol-id Statement

```
[edit interfaces interface-name unit logical-unit-number]
input-vlan-map {
    pop-swap;
    inner-tag-protocol-id tpid;
}
output-vlan-map {
    pop-swap;
    inner-tag-protocol-id tpid;
}
```

Input VLAN Map with inner-tag-protocol-id and inner-vlan-id Statements

```
[edit interfaces interface-name unit logical-unit-number]
input-vlan-map {
    pop-swap;
    inner-vlan-id number;
    inner-tag-protocol-id tpid;
}
```

RELATED DOCUMENTATION

- **inner-tag-protocol-id | 1192**
- **input-vlan-map | 1195**
- **output-vlan-map | 1294**
- **pop-swap | 1305**

unit

Ethernet Interfaces

Stacking a VLAN Tag

To stack a VLAN tag on all tagged frames entering or exiting the interface, include the **push**, **vlan-id**, and **tag-protocol-id** statements in the input VLAN map or the output VLAN map:

```
input-vlan-map {
    push;
    vlan-id number;
    tag-protocol-id tpid;
}
output-vlan-map {
    push;
    tag-protocol-id tpid;
}
```

You can include these statements at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number*]**
- **[edit interfaces *interface-name* unit *logical-unit-number*]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]**

If you include the **push** statement in an interface's input VLAN map, see Table 69 on page 384 for information about permissible rewrite operations,

The VLAN IDs you define in the input VLAN maps are stacked on top of the VLAN ID bound to the logical interface. For more information about binding a VLAN ID to the logical interface, see "802.1Q VLANs Overview" on page 295.

All TPIDs you include in input and output VLAN maps must be among those you specify at the **[edit interfaces *interface-name* gigether-options ethernet-switch-profile tag-protocol-id [*tpids*]]** hierarchy level. For more information, see "Configuring Inner and Outer TPIDs and VLAN IDs" on page 388.

RELATED DOCUMENTATION

tag-protocol-id | **1364**

unit

Table 69 | **384**

802.1Q VLANs Overview | **295**

Configuring Inner and Outer TPIDs and VLAN IDs | **388**

Ethernet Interfaces User Guide for Routing Devices

Stacking Two VLAN Tags

On Ethernet IQ, IQ2 and IQ2-E interfaces, on MX Series router Gigabit Ethernet, Tri-Rate Ethernet copper, and 10-Gigabit Ethernet interfaces, and on aggregated Ethernet interfaces using Gigabit Ethernet IQ2 and IQ2-E or 10-Gigabit Ethernet PICs on MX Series routers, to push two VLAN tags in front of tagged frames entering or exiting the interface, include the **push-push** statement in the input VLAN map or the output VLAN map:

```
push-push;
```

You can include this statement at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number* input-vlan-map]**
- **[edit interfaces *interface-name* unit *logical-unit-number* output-vlan-map]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* input-vlan-map]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* output-vlan-map]**

See "Rewrite Operations and Statement Usage for Input VLAN Maps" on page 376 and "Rewrite Operations and Statement Usage for Output VLAN Maps" on page 377 for information about configuring inner and outer VLAN ID values and inner and outer TPID values required for VLAN maps.

RELATED DOCUMENTATION

input-vlan-map | **1195**

output-vlan-map | **1294**

pop | **1303**

unit

> See Rewrite Operations and Statement Usage for Input VLAN Maps | 376 and Rewrite Operations and Statement Usage for Output VLAN Maps | 377 for information about configuring inner and outer VLAN ID values and inner and outer TPID values required for VLAN maps.
>
> *Ethernet Interfaces User Guide for Routing Devices*

Removing a VLAN Tag

To remove a VLAN tag from all tagged frames entering or exiting the interface, include the **pop** statement in the input VLAN map or output VLAN map:

```
pop;
```

You can include this statement at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number* input-vlan-map]**
- **[edit interfaces *interface-name* unit *logical-unit-number* output-vlan-map]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* input-vlan-map]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* output-vlan-map]**

RELATED DOCUMENTATION

> input-vlan-map | 1195
>
> output-vlan-map | 1294
>
> pop | 1303
>
> *unit*
>
> *Ethernet Interfaces User Guide for Routing Devices*

Removing the Outer and Inner VLAN Tags

On Ethernet IQ, IQ2 and IQ2-E interfaces, on MX Series router Gigabit Ethernet, Tri-Rate Ethernet copper, and 10-Gigabit Ethernet interfaces, and on aggregated Ethernet interfaces using Gigabit Ethernet IQ2 and

IQ2-E or 10-Gigabit Ethernet PICs on MX Series routers, to remove both the outer and inner VLAN tags of the frame, include the **pop-pop** statement in the input VLAN map or output VLAN map:

```
pop-pop;
```

You can include this statement at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number* input-vlan-map]**
- **[edit interfaces *interface-name* unit *logical-unit-number* output-vlan-map]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* input-vlan-map]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* output-vlan-map]**

See "Rewrite Operations and Statement Usage for Input VLAN Maps" on page 376 and "Rewrite Operations and Statement Usage for Output VLAN Maps" on page 377 for information about configuring inner and outer VLAN ID values and inner and outer TPID values required for VLAN maps.

RELATED DOCUMENTATION

input-vlan-map | **1195**
output-vlan-map | **1294**
pop-pop | **1304**
unit
See Rewrite Operations and Statement Usage for Input VLAN Maps | **376** and Rewrite Operations and Statement Usage for Output VLAN Maps | **377** for information about configuring inner and outer VLAN ID values and inner and outer TPID values required for VLAN maps.
Ethernet Interfaces User Guide for Routing Devices

Removing the Outer VLAN Tag and Rewriting the Inner VLAN Tag

On Ethernet IQ, IQ2 and IQ2-E interfaces, on MX Series router Gigabit Ethernet, Tri-Rate Ethernet copper, and 10-Gigabit Ethernet interfaces, and on aggregated Ethernet interfaces using Gigabit Ethernet IQ2 and IQ2-E or 10-Gigabit Ethernet PICs on MX Series routers, to remove the outer VLAN tag of the frame and

replace the inner VLAN tag of the frame with a user-specified VLAN tag value, include the **pop-swap** statement in the input VLAN map or output VLAN map:

```
pop-swap;
```

The inner tag becomes the outer tag in the final frame.

You can include this statement at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number* input-vlan-map]**
- **[edit interfaces *interface-name* unit *logical-unit-number* output-vlan-map]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* input-vlan-map]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* output-vlan-map]**

See "Rewrite Operations and Statement Usage for Input VLAN Maps" on page 376 and "Rewrite Operations and Statement Usage for Output VLAN Maps" on page 377 for information about configuring inner and outer VLAN ID values and inner and outer TPID values required for VLAN maps.

RELATED DOCUMENTATION

> input-vlan-map | **1195**
>
> output-vlan-map | **1294**
>
> pop-swap | **1305**
>
> *unit*
>
> See Rewrite Operations and Statement Usage for Input VLAN Maps | **376** and Rewrite Operations and Statement Usage for Output VLAN Maps | **377** for information about configuring inner and outer VLAN ID values and inner and outer TPID values required for VLAN maps.
>
> *Ethernet Interfaces User Guide for Routing Devices*

Rewriting the VLAN Tag on Tagged Frames

To rewrite the VLAN tag on all tagged frames entering the interface to a specified VLAN ID and TPID, include the **swap**, **tag-protocol-id**, and **vlan-id** statements in the input VLAN map:

```
input-vlan-map {
    swap;
    vlan-id number;
    tag-protocol-id tpid;
}
```

To rewrite the VLAN tag on all tagged frames exiting the interface to a specified VLAN ID and TPID, include the **swap** and **tag-protocol-id** statements in the output VLAN map:

```
output-vlan-map {
    swap;
    vlan-id number;
    tag-protocol-id tpid;
}
```

You can include these statements at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number* input-vlan-map]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* input-vlan-map]**

You cannot include both the **swap** statement and the **vlan-id** statement in the output VLAN map configuration. If you include the **swap** statement in the configuration, the VLAN ID in outgoing frames is rewritten to the VLAN ID bound to the logical interface. For more information about binding a VLAN ID to the logical interface, see “802.1Q VLANs Overview” on page 295.

The swap operation works on the outer tag only, whether or not you include the **stacked-vlan-tagging** statement in the configuration. For more information, see “Examples: Stacking and Rewriting Gigabit Ethernet IQ VLAN Tags” on page 405.

RELATED DOCUMENTATION

Ethernet Interfaces User Guide for Routing Devices

Rewriting a VLAN Tag on Untagged Frames

IN THIS SECTION

Overview

You can rewrite VLAN tags on untagged incoming and outgoing frames with the ethernet-ccc and the ethernet-vpls encapsulations for the following routers:

- M120 routers and M320 routers with:
  - Gigabit Ethernet IQ PIC with SFP
  - Gigabit Ethernet IQ2 PICs with SFP
  - Gigabit Ethernet Enhanced IQ2 (IQ2E) PICs with SFP
  - 10-Gigabit Ethernet IQ2 PIC with XFP
  - 10-Gigabit Ethernet Enhanced IQ2 (IQ2E) PIC with XFP
- MX240, MX480, and MX960 routers with:
  - Gigabit Ethernet Enhanced DPC with SFP
  - Gigabit Ethernet Enhanced Queuing IP Services DPCs with SFP
  - 10-Gigabit Ethernet Enhanced DPCs with XFP
  - 10-Gigabit Ethernet Enhanced Queuing IP Services DPC with XFP

On M Series routers with Gigabit Ethernet IQ2 PICs and Gigabit Ethernet Enhanced IQ2 (IQ2E) PICs, you can perform all the rewrite VLAN tag operations.

Consider a network where two provider edges (PE) are connected by a Layer 2 circuit. PE1 is receiving traffic on an untagged port while the corresponding port on PE2 is tagged. In the normal case, packets coming from PE1 will be dropped at PE2 because it is expecting tagged packets. However, if PE1 can push

a VLAN tag on the incoming packet before sending it across to PE2, you can ensure that packets are not dropped. To make it work in both directions, PE1 must strip the VLAN tag from outgoing packets. Therefore, a push on the ingress side is always paired with a pop on the egress side.

The rewrite operations represented by the following statement options are supported under **ethernet-ccc** and **ethernet-vpls** encapsulations:

- **push**—A VLAN tag is added to the incoming untagged frame.
- **pop**—VLAN tag is removed from the outgoing frame.
- **push-push**—An outer and inner VLAN tag are added to the incoming untagged frame.
- **pop-pop**—Both the outer and inner VLAN tags of the outgoing frame are removed.

IQ2 and 10-Gigabit Ethernet PICs support all rewrite operations described above. Details on the possible combinations of usage are explained later in this section.

> **NOTE:** The **push-push** and **pop-pop** operations are not supported on the Gigabit Ethernet IQ PIC.

For the **input-vlan-map** statement, only the **push** and **push-push** options are supported because it does not make sense to remove a VLAN tag from an incoming untagged frame. Similarly, only the **pop** and **pop-pop** options are supported for the **output-vlan-map** statement. Also, with the **push** and **push-push** options, the tag parameters have to be explicitly specified. Apart from this, the other rules for configuring the **input-vlan-map** and **output-vlan-map** statements are the same as for tagged frames. Table 72 on page 400 through Table 74 on page 401 explain the rules in more detail.

For the **input-vlan-map** statement, only the **push** and **push-push** options are supported because it does not make sense to remove a VLAN tag from an incoming untagged frame. Similarly, only the **pop** and **pop-pop** options are supported for the **output-vlan-map** statement. Also, with the **push** and **push-push** options, the **vlan-id** parameters (**vlan-id** for **push** and **vlan-id** or **inner-vlan-id** for **push-push**) have to be explicitly specified. TPID however, is optional and the default value of **0x8100** is set if not configured. Apart from this, the other rules for configuring the **input-vlan-map** and **output-vlan-map** statements are the same as for tagged frames.

Table 72: Input VLAN Map Statements Allowed for ethernet-ccc and ethernet-vpls Encapsulations

Operation	vlan-id	tag-protocol-id	inner-vlan-id	inner-tag-protocol-id
push	Yes	Optional	No	Optional
push-push	Yes	Optional	Yes	Optional

Table 73: Output VLAN Map Statements Allowed for ethernet-ccc and ethernet-vpls Encapsulations

Operation	vlan-id	tag-protocol-id	inner-vlan-id	inner-tag-protocol-id
pop	No	No	No	No
pop-pop	No	No	No	No

Table 74: Rules for Applying Rewrite Operations to VLAN Maps

	Output VLAN Map		
Input VLAN Map	**None**	**pop**	**pop-pop**
None	Yes	No	No
push	No	Yes	No
push-push	No	No	Yes

You can use the **show interface *interface-name*** command to display the status of a modified VLAN map for the specified interface.

Example: push and pop with Ethernet CCC Encapsulation

```
ge-3/1/0 {
    encapsulation ethernet-ccc;
    unit 0 {
        encapsulation ethernet-ccc;
        input-vlan-map {
            push;
            tag-protocol-id 0x8100;
            vlan-id 600;
        }
        output-vlan-map pop;
        family ccc;
    }
}
```

Example: push-push and pop-pop with Ethernet CCC Encapsulation

```
ge-3/1/0 {
    encapsulation ethernet-ccc;
    unit 0 {
        encapsulation ethernet-ccc;
        input-vlan-map {
            push-push;
            tag-protocol-id 0x8100;
            inner-tag-protocol-id 0x8100;
            vlan-id 600;
            inner-vlan-id 575;
        }
        output-vlan-map pop-pop;
        family ccc;
    }
}
```

Example: push and pop with Ethernet VPLS Encapsulation

```
ge-3/1/0 {
    encapsulation ethernet-vpls;
    unit 0 {
        encapsulation ethernet-vpls;
        input-vlan-map {
            push;
            tag-protocol-id 0x8100;
            vlan-id 700;
        }
        output-vlan-map pop;
        family vpls;
    }
}
```

Example: push-push and pop-pop with Ethernet VPLS Encapsulation

```
ge-3/1/0 {
    encapsulation ethernet-vpls;
    unit 0 {
        encapsulation ethernet-vpls;
        input-vlan-map {
            push-push;
            tag-protocol-id 0x8100;
            inner-tag-protocol-id 0x8100;
            vlan-id 600;
            inner-vlan-id 575;
        }
        output-vlan-map pop-pop;
        family vpls;
    }
}
```

RELATED DOCUMENTATION

- input-vlan-map | **1195**
- output-vlan-map | **1294**
- pop | **1303**
- pop-pop | **1304**
- push | **1332**
- push-push | **1333**
- *unit*
- *Ethernet Interfaces User Guide for Routing Devices*

Rewriting a VLAN Tag and Adding a New Tag

On Ethernet IQ, IQ2 and IQ2-E interfaces, on MX Series router Gigabit Ethernet, Tri-Rate Ethernet copper, and 10-Gigabit Ethernet interfaces, on aggregated Ethernet interfaces using Gigabit Ethernet IQ2 and IQ2-E or 10-Gigabit Ethernet PICs on MX Series routers, and on Gigabit Ethernet and 10-Gigabit Ethernet interfaces on EX Series switches, to replace the outer VLAN tag of the incoming frame with a user-specified VLAN tag value, include the **swap-push** statement in the input VLAN map or output VLAN map:

```
swap-push
```

A user-specified outer VLAN tag is pushed in front. The outer tag becomes an inner tag in the final frame. The stacked and rewriting Gigabit-Ethernet VLAN Tags are also referred to as Q-in-Q tunneling.

You can include this statement at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number* input-vlan-map]**
- **[edit interfaces *interface-name* unit *logical-unit-number* output-vlan-map]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* input-vlan-map]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* output-vlan-map]**

See "Rewrite Operations and Statement Usage for Input VLAN Maps" on page 376 and "Rewrite Operations and Statement Usage for Output VLAN Maps" on page 377 for information about configuring inner and outer VLAN ID values and inner and outer TPID values required for VLAN maps.

RELATED DOCUMENTATION

input-vlan-map | **1195**

output-vlan-map | **1294**

swap-push | **1356**

unit

Ethernet Interfaces User Guide for Routing Devices

Rewriting the Inner and Outer VLAN Tags

On Ethernet IQ, IQ2 and IQ2-E interfaces, on MX Series router Gigabit Ethernet, Tri-Rate Ethernet copper, and 10-Gigabit Ethernet interfaces, and on aggregated Ethernet interfaces using Gigabit Ethernet IQ2 and IQ2-E or 10-Gigabit Ethernet PICs on MX Series routers, to replace both the inner and the outer VLAN tags of the incoming frame with a user-specified VLAN tag value, include the **swap-swap** statement in the input VLAN map or output VLAN map: The stacked and rewriting Gigabit-Ethernet VLAN Tags are also referred to as Q-in-Q tunneling.

```
swap-swap;
```

You can include this statement at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number* input-vlan-map]**
- **[edit interfaces *interface-name* unit *logical-unit-number* output-vlan-map]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* input-vlan-map]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* output-vlan-map]**

See "Rewrite Operations and Statement Usage for Input VLAN Maps" on page 376 and "Rewrite Operations and Statement Usage for Output VLAN Maps" on page 377 for information about configuring inner and outer VLAN ID values and inner and outer TPID values required for VLAN maps.

RELATED DOCUMENTATION

> *Ethernet Interfaces User Guide for Routing Devices*

Examples: Stacking and Rewriting Gigabit Ethernet IQ VLAN Tags

Configure a VLAN CCC tunnel in which Ethernet frames enter the tunnel at interface **ge-4/0/0** and exit the tunnel at interface **ge-4/2/0**.

The following examples show how to perform the following tasks:

- Push a TPID and VLAN ID Pair on Ingress on page 406
- Stack Inner and Outer VLAN Tags on page 407
- Swap a VLAN ID on Ingress on page 407
- Swap a VLAN ID on Egress on page 408
- Swap a VLAN ID on Both Ingress and Egress on page 410
- Swap the Outer VLAN Tag and Push a New VLAN Tag on Ingress; Pop the Outer VLAN Tag and Swap the Inner VLAN Tag on Egress on page 411
- Swap a TPID and VLAN ID Pair for Both VLAN Tags on Ingress and on Egress on page 411
- Pop the Outer VLAN Tag and Swap the Inner VLAN Tag on Ingress; Swap the Outer VLAN Tag and Push a New VLAN Tag on Egress on page 412

- Pop a TPID and VLAN ID Pair on Ingress; Push a VLAN ID and TPID Pair on Egress on page 413
- Pop an Outer VLAN Tag to Connect an Untagged VPLS Interface to Tagged VPLS Interfaces on page 413

Push a TPID and VLAN ID Pair on Ingress

```
[edit interfaces]
ge-4/0/0 {
    vlan-tagging;
    encapsulation vlan-ccc;
    gigether-options {
        ethernet-switch-profile {
            tag-protocol-id 0x9909;
        }
    }
    unit 0 {
        encapsulation vlan-ccc;
        vlan-id 512;
        input-vlan-map {
            push;
            tag-protocol-id 0x9909;
            vlan-id 520;
        }
        output-vlan-map pop;
    }
}
ge-4/2/0 {
    vlan-tagging;
    encapsulation vlan-ccc;
    unit 0 {
        encapsulation vlan-ccc;
        vlan-id 515;
        input-vlan-map {
            swap-push;
            vlan-id 520;
            inner-vlan-id 512;
        }
        output-vlan-map {
            pop-swap;
        }
    }
    [edit protocols]
    mpls {
        interface ge-4/0/0.0;
```

```
        interface ge-4/2/0.0;
    }
    connections {
        interface-switch vlan-tag-push {
            interface ge-4/0/0.0;
            interface ge-4/2/0.0;
        }
    }
}
```

Stack Inner and Outer VLAN Tags

```
[edit interfaces]
ge-0/2/0 {
    stacked-vlan-tagging;
    mac 00.01.02.03.04.05;
    gigether-options {
        loopback;
    }
    unit 0 {
        vlan-tags outer 0x8100.200 inner 0x8100.200;
    }
}
```

Swap a VLAN ID on Ingress

```
[edit interfaces]
ge-4/0/0 {
    vlan-tagging;
    encapsulation vlan-ccc;
    gigether-options {
        ethernet-switch-profile {
            tag-protocol-id 0x9100;
        }
    }
    ...
    unit 1 {
```

```
        encapsulation vlan-ccc;
        vlan-id 1000;
        input-vlan-map {
            swap;
            tag-protocol-id 0x9100;
            vlan-id 2000;
        }
    }
}
ge-4/2/0 {
    vlan-tagging;
    encapsulation vlan-ccc;
    ...
    unit 1 {
        encapsulation vlan-ccc;
        vlan-id 2000;
        input-vlan-map {
            swap;
            tag-protocol-id 0x9100;
            vlan-id 1000;
        }
    }
    [edit protocols]
    mpls {
        ...
        interface ge-4/0/0.1;
        interface ge-4/2/0.1;
    }
    connections {
        ...
        interface-switch vlan-tag-swap {
            interface ge-4/2/0.1;
            interface ge-4/0/0.1;
        }
    }
}
```

Swap a VLAN ID on Egress

```
[edit interfaces]
```

```
ge-4/0/0 {
    vlan-tagging;
    encapsulation vlan-ccc;
    ...
    unit 1 {
        encapsulation vlan-ccc;
        vlan-id 1000;
    }
}
ge-4/2/0 {
    vlan-tagging;
    encapsulation vlan-ccc;
    gigether-options {
        ethernet-switch-profile {
            tag-protocol-id 0x8800;
        }
    }
    ...
    unit 1 {
        encapsulation vlan-ccc;
        vlan-id 2000;
        output-vlan-map {
            swap;
            tag-protocol-id 0x8800;
        }
    }
}
[edit protocols]
mpls {
    ...
    interface ge-4/0/0.1;
    interface ge-4/2/0.1;
}
connections {
    ...
    interface-switch vlan-tag-swap {
        interface ge-4/2/0.1;
        interface ge-4/0/0.1;
    }
}
```

Swap a VLAN ID on Both Ingress and Egress

```
[edit interfaces]
ge-4/0/0 {
    vlan-tagging;
    encapsulation vlan-ccc;
    gigether-options {
        ethernet-switch-profile {
            tag-protocol-id [ 0x8800 0x9100 ];
        }
    }
    ...
    unit 1 {
        encapsulation vlan-ccc;
        vlan-id 1000;
        input-vlan-map {
            swap;
            tag-protocol-id 0x9100;
            vlan-id 2000;
        }
    }
}
ge-4/2/0 {
    vlan-tagging;
    encapsulation vlan-ccc;
    gigether-options {
        ethernet-switch-profile {
            tag-protocol-id [ 0x8800 0x9100 ];
        }
    }
    unit 1 {
        encapsulation vlan-ccc;
        vlan-id 2000;
        output-vlan-map {
            swap;
            tag-protocol-id 0x8800;
        }
    }
}
[edit protocols]
mpls {
    ...
    interface ge-4/0/0.1;
    interface ge-4/2/0.1;
```

```
}
connections {
    ...
    interface-switch vlan-tag-swap {
        interface ge-4/2/0.1;
        interface ge-4/0/0.1;
    }
}
```

Swap the Outer VLAN Tag and Push a New VLAN Tag on Ingress; Pop the Outer VLAN Tag and Swap the Inner VLAN Tag on Egress

```
[edit interfaces]
ge-1/1/0 {
    unit 1 {
        vlan-id 200;
        input-vlan-map {
            swap-push;
            tag-protocol-id 0x9100;
            vlan-id 400;
            inner-tag-protocol-id 0x9100;
            inner-vlan-id 500;
        }
        output-vlan-map {
            pop-swap;
            inner-tag-protocol-id 0x9100;
        }
    }
}
```

Swap a TPID and VLAN ID Pair for Both VLAN Tags on Ingress and on Egress

```
[edit interfaces]
ge-1/1/0 {
    unit 0 {
        vlan-tags {
            inner 0x9100.425;
```

```
            outer 0x9200.525;
        }
        input-vlan-map {
            swap-swap;
            tag-protocol-id 0x9100;
            vlan-id 400;
            inner-tag-protocol-id 0x9100;
            inner-vlan-id 500;
        }
        output-vlan-map {
            swap-swap;
            tag-protocol-id 0x9200;
            inner-tag-protocol-id 0x9100;
        }
    }
}
```

Pop the Outer VLAN Tag and Swap the Inner VLAN Tag on Ingress; Swap the Outer VLAN Tag and Push a New VLAN Tag on Egress

```
[edit interfaces]
ge-1/1/0 {
    unit 0 {
        vlan-tags {
            inner 0x9100.425;
            outer 0x9200.525;
        }
        input-vlan-map {
            pop-swap;
            tag-protocol-id 0x9100;
            vlan-id 400;
        }
        output-vlan-map{
            swap-push;
            tag-protocol-id 0x9200;
            inner-tag-protocol-id 0x9100;
        }
    }
}
```

Pop a TPID and VLAN ID Pair on Ingress; Push a VLAN ID and TPID Pair on Egress

```
[edit interfaces]
ge-1/1/0 {
    unit 0 {
        vlan-tags {
            inner 0x9100.425;
            outer 0x9200.525;
        }
        input-vlan-map{
            pop-pop;
        }
        output-vlan-map {
            push-push;
            tag-protocol-id 0x9200;
            inner-tag-protocol-id 0x9100;
        }
    }
}
```

Pop an Outer VLAN Tag to Connect an Untagged VPLS Interface to Tagged VPLS Interfaces

```
[edit interfaces]
ge-1/1/0 {
    vlan-tagging;
    encapsulation extended-vlan-vpls;
    unit 0 {
        vlan-id 0;
        input-vlan-map {
            push;
            vlan-id 0;
        }
        output-vlan-map pop;
        family vpls;
    }
}
```

RELATED DOCUMENTATION

input-vlan-map | 1195

output-vlan-map | 1294

inner-tag-protocol-id | 1192

inner-vlan-id | 1193

pop | 1303

pop-pop | 1304

pop-swap | 1305

push | 1332

push-push | 1333

swap | 1354

swap-push | 1356

swap-swap | 1357

unit

Ethernet Interfaces User Guide for Routing Devices

Understanding Transparent Tag Operations and IEEE 802.1p Inheritance

When **swap-by-poppush** is configured on IQ2 interfaces, 10-Gigabit Ethernet LAN/WAN PIC, IQ2-E interfaces, and MX Series interfaces, during a swap operation, the inner VLAN IEEE 802.1p bits are copied to the IEEE bits of the tag being swapped. If swap-by-poppush is not configured on the interface, the VLAN IEEE 802.1p bits of the tag being swapped remains same.

When **swap-by-poppush** is configured but the incoming packet has no inner VLAN tag (transparent tag), the IEEE 802.1p bits are set to zero .

Table 75 on page 415 describes the relationship between the VLAN map operation and the inheritance of IEEE 802.1p from the transparent tag. It assumes the presence of the transparent tag in the incoming packet. If the transparent tag is not present, the IEEE 802.1p value is set to 0.

Table 75: VLAN Map Operation and IEEE 802.1p Inheritance

Rewrite Operation	Untagged Logical Interface	Transparent tag IEEE 802.1p Inheritance	Single-tagged Logical Interface	Transparent tag IEEE 802.1p Inheritance	Change in number of tags
push-push	yes	OUTER, INNER	NA	no operation	+2
swap-push	NA	no operation	yes	OUTER, INNER	+1
push	yes	OUTER	yes	*none	+1
swap	NA	NA	yes	OUTER	0

NOTE: *In a **push** operation on a single-tagged logical interface, none of the tags (inner, or outer) inherit the IEEE 802.1p bits from the transparent tag.

The following section shows four different examples of the inheritance of the transparent IEEE 802.1p values into the outer and inner VLAN tags.

Figure 4 on page 415 shows an incoming packet with a transparent tag. A swap-push operation swaps the outer VLAN tag and pushes another VLAN tag. The IEEE 802.1p values are inherited from the transparent tag.

Figure 4: swap-push (transparent tag)

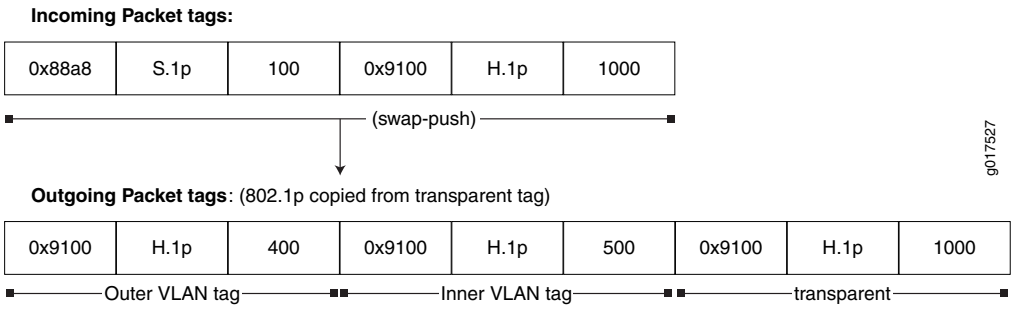


Figure 5 on page 416 shows an incoming packet with no transparent tag. A swap-push operation swaps the outer VLAN tag and pushes another VLAN tag. The IEEE 802.1p value is set to zero, as there is no transparent tag.

Figure 5: swap-push (no transparent tag)

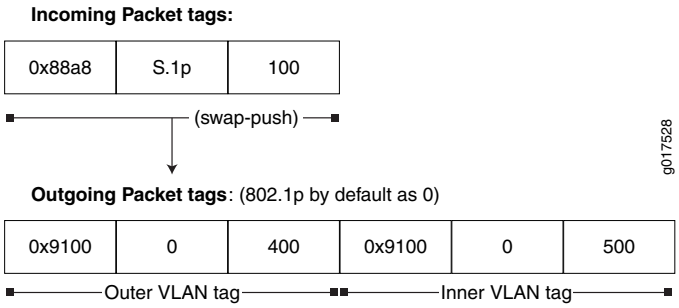


Figure 6 on page 416 shows an incoming packet with a transparent tag. A push operation pushes another VLAN tag. The IEEE 802.1p value is inherited from the transparent tag.

Figure 6: push (transparent tag)

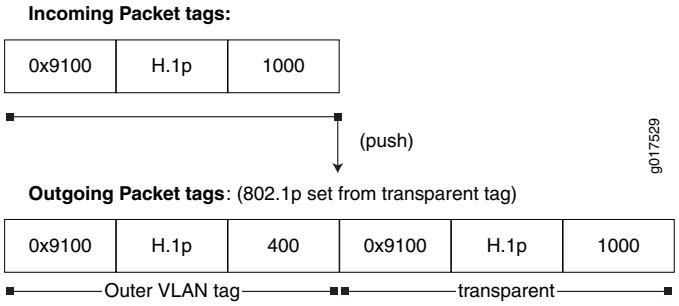
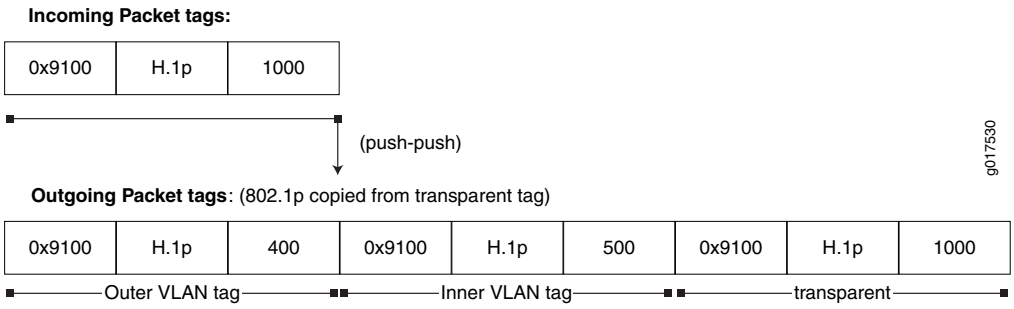


Figure 7 on page 416 shows an incoming packet with a transparent tag. A push-push operation pushes the outer and inner VLAN tags, respectively. The IEEE 802.1p values are inherited from the transparent tag.

Figure 7: push-push (transparent tag)



RELATED DOCUMENTATION

Understanding IEEE 802.1p Inheritance push and swap from a Transparent Tag

Configuring IEEE 802.1p Inheritance push and swap from the Transparent Tag | **417**

Understanding swap-by-poppush

By default, during a swap operation, the IEEE 802.1p bits of the VLAN tag remain unchanged. When the **swap-by-poppush** operation is enabled on a logical interface, the swap operation is treated as a **pop** operation followed by **push** operation. The **pop** operation removes the existing tag and the associated IEEE 802.1p bits and the push operation copies the inner VLAN IEEE 802.1p bits to the IEEE bits of the VLAN or VLANs being pushed. As a result, the IEEE 802.1p bits are inherited from the incoming transparent tag.

In effect, **swap-by-poppush** serves as a VLAN operation property and is used along with a **swap** or **swap-push** VLAN rewrite operation, indicating the nature of the swap operation being performed.

RELATED DOCUMENTATION

Configuring IEEE 802.1p Inheritance push and swap from the Transparent Tag

To classify incoming packets based on the IEEE 802.1p bits from the transparent tag, include the **transparent** statement at the **[edit class-of-service interfaces** *interface-name* **unit** *logical-unit-number* **classifiers ieee-802.1 vlan-tag]** hierarchy level.

Tagged Interface Example

The following example configuration specifies the classification based on the transparent VLAN tag.

```
edit
class-of-service {
   interfaces {
      ge-3/0/1 {
         unit 0 {
            classifiers {
               ieee-802.1 default vlan-tag transparent;
            }
         }
      }
   }
}
```

To configure Junos OS to inherit the IEEE 802.1p bits from the transparent tag, include the **swap-by-poppush** statement at the **[edit interfaces *interface-name* unit *logical-unit-number*]** hierarchy level.

The following is a configuration to swap and push VLAN tags and allow inheritance of the IEEE 802.1p value from the transparent VLAN tag in incoming packets.

```
edit
   ge-3/0/0 {
   vlan-tagging;
   encapsulation vlan-ccc;
   unit 0 {
      encapsulation vlan-ccc;
      vlan-id 100;
      swap-by-poppush;
      input-vlan-map {
         swap-push;
         tag-protocol-id 0x9100;
         inner-tag-protocol-id 0x9100;
         vlan-id 500;
         inner-vlan-id 400;
      }
      output-vlan-map {
         pop-swap;
         inner-vlan-id 100;
         inner-tag-protocol-id 0x88a8;
      }
```

```
    }
}
```

The **swap-by-poppush** statement causes a swap operation to be done as a pop followed by a push operation. So for the outer tag, the incoming S-Tag is popped and a new tag is pushed. As a result, the S-Tag inherits the IEEE 802.1p bits from the transparent tag. The inner tag is then pushed, which results in the inner tag inheriting the IEEE 802.1p bits from the transparent tag.

Untagged Interface Example

The following is a configuration to push two VLAN tags and allow inheritance of the IEEE 802.1p value from the transparent VLAN tag in the incoming packet.

```
[edit]
ge-3/0/1 {
   encapsulation ccc;
   unit 0 {
      input-vlan-map {
         push-push;
         tag-protocol-id 0x9100;
         inner-tag-protocol-id 0x9100;
         vlan-id 500;
         inner-vlan-id 400;
      }
      output-vlan-map{
         pop-pop;
      }
   }
}
```

No additional configuration is required to inherit the IEEE 802.1p value, as the **push** operation inherits the IEEE 802.1p values by default.

The following configuration specifies the classification based on the transparent VLAN tag.

```
[edit]
class-of-service {
   interfaces {
      ge-3/0/1 {
         unit 0 {
            classifiers {
```

```
                ieee-802.1 default vlan-tag transparent;
            }
        }
    }
}
}
```

RELATED DOCUMENTATION

transparent

swap-by-poppush | **1355**

Understanding IEEE 802.1p Inheritance push and swap from a Transparent Tag

Understanding swap-by-poppush | **417**

Understanding Transparent Tag Operations and IEEE 802.1p Inheritance | **414**

17

CHAPTER

Configuring Private VLANs

Private VLANs

IN THIS SECTION

Understanding Private VLANs

IN THIS SECTION

VLANs limit broadcasts to specified users. Private VLANs (PVLANs) take this concept a step further by limiting communication within a VLAN. PVLANs accomplish this by restricting traffic flows through their member switch ports (which are called *private ports*) so that these ports communicate only with a specified uplink trunk port or with specified ports within the same VLAN. The uplink trunk port or link aggregation group (LAG) is usually connected to a router, firewall, server, or provider network. Each PVLAN typically contains many private ports that communicate only with a single uplink port, thereby preventing the ports from communicating with each other.

PVLANs provide Layer 2 isolation between ports within a VLAN, splitting a broadcast domain into multiple discrete broadcast subdomains by creating secondary VLANs (community VLANs and an isolated VLAN) inside a primary VLAN. Ports within the same community VLAN can communicate with each other. Ports within an isolated VLAN can communicate *only* with a single uplink port.

Just like regular VLANs, PVLANs are isolated on Layer 2 and require one of the following options to route Layer 3 traffic among the secondary VLANs:

- A promiscuous port connection with a router
- A routed VLAN interface (RVI)

> **NOTE:** To route Layer 3 traffic among secondary VLANs, a PVLAN needs only one of the options mentioned above. If you use an RVI, you can still implement a promiscuous port connection to a router with the promiscuous port set up to handle only traffic that enters and exits the PVLAN.

PVLANs are useful for restricting the flow of broadcast and unknown unicast traffic and for limiting the communication between known hosts. Service providers use PVLANs to keep their customers isolated from each other. Another typical use for a PVLAN is to provide per-room Internet access in a hotel.

NOTE: You can configure a PVLAN to span switches that support PVLANs.

This topic explains the following concepts regarding PVLANs on EX Series switches:

Benefits of PVLANs

The need to segregate a single VLAN is particularly useful in the following deployment scenarios:

- Server farms—A typical Internet service provider uses a server farm to provide Web hosting for numerous customers. Locating the various servers within a single server farm provides ease of management. Security concerns arise if all servers are in the same VLAN because Layer 2 broadcasts go to all servers in the VLAN.
- Metropolitan Ethernet networks—A metro service provider offers Layer 2 Ethernet access to assorted homes, rental communities, and businesses. The traditional solution of deploying one VLAN per customer is not scalable and is difficult to manage, leading to potential waste of IP addresses. PVLANs provide a more secure and more efficient solution.

Typical Structure and Primary Application of PVLANs

A PVLAN can be configured on a single switch or can be configured to span multiple switches. The types of domains and ports are:

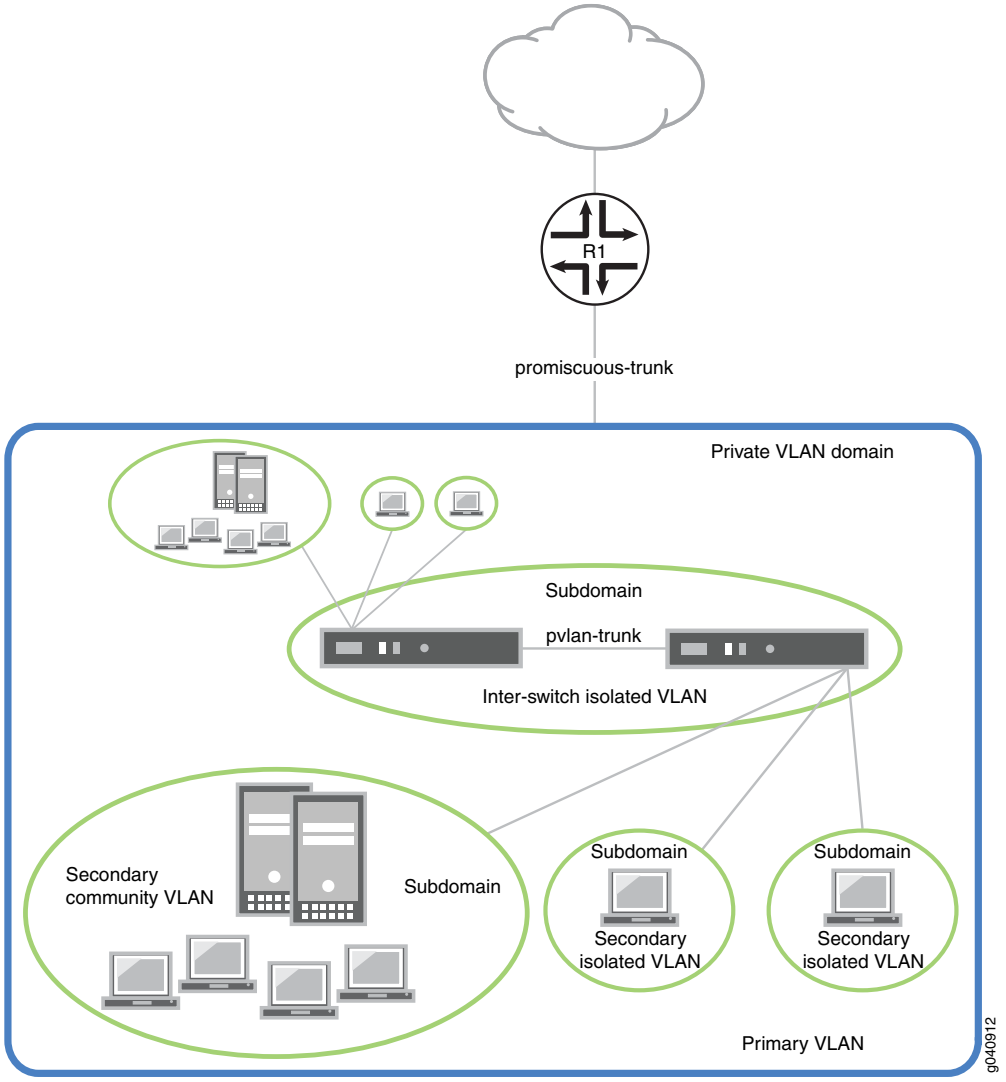
- Primary VLAN—The primary VLAN of the PVLAN is defined with an 802.1Q tag (VLAN ID) for the complete PVLAN. The primary PVLAN can contain multiple secondary VLANs (one isolated VLAN and multiple community VLANs).
- Isolated VLAN/isolated port—A primary VLAN can contain only one isolated VLAN. An interface within an isolated VLAN can forward packets only to a promiscuous port or the Inter-Switch Link (ISL) port. An isolated interface cannot forward packets to another isolated interface; and an isolated interface cannot receive packets from another isolated interface. If a customer device needs to have access *only* to a gateway router, the device must be attached to an isolated trunk port.
- Community VLAN/community port—You can configure multiple community VLANs within a single PVLAN. An interface within a specific community VLAN can establish Layer 2 communications with any other interface that belongs to the same community VLAN. An interface within a community VLAN can also communicate with a promiscuous port or the ISL port. If you have, for example, two customer devices that you need to isolate from other customer devices but that must be able to communicate with one another, use community ports.
- Promiscuous port—A promiscuous port has Layer 2 communications with all interfaces in the PVLAN, regardless of whether an interface belongs to an isolated VLAN or a community VLAN. A promiscuous port is a member of the primary VLAN but is not included within any secondary subdomain. Layer 3

gateways, DHCP servers, and other trusted devices that need to communicate with endpoint devices are typically connected to a promiscuous port.

- Inter-Switch Link (ISL)—An ISL is a trunk port that connects multiple switches in a PVLAN and contains two or more VLANs. It is required only when a PVLAN spans multiple switches.

The configured PVLAN is the *primary* domain (primary VLAN). Within the PVLAN, you configure *secondary* VLANs, which become subdomains nested within the primary domain. A PVLAN can be configured on a single switch or can be configured to span multiple switches. The PVLAN shown in Figure 8 on page 426 includes two switches, with a primary PVLAN domain and various subdomains.

Figure 8: Subdomains in a PVLAN

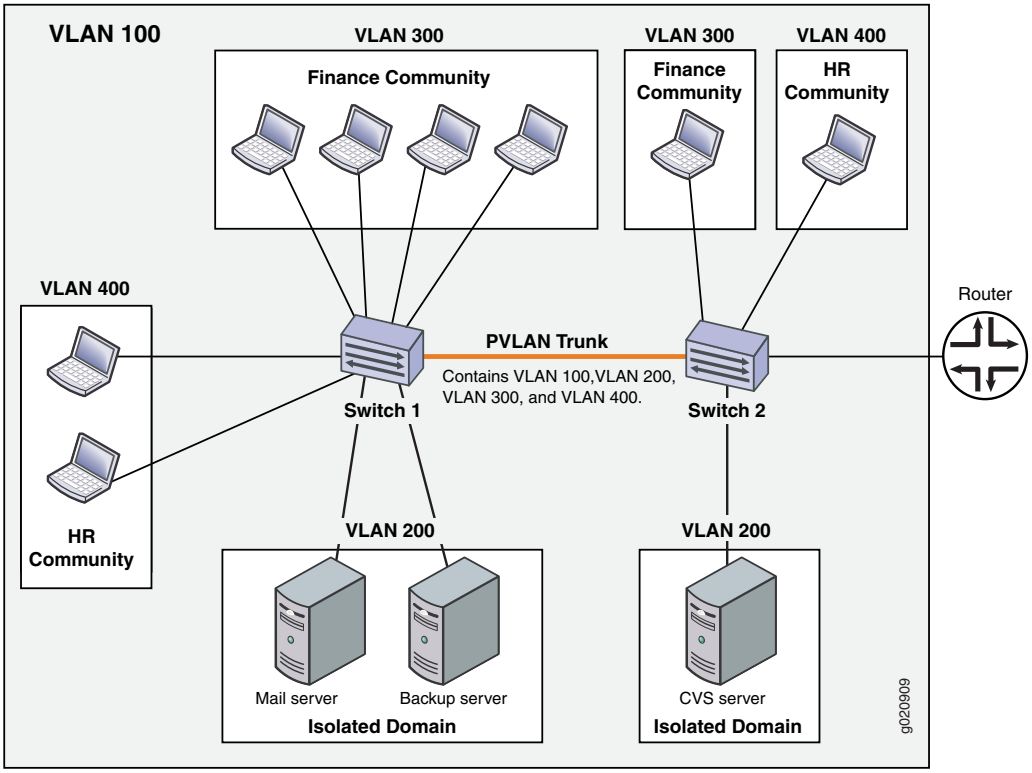


As shown in Figure 10 on page 429, a PVLAN has only one primary domain and multiple secondary domains. The types of domains are:

- Primary VLAN—VLAN used to forward frames downstream to isolated and community VLANs. The primary VLAN of the PVLAN is defined with an 802.1Q tag (VLAN ID) for the complete PVLAN. The primary PVLAN can contain multiple secondary VLANs (one isolated VLAN and multiple community VLANs).
- Secondary isolated VLAN—VLAN that receives packets only from the primary VLAN and forwards frames upstream to the primary VLAN. The isolated VLAN is a secondary VLAN nested within the primary VLAN. A primary VLAN can contain only one isolated VLAN. An interface within an isolated VLAN (isolated interface) can forward packets only to a promiscuous port or the PVLAN trunk port. An isolated interface cannot forward packets to another isolated interface; nor can an isolated interface receive packets from another isolated interface. If a customer device needs to have access *only* to a router, the device must be attached to an isolated trunk port.
- Secondary interswitch isolated VLAN—VLAN used to forward isolated VLAN traffic from one switch to another through PVLAN trunk ports. 802.1Q tags are required for interswitch isolated VLANs because IEEE 802.1Q uses an internal tagging mechanism by which a trunking device inserts a 4-byte VLAN frame identification tab into the packet header. An interswitch isolated VLAN is a secondary VLAN nested within the primary VLAN.
- Secondary community VLAN—VLAN used to transport frames among members of a community (a subset of users within the VLAN) and to forward frames upstream to the primary VLAN. A community VLAN is a secondary VLAN nested within the primary VLAN. You can configure multiple community VLANs within a single PVLAN. An interface within a specific community VLAN can establish Layer 2 communications with any other interface that belongs to the same community VLAN. An interface within a community VLAN can also communicate with a promiscuous port or the PVLAN trunk port.

Figure 9 on page 428 shows a PVLAN spanning multiple switches, where the primary VLAN (**100**) contains two community domains (**300** and **400**) and one interswitch isolated domain.

Figure 9: PVLAN Spanning Multiple Switches

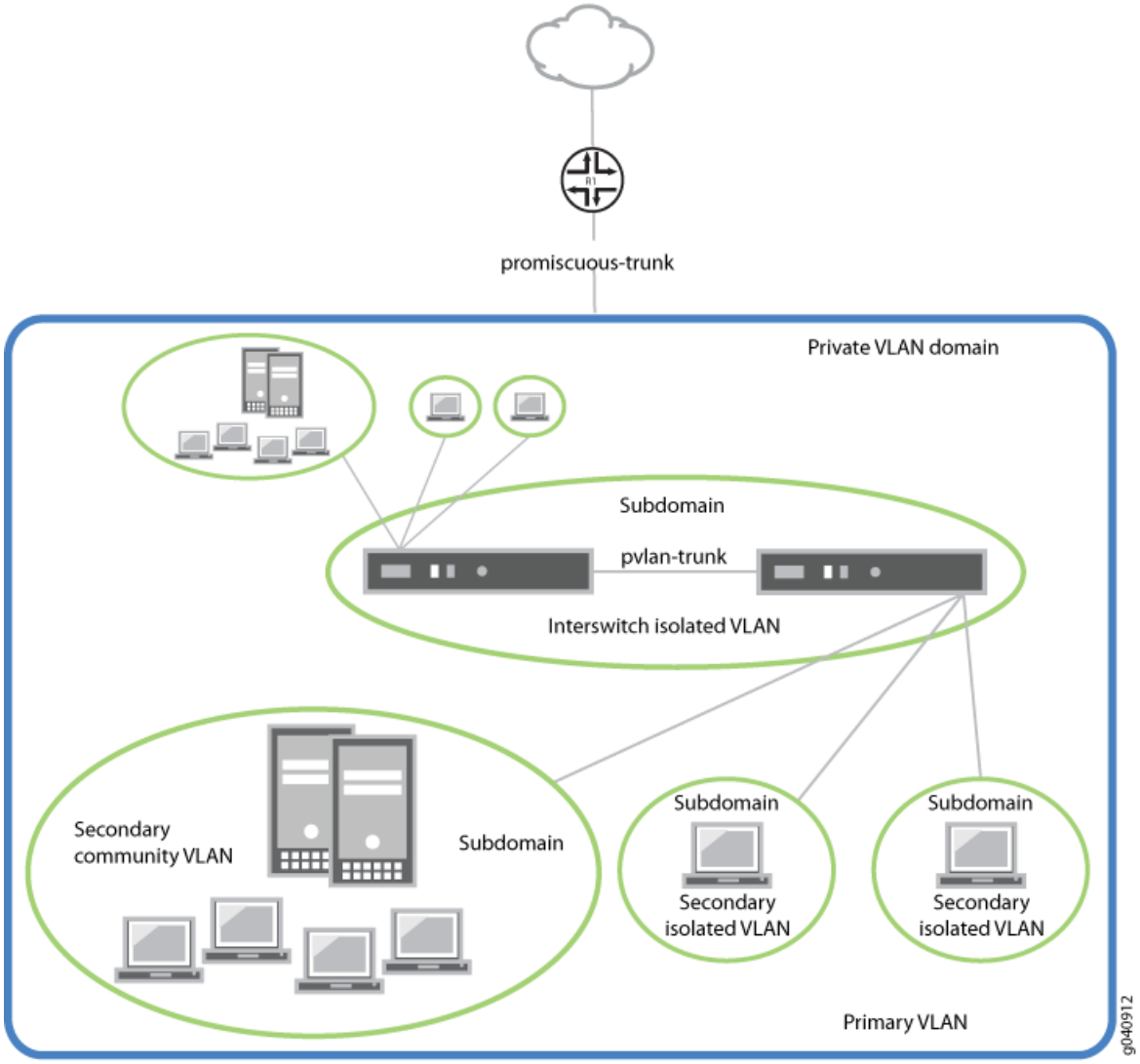


NOTE: Primary and secondary VLANs count against the limit of 4089 VLANs supported on the QFX Series. For example, each VLAN in Figure 9 on page 428 counts against this limit.

Typical Structure and Primary Application of PVLANs on MX Series Routers

The configured PVLAN becomes the primary domain, and secondary VLANs become subdomains that are nested inside the primary domain. A PVLAN can be created on a single router. The PVLAN shown in Figure 10 on page 429 includes one router, with one primary PVLAN domain and multiple secondary subdomains.

Figure 10: Subdomains in a PVLAN With One Router



The types of domains are:

- Primary VLAN—VLAN used to forward frames downstream to isolated and community VLANs.
- Secondary isolated VLAN—VLAN that receives packets only from the primary VLAN and forwards frames upstream to the primary VLAN.
- Secondary interswitch isolated VLAN—VLAN used to forward isolated VLAN traffic from one router to another through PVLAN trunk ports.
- Secondary community VLAN—VLAN used to transport frames among members of a community, which is a subset of users within the VLAN, and to forward frames upstream to the primary VLAN.

NOTE: PVLANs are supported on MX80 routers, on MX240, MX480, and MX960 routers with DPCs in enhanced LAN mode, on MX Series routers with MPC1, MPC2, and Adaptive Services PICs.

Typical Structure and Primary Application of PVLANs on EX Series Switches

NOTE: The primary VLAN of the PVLAN is defined with an 802.1Q tag (VLAN ID) for the complete PVLAN. On EX9200 switches, each secondary VLAN must also be defined with its own separate VLAN ID.

Figure 11 on page 430 shows a PVLAN on a single switch, where the primary VLAN (VLAN **100**) contains two community VLANs (VLAN **300** and VLAN **400**) and one isolated VLAN (VLAN **50**).

Figure 11: Private VLAN on a Single EX Switch

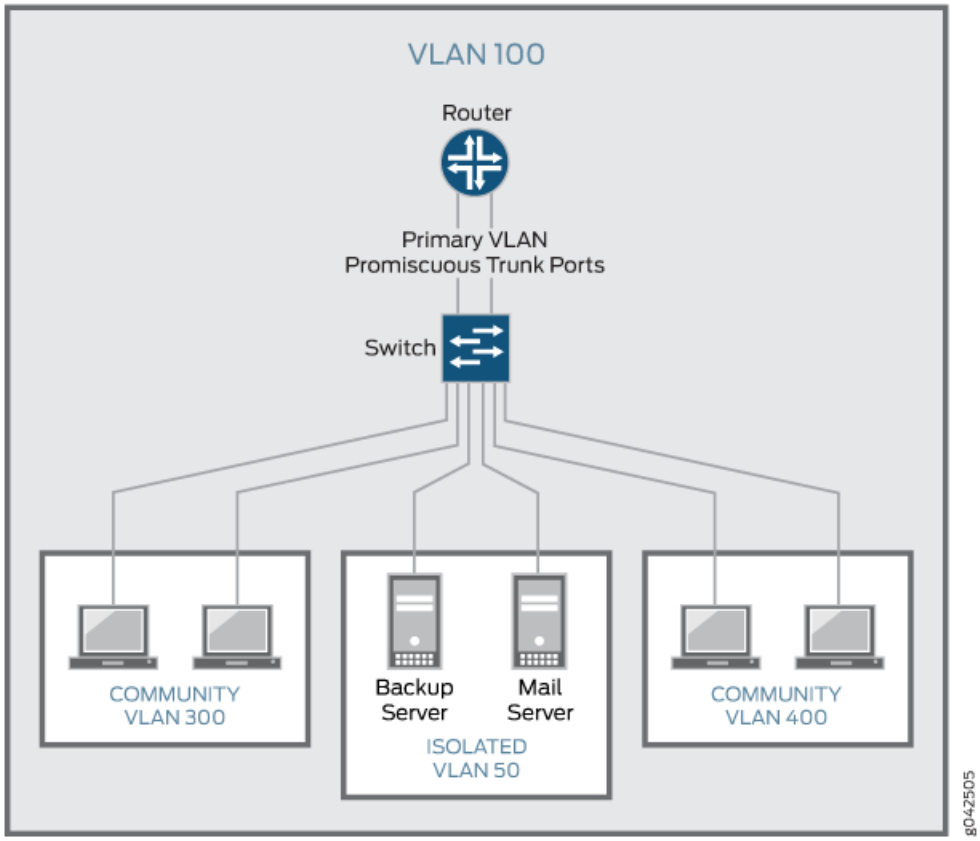
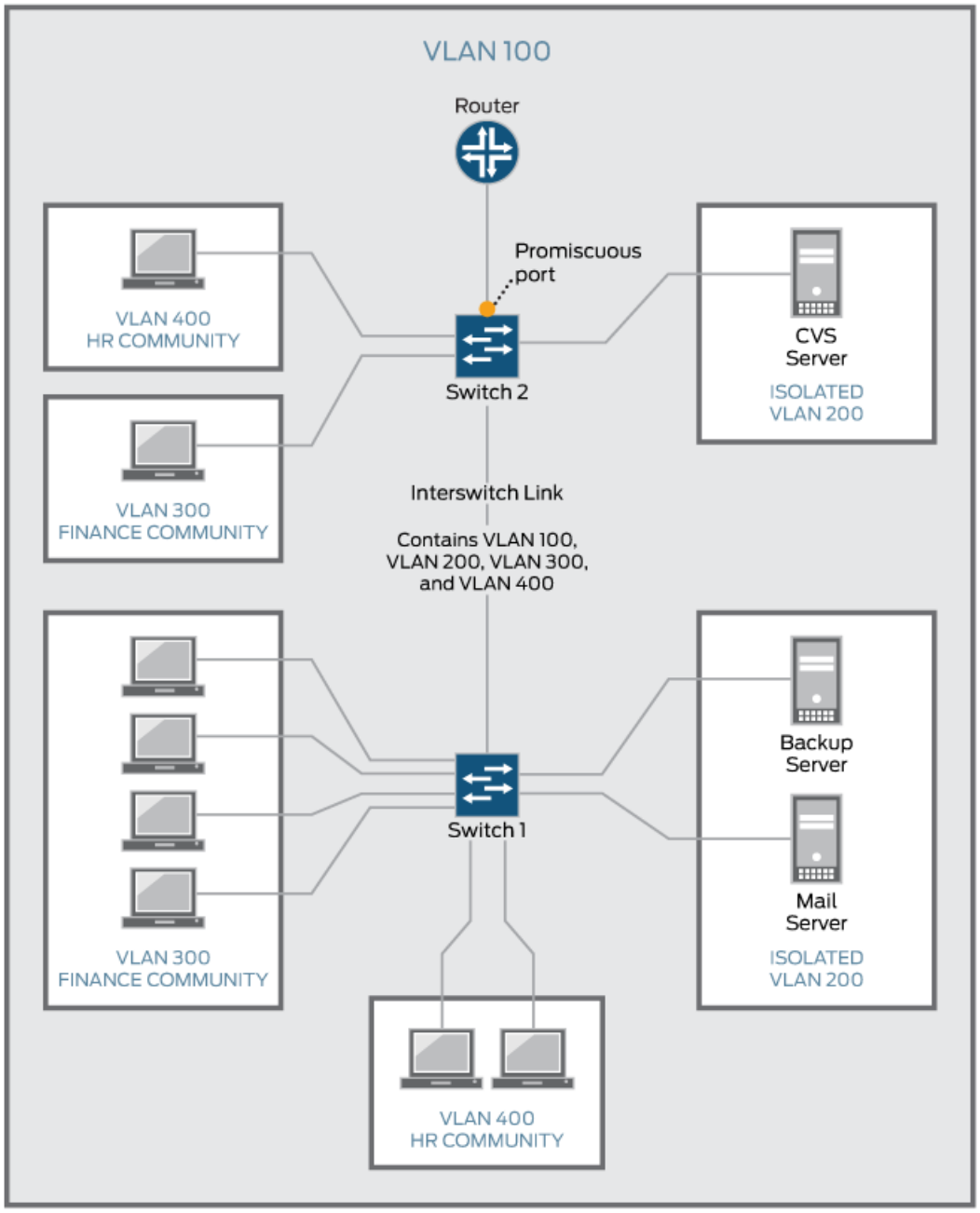


Figure 12 on page 431 shows a PVLAN spanning multiple switches, where the primary VLAN (VLAN **100**) contains two community VLANs (VLAN **300** and VLAN **400**) and one isolated VLAN (VLAN 200). It also shows that Switches 1 and 2 are connected through an interswitch link (PVLAN trunk link).

Figure 12: PVLAN Spanning Multiple EX Series Switches



Also, the PVLANs shown in Figure 11 on page 430 and Figure 12 on page 431 use a promiscuous port connected to a router as the means to route Layer 3 traffic among the community and isolated VLANs.

Instead of using the promiscuous port connected to a router, you can configure an RVI on the switch in Figure 11 on page 430 or one of the switches shown in Figure 12 on page 431 (on some EX switches).

To route Layer 3 traffic between isolated and community VLANs, you must either connect a router to a promiscuous port, as shown in Figure 11 on page 430 and Figure 12 on page 431, or configure an RVI.

If you choose the RVI option, you must configure one RVI for the primary VLAN in the PVLAN domain. This RVI serves the entire PVLAN domain regardless of whether the domain includes one or more switches. After you configure the RVI, Layer 3 packets received by the secondary VLAN interfaces are mapped to and routed by the RVI.

When setting up the RVI, you must also enable proxy Address Resolution Protocol (ARP) so that the RVI can handle ARP requests received by the secondary VLAN interfaces.

For information about configuring PVLANs on a single switch and on multiple switches, see "Creating a Private VLAN on a Single EX Series Switch (CLI Procedure)" on page 477. For information about configuring an RVI, see "Configuring a Routed VLAN Interface in a Private VLAN on an EX Series Switch" on page 723.

Routing Between Isolated and Community VLANs

To route Layer 3 traffic between isolated and community VLANs, you must connect an external router or switch to a trunk port of the primary VLAN. The trunk port of the primary VLAN is a *promiscuous* port; therefore, it can communicate with *all* the ports in the PVLAN.

PVLANs Use 802.1Q Tags to Identify Packets

When packets are marked with a customer-specific 802.1Q tag, that tag identifies ownership of the packets for any switch or router in the network. Sometimes, 802.1Q tags are needed within PVLANs to keep track of packets from different subdomains. Table 76 on page 432 indicates when a VLAN 802.1Q tag is needed on the primary VLAN or on secondary VLANs.

Table 76: When VLANs in a PVLAN Need 802.1Q Tags

	On a Single Switch	On Multiple Switches
Primary VLAN	Specify an 802.1Q tag by setting a VLAN ID.	Specify an 802.1Q tag by setting a VLAN ID.
Secondary VLAN	No tag needed on VLANs.	VLANs need 802.1Q tags: • Specify an 802.1Q tag for each community VLAN by setting a VLAN ID. • Specify the 802.1Q tag for an isolation VLAN ID by setting an isolation ID.

PVLANs Use IP Addresses Efficiently

PVLANs provide IP address conservation and efficient allocation of IP addresses. In a typical network, VLANs usually correspond to a single IP subnet. In PVLANs, the hosts in all secondary VLANs belong to the same IP subnet because the subnet is allocated to the primary VLAN. Hosts within the secondary VLAN are assigned IP addresses based on IP subnets associated with the primary VLAN, and their IP subnet masking information reflects that of the primary VLAN subnet. However, each secondary VLAN is a separate broadcast domain.

PVLAN Port Types and Forwarding Rules

PVLANs can use up to six different port types. The network depicted inFigure 9 on page 428 uses a promiscuous port to transport information to the router, community ports to connect the finance and HR communities to their respective switches, isolated ports to connect the servers, and a PVLAN trunk port to connect the two switches. PVLAN ports have different restrictions:

- Promiscuous trunk port—A promiscuous port has Layer 2 communications with all the interfaces that are in the PVLAN, regardless of whether the interface belongs to an isolated VLAN or a community VLAN. A promiscuous port is a member of the primary VLAN, but is not included within one of the secondary subdomains. Layer 3 gateways, DHCP servers, and other trusted devices that need to communicate with endpoint devices are typically connected to a promiscuous port.
- PVLAN trunk link—The PVLAN trunk link, which is also known as the interswitch link, is required only when a PVLAN is configured to span multiple switches. The PVLAN trunk link connects the multiple switches that compose the PVLAN.
- PVLAN trunk port—A PVLAN trunk port is required in multiswitch PVLAN configurations to span the switches. The PVLAN trunk port is a member of all VLANs within the PVLAN (that is, the primary VLAN, the community VLANs, and the interswitch isolated VLAN), and it carries traffic from the primary VLAN and all secondary VLANs. It can communicate with all ports other than the isolated ports.

  Communication between a PVLAN trunk port and an isolated port is usually unidirectional. A PVLAN trunk port's membership in the interswitch isolated VLAN is egress-only, meaning that an isolated port can forward packets to a PVLAN trunk port, but a PVLAN trunk port does not forward packets to an isolated port (unless the packets ingressed on a promiscuous access port and are therefore being forwarded to all the secondary VLANs in the same primary VLAN as the promiscuous port).
- Secondary VLAN trunk port (not shown)—Secondary trunk ports carry secondary VLAN traffic. For a given private VLAN, a secondary VLAN trunk port can carry traffic for only one secondary VLAN. However, a secondary VLAN trunk port can carry traffic for multiple secondary VLANs as long as each secondary VLAN is a member of a different primary VLAN. For example, a secondary VLAN trunk port can carry traffic for a community VLAN that is part of primary VLAN pvlan100 and also carry traffic for an isolated VLAN that is part of primary VLAN pvlan400.

- Community port—Community ports communicate among themselves and with their promiscuous ports. Community ports serve only a select group of users. These interfaces are separated at Layer 2 from all other interfaces in other communities or isolated ports within their PVLAN.
- Isolated access port—Isolated ports have Layer 2 connectivity only with promiscuous ports and PVLAN trunk ports—an isolated port cannot communicate with another isolated port even if these two ports are members of the same isolated VLAN (or interswitch isolated VLAN) domain. Typically, a server, such as a mail server or a backup server, is connected on an isolated port. In a hotel, each room would typically be connected on an isolated port, meaning that room-to-room communication is not possible, but each room can access the Internet on the promiscuous port.
- Promiscuous access port (not shown)—These ports carry untagged traffic. Traffic that ingresses on a promiscuous access port is forwarded to all secondary VLAN ports on the device. If traffic ingresses into the device on a VLAN-enabled port and egresses on a promiscuous access port, the traffic is untagged on egress. If tagged traffic ingresses on a promiscuous access port, the traffic is discarded.
- Interswitch link port—An interswitch link (ISL) port is a trunk port that connects two routers when a PVLAN spans those routers. The ISL port is a member of all VLANs within the PVLAN (that is, the primary VLAN, the community VLANs, and the isolated VLAN).

  Communication between an ISL port and an isolated port is unidirectional. An ISL port's membership in the interswitch isolated VLAN is egress-only, meaning that incoming traffic on the ISL port is never assigned to the isolated VLAN. An isolated port can forward packets to a PVLAN trunk port, but a PVLAN trunk port cannot forward packets to an isolated port. Table 78 on page 434 summarizes whether Layer 2 connectivity exists between the different types of ports.

Table 77 on page 434 summarizes Layer 2 connectivity between the different types of ports within a PVLAN on EX Series switches that support ELS.

Table 77: PVLAN Ports and Layer 2 Forwarding on EX Series switches that support ELS

From Port Type	To Isolated Ports?	To Promiscuous Ports?	To Community Ports?	To Inter-Switch Link Port?
Isolated	Deny	Permit	Deny	Permit
Promiscuous	Permit	Permit	Permit	Permit
Community 1	Deny	Permit	Permit	Permit

Table 78: PVLAN Ports and Layer 2 Connectivity

Port Type	Promiscuous Trunk	PVLAN Trunk	Secondary Trunk	Community	Isolated Access	Promiscuous access
Promiscuous trunk	Yes	Yes	Yes	Yes	Yes	Yes

Table 78: PVLAN Ports and Layer 2 Connectivity *(continued)*

Port Type	Promiscuous Trunk	PVLAN Trunk	Secondary Trunk	Community	Isolated Access	Promiscuous access
PVLAN trunk	Yes	Yes	Yes	Yes—same community only	Yes	Yes
Secondary Trunk	Yes	Yes	No	Yes	No	Yes
Community	Yes	Yes	Yes	Yes—same community only	No	Yes
Isolated access	Yes	Yes—unidirectional only	No	No	No	Yes
Promiscuous access	Yes	Yes	Yes	Yes	Yes	No

Table 79 on page 435 summarizes whether or not Layer 2 connectivity exists between the different types of ports within a PVLAN.

Table 79: PVLAN Ports and Layer 2 Connectivity on EX Series Switches without ELS Support

Port Type To: → From:↓	Promiscuous	Community	Isolated	PVLAN Trunk	RVI
Promiscuous	Yes	Yes	Yes	Yes	Yes
Community	Yes	Yes—same community only	No	Yes	Yes
Isolated	Yes	No	No	Yes **NOTE:** This communication is unidirectional.	Yes

Table 79: PVLAN Ports and Layer 2 Connectivity on EX Series Switches without ELS Support *(continued)*

Port Type To: → From:↓	Promiscuous	Community	Isolated	PVLAN Trunk	RVI
PVLAN trunk	Yes	Yes—same community only	Yes **NOTE:** This communication is unidirectional.	Yes	Yes
RVI	Yes	Yes	Yes	Yes	Yes

As noted in Table 79 on page 435, Layer 2 communication between an isolated port and a PVLAN trunk port is unidirectional. That is, an isolated port can only send packets to a PVLAN trunk port, and a PVLAN trunk port can only receive packets from an isolated port. Conversely, a PVLAN trunk port cannot send packets to an isolated port, and an isolated port cannot receive packets from a PVLAN trunk port.

NOTE: If you enable **no-mac-learning** on a primary VLAN, all isolated VLANs (or the interswitch isolated VLAN) in the PVLAN inherit that setting. However, if you want to disable MAC address learning on any community VLANs, you must configure **no-mac-learning** on each of those VLANs.

Creating a PVLAN

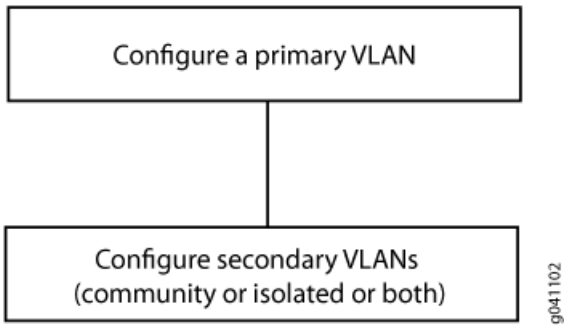
The flowchart shown in Figure 13 on page 437 gives you a general idea of the process for creating PVLANs. If you complete your configuration steps in the order shown, you will not violate these PVLAN rules. (In the PVLAN rules, configuring the PVLAN trunk port applies only to a PVLAN that spans multiple routers.)

- The primary VLAN must be a tagged VLAN.
- If you are going to configure a community VLAN ID, you must first configure the primary VLAN.
- If you are going to configure an isolation VLAN ID, you must first configure the primary VLAN.

NOTE: Configuring a voice over IP (VoIP) VLAN on PVLAN interfaces is not supported.

Configuring a VLAN on a single router is relatively simple, as shown in Figure 13 on page 437.

Figure 13: Configuring a PVLAN on a Single Switch



Configuring a primary VLAN consists of these steps:

1. Configure the primary VLAN name and 802.1Q tag.
2. Set **no-local-switching** on the primary VLAN.
3. Configure the promiscuous trunk port and access ports.
4. Make the promiscuous trunk and access ports members of the primary VLAN.

Within a primary VLAN, you can configure secondary community VLANs or secondary isolated VLANs or both. Configuring a secondary community VLAN consists of these steps:

1. Configure a VLAN using the usual process.
2. Configure access interfaces for the VLAN.
3. Assign a primary VLAN to the community VLAN,

Isolated VLANs are created internally when the isolated VLAN has access interfaces as members and the option **no-local-switching** is enabled on the primary VLAN.

802.1Q tags are required for interswitch isolated VLANs because IEEE 802.1Q uses an internal tagging mechanism by which a trunking device inserts a 4-byte VLAN frame identification tab into the packet header.

Trunk ports are only needed for multirouter PVLAN configurations—the trunk port carries traffic from the primary VLAN and all secondary VLANs.

Limitations of Private VLANs

The following constraints apply to private VLAN configurations:

- An access interface can belong to only one PVLAN domain, that is, it cannot participate in two different primary VLANs.
- A trunk interface can be a member of two secondary VLANs as long as the secondary VLANs are in two *different* primary VLANs. A trunk interface cannot be a member of two secondary VLANs that are in the *same* primary VLAN.
- A single region of Multiple Spanning Tree Protocol (MSTP) must be configured on all VLANs that are included within the PVLAN.
- VLAN Spanning Tree Protocol (VSTP) is not supported.
- IGMP snooping is not supported with private VLANs.
- Routed VLAN interfaces are not supported on private VLANs
- Routing between secondary VLANs in the same primary VLAN is not supported.
- Some configuration statements cannot be specified on a secondary VLAN. You can configure the following statements at the **[edit vlans *vlan-name* switch-options]** hierarchy level *only* on the primary PVLAN.
- If you want to change a primary VLAN to be a secondary VLAN, you must first change it to a normal VLAN and commit the change. For example, you would follow this procedure:

  1. Change the primary VLAN to be a normal VLAN.
  2. Commit the configuration.
  3. Change the normal VLAN to be a secondary VLAN.
  4. Commit the configuration.

  Follow the same sequence of commits if you want to change a secondary VLAN to be a primary VLAN. That is, make the secondary VLAN a normal VLAN and commit that change and then change the normal VLAN to be a primary VLAN.

The following features are *not* supported on PVLANs on Junos switches with support for the ELS configuration style:

- DHCP security features (DHCP snooping, dynamic ARP inspection, IP source guard)
- Egress VLAN firewall filters
- Ethernet ring protection (ERP)
- Flexible VLAN tagging

- **global-mac-statistics**
- Integrated routing and bridging (IRB) interface
- Multicast snooping or IGMP snooping
- Multichassis link aggregation groups (MC-LAGs)
- Port mirroring
- Q-in-Q tunneling
- VLAN Spanning Tree Protocol (VSTP)
- Voice over IP (VoIP)

You can configure the following statements at the **[edit vlans *vlan-name* switch-options]** hierarchy level only on the primary PVLAN:

- **mac-table-size**
- **no-mac-learning**
- **mac-statistics**
- **interface-mac-limit**

RELATED DOCUMENTATION

Understanding Bridging and VLANs on Switches | **168**

Understanding PVLAN Traffic Flows Across Multiple Switches

IN THIS SECTION

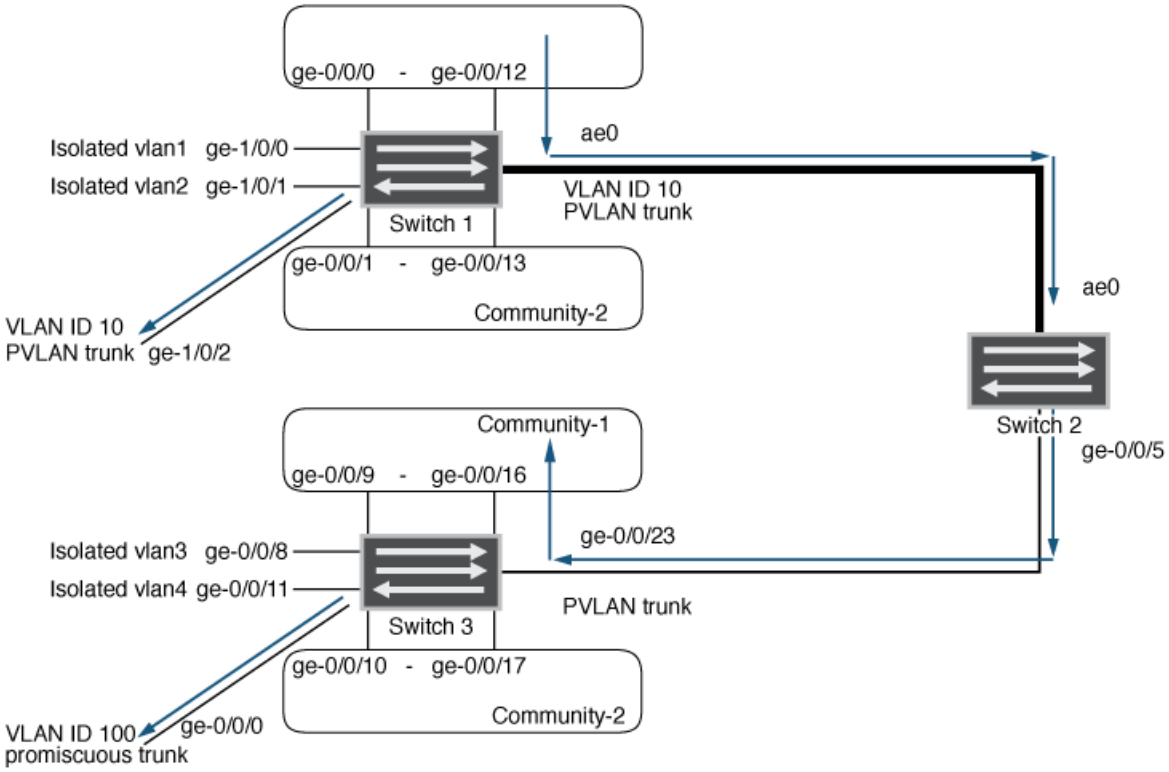
This topic illustrates and explains three different traffic flows on a sample multiswitch network configured with a private VLAN (PVLAN). PVLANs restrict traffic flows through their member switch ports (which are called “private ports”) so that they communicate only with a specific uplink trunk port or with specified ports within the same VLAN.

This topic describes:

Community VLAN Sending Untagged Traffic

In this scenario, a VLAN in Community-1 of Switch 1 at interface ge-0/0/0 sends untagged traffic. The arrows in Figure 14 on page 440 represent this traffic flow.

Figure 14: Community VLAN Sends Untagged Traffic



In this scenario, the following activity takes place on Switch 1:

- Community-1 VLAN on interface ge-0/0/0: Learning
- pvlan100 on interface ge-0/0/0: Replication
- Community-1 VLAN on interface ge-0/0/12: Receives traffic
- PVLAN trunk port: Traffic exits from ge-1/0/2 and from ae0 with tag 10

- Community-2: Interface receives no traffic
- Isolated VLANs: Interfaces receive no traffic

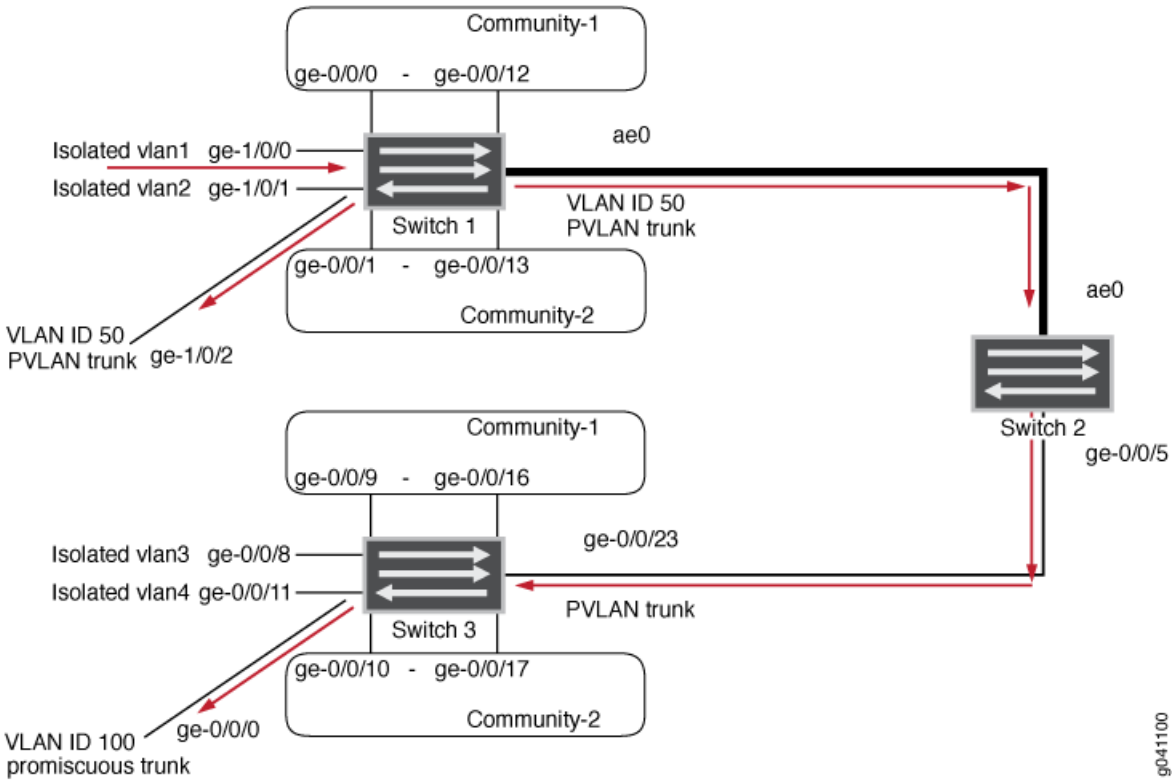
In this scenario, this activity takes place on Switch 3:

- Community-1 VLAN on interface ge-0/0/23 (PVLAN trunk): Learning
- pvlan100 on interface ge-0/0/23: Replication
- Community-1 VLAN on interface ge-0/0/9 and ge-0/0/16: Receives traffic
- Promiscuous trunk port: Traffic exits from ge-0/0/0 with tag 100
- Community-2: Interface receives no traffic
- Isolated VLANs: Interfaces receive no traffic

Isolated VLAN Sending Untagged Traffic

In this scenario, isolated VLAN1 on Switch 1 at interface ge-1/0/0 sends untagged traffic. The arrows in Figure 15 on page 441 represent this traffic flow.

Figure 15: Isolated VLAN Sends Untagged Traffic



In this scenario, the following activity takes place on Switch 1:

- Isolated VLAN1 on interface ge-1/0/0: Learning
- pvlan100 on interface ge-1/0/0: Replication
- Traffic exits from pvlan-trunk ge-1/0/2 and ae0 with tag 50
- Community-1 and Community-2: Interfaces receive no traffic
- Isolated VLANs: Interfaces receive no traffic

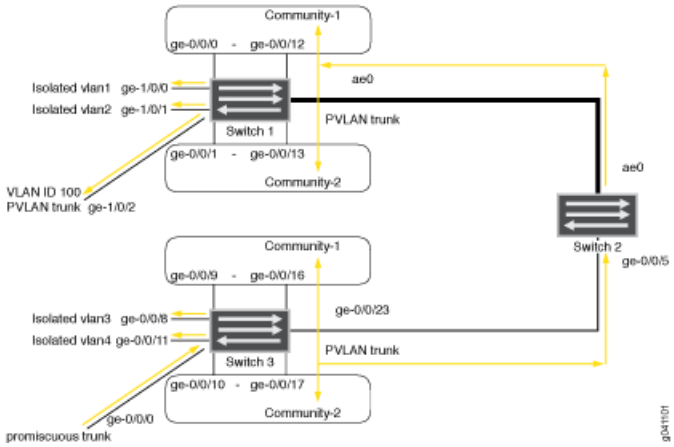
In this scenario, this activity takes place on Switch 3:

- VLAN on interface ge-0/0/23 (PVLAN trunk port): Learning
- pvlan100 on interface ge0/0/23: Replication
- Promiscuous trunk port: Traffic exits from ge-0/0/0 with tag 100
- Community-1 and Community-2: Interfaces receive no traffic
- Isolated VLANs: Receive no traffic

PVLAN Tagged Traffic Sent on a Promiscuous Port

In this scenario, PVLAN tagged traffic is sent on a promiscuous port. The arrows in Figure 16 on page 442 represent this traffic flow.

Figure 16: PVLAN Tagged Traffic Sent on a Promiscuous Port



In this scenario, the following activity takes place on Switch 1:

- pvlan100 VLAN on interface ae0 (PVLAN trunk): Learning
- Community-1, Community-2, and all isolated VLANs on interface ae0: Replication
- VLAN on interface ae0: Replication
- Traffic exits from pvlan-trunk ge-1/0/2 with tag 100

- Community-1 and Community-2: Interfaces receive traffic
- Isolated VLANs: Receive traffic

In this scenario, this activity takes place on Switch 3:

- pvlan100 on interface ge-0/0/0: Learning
- Community-1, Community-2 and all isolated VLANs on interface ge-0/0/0: Replication
- VLAN on interface ge-0/0/0: Replication
- Community-1 and Community-2: Interfaces receive traffic
- Isolated VLANs: Receive traffic

Understanding Secondary VLAN Trunk Ports and Promiscuous Access Ports on PVLANs

IN THIS SECTION

VLANs limit broadcasts to specified users. Private VLANs (PVLANs) take this concept a step further by splitting a VLAN into multiple broadcast subdomains and essentially putting secondary VLANs inside a primary VLAN. PVLANs restrict traffic flows through their member ports so that these ports communicate only with a specified uplink trunk port or with specified ports within the same VLAN. The uplink trunk port is usually connected to a router, firewall, server, or provider network. A PVLAN typically contains many private ports that communicate only with a single uplink, thereby preventing the ports from communicating with each other.

Secondary trunk ports and promiscuous access ports extend the functionality of PVLANs for use in complex deployments, such as:

- Enterprise VMWare Infrastructure environments
- Multitenant cloud services with VM management
- Web hosting services for multiple customers

For example, you can use secondary VLAN trunk ports to connect QFX devices to VMware servers that are configured with private VLANs. You can use promiscuous access ports to connect QFX devices to systems that do not support trunk ports but do need to participate in private VLANs.

This topic explains the following concepts regarding PVLANs on the QFX Series:

PVLAN Port Types

PVLANs can use the following different port types:

- Promiscuous trunk port—A promiscuous port is an upstream trunk port connected to a router, firewall, server, or provider network. A promiscuous trunk port can communicate with all interfaces, including the isolated and community ports within a PVLAN.
- PVLAN trunk port—A PVLAN trunk port is required in multiswitch PVLAN configurations to span the switches. The PVLAN trunk port is a member of all VLANs within the PVLAN (that is, the primary VLAN, the community VLANs, and the interswitch isolated VLAN), and it carries traffic from the primary VLAN and all secondary VLANs. It can communicate with all ports.

  Communication between a PVLAN trunk port and an isolated port is usually unidirectional. A PVLAN trunk port's membership in the interswitch isolated VLAN is egress-only, meaning that an isolated port can forward packets to a PVLAN trunk port, but a PVLAN trunk port does not forward packets to an isolated port (unless the packets ingressed on a promiscuous access port and are therefore being forwarded to all the secondary VLANs in the same primary VLAN as the promiscuous port).
- Secondary VLAN trunk port—Secondary VLAN trunk ports carry secondary VLAN traffic. For a given private (primary) VLAN, a secondary VLAN trunk port can carry traffic for only one secondary VLAN. However, a secondary VLAN trunk port can carry traffic for multiple secondary VLANs as long as each secondary VLAN is a member of a different primary VLAN. For example, a secondary VLAN trunk port can carry traffic for a community VLAN that is part of primary VLAN pvlan100 and also carry traffic for an isolated VLAN that is part of primary VLAN pvlan400.

  > **NOTE:** When traffic egresses from a secondary VLAN trunk port, it normally carries the tag of the primary VLAN that the secondary port is a member of. If you want traffic that egresses from a secondary VLAN trunk port to retain its secondary VLAN tag, use the **extend-secondary-vlan-id** statement.

- Community port—Community ports communicate among themselves and with their promiscuous ports. Community ports serve only a select group of users. These interfaces are separated at Layer 2 from all other interfaces in other communities or isolated ports within their PVLAN.

- Isolated access port—Isolated ports have Layer 2 connectivity only with promiscuous ports and PVLAN trunk ports. An isolated access port cannot communicate with another isolated port even if these two ports are members of the same isolated VLAN.
- Promiscuous access port—These ports carry untagged traffic and can be a member of only one primary VLAN. Traffic that ingresses on a promiscuous access port is forwarded to the ports of the secondary VLANs that are members of the primary VLAN that the promiscuous access port is a member of. In this case, the traffic carries the appropriate secondary VLAN tag when it egresses from the secondary VLAN port if the secondary VLAN port is a trunk port. If traffic ingresses on a secondary VLAN port and egresses on a promiscuous access port, the traffic is untagged on egress. If tagged traffic ingresses on a promiscuous access port, the traffic is discarded.

Secondary VLAN Trunk Port Details

When using a secondary VLAN trunk port, be aware of the following:

- You must configure an isolation VLAN ID for each primary VLAN that the secondary VLAN trunk port will participate in. This is true even if the secondary VLANs that the secondary VLAN trunk port will carry are confined to a single device.
- If you configure a port to be a secondary VLAN trunk port for a given primary VLAN, you can also configure the same physical port to be any of the following:
  - Secondary VLAN trunk port for another primary VLAN
  - PVLAN trunk for another primary VLAN
  - Promiscuous trunk port
  - Access port for a non-private VLAN
- Traffic that ingresses on a secondary VLAN trunk port (with a secondary VLAN tag) and egresses on a PVLAN trunk port retains the secondary VLAN tag on egress.
- Traffic that ingresses on a secondary VLAN trunk port and egresses on a promiscuous trunk port has the appropriate primary VLAN tag on egress.
- Traffic that ingresses on a secondary VLAN trunk port and egresses on a promiscuous access port is untagged on egress.
- Traffic that ingresses on a promiscuous trunk port with a primary VLAN tag and egresses on a secondary VLAN trunk port carries the appropriate secondary VLAN tag on egress. For example, assume that you have configured the following on a switch:
  - Primary VLAN 100
  - Community VLAN 200 as part of the primary VLAN
  - Promiscuous trunk port
  - Secondary trunk port that carries community VLAN 200

If a packet ingresses on the promiscuous trunk port with primary VLAN tag 100 and egresses on the secondary VLAN trunk port, it carries tag 200 on egress.

Use Cases

IN THIS SECTION

On the same physical interface, you can configure multiple secondary VLAN trunk ports (in different primary VLANs) or combine a secondary VLAN trunk port with other types of VLAN ports. The following use cases provide examples of doing this and show how traffic would flow in each case:

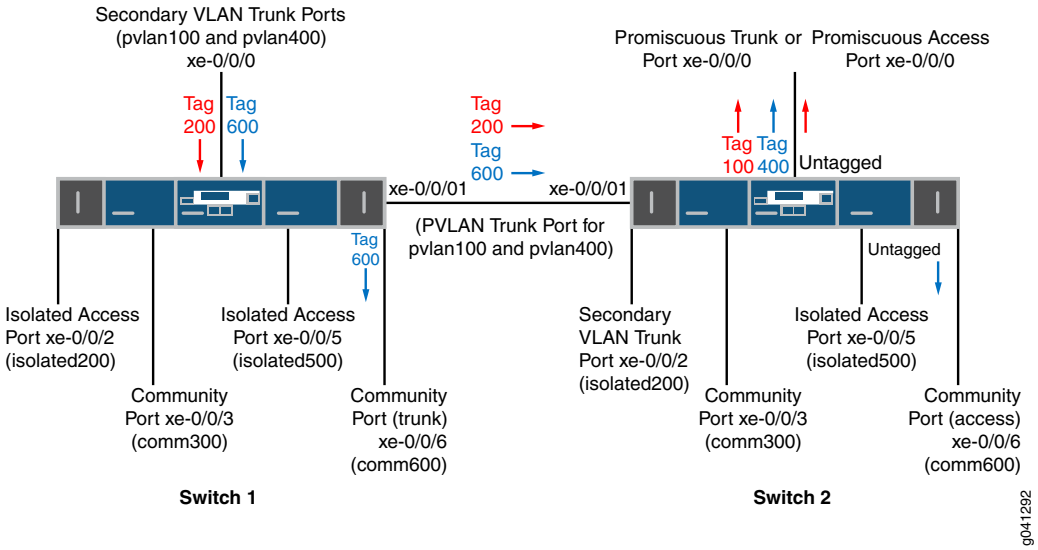
Secondary VLAN Trunks In Two Primary VLANS

For this use case, assume you have two switches with the following configuration:

- Primary VLAN pvlan100 with tag 100.
  - Isolated VLAN isolated200 with tag 200 is a member of pvlan100.
  - Community VLAN comm300 with tag 300 is a member of pvlan100.
- Primary VLAN pvlan400 with tag 400.
  - Isolated VLAN isolated500 with tag 500 is a member of pvlan400.
  - Community VLAN comm600 with tag 600 is a member of pvlan400.
- Interface xe-0/0/0 on Switch 1 connects to a VMware server (not shown) that is configured with the private VLANs used in this example. This interface is configured with secondary VLAN trunk ports to carry traffic for secondary VLAN comm600 and the isolated VLAN (tag 200) that is a member of pvlan100.
- Interface xe-0/0/0 on Switch 2 is shown configured as a promiscuous trunk port or promiscuous access port. In the latter case, you can assume that it connects to a system (not shown) that does not support trunk ports but is configured with the private VLANs used in this example.
- On Switch 1, xe-0/0/6 is a member of comm600 and is configured as a trunk port.
- On Switch 2, xe-0/0/6 is a member of comm600 and is configured as an access port.

Figure 17 on page 447 shows this topology and how traffic for isolated200 and comm600 would flow after ingressing on xe-0/0/0 on Switch 1. Note that traffic would flow only where the arrows indicate. For example, there are no arrows for interfaces xe-0/0/2, xe-0/0/3, and xe-0/0/5 on Switch 1 because no packets would egress on those interfaces.

Figure 17: Two Secondary VLAN Trunk Ports on One Interface



Here is the traffic flow for VLAN isolated200:

1. After traffic for isolated200 ingresses on the secondary VLAN trunk port on Switch 1, it egresses on the PVLAN trunk port because the PVLAN trunk port is a member of all the VLANs. The packets keep the secondary VLAN tag (200) when egressing.

2. After traffic for isolated200 ingresses on the secondary VLAN trunk port on Switch 2, it egresses on xe-0/0/0, which is configured as a promiscuous trunk port or promiscuous access port.

   - If xe-0/0/0 on Switch 2 is configured as a promiscuous trunk port, the packets egress on this port with the primary VLAN tag (100).
   - If xe-0/0/0 on Switch 2 is configured as a promiscuous access port, the packets egress on this port untagged.

Note that traffic for VLAN isolated200 does not egress on isolated access port xe-0/0/2 on Switch 1 or secondary VLAN trunk port xe-0/0/2 on Switch 2 even though these two ports are members of the same isolated VLAN.

Here is the traffic flow for VLAN comm600:

1. After traffic for comm600 ingresses on the secondary VLAN trunk port on Switch 1, it egresses on the PVLAN trunk port because the PVLAN trunk port is a member of all the VLANs. The packets keep the secondary VLAN tag (600) when egressing.

2. Traffic for comm600 also egresses on community port xe-0/0/6 on Switch 1. The traffic is tagged because the port is configured as a trunk.

3. After traffic for comm600 ingresses on the PVLAN trunk port on Switch 2, it egresses on xe-0/0/0, if this interface is configured as a promiscuous trunk port.

   NOTE: If xe-0/0/0 on Switch 2 is configured as a promiscuous access port, the port can participate in only one primary VLAN. In this case, the promiscuous access port is part of pvlan100, so traffic for comm600 does not egress from it

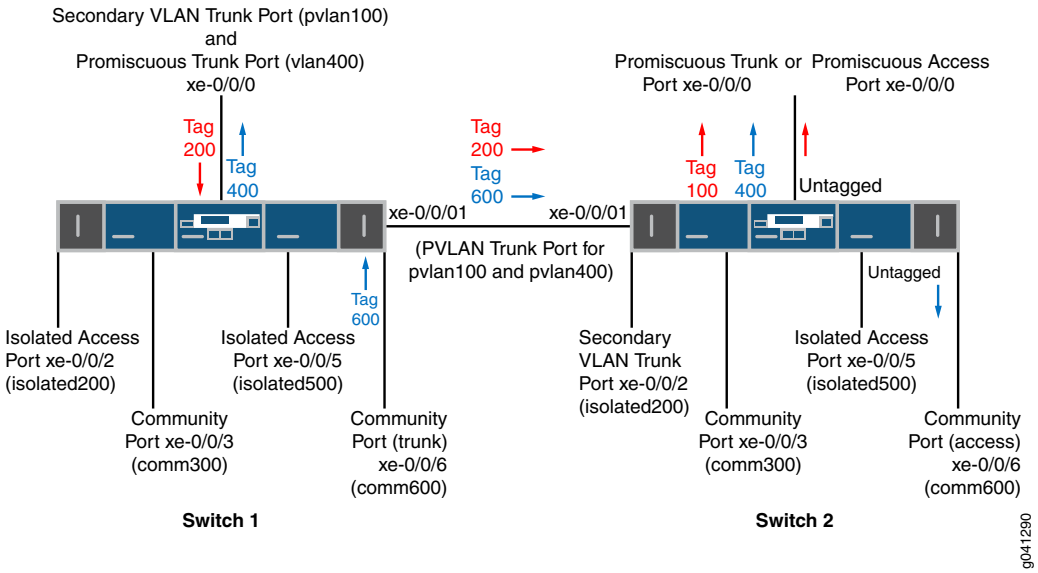
4. Traffic for comm600 also egresses on community port xe-0/0/6 on Switch 2. In this case, the traffic is untagged because the port mode is access.

Secondary VLAN Trunk and Promiscuous Trunk

For this use case, assume you have two switches configured with the same ports and VLANs as in the previous use case, with one exception: In this case, xe-0/0/0 on Switch 1 is configured as a secondary VLAN trunk port for VLAN pvlan100 and is also configured as a promiscuous trunk port for pvlan400.

Figure 18 on page 449 shows this topology and how traffic for isolated200 (member of pvlan100) and comm600 (member of pvlan400) would flow after ingressing on Switch 1.

Figure 18: Secondary VLAN Trunk and Promiscuous Trunk on One Interface



The traffic flow for VLAN isolated200 is the same as in the previous use case, but the flow for comm600 is different. Here is the traffic flow for VLAN comm600:

1. After traffic for comm600 ingresses on community VLAN port xe-0/0/6 on Switch 1, it egresses on promiscuous trunk port xe-0/0/0 on Switch 1. In this case it carries the primary VLAN tag (400).

2. Traffic for comm600 also egresses on the PVLAN trunk port because the PVLAN trunk port is a member of all the VLANs. The packets keep the secondary VLAN tag (600) when egressing.

3. After traffic for comm600 ingresses on the PVLAN trunk port on Switch 2, it egresses on xe-0/0/0, if this interface is configured as a promiscuous trunk port.

   It does not egress on xe-0/0/0 if this interface is configured as a promiscuous access port because the port can participate only in pvlan100.

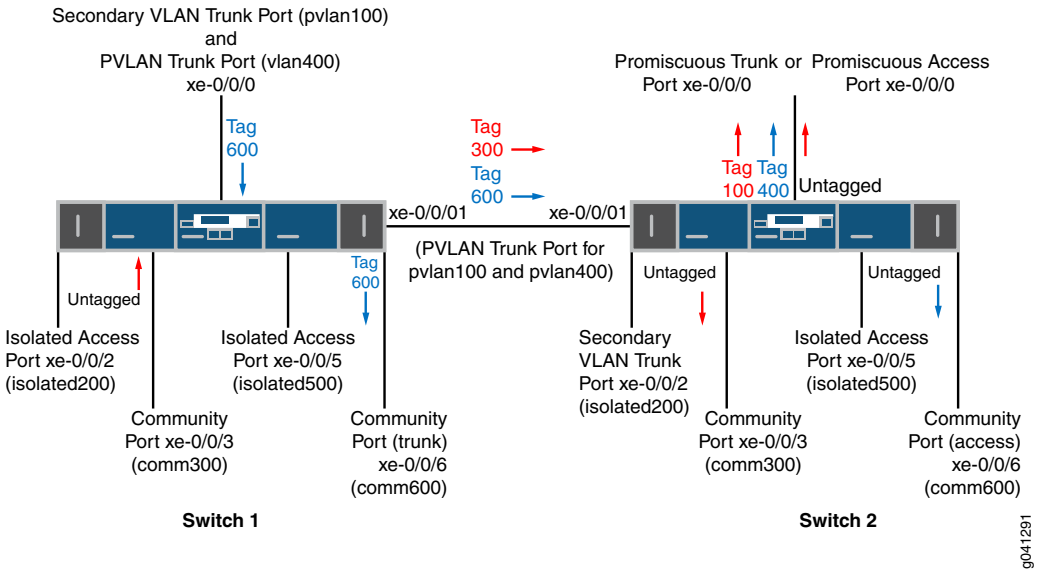
4. Traffic for comm600 also egresses on community port xe-0/0/6 on Switch 2.

Secondary VLAN Trunk and PVLAN Trunk

For this use case, assume you have two switches configured with the same ports and VLANs as in the previous use cases except that xe-0/0/0 on Switch 1 is configured as a secondary VLAN trunk port for VLAN pvlan100 and is also configured as a PVLAN trunk port for pvlan400.

Figure 19 on page 450 shows this topology and how traffic for comm300 (member of pvlan100) and comm600 (member of pvlan400) would flow after ingressing on Switch 1.

Figure 19: Secondary VLAN Trunk and PVLAN Trunk on One Interface



Here is the traffic flow for VLAN comm300:

1. After traffic for comm300 ingresses on community port xe-0/0/3 on Switch 1, it egresses on PVLAN trunk port xe-0/0/1 because that PVLAN trunk port is a member of all the VLANs. The packets keep the secondary VLAN tag (300) when egressing.

   NOTE: Traffic for comm300 does not egress on xe-0/0/0 because the secondary VLAN trunk port on this interface carries isolated200, not comm300.

2. After traffic for comm300 ingresses on the PVLAN trunk port on Switch 2, it egresses on xe-0/0/0, which is configured as a promiscuous trunk port or promiscuous access port.

   - If xe-0/0/0 on Switch 2 is configured as a promiscuous trunk port, the packets egress on this port with the primary VLAN tag (100).
   - If xe-0/0/0 on Switch 2 is configured as a promiscuous access port, the packets egress on this port untagged.

3. Traffic for comm300 also egresses on community port xe-0/0/3 on Switch 2.

Here is the traffic flow for VLAN comm600:

1. After traffic for comm600 ingresses on the PVLAN port xe-0/0/0 on Switch 1, it egresses on the community port xe-0/0/6 on Switch 1. The packets keep the secondary VLAN tag (600) when egressing because xe-0/0/6 is a trunk port.

2. Traffic for comm600 also egresses on PVLAN trunk port xe-0/0/1 because that PVLAN trunk port is a member of all the VLANs. The packets keep the secondary VLAN tag (600) when egressing.

3. After traffic for comm600 ingresses on the PVLAN trunk port on Switch 2, it egresses on xe-0/0/0, if this interface is configured as a promiscuous trunk port.

   It does not egress on xe-0/0/0 if this interface is configured as a promiscuous access port because the port can participate only in pvlan100.

4. Traffic for comm600 also egresses on community port xe-0/0/6 on Switch 2. This traffic is untagged on egress because xe-0/0/6 is an access port.

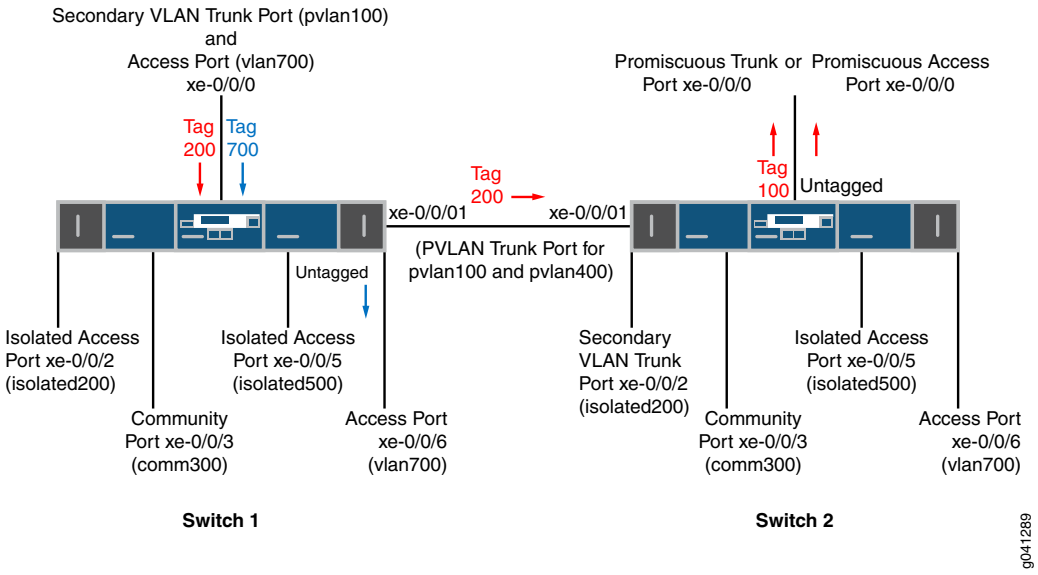
Secondary VLAN Trunk and Non-Private VLAN Interface

For this use case, assume you have two switches configured with the same ports and VLANs as in the previous use cases except for these differences:

- Configuration for xe-0/0/0 on Switch 1:
  - Secondary VLAN trunk port for VLAN pvlan100
  - Access port for vlan700
- Port xe-0/0/6 on both switches is an access port for vlan700.

Figure 20 on page 452 shows this topology and how traffic for isolated200 (member of pvlan100) and vlan700 would flow after ingressing on Switch 1.

Figure 20: Secondary VLAN Trunk and Non-Private VLAN Port on One Interface



Here is the traffic flow for VLAN isolated200:

1. After traffic for isolated200 ingresses on the secondary VLAN trunk port on Switch 1, it egresses on the PVLAN trunk port. The packets keep the secondary VLAN tag (200) when egressing.

2. After traffic for isolated200 ingresses on the PVLAN trunk port on Switch 2, it egresses on xe-0/0/0, which is configured as a promiscuous trunk port or promiscuous access port.

   - If xe-0/0/0 on Switch 2 is configured as a promiscuous trunk port, the packets egress on this port with the primary VLAN tag (100).
   - If xe-0/0/0 on Switch 2 is configured as a promiscuous access port, the packets egress on this port untagged.

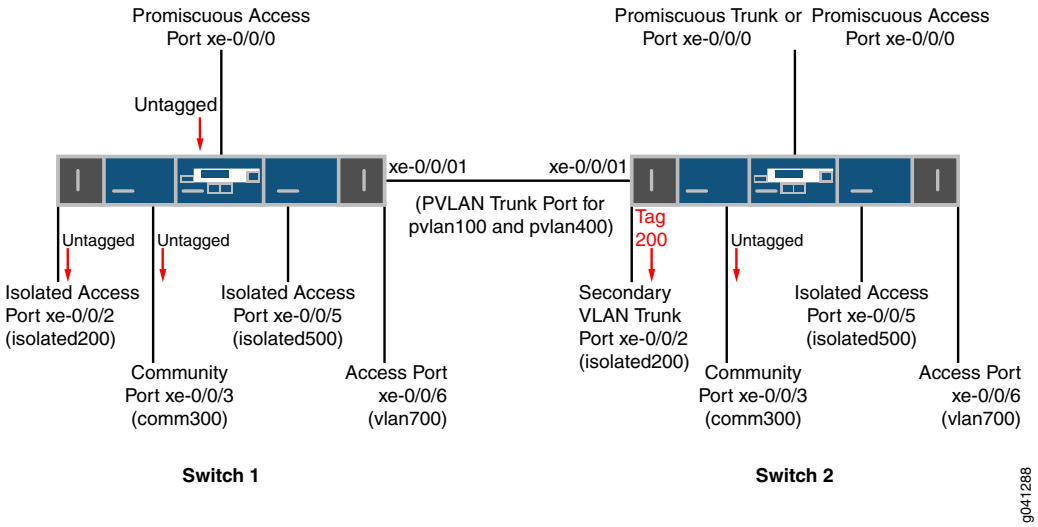
Note that traffic for VLAN isolated200 does not egress on isolated access port xe-0/0/2 on Switch 1 or secondary VLAN trunk port xe-0/0/2 on Switch 2 even though these two ports are members of the same isolated VLAN.

After traffic for vlan700 ingresses on the access port configured on xe-0/0/0 on Switch 1, it egresses on access port xe-0/0/6 because that port is a member of the same VLAN. Traffic for vlan700 is not forwarded to Switch 2 (even though xe-0/0/6 on Switch 2 is a member of vlan700) because the PVLAN trunk on xe-0/0/1 does not carry this VLAN.

Traffic Ingressing on Promiscuous Access Port

For this use case, assume you have two switches configured with the same ports and VLANs as in the previous use case except that xe-0/0/0 on Switch 1 is configured as a promiscuous access port and is a member of pvlan100. Figure 21 on page 453 shows this topology and how untagged traffic would flow after ingressing through this interface on Switch 1.

Figure 21: Traffic Ingressing on Promiscuous Access Port



As the figure shows, untagged traffic that ingresses on a promiscuous access port is forwarded to all the secondary VLAN ports that are members of the same primary VLAN that the promiscuous access port is a member of. The traffic is untagged when it egresses from access ports and tagged on egress from a trunk port (xe-0/0/2 on Switch 2).

RELATED DOCUMENTATION

Understanding Egress Firewall Filters with PVLANs

Using 802.1X Authentication and Private VLANs Together on the Same Interface

IN THIS SECTION

Understanding Using 802.1X Authentication and PVLANs Together on the Same Interface

You can now configure both 802.1X authentication and private VLANs (PVLANs) on the same interface.

IEEE 802.1X authentication provides network edge security, protecting Ethernet LANs from unauthorized user access by blocking all traffic to and from a supplicant (client) at the interface until the supplicant's credentials are presented and matched on the *authentication server* (a RADIUS server).

Private VLANs (PVLANs) provide Layer 2 isolation between ports within a VLAN, splitting a broadcast domain into multiple discrete broadcast subdomains by creating secondary VLANs. PVLANs are useful for restricting the flow of broadcast and unknown unicast traffic and for limiting the communication between known hosts.

On a switch that is configured with both 802.1X authentication and PVLANs, when a new device is attached to the PVLAN network, the device is authenticated and then is assigned to a secondary VLAN based on the PVLAN configuration or RADIUS profile. The device then obtains an IP address and is given access to the PVLAN network.

> **NOTE:** This document does not provide detailed information about 802.1X authentication or private VLANs. For those details, see the feature documentation that is specific to those individual features. For 802.1X, see User Access and Authentication User Guide. For PVLANs, see Ethernet Switching User Guide.

Configuration Guidelines for Combining 802.1X Authentication with PVLANs

Keep the following guidelines and limitations in mind for configuring these two features on the same interface:

- You cannot configure an 802.1X-enabled interface as a promiscuous interface (an interface that is a member of the primary VLAN by configuration) or as an interswitch-link (ISL) interface.
- Multiple users cannot be authenticated over different VLANs belonging to the same PVLAN domain on a logical interface—for example, if interface ge-0/0/0 is configured as **supplicant multiple** and clients C1 and C2 are authenticated and are added to dynamic VLANs V1 and V2, respectively, then V1 and V2 must belong to different PVLAN domains.
- If the VoIP VLAN and the data VLAN are different, those two VLANs must be in different PVLAN domains.
- When PVLAN membership is changed (that is, an interface is reconfigured in a different PVLAN), clients must be reauthenticated.

Example: Configuring 802.1X Authentication with Private VLANs in One Configuration

IN THIS SECTION

Requirements

- Junos OS Release 18.2R1 or later
- EX2300, EX3400, or EX4300 switch

Before you begin, specify the RADIUS server or servers to be used as the authentication server. See *Specifying RADIUS Server Connections on Switches (CLI Procedure)*.

Overview

The following configuration section shows the access profile configuration, the 802.1X authentication configuration, and finally the VLANs (including PVLANs) configuration.

Configuring 802.1X Authentication with Private VLANs in One Configuration

CLI Quick Configuration

```
[edit]
set access radius-server 10.20.9.199 port 1812
set access radius-server 10.20.9.199 secret
"$9$Lqa7dsaZjP5F245Fn/0OX7-V24JGDkmf"
set access profile dot1x-auth authentication-order radius
set access profile authp authentication-order radius
set access profile authp radius authentication-server 10.204.96.165
set switch-options voip interface ge-0/0/8.0 vlan voip
set interfaces ge-0/0/8 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/8 unit 0 family ethernet-switching vlan members data
set protocols dot1x authenticator authentication-profile-name authp
```

```
set protocols dot1x authenticator interface ge-0/0/8.0 supplicant multiple
set protocols dot1x authenticator interface ge-0/0/8.0 mac-radius
set vlans community vlan-id 20
set vlans community private-vlan community
set vlans community-one vlan-id 30
set vlans community-one private-vlan community
set vlans isolated vlan-id 200
set vlans isolated private-vlan isolated
set vlans pvlan vlan-id 2000
set vlans pvlan isolated-vlan isolated
set vlans pvlan community-vlans [community community-one]
set vlans data vlan-id 43
set vlans voip vlan-id 33
```

Step-by-Step Procedure

To configure 802.1X authentication and PVLANs in one configuration:

1. Configure the access profile:

   [edit access]
   set radius-server 10.20.9.199 port 1812
   set radius-server 10.20.9.199 secret "9Lqa7dsaZjP5F245Fn/0OX7-V24JGDkmf"
   set profile dot1x-auth authentication-order radius
   set profile authp authentication-order radius
   set profile authp radius authentication-server 10.204.96.165
   [edit switch-options]
   set voip interface ge-0/0/8.0 vlan voip

   NOTE: The configured VoIP VLAN cannot be a PVLAN (primary, community, or isolated).

2. Configure the 802.1X settings:

   [edit interfaces]

set ge-0/0/8 unit 0 family ethernet-switching interface-mode access
set ge-0/0/8 unit 0 family ethernet-switching vlan members data
[edit protocols]
set dot1x authenticator authentication-profile-name authp
set dot1x authenticator interface ge-0/0/8.0 supplicant multiple
set dot1x authenticator interface ge-0/0/8.0 mac-radius

> **NOTE:** The configured data VLAN could also be a community VLAN or an isolated VLAN.

3. Configure the VLANs (including the PVLANs):

[edit vlans]
set community vlan-id 20
set community private-vlan community
set community-one vlan-id 30
set community-one private-vlan community
set isolated vlan-id 200
set isolated private-vlan isolated
set pvlan vlan-id 2000
set pvlan isolated-vlan isolated
set pvlan community-vlans [community community-one]
set data vlan-id 43
set voip vlan-id 33

Results

From configuration mode, confirm your configuration by entering the following **show** commands on the switch. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@switch# show access
radius-server {
    10.20.9.199 {
        port 1812;
        secret "$9$Lqa7dsaZjP5F245Fn/0OX7-V24JGDkmf"; ## SECRET-DATA
    }
}
profile dot1x-auth {
```

```
    authentication-order radius;
}
profile authp {
    authentication-order radius;
    radius {
        authentication-server 10.204.96.165;
    }
}
user@switch# show interfaces
ge-0/0/8 {
    unit 0 {
        family ethernet-switching {
            interface-mode access;
            vlan {
                members data;
            }
        }
    }
}
user@switch# show protocols
dot1x {
    authenticator {
        authentication-profile-name authp;
        interface {
            ge-0/0/8.0 {
                supplicant multiple;
                mac-radius;
            }
        }
    }
}
user@switch# show switch-options
voip {
    interface ge-0/0/8.0 {
        vlan voip;
    }
}
user@switch# show vlans
community {
    vlan-id 20;
    private-vlan community;
}
community-one {
    vlan-id 30;
```

```
    private-vlan community;
  }
  data {
    vlan-id 43;
  }
  isolated {
    vlan-id 200;
    private-vlan isolated;
  }
  pvlan {
    vlan-id 2000;
    isolated-vlan isolated;
    community-vlans [community community-one];
  }
  voip {
    vlan-id 33;
  }
```

Verification

IN THIS SECTION

Verify That Client MAC Addresses Are Learned on the Primary VLAN

Purpose

Show that a client MAC address has been learned on the primary VLAN.

Action

`user@switch>` **show ethernet-switching table**

```
MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent
static, C - Control MAC, SE - statistics enabled, NM - non configured MAC, R -
remote PE MAC, O - ovsdb MAC)
Ethernet switching table : 1 entries, 1 learned
Routing instance : default-switch
    Vlan        MAC               MAC    Age  Logical      NH     RTR
```

```
    name        address            flags       interface    Index  ID
    pvlan       00:30:48:8C:66:BD  D      -    ge-0/0/8.0   0      0
```

Verify That the Primary VLAN Is an Authenticated VLAN

Purpose

Show that the primary VLAN is shown as an authenticated VLAN.

Action

`user@switch>` **show dot1x interface ge-0/0/8.0 detail**

```
ge-0/0/8.0
  Role: Authenticator
  Administrative state: Auto
  Supplicant mode: Multiple
  Number of retries: 3
  Quiet period: 60 seconds
  Transmit period: 30 seconds
  Mac Radius: Enabled
  Mac Radius Strict: Disabled
  Reauthentication: Enabled Reauthentication interval: 40 seconds
  Supplicant timeout: 30 seconds
  Server timeout: 30 seconds
  Maximum EAPOL requests: 1
  Guest VLAN member: <not configured>
  Number of connected supplicants: 1
    Supplicant: user5, 00:30:48:8C:66:BD
      Operational state: Authenticated
      Authentication method: Radius
      Authenticated VLAN: pvlan
      Reauthentication due in 17 seconds
```

Putting Access Port Security on Private VLANs

IN THIS SECTION

Understanding Access Port Security on PVLANs

You can now enable access port security features, such as DHCP snooping, on private VLANs (PVLANs).

For security reasons, it is often useful to restrict the flow of broadcast and unknown unicast traffic and to even limit the communication between known hosts. The PVLAN feature allows you to split a broadcast domain into multiple isolated broadcast subdomains, essentially putting a VLAN inside a VLAN.

Ethernet LANs are vulnerable to attacks such as address spoofing (forging) and Layer 2 denial of service (DoS) on network devices. The following access port security features help protect your device against losses of information and productivity that such attacks can cause, and you can now configure these security features on a PVLAN:

- DHCP snooping—Filters and blocks ingress DHCP server messages on untrusted ports. DHCP snooping builds and maintains a database of DHCP lease information, which is called the DHCP snooping database.
- DHCPv6 snooping—DHCP snooping for IPv6.
- DHCP option 82—Also known as the DHCP Relay Agent Information option. Helps protect the switch against attacks such as spoofing of IP addresses and MAC addresses and DHCP IP address starvation. Option 82 provides information about the network location of a DHCP client. The DHCP server uses this information to implement IP addresses or other parameters for the client.
- DHCPv6 options:
  - Option 37—Remote ID option for DHCPv6; inserts information about the network location of the remote host into DHCPv6 packets.
  - Option 18—Circuit ID option for DHCPv6; inserts information about the client port into DHCPv6 packets.
  - Option 16—Vendor ID option for DHCPv6; inserts information about the vendor of the client hardware into DHCPv6 packets.

- Dynamic ARP inspection (DAI)—Prevents Address Resolution Protocol (ARP) spoofing attacks. ARP requests and replies are compared against entries in the DHCP snooping database, and filtering decisions are made on the basis of the results of those comparisons.
- IP source guard—Mitigates the effects of IP address spoofing attacks on the Ethernet LAN; validates the source IP address in the packet sent from an untrusted access interface against the DHCP snooping database. If the packet cannot be validated, it is discarded.
- IPv6 source guard—IP source guard for IPv6.
- IPv6 neighbor discovery inspection—Prevents IPv6 address spoofing attacks; compares neighbor discovery requests and replies against entries in the DHCPv6 snooping database, and filtering decisions are made on the basis of the results of those comparisons.

> **NOTE:** This document does not provide detailed information about access port security features or PVLANs. For those details, see the feature documentation that is specific to those individual features. For access port security, see Security Services Administration Guide. For PVLANs, see Ethernet Switching User Guide.

Configuration Guidelines for Putting Access Port Security Features on PVLANs

Keep the following guidelines and limitations in mind for configuring access port security features on PVLANs:

- You must apply the *same* access port security features on both the primary vlan and all its secondary VLANs.
- A PVLAN can have only one integrated routing and bridging (IRB) interface, and the IRB interface must be on the primary VLAN.
- Limitations on access port security configurations on PVLANs are the same as those for access port security features configurations that are not in PVLANs. See the access port security documentation at Security Services Administration Guide.

Example: Configuring Access Port Security on a PVLAN

IN THIS SECTION

Requirements

- Junos OS Release 18.2R1 or later
- EX4300 switch

Overview

The following configuration section shows:

- Configuration of a private VLAN, with the primary VLAN (**vlan-pri**) and its three secondary VLANs—community VLANs (**vlan-hr** and **vlan-finance**) and isolated VLAN (**vlan-iso**).
- Configuration of the interfaces that are used to send communications between the interfaces on those VLANs.
- Configuration of access security features on the primary and secondary VLANs that make up the PVLAN.

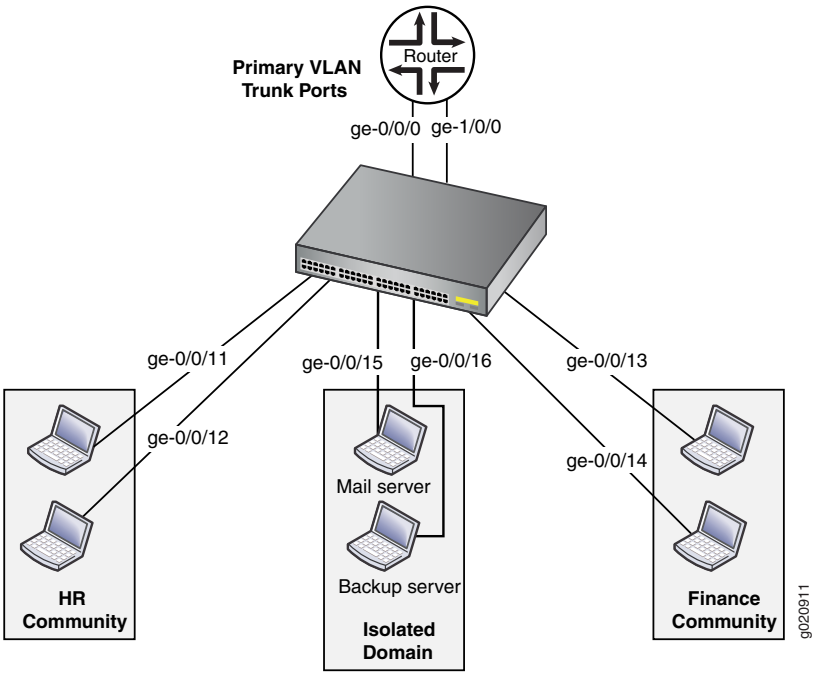


Table 80 on page 464 lists the settings for the example topology.

Table 80: Components of the Topology for Configuring a PVLAN with Access Port Security Features

Interface	Description
ge-0/0/0.0	Primary VLAN (vlan1-pri) trunk interface
ge-0/0/11.0	User 1, HR Community (vlan-hr)
ge-0/0/12.0	User 2, HR Community (vlan-hr)
ge-0/0/13.0	User 3, Finance Community (vlan-finance)
ge-0/0/14.0	User 4, Finance Community (vlan-finance)
ge-0/0/15.0	Mail server, Isolated (vlan-iso)
ge-0/0/16.0	Backup server, Isolated (vlan-iso)
ge-1/0/0.0	Primary VLAN (vlan-pri) trunk interface

Configuring Access Port Security on a PVLAN

CLI Quick Configuration

```
set vlans vlan-pri vlan-id 100
set vlans vlan-hr private-vlan community vlan-id 200
set vlans vlan-finance private-vlan community vlan-id 300
set vlans vlan-iso private-vlan isolated vlan-id 400
set vlans vlan-pri community-vlan vlan-hr
set vlans vlan-pri community-vlan vlan-finance
set vlans vlan-pri isolated-vlan vlan-iso
set interfaces ge-0/0/11 unit 0 family ethernet-switching interface-mode access vlan members vlan-hr
set interfaces ge-0/0/12 unit 0 family ethernet-switching interface-mode trunk vlan members vlan-hr
set interfaces ge-0/0/13 unit 0 family ethernet-switching interface-mode access vlan members vlan-finance
set interfaces ge-0/0/14 unit 0 family ethernet-switching interface-mode trunk vlan members vlan-finance
set interfaces ge-0/0/15 unit 0 family ethernet-switching interface-mode access vlan members vlan-iso
set interfaces ge-0/0/16 unit 0 family ethernet-switching interface-mode access vlan members vlan-iso
set interfaces ge-0/0/0 unit 0 family ethernet-switching interface-mode trunk vlan members vlan-pri
set interfaces ge-1/0/0 unit 0 family ethernet-switching interface-mode trunk vlan members vlan-pri
set vlans vlan-pri forwarding-options dhcp-security arp-inspection
set vlans vlan-pri forwarding-options dhcp-security ip-source-guard
set vlans vlan-pri forwarding-options dhcp-security ipv6-source-guard
set vlans vlan-pri forwarding-options dhcp-security neighbor-discovery-inspection
set vlans vlan-pri forwarding-options dhcp-security option-82
set vlans vlan-pri forwarding-options dhcp-security dhcpv6-options option-16
```

```
set vlans vlan-pri forwarding-options dhcp-security dhcpv6-options light-weight-dhcpv6-relay
set vlans vlan-hr forwarding-options dhcp-security arp-inspection
set vlans vlan-hr forwarding-options dhcp-security ip-source-guard
set vlans vlan-hr forwarding-options dhcp-security ipv6-source-guard
set vlans vlan-hr forwarding-options dhcp-security neighbor-discovery-inspection
set vlans vlan-hr forwarding-options dhcp-security option-82
set vlans vlan-hr forwarding-options dhcp-security dhcpv6-options option-16
set vlans vlan-hr forwarding-options dhcp-security dhcpv6-options light-weight-dhcpv6-relay
set vlans vlan-finance forwarding-options dhcp-security arp-inspection
set vlans vlan-finance forwarding-options dhcp-security ip-source-guard
set vlans vlan-finance forwarding-options dhcp-security ipv6-source-guard
set vlans vlan-finance forwarding-options dhcp-security neighbor-discovery-inspection
set vlans vlan-finance forwarding-options dhcp-security option-82
set vlans vlan-finance forwarding-options dhcp-security dhcpv6-options option-16
set vlans vlan-finance forwarding-options dhcp-security dhcpv6-options light-weight-dhcpv6-relay
set vlans vlan-iso forwarding-options dhcp-security arp-inspection
set vlans vlan-iso forwarding-options dhcp-security ip-source-guard
set vlans vlan-iso forwarding-options dhcp-security ipv6-source-guard
set vlans vlan-iso forwarding-options dhcp-security neighbor-discovery-inspection
set vlans vlan-iso forwarding-options dhcp-security option-82
set vlans vlan-iso forwarding-options dhcp-security dhcpv6-options option-16
set vlans vlan-iso forwarding-options dhcp-security dhcpv6-options light-weight-dhcpv6-relay
```

Step-by-Step Procedure

To configure a private VLAN (PVLAN) and then configure access port security features on that PVLAN:

1. Configure the PVLAN—Create the primary VLAN and its secondary VLANs and assign VLAN IDs to them. Associate interfaces with the VLANs. (For details on configuring VLANs, see *Configuring VLANs for EX Series Switches with ELS Support (CLI Procedure)*.)

   1.
   ```
   [edit vlans]
   user@switch# set vlan-pri vlan-id 100
   user@switch# set vlan-hr private-vlan community vlan-id 200
   user@switch# set vlan-finance private-vlan community vlan-id 300
   user@switch# set vlan-iso private-vlan isolated vlan-id 400
   user@switch# set vlan-pri community-vlan vlan-hr
   user@switch# set vlan-pri community-vlan vlan-finance
   user@switch# set vlan-pri isolated-vlan vlan-iso
   ```

   2.
   ```
   [edit interfaces]
   user@switch# set ge-0/0/11 unit 0 family ethernet-switching interface-mode access vlan members
     vlan-hr
   user@switch# set ge-0/0/12 unit 0 family ethernet-switching interface-mode trunk vlan members vlan-hr
   ```

```
user@switch# set ge-0/0/13 unit 0 family ethernet-switching interface-mode access vlan members
  vlan-finance
user@switch# set ge-0/0/14 unit 0 family ethernet-switching interface-mode trunk vlan members
  vlan-finance
user@switch# set ge-0/0/15 unit 0 family ethernet-switching interface-mode access vlan members
  vlan-iso
user@switch# set ge-0/0/16 unit 0 family ethernet-switching interface-mode access vlan members
  vlan-iso
user@switch# set ge-0/0/0 unit 0 family ethernet-switching interface-mode trunk vlan members vlan-pri
user@switch# set ge-1/0/0 unit 0 family ethernet-switching interface-mode trunk vlan members vlan-pri
```

2. Configure access port security features on the primary VLAN and all its secondary VLANs:

> **NOTE:** When you configure ARP inspection, IP source guard, IPv6 source guard, neighbor discovery inspection, DHCP option 82, or DHCPv6 options, then DHCP snooping and DHCPv6 snooping are automatically configured.

```
[edit vlans]
user@switch# set vlan-pri forwarding-options dhcp-security arp-inspection
user@switch# set vlan-pri forwarding-options dhcp-security ip-source-guard
user@switch# set vlan-pri forwarding-options dhcp-security ipv6-source-guard
user@switch# set vlan-pri forwarding-options dhcp-security neighbor-discovery-inspection
user@switch# set vlan-pri forwarding-options dhcp-security option-82
user@switch# set vlan-pri forwarding-options dhcp-security dhcpv6-options option-16
user@switch# set vlan-pri forwarding-options dhcp-security dhcpv6-options light-weight-dhcpv6-relay
user@switch# set vlan-hr forwarding-options dhcp-security arp-inspection
user@switch# set vlan-hr forwarding-options dhcp-security ip-source-guard
user@switch# set vlan-hr forwarding-options dhcp-security ipv6-source-guard
user@switch# set vlan-hr forwarding-options dhcp-security neighbor-discovery-inspection
user@switch# set vlan-hr forwarding-options dhcp-security option-82
user@switch# set vlan-hr forwarding-options dhcp-security dhcpv6-options option-16
user@switch# set vlan-hr forwarding-options dhcp-security dhcpv6-options light-weight-dhcpv6-relay
user@switch# set vlan-finance forwarding-options dhcp-security arp-inspection
user@switch# set vlan-finance forwarding-options dhcp-security ip-source-guard
user@switch# set vlan-finance forwarding-options dhcp-security ipv6-source-guard
user@switch# set vlan-finance forwarding-options dhcp-security neighbor-discovery-inspection
user@switch# set vlan-finance forwarding-options dhcp-security option-82
user@switch# set vlan-finance forwarding-options dhcp-security dhcpv6-options option-16
user@switch# set vlan-finance forwarding-options dhcp-security dhcpv6-options light-weight-dhcpv6-relay
user@switch# set vlan-iso forwarding-options dhcp-security arp-inspection
user@switch# set vlan-iso forwarding-options dhcp-security ip-source-guard
```

```
user@switch# set vlan-iso forwarding-options dhcp-security ipv6-source-guard
user@switch# set vlan-iso forwarding-options dhcp-security neighbor-discovery-inspection
user@switch# set vlan-iso forwarding-options dhcp-security option-82
user@switch# set vlan-iso forwarding-options dhcp-security dhcpv6-options option-16
user@switch# set vlan-iso forwarding-options dhcp-security dhcpv6-options light-weight-dhcpv6-relay
```

Results

From configuration mode, confirm your configuration by entering the following **show** commands on the switch. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@switch#  show interfaces
ge-0/0/0 {
    unit 0 {
        family ethernet-switching {
            interface-mode trunk;
            vlan {
                members vlan-pri;
            }
        }
    }
}
ge-1/0/0 {
    unit 0 {
        family ethernet-switching {
            interface-mode trunk;
            vlan {
                members vlan-pri;
            }
        }
    }
}
ge-0/0/11 {
    unit 0 {
        family ethernet-switching {
            interface-mode access;
            vlan {
                members vlan-hr;
            }
        }
    }
}
```

```
ge-0/0/12 {
    unit 0 {
        family ethernet-switching {
            interface-mode access;
            vlan {
                members vlan-hr;
            }
        }
    }
}
ge-0/0/13 {
    unit 0 {
        family ethernet-switching {
            interface-mode access;
            vlan {
                members vlan-hr;
            }
        }
    }
}
ge-0/0/14 {
    unit 0 {
        family ethernet-switching {
            interface-mode access;
            vlan {
                members vlan-hr;
            }
        }
    }
}
ge-0/0/15 {
    unit 0 {
        family ethernet-switching {
            interface-mode access;
            vlan {
                members vlan-iso;
            }
        }
    }
}
ge-0/0/16 {
    unit 0 {
        family ethernet-switching {
            interface-mode access;
```

```
            vlan {
                members vlan-iso;
            }
        }
    }
}
user@switch#  show vlans
vlan-finance {
    vlan-id 300;
    private-vlan community;
    interface {
        ge-0/0/13.0;
        ge-0/0/14.0;
    }
    forwarding-options {
        dhcp-security {
            arp-inspection;
                ip-source-guard;
                neighbor-discovery-inspection;
                ipv6-source-guard;
                option-82;
                dhcpv6-options light-weight-dhcpv6-relay;
            dhcpv6-options option-16;
        }
    }
}
vlan-hr {
    vlan-id 200;
    private-vlan community;
    interface {
        ge-0/0/11.0;
        ge-0/0/12.0;
    }
    forwarding-options {
        dhcp-security {
            arp-inspection;
                ip-source-guard;
                neighbor-discovery-inspection;
                ipv6-source-guard;
                option-82;
                dhcpv6-options light-weight-dhcpv6-relay;
            dhcpv6-options option-16;
        }
    }
```

```
}
vlan-iso {
    vlan-id 400;
    private-vlan isolated;
    interface {
        ge-0/0/15.0;
        ge-0/0/16.0;
    }
    forwarding-options {
        dhcp-security {
            arp-inspection;
                ip-source-guard;
                neighbor-discovery-inspection;
                ipv6-source-guard;
                option-82;
                dhcpv6-options light-weight-dhcpv6-relay;
            dhcpv6-options option-16;
        }
    }
}
vlan-pri {
    vlan-id 100;
        community-vlan vlan-finance;
        community-vlan vlan-hr;
    isolated-vlan vlan-iso;
    interface {
        ge-0/0/0.0;
        ge-1/0/0.0;
    }
    forwarding-options {
        dhcp-security {
            arp-inspection;
                ip-source-guard;
                neighbor-discovery-inspection;
                ipv6-source-guard;
                option-82;
                dhcpv6-options light-weight-dhcpv6-relay;
            dhcpv6-options option-16;
        }
    }
}
```

Verification

Verify That Access Security Features Are Working as Expected

Purpose

Verify that the access port security features that you configured on your PVLAN are working as expected.

Action

Use the **show dhcp-security** and the **clear dhcp-security** CLI commands to verify that the features are working as expected. See details about those commands in Security Services Administration Guide.

Creating a Private VLAN on a Single Switch with ELS Support (CLI Procedure)

> **NOTE:** This task uses Junos OS for switches with support for the Enhanced Layer 2 Software (ELS) configuration style. If your EX Series switch runs software that does not support ELS, see "Creating a Private VLAN on a Single EX Series Switch (CLI Procedure)" on page 477. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50.

> **NOTE:** Private VLANs are not supported on QFX5100 switches and QFX10002 switches running Junos OS Release 15.1X53.

For security reasons, it is often useful to restrict the flow of broadcast and unknown unicast traffic or limit the communication between known hosts. Private VLANs (PVLANs) enable you to split a broadcast domain (primary VLAN) into multiple isolated broadcast subdomains (secondary VLANs), essentially putting a VLAN inside a VLAN. This procedure describes how to create a PVLAN on a single switch.

> **NOTE:** You must specify a VLAN ID for each secondary VLAN even if the PVLAN is configured on a single switch.

You do not need to preconfigure the primary VLAN. This topic shows the primary VLAN being configured as part of this PVLAN configuration procedure.

For a list of guidelines on configuring PVLANs, see "Understanding Private VLANs" on page 423.

To configure a private VLAN on a single switch:

1. Set the VLAN ID for the primary VLAN:

   [edit vlans]
   `user@switch#` **set *primary-vlan-name* vlan-id *vlan-id-number***

2. Configure at least one interface within the primary VLAN so that it communicates with all the subdomains of the PVLAN. This interface functions as a *promiscuous* port. It can be either a trunk port or an access port.

   [edit interfaces]
   `user@switch#` **set *interface-name* unit 0 family ethernet-switching**
   `user@switch#` **set *interface-name* unit 0 family ethernet-switching vlan members *primary-vlan-name***

3. Configure another promiscuous interface of the primary VLAN as a trunk port to connect the PVLAN to the external router or switch:

   [edit interfaces]

   `user@switch#` **set *interface-name* unit 0 family ethernet-switching interface-mode trunk**

   `user@switch#` **set *interface-name* unit 0 family ethernet-switching vlan members *primary-vlan-name***

4. Create an isolated VLAN by selecting the **isolated** option for **private-vlan**, and setting a VLAN ID for the isolated VLAN:

   [edit vlans]

   `user@switch#` **set *isolated-vlan-name* private-vlan isolated vlan-id *isolated-vlan-id***

   > **NOTE:** You can create only one isolated VLAN within a private VLAN. Setting the VLAN name for the isolated VLAN is optional. Configuring the VLAN ID is required.

5. Create a community VLAN by selecting the **community** option for **private-vlan**, and setting a VLAN ID for this community VLAN:

   [edit vlans]

   `user@switch#` **set *community-vlan-name* private-vlan community vlan-id *community-vlan-id***

   > **NOTE:** To create additional community VLANs, repeat this step and specify a different name for the community VLAN. Setting the VLAN name for the community VLAN is optional. Configuring the VLAN ID is required.

6. Associate the isolated VLAN with the primary VLAN:

   [edit vlans]

   `user@switch#` **set *primary-vlan-name* vlan-id *primary-vlan-id* isolated-vlan*isolated-vlan-name***

7. Associate each community VLAN with the primary VLAN:

   [edit vlans]

   `user@switch#` **set *primary-vlan-name* vlan-id *primary-vlan-id* community-vlan*community-vlan-name***

8. If you have not already done so, configure at least one interface of the isolated VLAN.

   [edit interfaces]

```
user@switch# set interface-name unit logical-unit-number family ethernet-switching interface-mode
access vlan members isolated-vlan-name
```

9. If you have not already done so, configure at least one interface of the community VLAN.

   [edit interfaces]

   ```
   user@switch# set interface-name unit logical-unit-number family ethernet-switching interface-mode
   access vlan members community-vlan-name
   ```

   > **NOTE:** Repeat the same step on other community VLANs that you want to include in the PVLAN.

Creating a Private VLAN on a Single QFX Switch

For security reasons, it is often useful to restrict the flow of broadcast and unknown unicast traffic and to even limit the communication between known hosts. The private VLAN (PVLAN) feature allows you to split a broadcast domain into multiple isolated broadcast subdomains, essentially putting a secondary VLAN inside a primary VLAN. This topic describes how to configure a PVLAN on a single switch.

Before you begin, configure names for all secondary VLANs that will be part of the primary VLAN. (You do not need to preconfigure the primary VLAN—it is configured as part of this procedure.) You do not need to create VLAN IDs (tags) for the secondary VLANs. It does not impair functioning if you tag the secondary VLANS, but tags are not used when secondary VLANs are configured on a single switch.

Keep these rules in mind when configuring a PVLAN:

- The primary VLAN must be a tagged VLAN.
- If you are going to configure a community VLAN, you must first configure the primary VLAN and the PVLAN trunk port. You must also configure the primary VLAN to be private using the **pvlan** statement.
- If you are going to configure an isolated VLAN, you must first configure the primary VLAN and the PVLAN trunk port.

If you complete your configuration steps in the order shown, you will not violate these PVLAN rules. To configure a private VLAN on a single switch:

1. Set the name and VLAN ID (802.1Q tag) for the primary VLAN:

   ```
   [edit vlans]
   user@switch# set primary-vlan-name vlan-id vlan-id-number
   ```

2. Configure the VLAN to be private:

   ```
   [edit vlans]
   user@switch# set primary-vlan-name pvlan
   ```

3. Configure the trunk interfaces for the primary VLAN:

   ```
   [edit interfaces]
   user@switch# set interface-name unit 0 family ethernet-switching port-mode trunk
   user@switch# set interface-name unit 0 family ethernet-switching vlan members primary-vlan-name
   ```

4. Add the trunk interfaces to the primary VLAN:

   ```
   [edit vlans]
   user@switch# set primary-vlan-name interface interface-name
   ```

5. Configure the access interfaces for the community (secondary) VLANs:

```
[edit interfaces]
user@switch# set interface-name unit 0 family ethernet-switching port-mode access
```

6. Add the access interfaces to the community VLANs:

```
[edit vlans]
user@switch# set community-vlan-name interface interface-name
```

7. For each community VLAN, set the primary VLAN:

```
[edit vlans]
user@switch# set community-vlan-name primary-vlan primary-vlan-name
```

8. Configure isolated ports:

```
[edit vlans]
user@switch# set primary-vlan-name interface interface-name isolated
```

Creating a Private VLAN on a Single EX Series Switch (CLI Procedure)

For security reasons, it is often useful to restrict the flow of broadcast and unknown unicast traffic and to even limit the communication between known hosts. The private VLAN (PVLAN) feature on EX Series switches enables you to split a broadcast domain, also known as a primary VLAN, into multiple isolated broadcast subdomains, also known as secondary VLANs. Splitting the primary VLAN into secondary VLANs essentially nests a VLAN inside another VLAN. This topic describes how to configure a PVLAN on a single switch.

Before you begin, configure names for all secondary VLANs that will be part of the primary VLAN. (Unlike the secondary VLANs, you do not need to preconfigure the primary VLAN—this procedure provides the complete configuration of the primary VLAN.) Although tags are not needed when a secondary VLAN is configured on a single switch, configuring a secondary VLAN as tagged does not adversely affect its functionality. For instructions on configuring the secondary VLANs, see "Configuring VLANs for EX Series Switches" on page 183.

Keep these rules in mind when configuring a PVLAN on a single switch:

- The primary VLAN must be a tagged VLAN.
- Configuring a VoIP VLAN on PVLAN interfaces is not supported.

To configure a private VLAN on a single switch:

1. Set the VLAN ID for the primary VLAN:

   [edit vlans]
   `user@switch#` **set *primary-vlan-name* vlan-id *vlan-id-number***

2. Set the interfaces and port modes:

   [edit interfaces]
   `user@switch#` **set *interface-name* unit 0 family ethernet-switching port-mode *mode***
   `user@switch#` **set *interface-name* unit 0 family ethernet-switching vlan members (all | *vlan-id* | *vlan-number*)**

3. Configure the access ports in the primary VLAN to not forward packets to one another:

   `[edit vlans]`
   `user@switch#` **set vlan-id *vlan-id-number* no-local-switching**

4. For each community VLAN, configure access interfaces:

   `[edit vlans]`

```
user@switch# set community-vlan-name interface-mac-limit interface-name
```

5. For each community VLAN, set the primary VLAN:

```
[edit vlans]
user@switch# set community-vlan-name primary-vlan primary-vlan-name
```

Isolated VLANs are not configured as part of this process. Instead, they are created internally if **no-local-switching** is enabled on the primary VLAN and the isolated VLAN has access interfaces as members.

To optionally enable routing between isolated and community VLANs by using a routed VLAN interface (RVI) instead of a promiscuous port connected to a router, see "Configuring a Routed VLAN Interface in a Private VLAN on an EX Series Switch" on page 723.

> **NOTE:** Only an EX8200 switch or EX8200 Virtual Chassis support the use of an RVI to route Layer 3 traffic between isolated and community VLANs in a PVLAN domain.

Creating a Private VLAN Spanning Multiple QFX Series Switches

For security reasons, it is often useful to restrict the flow of broadcast and unknown unicast traffic and to even limit the communication between known hosts. The private VLAN (PVLAN) feature allows you to split a broadcast domain into multiple isolated broadcast subdomains, essentially putting a secondary VLAN inside a primary VLAN. This topic describes how to configure a PVLAN to span multiple switches.

Before you begin, configure names for all secondary VLANs that will be part of the primary VLAN. (You do not need to preconfigure the primary VLAN—it is configured as part of this procedure.) You do not need to create VLAN IDs (tags) for the secondary VLANs. It does not impair functioning if you tag the secondary VLANS, but tags are not used when secondary VLANs are configured on a single switch.

The following rules apply to creating PVLANs:

- The primary VLAN must be a tagged VLAN.
- If you are going to configure a community VLAN, you must first configure the primary VLAN and the PVLAN trunk port. You must also configure the primary VLAN to be private using the **pvlan** statement.
- If you are going to configure an isolated VLAN, you must first configure the primary VLAN and the PVLAN trunk port.

If you complete your configuration steps in the order shown, you will not violate these PVLAN rules. To configure a private VLAN to span multiple switches:

1. Set the name and VLAN ID (802.1Q tag) for the primary VLAN:

   ```
   [edit vlans]
   user@switch# set primary-vlan-name vlan-id vlan-id-number
   ```

2. Configure the VLAN to be private:

   ```
   [edit vlans]
   user@switch# set primary-vlan-name pvlan
   ```

3. Configure the trunk interfaces for the primary VLAN:

   ```
   [edit interfaces]
   user@switch# set interface-name unit 0 family ethernet-switching port-mode trunk
   user@switch# set interface-name unit 0 family ethernet-switching vlan members primary-vlan-name
   ```

4. Add the trunk interfaces to the primary VLAN:

   ```
   [edit vlans]
   user@switch# set primary-vlan-name interface interface-name
   ```

5. Configure the access interfaces for the community (secondary) VLANs:

```
[edit interfaces]
user@switch# set interface-name unit 0 family ethernet-switching port-mode access
```

6. Add the access interfaces to the community VLANs:

```
[edit vlans]
user@switch# set community-vlan-name interface interface-name
```

7. For each community VLAN, set the primary VLAN:

```
[edit vlans]
user@switch# set community-vlan-name primary-vlan primary-vlan-name
```

8. Configure an isolated VLAN ID to create an interswitch isolated domain that spans the switches:

```
[edit vlans]
user@switch# set primary-vlan-name isolation-vlan-id number
```

9. Configure isolated ports:

```
[edit vlans]
user@switch# set primary-vlan-name interface interface-name isolated
```

Creating a Private VLAN Spanning Multiple EX Series Switches with ELS Support (CLI Procedure)

> **NOTE:** This task uses Junos OS for EX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style If your switch runs software that does not support ELS, see "Creating a Private VLAN Spanning Multiple EX Series Switches (CLI Procedure)" on page 484. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50.

> **NOTE:** Private VLANs are not supported on QFX5100 switches and QFX10002 switches running Junos OS Release 15.1X53.

For security reasons, it is often useful to restrict the flow of broadcast and unknown unicast traffic or limit the communication between known hosts. Private VLANs (PVLANs) enable you to split a broadcast domain (primary VLAN) into multiple isolated broadcast subdomains (secondary VLANs), essentially putting a VLAN inside a VLAN. This procedure describes how to configure a PVLAN to span multiple switches.

For a list of guidelines on configuring PVLANs, see "Understanding Private VLANs" on page 423.

To configure a PVLAN to span multiple switches, perform the following procedure on all the switches that will participate in the PVLAN::

1. Create the primary VLAN by setting the unique VLAN name and specify an 802.1Q tag for the VLAN:

   ```
   [edit vlans]
   user@switch# set primary-vlan-name vlan-id number
   ```

2. On the switch that will connect to a router, configure a promiscuous interface as a trunk port to connect the PVLAN to the router:

   [edit interfaces]

   ```
   user@switch# set interface-name unit 0 family ethernet-switching interface-mode trunk
   user@switch# set interface-name unit 0 family ethernet-switching vlan members primary-vlan-name
   ```

3. On all the switches, configure a trunk interface as the Inter-Switch Link (ISL) that will be used to connect the switches to each other:

   [edit interfaces]

   ```
   user@switch# set interface-name unit 0 family ethernet-switching interface-mode trunk
   inter-switch-link
   ```

```
user@switch# set interface-name unit 0 family ethernet-switching vlan members name-of-primary-vlan
```

4. Create an isolated VLAN within the primary VLAN by selecting the **isolated** option for **private-vlan**, and setting a VLAN ID for the isolated VLAN:

   [edit vlans]

   ```
   user@switch# set isolated-vlan-name private-vlan isolated vlan-id isolated-vlan-id
   ```

   > **NOTE:** You can create only one isolated VLAN within a private VLAN. The isolated VLAN can contain member interfaces from the multiple switches that compose the PVLAN. Setting the VLAN name for the isolated VLAN is optional. Configuring the VLAN ID is required.

5. Create a community VLAN within the primary VLAN by selecting the **community** option for **private-vlan**, and setting a VLAN ID for this community VLAN::

   [edit vlans]

   ```
   user@switch# set community-vlan-name private-vlan community vlan-id community-vlan-id
   ```

   > **NOTE:** To create additional community VLANs, repeat this step and specify a different name for the community VLAN. Setting the VLAN name for the community VLAN is optional. Configuring the VLAN ID is required.

6. Associate the isolated VLAN with the primary VLAN:

   [edit vlans *primary-vlan-name* vlan-id *primary-vlan-id*]

   ```
   user@switch# set isolated-vlanisolated-vlan-name
   ```

7. Associate each community VLAN with the primary VLAN:

   [edit vlans *primary-vlan-name* vlan-id *primary-vlan-id*]

   ```
   user@switch# set community-vlancommunity-vlan-name
   ```

8. If you have not already done so, configure at least one access interface to be a member of the isolated VLAN.

   [edit interface]

```
user@switch# set interface-name unit logical-unit-number family ethernet-switching interface-mode
access vlan members isolated-vlan-name
```

9. If you have not already done so, configure at least one access interface to be a member of the community VLAN.

   [edit interface]

   ```
   user@switch# set interface-name unit logical-unit-number family ethernet-switching interface-mode
   access vlan members community-vlan-name
   ```

   > **NOTE:** Repeat this step for the other community VLANs that you are including in the PVLAN.

Creating a Private VLAN Spanning Multiple EX Series Switches (CLI Procedure)

For security reasons, it is often useful to restrict the flow of broadcast and unknown unicast traffic and to even limit the communication between known hosts. The private VLAN (PVLAN) feature on EX Series switches enables an administrator to split a broadcast domain, also known as a primary VLAN, into multiple isolated broadcast subdomains, also known as secondary VLANs. Splitting the primary VLAN into secondary VLANs essentially nests a VLAN inside another VLAN. This topic describes how to configure a PVLAN to span multiple switches.

Before you begin, configure names for all secondary VLANs that will be part of the primary VLAN. (Unlike the secondary VLANs, you do not need to preconfigure the primary VLAN—this procedure provides the complete configuration of the primary VLAN.) For instructions on configuring the secondary VLANs, see "Configuring VLANs for EX Series Switches" on page 183.

The following rules apply to creating PVLANs:

- The primary VLAN must be a tagged VLAN.
- You must configure the primary VLAN and the PVLAN trunk port before configuring the secondary VLANs.
- Configuring a VoIP VLAN on PVLAN interfaces is not supported.
- If the Multiple VLAN Registration Protocol (MVRP) is configured on the PVLAN trunk port, the configuration of secondary VLANs and the PVLAN trunk port must be committed with the same commit operation.

To configure a private VLAN to span multiple switches:

1. Configure a name and an 802.1Q tag for the primary VLAN:.

   [edit vlans]
   `user@switch#` **set *primary-vlan-name* vlan-id *number***

2. Set the primary VLAN to have no local switching:

   ```
   [edit vlans]
   user@switch# set primary-vlan-name no-local-switching
   ```

3. Set the PVLAN trunk interface that will connect the primary VLAN to the neighboring switch:

   [edit vlans]
   `user@switch#` **set *primary-vlan-name* interface *interface-name* pvlan-trunk**

4. Configure a name and 802.1Q tag for a community VLAN that spans the switches:

[edit vlans]

```
user@switch# set community-vlan-name vlan-id number
```

5. Add access interfaces to the community VLAN:

```
[edit vlans]
user@switch# set community-vlan-name interface interface-name
```

6. Specify the primary VLAN of the specified community VLAN:

```
[edit vlans]
user@switch# set community-vlan-name primary-vlan primary-vlan-name
```

7. Add the isolated interface to the specified primary VLAN:

[edit vlans]

```
user@switch# set primary-vlan-name interface interface-name
```

> **NOTE:** To configure an isolated interface, include it as one of the members of the primary VLAN, but do not configure it as belonging to one of the community VLANs.

8. Set the 802.1Q tag of the interswitch isolated VLAN:

```
[edit vlans]
user@switch# set primary-vlan-name isolation-id number
```

802.1Q tags are required for interswitch isolated VLANs because IEEE 802.1Q uses an internal tagging mechanism by which a trunking device inserts a 4-byte VLAN frame identification tab into the packet header.

To optionally enable routing between isolated and community VLANs by using a routed VLAN interface (RVI) instead of a promiscuous port connected to a router, see "Configuring a Routed VLAN Interface in a Private VLAN on an EX Series Switch" on page 723.

> **NOTE:** Only an EX8200 switch or EX8200 Virtual Chassis support the use of an RVI to route Layer 3 traffic between isolated and community VLANs in a PVLAN domain.

Example: Configuring a Private VLAN on a Single Switch with ELS Support

IN THIS SECTION

NOTE: This example uses Junos OS for switches with support for the Enhanced Layer 2 Software (ELS) configuration style. If your EX switch runs software that does not support ELS, see "Example: Configuring a Private VLAN on a Single EX Series Switch" on page 498. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50.

NOTE: Private VLANs are not supported on QFX5100 switches and QFX10002 switches running Junos OS Release 15.1X53.

For security reasons, it is often useful to restrict the flow of broadcast and unknown unicast traffic or limit the communication between known hosts. Private VLANs (PVLANs) enable you to split a broadcast domain (primary VLAN) into multiple isolated broadcast subdomains (secondary VLANs), essentially putting a VLAN inside a VLAN.

This example describes how to create a PVLAN on a single switch:

Requirements

This example uses the following hardware and software components:

- One Junos OS switch
- Junos OS Release 14.1X53-D10 or later for EX Series switches

  Junos OS Release 14.1X53-D15 or later for QFX Series switches

Overview and Topology

You can isolate groups of subscribers for improved security and efficiency. This configuration example uses a simple topology to illustrate how to create a PVLAN with one primary VLAN and three secondary VLANs (one isolated VLAN, and two community VLANs).

Table 81 on page 487 lists the interfaces of the topology used in the example.

Table 81: Interfaces of the Topology for Configuring a PVLAN

Interface	Description
ge-0/0/0 **ge-1/0/0**	Promiscuous member ports
ge-0/0/11, **ge-0/0/12**	HR community VLAN member ports
ge-0/0/13, **ge-0/0/14**	Finance community VLAN member ports
ge-0/0/15, **ge-0/0/16**	Isolated member ports

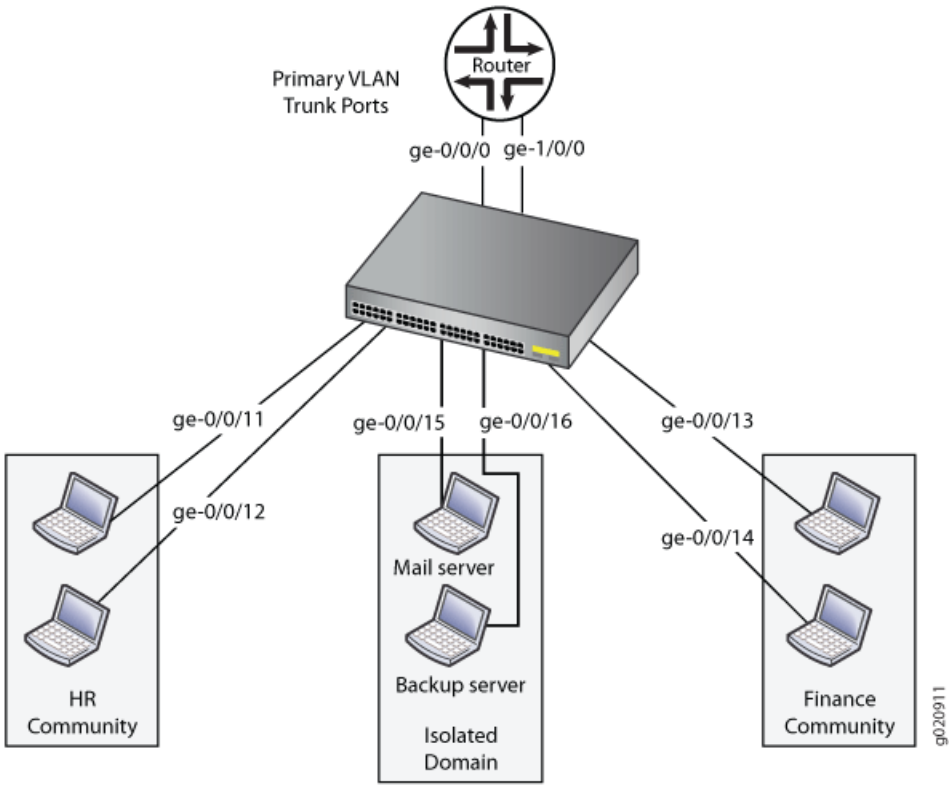
Table 82 on page 487 lists the VLAN IDs of the topology used in the example.

Table 82: VLAN IDs in the Topology for Configuring a PVLAN

VLAN ID	Description
100	Primary VLAN
200	HR community VLAN
300	Finance community VLAN
400	Isolated VLAN

Figure 22 on page 488 shows the topology for this example.

Figure 22: Topology of a Private VLAN on a Single EX Series Switch



Configuration

You can use an existing VLAN as the basis for your private PVLAN and create subdomains within it. This example creates a primary VLAN—using the VLAN name **vlan-pri**—as part of the procedure.

To configure a PVLAN, perform these tasks:

CLI Quick Configuration

To quickly create and configure a PVLAN, copy the following commands and paste them into the switch terminal window:

```
[edit]
```

set vlans vlan-pri vlan-id 100

set vlans vlan-iso private-vlan isolated vlan-id 400

set vlans vlan-hr private-vlan community vlan-id 200

set vlans vlan-finance private-vlan community vlan-id 300

set vlans vlan-pri vlan-id 100 isolated-vlan vlan-iso community-vlan vlan-hr community-vlan vlan-finance

set interface ge-0/0/11 unit 0 family ethernet-switching interface-mode access vlan members vlan-hr

set interface ge-0/0/12 unit 0 family ethernet-switching interface-mode trunk vlan members vlan-hr

set interface ge-0/0/13 unit 0 family ethernet-switching interface-mode access vlan members vlan-finance

set interface ge-0/0/14 unit 0 family ethernet-switching interface-mode trunk vlan members vlan-finance

set interface ge-0/0/15 unit 0 family ethernet-switching interface-mode access vlan members vlan-iso

set interface ge-0/0/16 unit 0 family ethernet-switching interface-mode access vlan members vlan-iso

set interface ge-0/0/0 unit 0 family ethernet-switching interface-mode trunk vlan members vlan-pri

set interface ge-1/0/0 unit 0 family ethernet-switching interface-mode trunk vlan members vlan-pri

Step-by-Step Procedure

To configure the PVLAN:

1. Create the primary VLAN (in this example, the name is **vlan-pri**) of the private VLAN:

   [edit vlans]

   `user@switch#` **set vlan-pri vlan-id 100**

2. Create an isolated VLAN and assign it a VLAN ID:

   [edit vlans]

   `user@switch#` **set vlan-iso private-vlan isolated vlan-id 400**

3. Create the HR community VLAN and assign it a VLAN ID:

   [edit vlans]

   `user@switch#` **set vlan-hr private-vlan community vlan-id 200**

4. Create the finance community VLAN and assign it a VLAN ID:

   [edit vlans]

   `user@switch#` **set vlan-finance private-vlan community vlan-id 300**

5. Associate the secondary VLANs with the primary VLAN:

   [edit vlans]

   `user@switch#` **set vlan-pri vlan-id 100 isolated-vlan vlan-iso community-vlan vlan-hr community-vlan vlan-finance**

6. Set the interfaces to the appropriate interface modes:

```
[edit interfaces]
user@switch# set ge-0/0/11 unit 0 family ethernet-switching interface-mode access vlan members vlan-hr
user@switch# set ge-0/0/12 unit 0 family ethernet-switching interface-mode access vlan members vlan-hr
user@switch# set ge-0/0/13 unit 0 family ethernet-switching interface-mode access vlan members vlan-finance
user@switch# set ge-0/0/14 unit 0 family ethernet-switching interface-mode trunk vlan members vlan-finance
user@switch# set ge-0/0/15 unit 0 family ethernet-switching interface-mode access vlan members vlan-iso
user@switch# set ge-0/0/16 unit 0 family ethernet-switching interface-mode trunk vlan members vlan-iso
```

7. Configure a promiscuous trunk interface of the primary VLAN. This interface is used by the primary VLAN to communicate with the secondary VLANs.

```
user@switch# set ge-0/0/0 unit 0 family ethernet-switching interface-mode trunk vlan members vlan-pri
```

8. Configure another trunk interface (it is also a promiscuous interface) of the primary VLAN, connecting the PVLAN to the router.

```
user@switch# set ge-1/0/0 unit 0 family ethernet-switching interface-mode trunk vlan members vlan-pri
```

Example: Configuring a Private VLAN on a Single QFX Series Switch

IN THIS SECTION

For security reasons, it is often useful to restrict the flow of broadcast and unknown unicast traffic and even to limit the communication between known hosts. The private VLAN (PVLAN) feature allows an administrator to split a broadcast domain into multiple isolated broadcast subdomains, essentially putting a VLAN inside a VLAN.

This example describes how to create a PVLAN on a single switch:

Requirements

This example uses the following hardware and software components:

- One QFX3500 device
- Junos OS Release 12.1 or later for the QFX Series

Before you begin configuring a PVLAN, make sure you have created and configured the necessary VLANs. See "Configuring VLANs on Switches" on page 182.

Overview and Topology

In a large office with multiple buildings and VLANs, you might need to isolate some workgroups or other endpoints for security reasons or to partition the broadcast domain. This configuration example shows a simple topology to illustrate how to create a PVLAN with one primary VLAN and two community VLANs, one for HR and one for finance, as well as two isolated ports—one for the mail server and the other for the backup server.

Table 83 on page 491 lists the settings for the sample topology.

Table 83: Components of the Topology for Configuring a PVLAN

Interface	Description
ge-0/0/0.0	Primary VLAN (**pvlan100**) trunk interface
ge-0/0/11.0	User 1, HR Community (**hr-comm**)
ge-0/0/12.0	User 2, HR Community (**hr-comm**)
ge-0/0/13.0	User 3, Finance Community (**finance-comm**)
ge-0/0/14.0	User 4, Finance Community (**finance-comm**)
ge-0/0/15.0	Mail server, Isolated (**isolated**)
ge-0/0/16.0	Backup server, Isolated (**isolated**)
ge-1/0/0.0	Primary VLAN (**pvlan100**) trunk interface

Configuration

CLI Quick Configuration

To quickly create and configure a PVLAN, copy the following commands and paste them into the switch terminal window:

```
[edit]
set vlans pvlan100 vlan-id 100
set interfaces ge-0/0/0 unit 0 family ethernet-switching port-mode trunk
set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members pvlan
set interfaces ge-1/0/0 unit 0 family ethernet-switching port-mode trunk
set interfaces ge-1/0/0 unit 0 family ethernet-switching vlan members pvlan
set interfaces ge-0/0/11 unit 0 family ethernet-switching port-mode access
set interfaces ge-0/0/12 unit 0 family ethernet-switching port-mode access
set interfaces ge-0/0/13 unit 0 family ethernet-switching port-mode access
set interfaces ge-0/0/14 unit 0 family ethernet-switching port-mode access
set interfaces ge-0/0/15 unit 0 family ethernet-switching port-mode access
set interfaces ge-0/0/16 unit 0 family ethernet-switching port-mode access
set vlans pvlan100 pvlan
set vlans pvlan100 interface ge-0/0/0.0
set vlans pvlan100 interface ge-1/0/0.0
set vlans hr-comm interface ge-0/0/11.0
set vlans hr-comm interface ge-0/0/12.0
set vlans finance-comm interface ge-0/0/13.0
set vlans finance-comm interface ge-0/0/14.0
set vlans hr-comm primary-vlan pvlan100
set vlans finance-comm primary-vlan pvlan100
set pvlan100 interface ge-0/0/15.0 isolated
```

set pvlan100 interface ge-0/0/16.0 isolated

Step-by-Step Procedure

To configure the PVLAN:

1. Set the VLAN ID for the primary VLAN:

   [edit vlans]

   ```
   user@switch#  set pvlan vlan-id 100
   ```

2. Set the interfaces and port modes:

   [edit interfaces]

   ```
   user@switch#  set ge-0/0/0 unit 0 family ethernet-switching port-mode trunk
   user@switch#  set ge-0/0/0 unit 0 family ethernet-switching vlan members pvlan
   user@switch#  set ge-1/0/0 unit 0 family ethernet-switching port-mode trunk
   user@switch#  set ge-1/0/0 unit 0 family ethernet-switching vlan members pvlan
   user@switch#  set ge-0/0/11 unit 0 family ethernet-switching port-mode access
   user@switch#  set ge-0/0/12 unit 0 family ethernet-switching port-mode access
   user@switch#  set ge-0/0/13 unit 0 family ethernet-switching port-mode access
   user@switch#  set ge-0/0/14 unit 0 family ethernet-switching port-mode access
   user@switch#  set ge-0/0/15 unit 0 family ethernet-switching port-mode access
   user@switch#  set ge-0/0/16 unit 0 family ethernet-switching port-mode access
   ```

3. Set the primary VLAN to have no local switching:

   > **NOTE:** The primary VLAN must be a tagged VLAN.

   [edit vlans]

   ```
   user@switch#  set pvlan100 pvlan
   ```

4. Add the trunk interfaces to the primary VLAN:

   [edit vlans]

   ```
   user@switch#  set pvlan100 interface ge-0/0/0.0
   user@switch#  set pvlan100 interface ge-1/0/0.0
   ```

5. For each secondary VLAN, configure access interfaces:

> **NOTE:** We recommend that the secondary VLANs be untagged VLANs. It does not impair functioning if you tag the secondary VLANS. However, the tags are not used when a secondary VLAN is configured on a single switch.

```
[edit vlans]
user@switch# set hr-comm interface ge-0/0/11.0
user@switch# set hr-comm interface ge-0/0/12.0
user@switch# set finance-comm interface ge-0/0/13.0
user@switch# set finance-comm interface ge-0/0/14.0
```

6. For each community VLAN, set the primary VLAN:

```
[edit vlans]
user@switch# set hr-comm primary-vlan pvlan100
user@switch# set finance-comm primary-vlan pvlan100
```

7. Configure the isolated interfaces in the primary VLAN:

```
[edit vlans]
user@switch# set pvlan100 interface ge-0/0/15.0 isolated
user@switch# set pvlan100 interface ge-0/0/16.0 isolated
```

Results

Check the results of the configuration:

```
[edit]
user@switch# show
interfaces {
    ge-0/0/0 {
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
                vlan {
                    members pvlan100;
                }
            }
        }
    }
    ge-1/0/0 {
```

```
        unit 0 {
            family ethernet-switching;
        }
    }
    ge-0/0/11 {
        unit 0 {
            family ethernet-switching {
                port-mode access;
            }
        }
    }
    ge-0/0/12 {
        unit 0 {
            family ethernet-switching {
                port-mode access;
            }
        }
    }
    ge-0/0/13 {
        unit 0 {
            family ethernet-switching {
                port-mode access;
            }
        }
    }
    ge-0/0/14 {
        unit 0 {
            family ethernet-switching {
                port-mode access;
            }
        }
    }
vlans {
    finance-comm {
    interface {
        ge-0/0/13.0;
        ge-0/0/14.0;
    }
    primary-vlan pvlan100;
    }
    hr-comm {
        interface {
            ge-0/0/11.0;
            ge-0/0/12.0;
```

```
        }
        primary-vlan pvlan100;
    }
    pvlan100 {
        vlan-id 100;
        interface {
            ge-0/0/15.0;
            ge-0/0/16.0;
            ge-0/0/0.0;
            ge-1/0/0.0;
        }
        pvlan;
    }
}
```

Verification

IN THIS SECTION

- Verifying That the Private VLAN and Secondary VLANs Were Created | **496**

To confirm that the configuration is working properly, perform these tasks:

Verifying That the Private VLAN and Secondary VLANs Were Created

Purpose

Verify that the primary VLAN and secondary VLANs were properly created on the switch.

Action

Use the **show vlans** command:

```
user@switch> show vlans pvlan100 extensive
```

```
VLAN: pvlan100, Created at: Tue Sep 16 17:59:47 2008
802.1Q Tag: 100, Internal index: 18, Admin State: Enabled, Origin: Static
Private VLAN Mode: Primary
Protocol: Port Mode
Number of interfaces: Tagged 2 (Active = 0), Untagged  6 (Active = 0)
      ge-0/0/0.0, tagged, trunk
      ge-0/0/11.0, untagged, access
```

```
      ge-0/0/12.0, untagged, access
      ge-0/0/13.0, untagged, access
      ge-0/0/14.0, untagged, access
      ge-0/0/15.0, untagged, access
      ge-0/0/16.0, untagged, access
      ge-1/0/0.0, tagged, trunk
Secondary VLANs: Isolated 2, Community  2
  Isolated VLANs :
      __pvlan_pvlan_ge-0/0/15.0__
      __pvlan_pvlan_ge-0/0/16.0__
  Community VLANs :
      finance-comm
      hr-comm
```

user@switch> **show vlans hr-comm extensive**

```
VLAN: hr-comm, Created at: Tue Sep 16 17:59:47 2008
Internal index: 22, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: pvlan100
Protocol: Port Mode
Number of interfaces: Tagged 2 (Active = 0), Untagged  2 (Active = 0)
      ge-0/0/0.0, tagged, trunk
      ge-0/0/11.0, untagged, access
      ge-0/0/12.0, untagged, access
      ge-1/0/0.0, tagged, trunk
```

user@switch> **show vlans finance-comm extensive**

```
VLAN: finance-comm, Created at: Tue Sep 16 17:59:47 2008
Internal index: 21, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: pvlan100
Protocol: Port Mode
Number of interfaces: Tagged 2 (Active = 0), Untagged  2 (Active = 0)
      ge-0/0/0.0, tagged, trunk
      ge-0/0/13.0, untagged, access
      ge-0/0/14.0, untagged, access
      ge-1/0/0.0, tagged, trunk
```

user@switch> **show vlans __pvlan_pvlan_ge-0/0/15.0__ extensive**

```
VLAN: __pvlan_pvlan_ge-0/0/15.0__, Created at: Tue Sep 16 17:59:47 2008
Internal index: 19, Admin State: Enabled, Origin: Static
```

```
Private VLAN Mode: Isolated, Primary VLAN: pvlan100
Protocol: Port Mode
Number of interfaces: Tagged 2 (Active = 0), Untagged  1 (Active = 0)
      ge-0/0/0.0, tagged, trunk
      ge-0/0/15.0, untagged, access
      ge-1/0/0.0, tagged, trunk
```

user@switch> **show vlans __pvlan_pvlan_ge-0/0/16.0__ extensive**

```
VLAN: __pvlan_pvlan_ge-0/0/16.0__, Created at: Tue Sep 16 17:59:47 2008
Internal index: 20, Admin State: Enabled, Origin: Static
Private VLAN Mode: Isolated, Primary VLAN: pvlan100
Protocol: Port Mode
Number of interfaces: Tagged 2 (Active = 0), Untagged  1 (Active = 0)
      ge-0/0/0.0, tagged, trunk
      ge-0/0/16.0, untagged, access
      ge-1/0/0.0, tagged, trunk
```

Meaning

The output shows that the primary VLAN was created and identifies the interfaces and secondary VLANs associated with it.

Example: Configuring a Private VLAN on a Single EX Series Switch

IN THIS SECTION

For security reasons, it is often useful to restrict the flow of broadcast and unknown unicast traffic and to even limit the communication between known hosts. The private VLAN (PVLAN) feature on EX Series switches allows an administrator to split a broadcast domain into multiple isolated broadcast subdomains, essentially putting a VLAN inside a VLAN.

This example describes how to create a PVLAN on a single EX Series switch:

> **NOTE:** Configuring a voice over IP (VoIP) VLAN on PVLAN interfaces is not supported.

Requirements

This example uses the following hardware and software components:

- One EX Series switch
- Junos OS Release 9.3 or later for EX Series switches

Before you begin configuring a PVLAN, make sure you have created and configured the necessary VLANs. See "Configuring VLANs for EX Series Switches" on page 183.

Overview and Topology

In a large office with multiple buildings and VLANs, you might need to isolate some workgroups or other endpoints for security reasons or to partition the broadcast domain. This configuration example shows a simple topology to illustrate how to create a PVLAN with one primary VLAN and two community VLANs, one for HR and one for finance, as well as two isolated ports—one for the mail server and the other for the backup server.

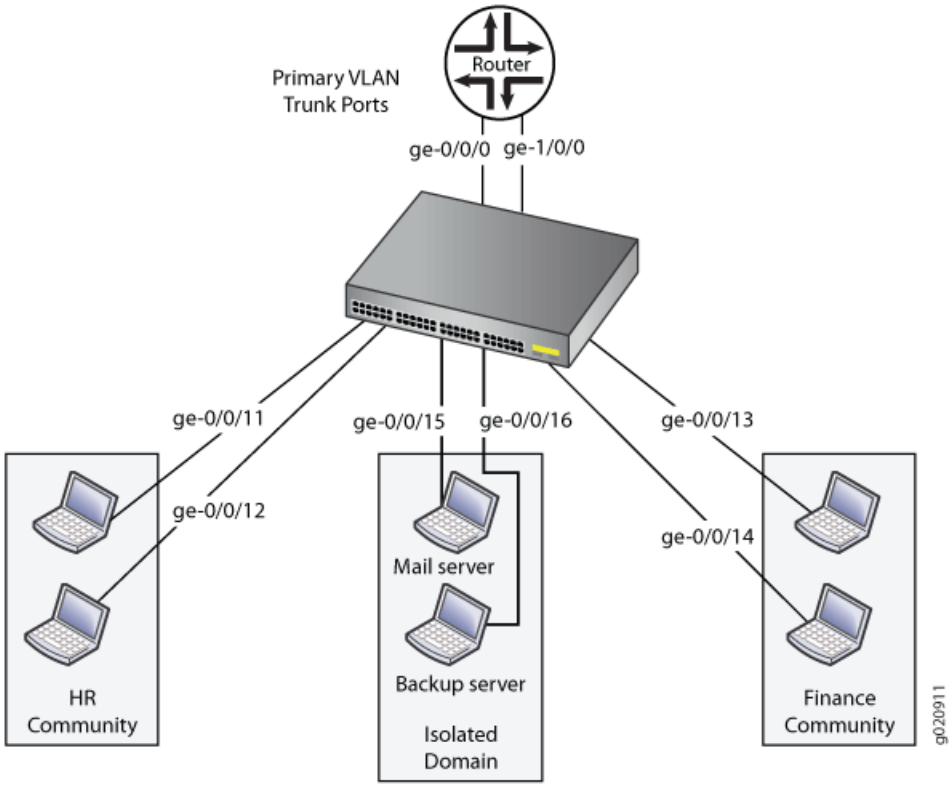
Table 84 on page 499 lists the settings for the example topology.

Table 84: Components of the Topology for Configuring a PVLAN

Interface	Description
ge-0/0/0.0	Primary VLAN (**vlan1**) trunk interface
ge-0/0/11.0	User 1, HR Community (**hr-comm**)
ge-0/0/12.0	User 2, HR Community (**hr-comm**)
ge-0/0/13.0	User 3, Finance Community (**finance-comm**)
ge-0/0/14.0	User 4, Finance Community (**finance-comm**)
ge-0/0/15.0	Mail server, Isolated (**isolated**)
ge-0/0/16.0	Backup server, Isolated (**isolated**)
ge-1/0/0.0	Primary VLAN (**pvlan**) trunk interface

Figure 23 on page 500 shows the topology for this example.

Figure 23: Topology of a Private VLAN on a Single EX Series Switch



Configuration

To configure a PVLAN, perform these tasks:

CLI Quick Configuration

To quickly create and configure a PVLAN, copy the following commands and paste them into the switch terminal window:

```
[edit]
```

set vlans vlan1 vlan-id 1000

set interfaces ge-0/0/0 unit 0 family ethernet-switching port-mode trunk

set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members vlan1

set interfaces ge-1/0/0 unit 0 family ethernet-switching port-mode trunk

set interfaces ge-1/0/0 unit 0 family ethernet-switching vlan members vlan1

set interfaces ge-0/0/11 unit 0 family ethernet-switching port-mode access

set interfaces ge-0/0/12 unit 0 family ethernet-switching port-mode access

set interfaces ge-0/0/13 unit 0 family ethernet-switching port-mode access

set interfaces ge-0/0/14 unit 0 family ethernet-switching port-mode access

set interfaces ge-0/0/15 unit 0 family ethernet-switching port-mode access

set interfaces ge-0/0/16 unit 0 family ethernet-switching port-mode access

set vlans vlan1 no-local-switching

set vlans vlan1 interface ge-0/0/0.0

set vlans vlan1 interface ge-1/0/0.0

set vlans hr-comm vlan-id 400

set vlans hr-comm interface ge-0/0/11.0

set vlans hr-comm interface ge-0/0/12.0

set vlans finance-comm vlan-id 300

set vlans finance-comm interface ge-0/0/13.0

set vlans finance-comm interface ge-0/0/14.0

set vlans hr-comm primary-vlan vlan1

set vlans finance-comm primary-vlan vlan1

set vlans vlan1 interface ge-0/0/15.0

set vlans vlan1 interface ge-0/0/16.0

Step-by-Step Procedure

To configure the PVLAN:

1. Set the VLAN ID for the primary VLAN:

   [edit vlans]

   ```
   user@switch# set vlan1 vlan-id 1000
   ```

2. Set the interfaces and port modes:

   [edit interfaces]

```
user@switch# set ge-0/0/0 unit 0 family ethernet-switching port-mode trunk
user@switch# set ge-0/0/0 unit 0 family ethernet-switching vlan members pvlan
user@switch# set ge-1/0/0 unit 0 family ethernet-switching port-mode trunk
user@switch# set ge-1/0/0 unit 0 family ethernet-switching vlan members vlan1
user@switch# set ge-0/0/11 unit 0 family ethernet-switching port-mode access
user@switch# set ge-0/0/12 unit 0 family ethernet-switching port-mode access
user@switch# set ge-0/0/13 unit 0 family ethernet-switching port-mode access
user@switch# set ge-0/0/14 unit 0 family ethernet-switching port-mode access
user@switch# set ge-0/0/15 unit 0 family ethernet-switching port-mode access
user@switch# set ge-0/0/16 unit 0 family ethernet-switching port-mode access
```

3. Set the primary VLAN to have no local switching:

> **NOTE:** The primary VLAN must be a tagged VLAN.

```
[edit vlans]
user@switch# set vlan1 no-local-switching
```

4. Add the trunk interfaces to the primary VLAN:

```
[edit vlans]
user@switch# set vlan1 interface ge-0/0/0.0
user@switch# set vlan1 interface ge-1/0/0.0
```

5. For each secondary VLAN, configure the VLAN IDs and the access interfaces:

> **NOTE:** We recommend that the secondary VLANs be untagged VLANs. It does not impair functioning if you tag the secondary VLANS. However, the tags are not used when a secondary VLAN is configured on a single switch.

```
[edit vlans]
user@switch# set hr-comm vlan-id 400
user@switch# set hr-comm interface ge-0/0/11.0
user@switch# set hr-comm interface ge-0/0/12.0
user@switch# set finance-comm vlan-id 300
user@switch# set finance-comm interface ge-0/0/13.0
```

```
user@switch# set finance-comm interface ge-0/0/14.0
```

6. For each community VLAN, set the primary VLAN:

```
[edit vlans]
user@switch# set hr-comm primary-vlan vlan1
user@switch# set finance-comm primary-vlan vlan1
```

7. Add each isolated interface to the primary VLAN:

```
[edit vlans]
user@switch# set vlan1 interface ge-0/0/15.0
user@switch# set vlan1 interface ge-0/0/16.0
```

Results

Check the results of the configuration:

```
[edit]
user@switch# show
interfaces {
    ge-0/0/0 {
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
                vlan {
                    members vlan1;
                }
            }
        }
    }
    ge-1/0/0 {
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
                vlan {
                    members vlan1;
                }
            }
        }
    }
    ge-0/0/11 {
        unit 0 {
            family ethernet-switching {
```

```
            port-mode access;
          }
        }
      }
      ge-0/0/12 {
        unit 0 {
          family ethernet-switching {
            port-mode access;
          }
        }
      }
      ge-0/0/13 {
        unit 0 {
          family ethernet-switching {
            port-mode access;
          }
        }
      }
      ge-0/0/14 {
        unit 0 {
          family ethernet-switching {
            port-mode access;
          }
        }
      }
    vlans {
      finance-comm {
        vlan-id 300;
        interface {
          ge-0/0/13.0;
          ge-0/0/14.0;
        }
        primary-vlan vlan1;
      }
      hr-comm {
        vlan-id 400;
        interface {
          ge-0/0/11.0;
          ge-0/0/12.0;
        }
        primary-vlan vlan1;
      }
      vlan1 {
        vlan-id 1000;
```

```
        interface {
            ge-0/0/15.0;
            ge-0/0/16.0;
            ge-0/0/0.0;
            ge-1/0/0.0;
        }
        no-local-switching;
    }
}
```

Verification

IN THIS SECTION

- Verifying That the Private VLAN and Secondary VLANs Were Created | 505

To confirm that the configuration is working properly, perform these tasks:

Verifying That the Private VLAN and Secondary VLANs Were Created

Purpose

Verify that the primary VLAN and secondary VLANs were properly created on the switch.

Action

Use the **show vlans** command:

```
user@switch> show vlans vlan1 extensive
```

```
VLAN: vlan1, Created at: Tue Sep 16 17:59:47 2008
802.1Q Tag: 1000, Internal index: 18, Admin State: Enabled, Origin: Static
Private VLAN Mode: Primary
Protocol: Port Mode
Number of interfaces: Tagged 2 (Active = 0), Untagged  6 (Active = 0)
      ge-0/0/0.0, tagged, trunk
      ge-0/0/11.0, untagged, access
      ge-0/0/12.0, untagged, access
      ge-0/0/13.0, untagged, access
      ge-0/0/14.0, untagged, access
      ge-0/0/15.0, untagged, access
      ge-0/0/16.0, untagged, access
```

```
      ge-1/0/0.0, tagged, trunk
Secondary VLANs: Isolated 2, Community  2
  Isolated VLANs :
      __vlan1_vlan1_ge-0/0/15.0__
      __vlan1_vlan1_ge-0/0/16.0__
  Community VLANs :
      finance-comm
      hr-comm
```

user@switch> **show vlans hr-comm extensive**

```
VLAN: hr-comm, Created at: Tue Sep 16 17:59:47 2008
802.1Q Tag: 400,Internal index: 22, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: vlan1
Protocol: Port Mode
Number of interfaces: Tagged 2 (Active = 0), Untagged  2 (Active = 0)
      ge-0/0/0.0, tagged, trunk
      ge-0/0/11.0, untagged, access
      ge-0/0/12.0, untagged, access
      ge-1/0/0.0, tagged, trunk
```

user@switch> **show vlans finance-comm extensive**

```
VLAN: finance-comm, Created at: Tue Sep 16 17:59:47 2008
802.1Q Tag: 300,Internal index: 21, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: vlan1
Protocol: Port Mode
Number of interfaces: Tagged 2 (Active = 0), Untagged  2 (Active = 0)
      ge-0/0/0.0, tagged, trunk
      ge-0/0/13.0, untagged, access
      ge-0/0/14.0, untagged, access
      ge-1/0/0.0, tagged, trunk
```

user@switch> **show vlans __vlan1_vlan1_ge-0/0/15.0__ extensive**

```
VLAN: __vlan1_vlan1_ge-0/0/15.0__, Created at: Tue Sep 16 17:59:47 2008
Internal index: 19, Admin State: Enabled, Origin: Static
Private VLAN Mode: Isolated, Primary VLAN: vlan1
Protocol: Port Mode
Number of interfaces: Tagged 2 (Active = 0), Untagged  1 (Active = 0)
      ge-0/0/0.0, tagged, trunk
```

```
      ge-0/0/15.0, untagged, access
      ge-1/0/0.0, tagged, trunk
```

```
user@switch> show vlans __vlan1_vlan1_ge-0/0/16.0__ extensive
```

```
VLAN: __vlan1_vlan1_ge-0/0/16.0__, Created at: Tue Sep 16 17:59:47 2008
Internal index: 20, Admin State: Enabled, Origin: Static
Private VLAN Mode: Isolated, Primary VLAN: vlan1
Protocol: Port Mode
Number of interfaces: Tagged 2 (Active = 0), Untagged  1 (Active = 0)
      ge-0/0/0.0, tagged, trunk
      ge-0/0/16.0, untagged, access
      ge-1/0/0.0, tagged, trunk
```

Meaning

The output shows that the primary VLAN was created and identifies the interfaces and secondary VLANs associated with it.

Example: Configuring a Private VLAN Spanning Multiple QFX Switches

IN THIS SECTION

For security reasons, it is often useful to restrict the flow of broadcast and unknown unicast traffic and even to limit the communication between known hosts. The private VLAN (PVLAN) feature allows an administrator to split a broadcast domain into multiple isolated broadcast subdomains, essentially putting a VLAN inside a VLAN. A PVLAN can span multiple switches.

This example describes how to create a PVLAN spanning multiple switches. The example creates one primary PVLAN containing multiple secondary VLANs:

Requirements

This example uses the following hardware and software components:

- Three QFX3500 devices
- Junos OS Release 12.1 or later for the QFX Series

Before you begin configuring a PVLAN, make sure you have created and configured the necessary VLANs. See “Configuring VLANs on Switches” on page 182.

Overview and Topology

In a large office with multiple buildings and VLANs, you might need to isolate some workgroups or other endpoints for security reasons or to partition the broadcast domain. This configuration example shows how to create a PVLAN spanning multiple QFX devices, with one primary VLAN containing two community VLANs (one for HR and one for Finance), and an interswitch isolated VLAN (for the mail server, the backup server, and the CVS server). The PVLAN comprises three switches, two access switches and one distribution switch. The PVLAN is connected to a router through a promiscuous port, which is configured on the distribution switch.

> **NOTE:** The isolated ports on Switch 1 and on Switch 2 do not have Layer 2 connectivity with one another even though they are included within the same domain. See “Understanding Private VLANs” on page 423.

Figure 24 on page 509 shows the topology for this example—two access switches connecting to a distribution switch, which has a connection (through a promiscuous port) to the router.

Figure 24: PVLAN Topology Spanning Multiple Switches

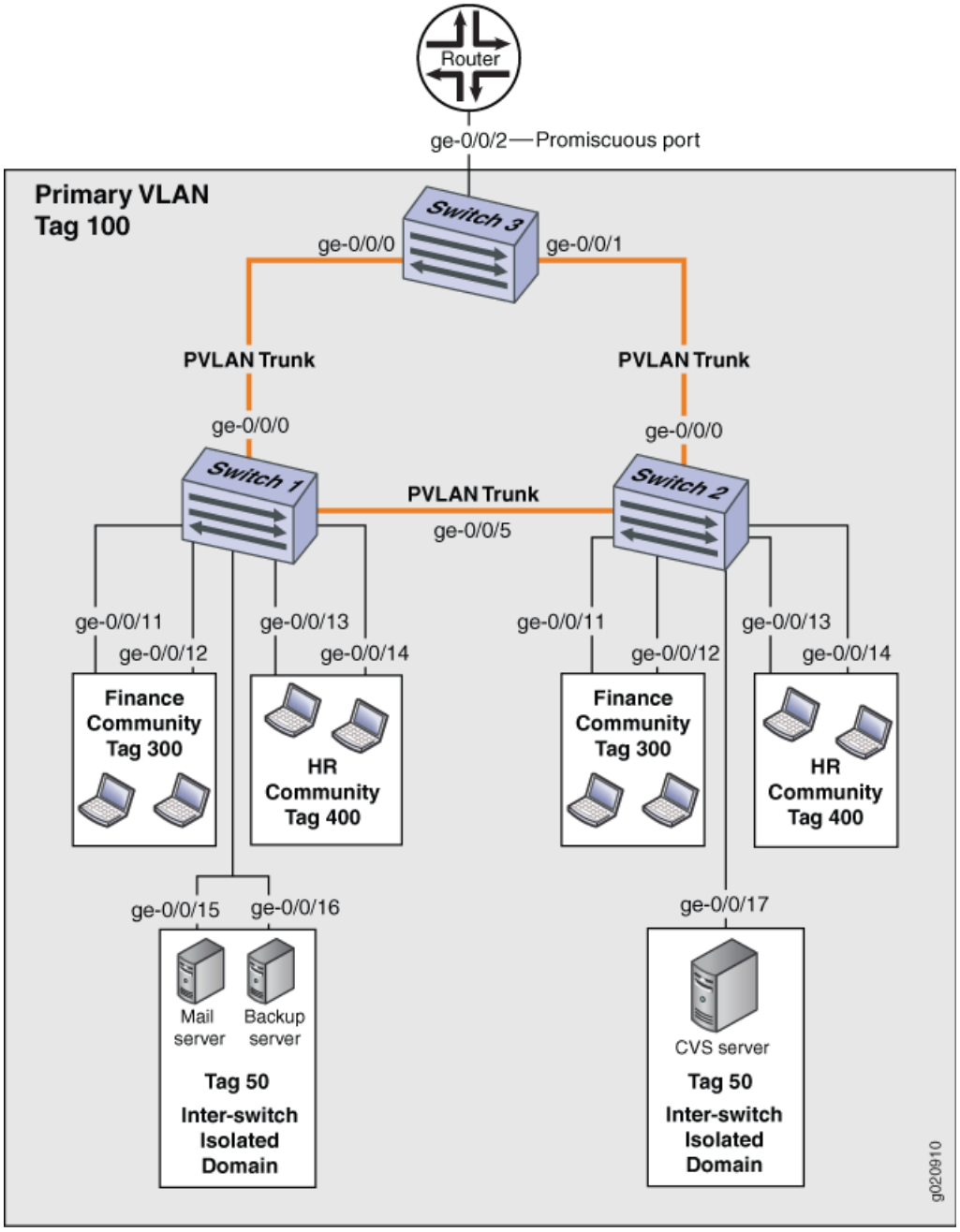


Table 85 on page 510, Table 86 on page 510, and Table 87 on page 511 list the settings for the example topology.

Table 85: Components of Switch 1 in the Topology for Configuring a PVLAN Spanning Multiple Devices

Property	Settings
VLAN names and tag IDs	**primary-vlan**, tag **100** **isolation-vlan-id**, tag **50** **finance-comm**, tag **300** **hr-comm**, tag **400**
PVLAN trunk interfaces	**ge-0/0/0.0**, connects Switch 1 to Switch 3 **ge-0/0/5.0**, connects Switch 1 to Switch 2
Isolated Interfaces in primary VLAN	**ge-0/0/15.0**, mail server **ge-0/0/16.0**, backup server
Interfaces in VLAN **finance-com**	**ge-0/0/11.0** **ge-0/0/12.0**
Interfaces in VLAN **hr-comm**	**ge-0/0/13.0** **ge-0/0/14.0**

Table 86: Components of Switch 2 in the Topology for Configuring a PVLAN Spanning Multiple Devices

Property	Settings
VLAN names and tag IDs	**primary-vlan**, tag **100** **isolation-vlan-id**, tag **50** **finance-comm**, tag **300** **hr-comm**, tag **400**
PVLAN trunk interfaces	**ge-0/0/0.0**, connects Switch 2 to Switch 3 **ge-0/0/5.0**, connects Switch 2 to Switch 1
Isolated Interface in primary VLAN	**ge-0/0/17.0**, CVS server
Interfaces in VLAN **finance-com**	**ge-0/0/11.0** **ge-0/0/12.0**
Interfaces in VLAN **hr-comm**	**ge-0/0/13.0** **ge-0/0/14.0**

Table 87: Components of Switch 3 in the Topology for Configuring a PVLAN Spanning Multiple Devices

Property	Settings
VLAN names and tag IDs	**primary-vlan**, tag **100** **isolation-vlan-id**, tag **50** **finance-comm**, tag **300** **hr-comm**, tag **400**
PVLAN trunk interfaces	**ge-0/0/0.0**, connects Switch 3 to Switch 1 **ge-0/0/1.0**, connects Switch 3 to Switch 2
Promiscuous port	**ge-0/0/2**, connects the PVLAN to the router **NOTE:** You must configure the trunk port that connects the PVLAN to another switch or router outside the PVLAN as a member of the PVLAN, which implicitly configures it as a promiscuous port.

Configuring a PVLAN on Switch 1

When configuring a PVLAN on multiple switches, these rules apply:

- The primary VLAN must be a tagged VLAN. We recommend that you configure the primary VLAN first.
- If you are going to configure a community VLAN ID, you must first configure the primary VLAN and the PVLAN trunk port. You must also configure the primary VLAN to be private using the **pvlan** statement.
- If you are going to configure an isolation VLAN ID, you must first configure the primary VLAN and the PVLAN trunk port.

CLI Quick Configuration

To quickly create and configure a PVLAN spanning multiple switches, copy the following commands and paste them into the terminal window of Switch 1:

```
[edit]
```

set vlans finance-comm vlan-id 300

set vlans finance-comm interface ge-0/0/11.0

set vlans finance-comm interface ge-0/0/12.0

set vlans finance-comm primary-vlan pvlan100

set vlans hr-comm vlan-id 400

set vlans hr-comm interface ge-0/0/13.0

set vlans hr-comm interface ge-0/0/14.0

set vlans hr-comm primary-vlan pvlan100

set vlans pvlan100 vlan-id 100

set vlans pvlan100 interface ge-0/0/15.0

set vlans pvlan100 interface ge-0/0/16.0

set vlans pvlan100 interface ge-0/0/0.0 pvlan-trunk

set vlans pvlan100 interface ge-0/0/5.0 pvlan-trunk

set vlans pvlan100 pvlan

set vlans pvlan100 pvlan isolation-vlan-id 50

set pvlan100 interface ge-0/0/15.0 isolated

set pvlan100 interface ge-0/0/16.0 isolated

Step-by-Step Procedure

1. Set the VLAN ID for the primary VLAN:

   [edit vlans]
   `user@switch#` **set pvlan100 vlan-id 100**

2. Set the PVLAN trunk interfaces to connect this VLAN across neighboring switches:

   [edit vlans]
   `user@switch#` **set pvlan100 interface ge-0/0/0.0 pvlan-trunk**
   `user@switch#` **set pvlan100 interface ge-0/0/5.0 pvlan-trunk**

3. Set the primary VLAN to be private and have no local switching:

   [edit vlans]
   `user@switch#` **set pvlan100 pvlan**

4. Set the VLAN ID for the **finance-comm** community VLAN that spans the switches:

   [edit vlans]
   `user@switch#` **set finance-comm vlan-id 300**

5. Configure access interfaces for the **finance-comm** VLAN:

   [edit vlans]
   `user@switch#` **set finance-comm interface ge-0/0/11.0**

   `user@switch#` **set finance-comm interface ge-0/0/12.0**

6. Set the primary VLAN of this secondary community VLAN, **finance-comm** :

   [edit vlans]
   `user@switch#` **set vlans finance-comm primary-vlan pvlan100**

7. Set the VLAN ID for the HR community VLAN that spans the switches.

   [edit vlans]
   `user@switch#` **set hr-comm vlan-id 400**

8. Configure access interfaces for the **hr-comm** VLAN:

   [edit vlans]
   `user@switch#` **set hr-comm interface ge-0/0/13.0**
   `user@switch#` **set hr-comm interface ge-0/0/14.0**

9. Set the primary VLAN of this secondary community VLAN, **hr-comm**:

   [edit vlans]
   `user@switch#` **set vlans hr-comm primary-vlan pvlan100**

10. Set the interswitch isolated ID to create an interswitch isolated domain that spans the switches:

    [edit vlans]
    `user@switch#` **set pvlan100 pvlan isolation-vlan-id 50**

11. Configure the isolated interfaces in the primary VLAN:

    [edit vlans]
    `user@switch#` **set pvlan100 interface ge-0/0/15.0 isolated**
    `user@switch#` **set pvlan100 interface ge-0/0/16.0 isolated**

> **NOTE:** When you configure an isolated port, include it as a member of the primary VLAN, but do not configure it as a member of any community VLAN.

Results

Check the results of the configuration:

```
[edit]
   user@switch# show
vlans {
   finance-comm {
      vlan-id 300;
      interface {
         ge-0/0/11.0;
         ge-0/0/12.0;
      }
      primary-vlan pvlan100;
   }
   hr-comm {
      vlan-id 400;
      interface {
         ge-0/0/13.0;
         ge-0/0/14.0;
      }
      primary-vlan pvlan100;
   }
   pvlan100 {
      vlan-id 100;
      interface {
         ge-0/0/15.0;
         ge-0/0/16.0;
         ge-0/0/0.0 {
            pvlan-trunk;
         }
         ge-0/0/5.0 {
            pvlan-trunk;
         }
      }
      pvlan;
      isolation-vlan-id 50;
   }
}
```

Configuring a PVLAN on Switch 2

CLI Quick Configuration

To quickly create and configure a private VLAN spanning multiple switches, copy the following commands and paste them into the terminal window of Switch 2:

NOTE: The configuration of Switch 2 is the same as the configuration of Switch 1 except for the interface in the interswitch isolated domain. For Switch 2, the interface is **ge-0/0/17.0**.

```
[edit]
set vlans finance-comm vlan-id 300
set vlans finance-comm interface ge-0/0/11.0
set vlans finance-comm interface ge-0/0/12.0
set vlans finance-comm primary-vlan pvlan100
set vlans hr-comm vlan-id 400
set vlans hr-comm interface ge-0/0/13.0
set vlans hr-comm interface ge-0/0/14.0
set vlans hr-comm primary-vlan pvlan100
set vlans pvlan100 vlan-id 100
set vlans pvlan100 interface ge-0/0/17.0
set vlans pvlan100 interface ge-0/0/0.0 pvlan-trunk
set vlans pvlan100 interface ge-0/0/5.0 pvlan-trunk
set vlans pvlan100 pvlan
set vlans pvlan100 pvlan isolation-vlan-id 50
set pvlan100 interface ge-0/0/17.0 isolated
```

Step-by-Step Procedure

To configure a PVLAN on Switch 2 that will span multiple switches:

1. Set the VLAN ID for the **finance-comm** community VLAN that spans the switches:

   [edit vlans]
   `user@switch#` **set finance-comm vlan-id 300**

2. Configure access interfaces for the **finance-comm** VLAN:

[edit vlans]

```
user@switch# set finance-comm interface ge-0/0/11.0

user@switch# set finance-comm interface ge-0/0/12.0
```

3. Set the primary VLAN of this secondary community VLAN, **finance-comm**:

   [edit vlans]

   ```
   user@switch# set vlans finance-comm primary-vlan pvlan100
   ```

4. Set the VLAN ID for the HR community VLAN that spans the switches.

   [edit vlans]

   ```
   user@switch# set hr-comm vlan-id 400
   ```

5. Configure access interfaces for the **hr-comm** VLAN:

   [edit vlans]

   ```
   user@switch# set hr-comm interface ge-0/0/13.0
   user@switch# set hr-comm interface ge-0/0/14.0
   ```

6. Set the primary VLAN of this secondary community VLAN, **hr-comm**:

   [edit vlans]

   ```
   user@switch# set vlans hr-comm primary-vlan pvlan100
   ```

7. Set the VLAN ID for the primary VLAN:

   [edit vlans]

   ```
   user@switch# set pvlan100 vlan-id 100
   ```

8. Set the PVLAN trunk interfaces that will connect this VLAN across neighboring switches:

   [edit vlans]

   ```
   user@switch# set pvlan100 interface ge-0/0/0.0 pvlan-trunk
   user@switch# set pvlan100 interface ge-0/0/5.0 pvlan-trunk
   ```

9. Set the primary VLAN to be private and have no local switching:

   [edit vlans]

   ```
   user@switch# set pvlan100 pvlan
   ```

10. Set the interswitch isolated ID to create an interswitch isolated domain that spans the switches:

[edit vlans]

```
user@switch# set pvlan100 pvlan isolation-vlan-id 50
```

> **NOTE:** To configure an isolated port, include it as one of the members of the primary VLAN, but do not configure it as belonging to one of the community VLANs.

11. Configure the isolated interface in the primary VLAN:

[edit vlans]

```
user@switch# set pvlan100 interface ge-0/0/17.0 isolated
```

Results

Check the results of the configuration:

```
[edit]
user@switch# show
vlans {
    finance-comm {
        vlan-id 300;
        interface {
            ge-0/0/11.0;
            ge-0/0/12.0;
        }
        primary-vlan pvlan100;
    }
    hr-comm {
        vlan-id 400;
        interface {
            ge-0/0/13.0;
            ge-0/0/14.0;
        }
        primary-vlan pvlan100;
    }
    pvlan100 {
        vlan-id 100;
        interface {
            ge-0/0/15.0;
            ge-0/0/16.0;
            ge-0/0/0.0 {
                pvlan-trunk;
```

```
            }
            ge-0/0/5.0 {
                pvlan-trunk;
            }
            ge-0/0/17.0;
        }
        pvlan;
        isolation-vlan-id 50;
    }
}
```

Configuring a PVLAN on Switch 3

CLI Quick Configuration

To quickly configure Switch 3 to function as the distribution switch of this PVLAN, copy the following commands and paste them into the terminal window of Switch 3:

> **NOTE:** Interface ge-0/0/2.0 is a trunk port connecting the PVLAN to a router.

```
[edit]
set vlans finance-comm vlan-id 300
set vlans finance-comm primary-vlan pvlan100
set vlans hr-comm vlan-id 400
set vlans hr-comm primary-vlan pvlan100
set vlans pvlan100 vlan-id 100
set vlans pvlan100 interface ge-0/0/0.0 pvlan-trunk
set vlans pvlan100 interface ge-0/0/1.0 pvlan-trunk
set vlans pvlan100 pvlan
set vlans pvlan100 pvlan isolation-vlan-id 50
```

Step-by-Step Procedure

To configure Switch 3 to function as the distribution switch for this PVLAN, use the following procedure:

1. Set the VLAN ID for the **finance-comm** community VLAN that spans the switches:

   [edit vlans]
   `user@switch#` **finance-comm vlan-id 300**

2. Set the primary VLAN of this secondary community VLAN, **finance-comm**:

   [edit vlans]
   `user@switch#` **set vlans finance-comm primary-vlan pvlan100**

3. Set the VLAN ID for the HR community VLAN that spans the switches:

   [edit vlans]
   `user@switch#` **set hr-comm vlan-id 400**

4. Set the primary VLAN of this secondary community VLAN, **hr-comm**:

   [edit vlans]
   `user@switch#` **set vlans hr-comm primary-vlan pvlan100**

5. Set the VLAN ID for the primary VLAN:

   [edit vlans]
   `user@switch#` **set pvlan100 vlan-id 100**

6. Set the PVLAN trunk interfaces that will connect this VLAN across neighboring switches:

   [edit vlans]
   `user@switch#` **set pvlan100 interface ge-0/0/0.0 pvlan-trunk**
   `user@switch#` **set pvlan100 interface ge-0/0/5.0 pvlan-trunk**

7. Set the primary VLAN to be private and have no local switching:

   [edit vlans]
   `user@switch#` **set pvlan100 pvlan**

8. Set the interswitch isolated ID to create an interswitch isolated domain that spans the switches:

   [edit vlans]
   `user@switch#` **set pvlan100 pvlan isolation-vlan-id 50**

NOTE: To configure an isolated port, include it as one of the members of the primary VLAN, but do not configure it as belonging to one of the community VLANs.

Results

Check the results of the configuration:

```
[edit]
user@switch# show
vlans {
    finance-comm {
        vlan-id 300;
        primary-vlan pvlan100;
    }
    hr-comm {
        vlan-id 400;
        primary-vlan pvlan100;
    }
    pvlan100 {
        vlan-id 100;
        interface {
            ge-0/0/0.0 {
                pvlan-trunk;
            }
            ge-0/0/1.0 {
                pvlan-trunk;
            }
            ge-0/0/2.0;
        }
        pvlan;
        isolation-vlan-id 50;
    }
}
```

Verification

IN THIS SECTION

To confirm that the configuration is working properly, perform these tasks:

Verifying That the Primary VLAN and Secondary VLANs Were Created on Switch 1

Purpose

Verify that the PVLAN configuration spanning multiple switches is working properly on Switch 1:

Action

Use the **show vlans extensive** command:

```
user@switch> show vlans extensive
```

```
VLAN: __pvlan_pvlan100_ge-0/0/15.0__, Created at: Thu Sep 16 23:15:27 2010
Internal index: 5, Admin State: Enabled, Origin: Static
Private VLAN Mode: Isolated, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  1 (Active = 1)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/5.0*, tagged, trunk, pvlan-trunk
      ge-0/0/15.0*, untagged, access

VLAN: __pvlan_pvlan100_ge-0/0/16.0__, Created at: Thu Sep 16 23:15:27 2010
Internal index: 6, Admin State: Enabled, Origin: Static
Private VLAN Mode: Isolated, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  1 (Active = 1)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/5.0*, tagged, trunk, pvlan-trunk
      ge-0/0/16.0*, untagged, access

VLAN: __pvlan_pvlan100_isiv__, Created at: Thu Sep 16 23:15:27 2010
802.1Q Tag: 50, Internal index: 7, Admin State: Enabled, Origin: Static
Private VLAN Mode: Inter-switch-isolated, Primary VLAN: pvlan100
```

```
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  0 (Active = 0)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/5.0*, tagged, trunk, pvlan-trunk

VLAN: default, Created at: Thu Sep 16 03:03:18 2010
Internal index: 2, Admin State: Enabled, Origin: Static
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 0 (Active = 0), Untagged  0 (Active = 0)

VLAN: finance-comm, Created at: Thu Sep 16 23:15:27 2010
802.1Q Tag: 300, Internal index: 8, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  2 (Active = 2)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/5.0*, tagged, trunk, pvlan-trunk
      ge-0/0/11.0*, untagged, access
      ge-0/0/12.0*, untagged, access

VLAN: hr-comm, Created at: Thu Sep 16 23:15:27 2010
802.1Q Tag: 400, Internal index: 9, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  2 (Active = 2)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/5.0*, tagged, trunk, pvlan-trunk
      ge-0/0/13.0*, untagged, access
      ge-0/0/14.0*, untagged, access

VLAN: pvlan100, Created at: Thu Sep 16 23:15:27 2010
802.1Q Tag: 100, Internal index: 4, Admin State: Enabled, Origin: Static
Private VLAN Mode: Primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  6 (Active = 6)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/5.0*, tagged, trunk, pvlan-trunk
      ge-0/0/11.0*, untagged, access
      ge-0/0/12.0*, untagged, access
      ge-0/0/13.0*, untagged, access
      ge-0/0/14.0*, untagged, access
      ge-0/0/15.0*, untagged, access
      ge-0/0/16.0*, untagged, access
Secondary VLANs: Isolated 2, Community  2, Inter-switch-isolated  1
```

```
  Isolated VLANs :
      __pvlan_pvlan100_ge-0/0/15.0__
      __pvlan_pvlan100_ge-0/0/16.0__
  Community VLANs :
      finance-comm
      hr-comm
  Inter-switch-isolated VLAN :
      __pvlan_pvlan100_isiv__
```

Meaning

The output shows that a PVLAN was created on Switch 1 and shows that it includes two isolated VLANs, two community VLANs, and an interswitch isolated VLAN. The presence of the pvlan-trunk and Inter-switch-isolated fields indicates that this PVLAN is spanning more than one switch.

Verifying That the Primary VLAN and Secondary VLANs Were Created on Switch 2

Purpose

Verify that the PVLAN configuration spanning multiple switches is working properly on Switch 2:

Action

Use the **show vlans extensive** command:

```
user@switch> show vlans extensive
```

```
VLAN: __pvlan_pvlan100_ge-0/0/17.0__, Created at: Thu Sep 16 23:19:22 2010
Internal index: 5, Admin State: Enabled, Origin: Static
Private VLAN Mode: Isolated, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  1 (Active = 1)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/5.0*, tagged, trunk, pvlan-trunk
      ge-0/0/17.0*, untagged, access

VLAN: __pvlan_pvlan100_isiv__, Created at: Thu Sep 16 23:19:22 2010
802.1Q Tag: 50, Internal index: 6, Admin State: Enabled, Origin: Static
Private VLAN Mode: Inter-switch-isolated, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  0 (Active = 0)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/5.0*, tagged, trunk, pvlan-trunk

VLAN: default, Created at: Thu Sep 16 03:03:18 2010
Internal index: 2, Admin State: Enabled, Origin: Static
```

```
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 0 (Active = 0), Untagged  0 (Active = 0)

VLAN: finance-comm, Created at: Thu Sep 16 23:19:22 2010
802.1Q Tag: 300, Internal index: 7, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  2 (Active = 2)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/5.0*, tagged, trunk, pvlan-trunk
      ge-0/0/11.0*, untagged, access
      ge-0/0/12.0*, untagged, access

VLAN: hr-comm, Created at: Thu Sep 16 23:19:22 2010
802.1Q Tag: 400, Internal index: 8, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  2 (Active = 2)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/5.0*, tagged, trunk, pvlan-trunk
      ge-0/0/13.0*, untagged, access
      ge-0/0/14.0*, untagged, access

VLAN: pvlan100, Created at: Thu Sep 16 23:19:22 2010
802.1Q Tag: 100, Internal index: 4, Admin State: Enabled, Origin: Static
Private VLAN Mode: Primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  5 (Active = 5)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/5.0*, tagged, trunk, pvlan-trunk
      ge-0/0/11.0*, untagged, access
      ge-0/0/12.0*, untagged, access
      ge-0/0/13.0*, untagged, access
      ge-0/0/14.0*, untagged, access
      ge-0/0/17.0*, untagged, access
Secondary VLANs: Isolated 1, Community  2, Inter-switch-isolated  1
  Isolated VLANs :
      __pvlan_pvlan100_ge-0/0/17.0__
  Community VLANs :
      finance-comm
      hr-comm
  Inter-switch-isolated VLAN :
      __pvlan_pvlan100_isiv__
```

Meaning

The output shows that a PVLAN was created on Switch 2 and shows that it includes one isolated VLAN, two community VLANs, and an interswitch isolated VLAN. The presence of the pvlan-trunk and Inter-switch-isolated fields indicates that this PVLAN is spanning more than one switch. When you compare this output to the output of Switch 1, you can see that both switches belong to the same PVLAN (**pvlan100**).

Verifying That the Primary VLAN and Secondary VLANs Were Created on Switch 3

Purpose

Verify that the PVLAN configuration spanning multiple switches is working properly on Switch 3:

Action

Use the **show vlans extensive** command:

```
user@switch> show vlans extensive
```

```
VLAN: __pvlan_pvlan100_isiv__, Created at: Thu Sep 16 23:22:40 2010
802.1Q Tag: 50, Internal index: 5, Admin State: Enabled, Origin: Static
Private VLAN Mode: Inter-switch-isolated, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  0 (Active = 0)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/1.0*, tagged, trunk, pvlan-trunk

VLAN: default, Created at: Thu Sep 16 03:03:18 2010
Internal index: 2, Admin State: Enabled, Origin: Static
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 0 (Active = 0), Untagged  0 (Active = 0)

VLAN: finance-comm, Created at: Thu Sep 16 23:22:40 2010
802.1Q Tag: 300, Internal index: 6, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  0 (Active = 0)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/1.0*, tagged, trunk, pvlan-trunk

VLAN: hr-comm, Created at: Thu Sep 16 23:22:40 2010
802.1Q Tag: 400, Internal index: 7, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  0 (Active = 0)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/1.0*, tagged, trunk, pvlan-trunk
```

```
VLAN: pvlan100, Created at: Thu Sep 16 23:22:40 2010
802.1Q Tag: 100, Internal index: 4, Admin State: Enabled, Origin: Static
Private VLAN Mode: Primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  0 (Active = 0)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/1.0*, tagged, trunk, pvlan-trunk
Secondary VLANs: Isolated 0, Community  2, Inter-switch-isolated  1
  Community VLANs :
      finance-comm
      hr-comm
  Inter-switch-isolated VLAN :
      __pvlan_pvlan100_isiv__
```

Meaning

The output shows that the PVLAN (**pvlan100**) is configured on Switch 3 and that it includes no isolated VLANs, two community VLANs, and an interswitch isolated VLAN. But Switch 3 is functioning as a distribution switch, so the output does not include access interfaces within the PVLAN. It shows only the **pvlan-trunk** interfaces that connect **pvlan100** from Switch 3 to the other switches (Switch 1 and Switch 2) in the same PVLAN.

Example: Configuring a Private VLAN Spanning Multiple Switches With an IRB Interface

IN THIS SECTION

For security reasons, it is often useful to restrict the flow of broadcast and unknown unicast traffic and even to limit the communication between known hosts. The private VLAN (PVLAN) feature allows an administrator to split a broadcast domain into multiple isolated broadcast subdomains, essentially putting a VLAN inside a VLAN. A PVLAN can span multiple switches. This example describes how to create a PVLAN spanning multiple switches. The example creates one primary PVLAN, containing multiple secondary VLANs.

Just like regular VLANs, PVLANs are isolated at Layer 2 and normally require that a Layer 3 device be used if you want to route traffic. Starting with Junos OS 14.1X53-D30, you can use an integrated routing and bridging (IRB) interface to route Layer 3 traffic between devices connected to a PVLAN. Using an IRB interface in this way can also allow the devices in the PVLAN to communicate at Layer 3 with devices in other community or isolated VLANs or with devices outside the PVLAN. This example also demonstrates how to include an IRB interface in a PVLAN configuration.

Requirements

This example uses the following hardware and software components:

- Three QFX Series or EX4600 switches
- Junos OS release with PVLAN for QFX Series or EX4600

Overview and Topology

In a large office with multiple buildings and VLANs, you might need to isolate some workgroups or other endpoints for security reasons or to partition the broadcast domain. This configuration example shows how to create a PVLAN spanning multiple switches, with one primary VLAN containing two community VLANs (one for HR and one for Finance), and an interswitch isolated VLAN (for the mail server, the backup server, and the CVS server). The PVLAN comprises three switches—two access switches and one distribution switch. The devices in the PVLAN are connected at Layer 3 to each other and to devices outside the PVLAN through an IRB interface configured on the distribution switch.

> **NOTE:** The isolated ports on Switch 1 and on Switch 2 do not have Layer 2 connectivity with one another even though they are included within the same domain. See "Understanding Private VLANs" on page 423.

Figure 25 on page 528 shows the topology for this example.

Figure 25: PVLAN Topology Spanning Multiple Switches with an IRB Interface

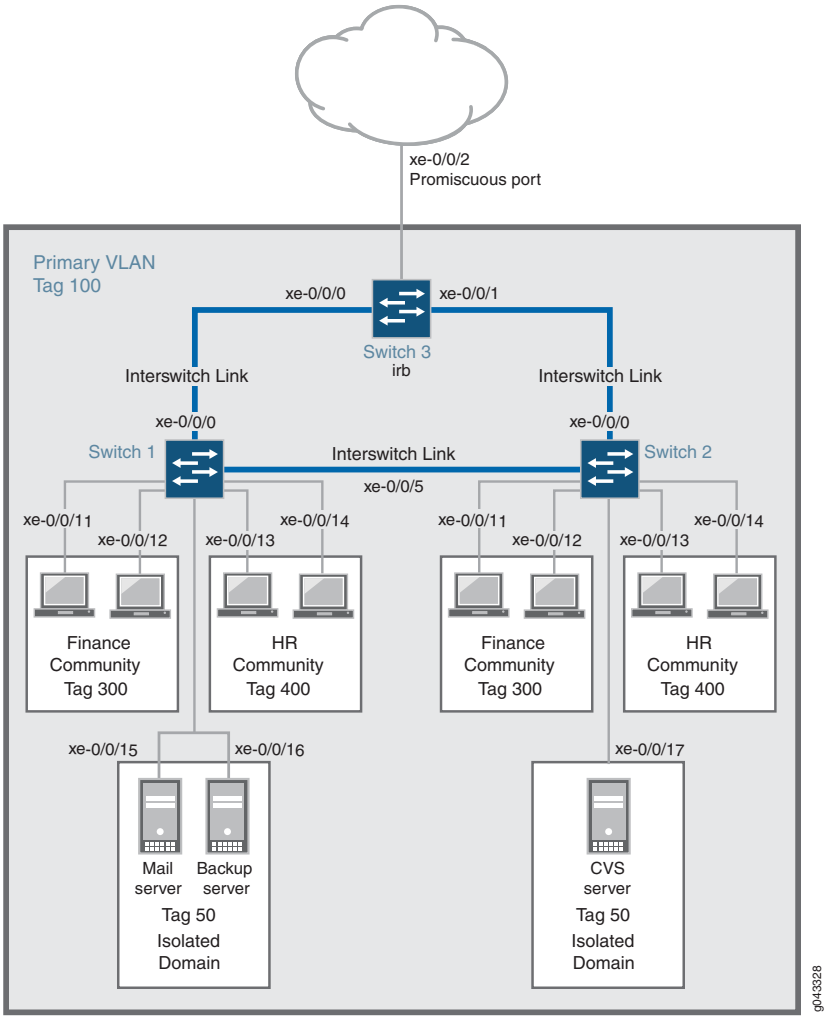


Table 88 on page 528, Table 89 on page 529, and Table 90 on page 530 list the settings for the example topology.

Table 88: Components of Switch 1 in the Topology for Configuring a PVLAN Spanning Multiple Devices

Property	Settings
VLAN names and tag IDs	**primary-vlan**, tag **100** **isolated-vlan-id**, tag **50** **finance-comm**, tag **300** **hr-comm**, tag **400**
Interswitch link interfaces	**xe-0/0/0.0**, connects Switch 1 to Switch 3 **xe-0/0/5.0**, connects Switch 1 to Switch 2

Table 88: Components of Switch 1 in the Topology for Configuring a PVLAN Spanning Multiple Devices *(continued)*

Property	Settings
Isolated Interfaces in primary VLAN	**xe-0/0/15.0**, mail server **xe-0/0/16.0**, backup server
Interfaces in VLAN **finance-com**	**xe-0/0/11.0** **xe-0/0/12.0**
Interfaces in VLAN **hr-comm**	**xe-0/0/13.0** **xe-0/0/14.0**

Table 89: Components of Switch 2 in the Topology for Configuring a PVLAN Spanning Multiple Devices

Property	Settings
VLAN names and tag IDs	**primary-vlan**, tag **100** **isolated-vlan-id**, tag **50** **finance-comm**, tag **300** **hr-comm**, tag **400**
Interswitch link interfaces	**xe-0/0/0.0**, connects Switch 2 to Switch 3 **xe-0/0/5.0**, connects Switch 2 to Switch 1
Isolated Interface in primary VLAN	**xe-0/0/17.0**, CVS server
Interfaces in VLAN **finance-com**	**xe-0/0/11.0** **xe-0/0/12.0**
Interfaces in VLAN **hr-comm**	**xe-0/0/13.0** **xe-0/0/14.0**

Table 90: Components of Switch 3 in the Topology for Configuring a PVLAN Spanning Multiple Devices

Property	Settings
VLAN names and tag IDs	**primary-vlan**, tag **100** **isolated-vlan-id**, tag **50finance-comm**, tag **300** **hr-comm**, tag **400**
Interswitch link interfaces	**xe-0/0/0.0**, connects Switch 3 to Switch 1. **xe-0/0/1.0**, connects Switch 3 to Switch 2.
Promiscuous port	**xe-0/0/2**, connects the PVLAN to another network. **NOTE:** You must configure the trunk port that connects the PVLAN to another switch or router outside the PVLAN as a member of the PVLAN, which implicitly configures it as a promiscuous port.
IRB interface	**xe-0/0/0** **xe-0/0/1** Configure unrestricted proxy ARP on the IRB interface to allow ARP resolution to occur so that devices that use IPv4 can communicate at Layer 3. For IPv6 traffic, you must explicitly map an IRB address to the destination address to allow ARP resolution.

Configuration Overview

When configuring a PVLAN on multiple switches, the following rules apply:

- The primary VLAN must be a tagged VLAN.
- The primary VLAN is the only VLAN that can be a member of an interswitch link interface.

When configuring an IRB interface in a PVLAN, these rules apply:

- You can create only one IRB interface in a PVLAN, regardless of how many switches participate in the PVLAN.
- The IRB interface must be a member of the primary VLAN in the PVLAN.
- Each host device that you want to connect at Layer 3 must use an IP address of the IRB as its default gateway address.

Configuring a PVLAN on Switch 1

CLI Quick Configuration

To quickly create and configure a PVLAN spanning multiple switches, copy the following commands and paste them into the terminal window of Switch 1:

```
[edit]
set interfaces xe-0/0/0 unit 0 family ethernet-switching interface-mode trunk
set interfaces xe-0/0/0 unit 0 family ethernet-switching inter-switch-link
set interfaces xe-0/0/0 unit 0 family ethernet-switching vlan members 100
set interfaces xe-0/0/5 unit 0 family ethernet-switching interface-mode trunk
set interfaces xe-0/0/5 unit 0 family ethernet-switching inter-switch-link
set interfaces xe-0/0/5 unit 0 family ethernet-switching vlan members 100
set vlans finance-comm vlan-id 300 private-vlan community
set vlans hr-comm vlan-id 400 private-vlan community
set vlans isolated-vlan vlan-id 50 private-vlan isolated
set vlans pvlan100 vlan-id 100 community-vlans [300 400] isolated-vlan 50
set interfaces xe-0/0/11 unit 0 family ethernet-switching vlan members 300
set interfaces xe-0/0/12 unit 0 family ethernet-switching vlan members 300
set interfaces xe-0/0/13 unit 0 family ethernet-switching vlan members 400
set interfaces xe-0/0/14 unit 0 family ethernet-switching vlan members 400
set interfaces xe-0/0/15 unit 0 family ethernet-switching vlan members 50
set interfaces xe-0/0/16 unit 0 family ethernet-switching vlan members 50
```

Step-by-Step Procedure

1. Configure interface xe-0/0/0 to be a trunk:

   [edit interfaces]
   `user@switch#` **set xe-0/0/0 unit 0 family ethernet-switching interface-mode trunk**

2. Configure interface xe-0/0/0 to be an interswitch link that carries all the VLANs:

   [edit interfaces]
   `user@switch#` **set xe-0/0/0 unit 0 family ethernet-switching inter-switch-link**

3. Configure pvlan100 (the primary VLAN) to be a member of interface xe-0/0/0:

   [edit interfaces]
   `user@switch#` **set xe-0/0/0 unit 0 family ethernet-switching vlan members 100**

4. Configure interface xe-0/0/5 to be a trunk:

   [edit interfaces]
   `user@switch#` **set xe-0/0/0 unit 0 family ethernet-switching interface-mode trunk**

5. Configure interface xe-0/0/5 to be an interswitch link that carries all the VLANs:

   [edit interfaces]
   `user@switch#` **set xe-0/0/5 unit 0 family ethernet-switching inter-switch-link**

6. Configure pvlan100 to be a member of interface xe-0/0/5:

   [edit interfaces]
   `user@switch#` **set xe-0/0/5 unit 0 family ethernet-switching vlan members 100**

7. Create the community VLAN for the finance organization:

   [edit vlans]
   set finance-comm vlan-id 300 private-vlan community

8. Create the community VLAN for the HR organization:

   [edit vlans]
   set hr-comm vlan-id 400 private-vlan community

9. Create the isolated VLAN for the mail and backup servers:

   [edit vlans]
   set isolated-vlan vlan-id 50 private-vlan isolated

10. Create the primary VLAN and make the community and isolated VLANs members of it:

    [edit vlans]
    set pvlan100 vlan-id 100 community-vlans [300 400] isolated-vlan 50

11. Configure VLAN 300 (the a community VLAN) to be a member of interface xe-0/0/11:

    [edit interfaces]
    `user@switch#` **set xe-0/0/11 unit 0 family ethernet-switching vlan members 300**

12. Configure VLAN 300 (a community VLAN) to be a member of interface xe-0/0/12:

    [edit interfaces]

    `user@switch#` **set xe-0/0/12 unit 0 family ethernet-switching vlan members 300**

13. Configure VLAN 400 (a community VLAN) to be a member of interface xe-0/0/13:

    [edit interfaces]

    `user@switch#` **set xe-0/0/13 unit 0 family ethernet-switching vlan members 400**

14. Configure VLAN 400 (a community VLAN) to be a member of interface xe-0/0/14:

    [edit interfaces]

    `user@switch#` **set xe-0/0/14 unit 0 family ethernet-switching vlan members 400**

15. Configure VLAN 50 (the isolated VLAN) to be a member of interface xe-0/0/15:

    [edit interfaces]

    `user@switch#` **set xe-0/0/15 unit 0 family ethernet-switching vlan members 50**

16. Configure VLAN 50 (the isolated VLAN) to be a member of interface xe-0/0/16:

    [edit interfaces]

    `user@switch#` **set xe-0/0/16 unit 0 family ethernet-switching vlan members 50**

Results

Check the results of the configuration:

```
[edit]
   user@switch# show
vlans {
   finance-comm {
      vlan-id 300;
      private-vlan community;
   }
   hr-comm {
      vlan-id 400;
      private-vlan community;
   }
   isolated-vlan{
      vlan-id 50;
      private-vlan isolated;
   }
```

```
    pvlan100 {
        vlan-id 100;
        isolated-vlan 50;
        community-vlans [300 400]
    }
}
```

Configuring a PVLAN on Switch 2

CLI Quick Configuration

To quickly create and configure a private VLAN spanning multiple switches, copy the following commands and paste them into the terminal window of Switch 2:

> **NOTE:** The configuration of Switch 2 is the same as the configuration of Switch 1 except for the isolated VLAN. For Switch 2, the isolated VLAN interface is **xe-0/0/17.0** .

```
[edit]
set interfaces xe-0/0/0 unit 0 family ethernet-switching interface-mode trunk
set interfaces xe-0/0/0 unit 0 family ethernet-switching inter-switch-link
set interfaces xe-0/0/0 unit 0 family ethernet-switching vlan members 100
set interfaces xe-0/0/5 unit 0 family ethernet-switching interface-mode trunk
set interfaces xe-0/0/5 unit 0 family ethernet-switching inter-switch-link
set interfaces xe-0/0/5 unit 0 family ethernet-switching vlan members 100
set vlans finance-comm vlan-id 300 private-vlan community
set vlans hr-comm vlan-id 400 private-vlan community
set vlans isolated-vlan vlan-id 50 private-vlan isolated
set vlans pvlan100 vlan-id 100 community-vlans [300 400] isolated-vlan 50
set interfaces xe-0/0/11 unit 0 family ethernet-switching vlan members 300
set interfaces xe-0/0/12 unit 0 family ethernet-switching vlan members 300
set interfaces xe-0/0/13 unit 0 family ethernet-switching vlan members 400
```

set interfaces xe-0/0/14 unit 0 family ethernet-switching vlan members 400

set interfaces xe-0/0/17 unit 0 family ethernet-switching vlan members 50

Step-by-Step Procedure

1. Configure interface xe-0/0/0 to be a trunk:

   [edit interfaces]
   `user@switch#` **set xe-0/0/0 unit 0 family ethernet-switching interface-mode trunk**

2. Configure interface xe-0/0/0 to be an interswitch link that carries all the VLANs:

   [edit interfaces]
   `user@switch#` **set xe-0/0/0 unit 0 family ethernet-switching inter-switch-link**

3. Configure pvlan100 (the primary VLAN) to be a member of interface xe-0/0/0:

   [edit interfaces]
   `user@switch#` **set xe-0/0/0 unit 0 family ethernet-switching vlan members 100**

4. Configure interface xe-0/0/5 to be a trunk:

   [edit interfaces]
   `user@switch#` **set xe-0/0/0 unit 0 family ethernet-switching interface-mode trunk**

5. Configure interface xe-0/0/5 to be an interswitch link that carries all the VLANs:

   [edit interfaces]
   `user@switch#` **set xe-0/0/5 unit 0 family ethernet-switching inter-switch-link**

6. Configure pvlan100 to be a member of interface xe-0/0/5:

   [edit interfaces]
   `user@switch#` **set xe-0/0/5 unit 0 family ethernet-switching vlan members 100**

7. Create the community VLAN for the finance organization:

   [edit vlans]
   set finance-comm vlan-id 300 private-vlan community

8. Create the community VLAN for the HR organization:

   [edit vlans]
   set hr-comm vlan-id 400 private-vlan community

9. Create the isolated VLAN for the mail and backup servers:

   [edit vlans]
   set isolated-vlan vlan-id 50 private-vlan isolated

10. Create the primary VLAN and make the community and isolated VLANs members of it:

    [edit vlans]
    set pvlan100 vlan-id 100 community-vlans [300 400] isolated-vlan 50

11. Configure VLAN 300 (the a community VLAN) to be a member of interface xe-0/0/11:

    [edit interfaces]
    `user@switch#` **set xe-0/0/11 unit 0 family ethernet-switching vlan members 300**

12. Configure VLAN 300 (a community VLAN) to be a member of interface xe-0/0/12:

    [edit interfaces]
    `user@switch#` **set xe-0/0/12 unit 0 family ethernet-switching vlan members 300**

13. Configure VLAN 400 (a community VLAN) to be a member of interface xe-0/0/13:

    [edit interfaces]
    `user@switch#` **set xe-0/0/13 unit 0 family ethernet-switching vlan members 400**

14. Configure VLAN 400 (a community VLAN) to be a member of interface xe-0/0/14:

    [edit interfaces]
    `user@switch#` **set xe-0/0/14 unit 0 family ethernet-switching vlan members 400**

15. Configure VLAN 50 (the isolated VLAN) to be a member of interface xe-0/0/17:

    [edit interfaces]
    `user@switch#` **set xe-0/0/17 unit 0 family ethernet-switching vlan members 50**

Results

Check the results of the configuration:

```
[edit]
   user@switch# show
vlans {
   finance-comm {
      vlan-id 300;
```

```
        private-vlan community;
    }
    hr-comm {
        vlan-id 400;
        private-vlan community;
    }
    isolated-vlan{
        vlan-id 50;
        private-vlan isolated;
    }
    pvlan100 {
        vlan-id 100;
        isolated-vlan 50;
        community-vlans [300 400]
    }
}
```

Configuring a PVLAN on Switch 3

CLI Quick Configuration

To quickly configure Switch 3 to function as the distribution switch of this PVLAN, copy the following commands and paste them into the terminal window of Switch 3:

> **NOTE:** Interface xe-0/0/2.0 is a trunk port connecting the PVLAN to another network.

```
[edit]
```

```
[edit]
```

set interfaces xe-0/0/0 unit 0 family ethernet-switching interface-mode trunk

set interfaces xe-0/0/0 unit 0 family ethernet-switching inter-switch-link

set interfaces xe-0/0/0 unit 0 family ethernet-switching vlan members 100

set interfaces xe-0/0/1 unit 0 family ethernet-switching interface-mode trunk

set interfaces xe-0/0/1 unit 0 family ethernet-switching inter-switch-link

set interfaces xe-0/0/1 unit 0 family ethernet-switching vlan members 100

set interfaces xe-0/0/2 unit 0 family ethernet-switching interface-mode trunk

set interfaces xe-0/0/2 unit 0 family ethernet-switching vlan members 100

set vlans pvlan100 vlan-id 100

set interfaces irb unit 100 family inet address 192.168.1.1/24

set vlans pvlan100 l3-interface irb.100

set interfaces irb unit 100 proxy-arp unrestricted

Step-by-Step Procedure

To configure Switch 3 to function as the distribution switch for this PVLAN, use the following procedure:

1. Configure interface xe-0/0/0 to be a trunk:

   [edit interfaces]
   `user@switch#` **set xe-0/0/0 unit 0 family ethernet-switching interface-mode trunk**

2. Configure interface xe-0/0/0 to be an interswitch link that carries all the VLANs:

   [edit interfaces]
   `user@switch#` **set xe-0/0/0 unit 0 family ethernet-switching inter-switch-link**

3. Configure pvlan100 (the primary VLAN) to be a member of interface xe-0/0/0:

   [edit interfaces]
   `user@switch#` **set xe-0/0/0 unit 0 family ethernet-switching vlan members 100**

4. Configure interface xe-0/0/5 to be a trunk:

   [edit interfaces]
   `user@switch#` **set xe-0/0/0 unit 0 family ethernet-switching interface-mode trunk**

5. Configure interface xe-0/0/5 to be an interswitch link that carries all the VLANs:

   [edit interfaces]
   `user@switch#` **set xe-0/0/5 unit 0 family ethernet-switching inter-switch-link**

6. Configure pvlan100 to be a member of interface xe-0/0/5:

   [edit interfaces]
   `user@switch#` **set xe-0/0/5 unit 0 family ethernet-switching vlan members 100**

7. Configure interface xe-0/0/2 (the promiscuous interface) to be a trunk:

[edit interfaces]

`user@switch#` **set xe-0/0/2 unit 0 family ethernet-switching interface-mode trunk**

8. Configure pvlan100 to be a member of interface xe-0/0/2:

   [edit interfaces]

   `user@switch#` **set xe-0/0/2 unit 0 family ethernet-switching vlan members 100**

9. Create the primary VLAN:

   [edit vlans]

   set vlans pvlan100 vlan-id 100

10. Create the IRB interface **irb** and assign it an address in the subnet used by the devices attached to Switches 1 and 2:

    [edit interfaces]

    set irb unit 100 family inet address 192.168.1.1/24

    > **NOTE:** Each host device that you want to connect at Layer 3 must be in the same subnet as the IRB interface and use the IP address of the IRB interface as its default gateway address.

11. Complete the IRB interface configuration by binding the interface to the primary VLAN **pvlan100**:

    [edit vlans]

    set pvlan100 l3-interface irb.100

12. Configure unrestricted proxy ARP for each unit of the IRB interface so that ARP resolution works for IPv4 traffic:

    [edit interfaces]

    set irb unit 100 proxy-arp unrestricted

    > **NOTE:** Because the devices in the community and isolated VLANs are isolated at Layer 2, this step is required to allow ARP resolution to occur between the VLANs so that devices using IPv4 can communicate at Layer 3. (For IPv6 traffic, you must explicitly map an IRB address to the destination address to allow ARP resolution.)

Results

Check the results of the configuration:

```
[edit]
user@switch# show
vlans {
   pvlan100{
      vlan-id 100;
   }
}
```

Verification

IN THIS SECTION

To confirm that the configuration is working properly, perform these tasks:

Verifying That the Primary VLAN and Secondary VLANs Were Created on Switch 1

Purpose

Verify that the PVLAN configuration spanning multiple switches is working properly on Switch 1:

Action

Use the **show vlans extensive** command:

```
user@switch> show vlans extensive
```

```
VLAN: __pvlan_pvlan100_xe-0/0/15.0__, Created at: Wed Sep 16 23:15:27 2015
Internal index: 5, Admin State: Enabled, Origin: Static
Private VLAN Mode: Isolated, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  1 (Active = 1)
     xe-0/0/0.0*, tagged, trunk     xe-0/0/5.0*, tagged, trunk     xe-0/0/15.0*,
 untagged, access
```

```
VLAN: __pvlan_pvlan100_xe-0/0/16.0__, Created at: Wed Sep 16 23:15:27 2015
Internal index: 6, Admin State: Enabled, Origin: Static
Private VLAN Mode: Isolated, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  1 (Active = 1)
     xe-0/0/0.0*, tagged, trunk     xe-0/0/5.0*, tagged, trunk     xe-0/0/16.0*,
 untagged, access

VLAN: __pvlan_pvlan100_isiv__, Created at: Wed Sep 16 23:15:27 2015
802.1Q Tag: 50, Internal index: 7, Admin State: Enabled, Origin: Static
Private VLAN Mode: Inter-switch-isolated, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  0 (Active = 0)
      xe-0/0/0.0*, tagged, trunk      xe-0/0/5.0*, tagged, trunk
VLAN: default, Created at: Wed Sep 16 03:03:18 2015
Internal index: 2, Admin State: Enabled, Origin: Static
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 0 (Active = 0), Untagged  0 (Active = 0)

VLAN: finance-comm, Created at: Wed Sep 16 23:15:27 2015
802.1Q Tag: 300, Internal index: 8, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  2 (Active = 2)
     xe-0/0/0.0*, tagged, trunk     xe-0/0/5.0*, tagged, trunk     xe-0/0/11.0*,
 untagged, access
      xe-0/0/12.0*, untagged, access

VLAN: hr-comm, Created at: Wed Sep 16 23:15:27 2015
802.1Q Tag: 400, Internal index: 9, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  2 (Active = 2)
     xe-0/0/0.0*, tagged, trunk     xe-0/0/5.0*, tagged, trunk     xe-0/0/13.0*,
 untagged, access
      xe-0/0/14.0*, untagged, access

VLAN: pvlan100, Created at: Wed Sep 16 23:15:27 2015
802.1Q Tag: 100, Internal index: 4, Admin State: Enabled, Origin: Static
Private VLAN Mode: Primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  6 (Active = 6)
      xe-0/0/0.0*, tagged, trunk
      xe-0/0/5.0*, tagged, trunk      xe-0/0/11.0*, untagged, access
```

```
      xe-0/0/12.0*, untagged, access
      xe-0/0/13.0*, untagged, access
      xe-0/0/14.0*, untagged, access
      xe-0/0/15.0*, untagged, access
      xe-0/0/16.0*, untagged, access
Secondary VLANs: Isolated 2, Community  2, Inter-switch-isolated  1
  Isolated VLANs :
      __pvlan_pvlan100_xe-0/0/15.0__
      __pvlan_pvlan100_xe-0/0/16.0__
  Community VLANs :
      finance-comm
      hr-comm
  Inter-switch-isolated VLAN :
      __pvlan_pvlan100_isiv__
```

Meaning

The output shows that a PVLAN was created on Switch 1 and shows that it includes two isolated VLANs, two community VLANs, and an interswitch isolated VLAN. The presence of the trunk and Inter-switch-isolated fields indicates that this PVLAN is spanning more than one switch.

Verifying That the Primary VLAN and Secondary VLANs Were Created on Switch 2

Purpose

Verify that the PVLAN configuration spanning multiple switches is working properly on Switch 2:

Action

Use the **show vlans extensive** command:

```
user@switch> show vlans extensive
```

```
VLAN: __pvlan_pvlan100_xe-0/0/17.0__, Created at: Wed Sep 16 23:19:22 2015
Internal index: 5, Admin State: Enabled, Origin: Static
Private VLAN Mode: Isolated, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  1 (Active = 1)
      xe-0/0/0.0*, tagged, trunk
      xe-0/0/5.0*, tagged, trunk
      xe-0/0/17.0*, untagged, access

VLAN: __pvlan_pvlan100_isiv__, Created at: Wed Sep 16 23:19:22 2015
802.1Q Tag: 50, Internal index: 6, Admin State: Enabled, Origin: Static
Private VLAN Mode: Inter-switch-isolated, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
```

```
Number of interfaces: Tagged 2 (Active = 2), Untagged  0 (Active = 0)
      xe-0/0/0.0*, tagged, trunk
      xe-0/0/5.0*, tagged, trunk

VLAN: default, Created at: Wed Sep 16 03:03:18 2015
Internal index: 2, Admin State: Enabled, Origin: Static
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 0 (Active = 0), Untagged  0 (Active = 0)

VLAN: finance-comm, Created at: Wed Sep 16 23:19:22 2015
802.1Q Tag: 300, Internal index: 7, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  2 (Active = 2)
      xe-0/0/0.0*, tagged, trunk
      xe-0/0/5.0*, tagged, trunk
      xe-0/0/11.0*, untagged, access
      xe-0/0/12.0*, untagged, access

VLAN: hr-comm, Created at: Wed Sep 16 23:19:22 2015
802.1Q Tag: 400, Internal index: 8, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  2 (Active = 2)
      xe-0/0/0.0*, tagged, trunk
      xe-0/0/5.0*, tagged, trunk
      xe-0/0/13.0*, untagged, access
      xe-0/0/14.0*, untagged, access

VLAN: pvlan100, Created at: Wed Sep 16 23:19:22 2015
802.1Q Tag: 100, Internal index: 4, Admin State: Enabled, Origin: Static
Private VLAN Mode: Primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  5 (Active = 5)
      xe-0/0/0.0*, tagged, trunk
      xe-0/0/5.0*, tagged, trunk
      xe-0/0/11.0*, untagged, access
      xe-0/0/12.0*, untagged, access
      xe-0/0/13.0*, untagged, access
      xe-0/0/14.0*, untagged, access
      xe-0/0/17.0*, untagged, access
Secondary VLANs: Isolated 1, Community  2, Inter-switch-isolated  1
  Isolated VLANs :
      __pvlan_pvlan100_xe-0/0/17.0__
```

```
  Community VLANs :
      finance-comm
      hr-comm
  Inter-switch-isolated VLAN :
      __pvlan_pvlan100_isiv__
```

Meaning

The output shows that a PVLAN was created on Switch 2 and shows that it includes one isolated VLAN, two community VLANs, and an interswitch isolated VLAN. The presence of the trunk and Inter-switch-isolated fields indicates that this PVLAN is spanning more than one switch. When you compare this output to the output of Switch 1, you can see that both switches belong to the same PVLAN (**pvlan100**).

Verifying That the Primary VLAN and Secondary VLANs Were Created on Switch 3

Purpose

Verify that the PVLAN configuration spanning multiple switches is working properly on Switch 3:

Action

Use the **show vlans extensive** command:

```
user@switch> show vlans extensive
```

```
VLAN: __pvlan_pvlan100_isiv__, Created at: Wed Sep 16 23:22:40 2015
802.1Q Tag: 50, Internal index: 5, Admin State: Enabled, Origin: Static
Private VLAN Mode: Inter-switch-isolated, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  0 (Active = 0)
      xe-0/0/0.0*, tagged, trunk
      xe-0/0/1.0*, tagged, trunk

VLAN: default, Created at: Wed Sep 16 03:03:18 2015
Internal index: 2, Admin State: Enabled, Origin: Static
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 0 (Active = 0), Untagged  0 (Active = 0)

VLAN: finance-comm, Created at: Wed Sep 16 23:22:40 2015
802.1Q Tag: 300, Internal index: 6, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  0 (Active = 0)
      xe-0/0/0.0*, tagged, trunk
      xe-0/0/1.0*, tagged, trunk
```

```
VLAN: hr-comm, Created at: Wed Sep 16 23:22:40 2015
802.1Q Tag: 400, Internal index: 7, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  0 (Active = 0)
      xe-0/0/0.0*, tagged, trunk
      xe-0/0/1.0*, tagged, trunk

VLAN: pvlan100, Created at: Wed Sep 16 23:22:40 2015
802.1Q Tag: 100, Internal index: 4, Admin State: Enabled, Origin: Static
Private VLAN Mode: Primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  0 (Active = 0)
      xe-0/0/0.0*, tagged, trunk
      xe-0/0/1.0*, tagged, trunk
Secondary VLANs: Isolated 0, Community  2, Inter-switch-isolated  1
  Community VLANs :
      finance-comm
      hr-comm
  Inter-switch-isolated VLAN :
      __pvlan_pvlan100_isiv__
```

Meaning

The output shows that the PVLAN (**pvlan100**) is configured on Switch 3 and that it includes no isolated VLANs, two community VLANs, and an interswitch isolated VLAN. But Switch 3 is functioning as a distribution switch, so the output does not include access interfaces within the PVLAN. It shows only the trunk interfaces that connect **pvlan100** from Switch 3 to the other switches (Switch 1 and Switch 2) in the same PVLAN.

Example: Configuring a Private VLAN Spanning Multiple EX Series Switches

IN THIS SECTION

For security reasons, it is often useful to restrict the flow of broadcast and unknown unicast traffic and to even limit the communication between known hosts. The private VLAN (PVLAN) feature on EX Series switches allows an administrator to split a broadcast domain into multiple isolated broadcast subdomains, essentially putting a VLAN inside a VLAN. A PVLAN can span multiple switches.

This example describes how to create a PVLAN spanning multiple EX Series switches. The example creates one primary PVLAN, containing multiple secondary VLANs:

NOTE: Configuring a voice over IP (VoIP) VLAN on PVLAN interfaces is not supported.

Requirements

This example uses the following hardware and software components:

- Three EX Series switches
- Junos OS Release 10.4 or later for EX Series switches

Before you begin configuring a PVLAN, make sure you have created and configured the necessary VLANs. See "Configuring VLANs for EX Series Switches" on page 183.

Overview and Topology

In a large office with multiple buildings and VLANs, you might need to isolate some workgroups or other endpoints for security reasons or to partition the broadcast domain. This configuration example shows how to create a PVLAN spanning multiple EX Series switches, with one primary VLAN containing two community VLANs (one for HR and one for Finance), and an Interswitch isolated VLAN (for the mail server, the backup server, and the CVS server). The PVLAN comprises three switches, two access switches and one distribution switch. The PVLAN is connected to a router through a promiscuous port, which is configured on the distribution switch.

NOTE: The isolated ports on Switch 1 and on Switch 2 do not have Layer 2 connectivity with each other even though they are included within the same domain. See "Understanding Private VLANs" on page 423.

Figure 26 on page 548 shows the topology for this example—two access switches connecting to a distribution switch, which has a connection (through a promiscuous port) to the router.

Figure 26: PVLAN Topology Spanning Multiple Switches

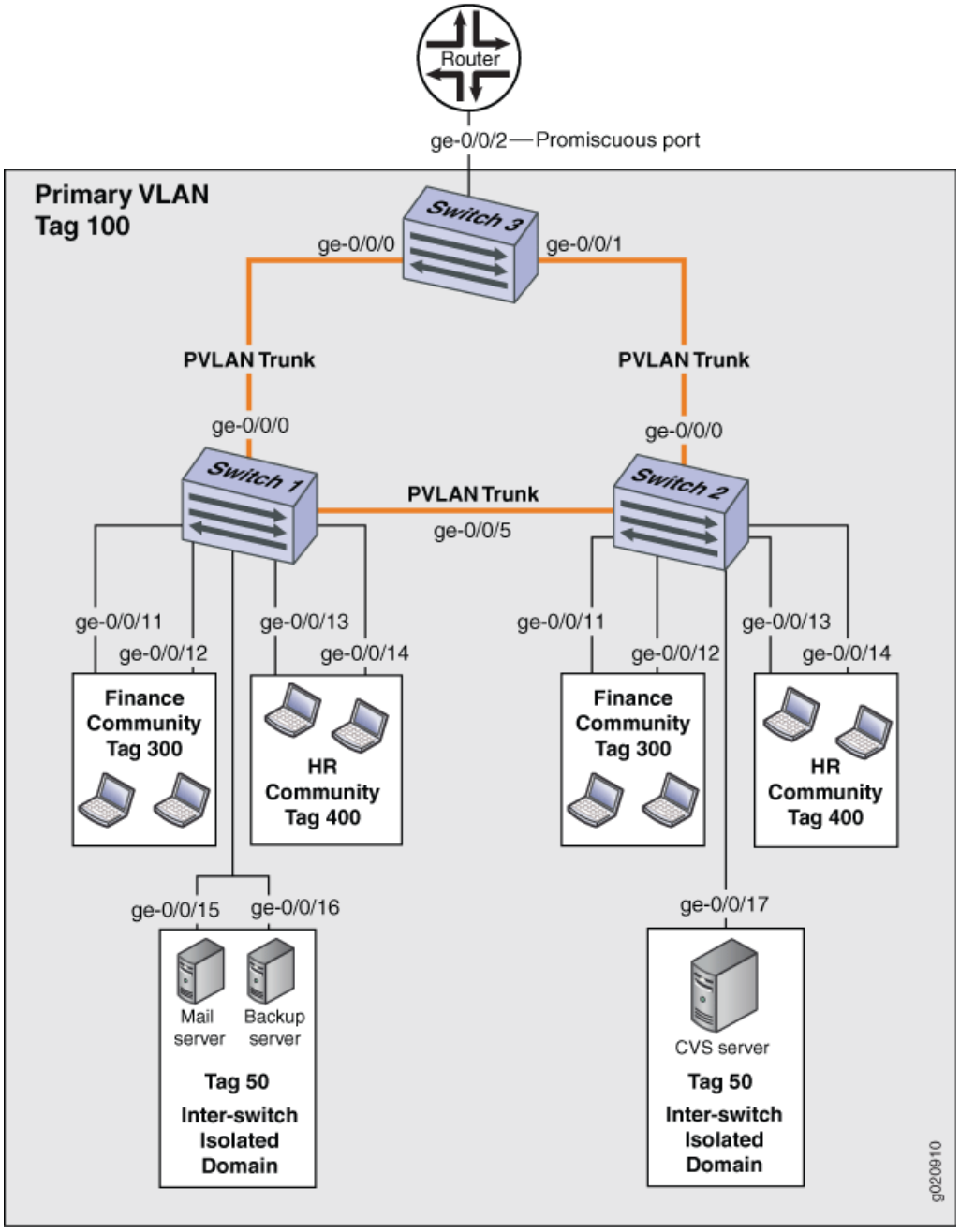


Table 91 on page 549, Table 92 on page 549, and Table 93 on page 550 list the settings for the example topology.

Table 91: Components of Switch 1 in the Topology for Configuring a PVLAN Spanning Multiple EX Series Switches

Property	Settings
VLAN names and tag IDs	**primary-vlan**, tag **100** **isolation-id**, tag **50** **finance-comm**, tag **300** **hr-comm**, tag **400**
PVLAN trunk interfaces	**ge-0/0/0.0**, Connects Switch 1 to Switch 3 **ge-0/0/5.0**, Connects Switch 1 to Switch 2
Interfaces in VLAN **isolation**	**ge-0/0/15.0**, Mail server **ge-0/0/16.0**, Backup server
Interfaces in VLAN **finance-com**	**ge-0/0/11.0** **ge-0/0/12.0**
Interfaces in VLAN **hr-comm**	**ge-0/0/13.0** **ge-0/0/14.0**

Table 92: Components of Switch 2 in the Topology for Configuring a PVLAN Spanning Multiple EX Series Switches

Property	Settings
VLAN names and tag IDs	**primary-vlan**, tag **100** **isolation-id**, tag **50** **finance-comm**, tag **300** **hr-comm**, tag **400**
PVLAN trunk interfaces	**ge-0/0/0.0**, Connects Switch 2 to Switch 3 **ge-0/0/5.0**, Connects Switch 2 to Switch 1
Interfaces in VLAN **isolation**	**ge-0/0/17.0**,CVS server
Interfaces in VLAN **finance-com**	**ge-0/0/11.0** **ge-0/0/12.0**

Table 92: Components of Switch 2 in the Topology for Configuring a PVLAN Spanning Multiple EX Series Switches *(continued)*

Property	Settings
Interfaces in VLAN **hr-comm**	**ge-0/0/13.0** **ge-0/0/14.0**

Table 93: Components of Switch 3 in the Topology for Configuring a PVLAN Spanning Multiple EX Series Switches

Property	Settings
VLAN names and tag IDs	**primary-vlan**, tag **100** **isolation-id**, tag **50** **finance-comm**, tag **300** **hr-comm**, tag **400**
PVLAN trunk interfaces	**ge-0/0/0.0**, Connects Switch 3 to Switch 1 **ge-0/0/1.0**, Connects Switch 3 to Switch 2
Promiscuous port	**ge-0/0/2**, Connects the PVLAN to the router **NOTE:** You must configure the trunk port that connects the PVLAN to another switch or router outside the PVLAN as a member of the PVLAN, which implicitly configures it as a promiscuous port.

Configuring a PVLAN on Switch 1

CLI Quick Configuration

When configuring a PVLAN on multiple switches, these rules apply:

- The primary VLAN must be a tagged VLAN. We recommend that you configure the primary VLAN first.
- Configuring a voice over IP (VoIP) VLAN on PVLAN interfaces is not supported.
- If you are going to configure a community VLAN ID, you must first configure the primary VLAN and the PVLAN trunk port.
- If you are going to configure an isolation VLAN ID, you must first configure the primary VLAN and the PVLAN trunk port.
- Secondary VLANs and the PVLAN trunk port must be committed on a single commit if MVRP is configured on the PVLAN trunk port.

To quickly create and configure a PVLAN spanning multiple switches, copy the following commands and paste them into the terminal window of Switch 1:

```
[edit]
```

set vlans finance-comm vlan-id 300

set vlans finance-comm interface ge-0/0/11.0

set vlans finance-comm interface ge-0/0/12.0

set vlans finance-comm primary-vlan pvlan100

set vlans hr-comm vlan-id 400

set vlans hr-comm interface ge-0/0/13.0

set vlans hr-comm interface ge-0/0/14.0

set vlans hr-comm primary-vlan pvlan100

set vlans pvlan100 vlan-id 100

set vlans pvlan100 interface ge-0/0/15.0

set vlans pvlan100 interface ge-0/0/16.0

set vlans pvlan100 interface ge-0/0/0.0 pvlan-trunk

set vlans pvlan100 interface ge-0/0/5.0 pvlan-trunk

set vlans pvlan100 no-local-switching

set vlans pvlan100 isolation-id 50

Step-by-Step Procedure

Complete the configuration steps below in the order shown—also, complete all steps before committing the configuration in a single commit. This is the easiest way to avoid error messages triggered by violating any of these three rules:

- If you are going to configure a community VLAN ID, you must first configure the primary VLAN and the PVLAN trunk port.
- If you are going to configure an isolation VLAN ID, you must first configure the primary VLAN and the PVLAN trunk port.
- Secondary vlans and a PVLAN trunk must be committed on a single commit.

To configure a PVLAN on Switch 1 that will span multiple switches:

1. Set the VLAN ID for the primary VLAN:

   [edit vlans]
   `user@switch#` **set pvlan100 vlan–id 100**

2. Set the PVLAN trunk interfaces that will connect this VLAN across neighboring switches:

   [edit vlans]
   `user@switch#` **set pvlan100 interface ge-0/0/0.0 pvlan-trunk**
   `user@switch#` **set pvlan100 interface ge-0/0/5.0 pvlan-trunk**

3. Set the primary VLAN to have no local switching:

   [edit vlans]
   `user@switch#` **set pvlan100 no-local-switching**

4. Set the VLAN ID for the **finance-comm** community VLAN that spans the switches:

   [edit vlans]
   `user@switch#` **finance-comm vlan-id 300**

   `user@switch#` **set pvlan100 vlan–id 100**

5. Configure access interfaces for the **finance-comm** VLAN:

   [edit vlans]
   `user@switch#` **set finance-comm interface interfacege-0/0/11.0**

   `user@switch#` **set finance-comm interface ge-0/0/12.0**

6. Set the primary VLAN of this secondary community VLAN, **finance-comm** :

   [edit vlans]

```
user@switch# set vlans finance-comm primary-vlan pvlan100
```

7. Set the VLAN ID for the HR community VLAN that spans the switches.

   [edit vlans]

   ```
   user@switch# hr-comm vlan-id 400
   ```

8. Configure access interfaces for the **hr-comm** VLAN:

   [edit vlans]

   ```
   user@switch# set hr-comm interface ge-0/0/13.0
   user@switch# set hr-comm interface ge-0/0/14.0
   ```

9. Set the primary VLAN of this secondary community VLAN, **hr-comm** :

   [edit vlans]

   ```
   user@switch# set vlans hr-comm primary-vlan pvlan100
   ```

10. Set the inter-switch isolated ID to create an inter-switch isolated domain that spans the switches:

    [edit vlans]

    ```
    user@switch# set pvlan100 isolation-id 50
    ```

> **NOTE:** To configure an isolated port, include it as one of the members of the primary VLAN but do not configure it as belonging to one of the community VLANs.

Results

Check the results of the configuration:

```
[edit]
   user@switch# show
vlans {
   finance-comm {
      vlan-id 300;
      interface {
         ge-0/0/11.0;
         ge-0/0/12.0;
      }
      primary-vlan pvlan100;
   }
```

```
    hr-comm {
        vlan-id 400;
        interface {
            ge-0/0/13.0;
            ge-0/0/14.0;
        }
        primary-vlan pvlan100;
    }
    pvlan100 {
        vlan-id 100;
        interface {
            ge-0/0/15.0;
            ge-0/0/16.0;
            ge-0/0/0.0 {
                pvlan-trunk;
            }
            ge-0/0/5.0 {
                pvlan-trunk;
            }
        }
        no-local-switching;
        isolation-id 50;
    }
}
```

Configuring a PVLAN on Switch 2

CLI Quick Configuration

To quickly create and configure a private VLAN spanning multiple switches, copy the following commands and paste them into the terminal window of Switch 2:

> **NOTE:** The configuration of Switch 2 is the same as the configuration of Switch 1 except for the interface in the inter-switch isolated domain. For Switch 2, the interface is **ge-0/0/17.0**.

```
[edit]
```

set vlans finance-comm vlan-id 300

set vlans finance-comm interface ge-0/0/11.0

set vlans finance-comm interface ge-0/0/12.0

set vlans finance-comm primary-vlan pvlan100

set vlans hr-comm vlan-id 400

set vlans hr-comm interface ge-0/0/13.0

set vlans hr-comm interface ge-0/0/14.0

set vlans hr-comm primary-vlan pvlan100

set vlans pvlan100 vlan-id 100

set vlans pvlan100 interface ge-0/0/17.0

set vlans pvlan100 interface ge-0/0/0.0 pvlan-trunk

set vlans pvlan100 interface ge-0/0/5.0 pvlan-trunk

set vlans pvlan100 no-local-switching

set vlans pvlan100 isolation-id 50

Step-by-Step Procedure

To configure a PVLAN on Switch 2 that will span multiple switches:

1. Set the VLAN ID for the **finance-comm** community VLAN that spans the switches:

   [edit vlans]

   ```
   user@switch# finance-comm vlan-id 300

   user@switch# set pvlan100 vlan–id 100
   ```

2. Configure access interfaces for the **finance-comm** VLAN:

   [edit vlans]

   ```
   user@switch# set finance-comm interface ge-0/0/11.0

   user@switch# set finance-comm interface ge-0/0/12.0
   ```

3. Set the primary VLAN of this secondary community VLAN, **finance-comm** :

   [edit vlans]

   ```
   user@switch# set vlans finance-comm primary-vlan pvlan100
   ```

4. Set the VLAN ID for the HR community VLAN that spans the switches.

   [edit vlans]

```
user@switch# hr-comm vlan-id 400
```

5. Configure access interfaces for the **hr-comm** VLAN:

```
[edit vlans]
user@switch# set hr-comm interface ge-0/0/13.0
user@switch# set hr-comm interface ge-0/0/14.0
```

6. Set the primary VLAN of this secondary community VLAN, **hr-comm** :

```
[edit vlans]
user@switch# set vlans hr-comm primary-vlan pvlan100
```

7. Set the VLAN ID for the primary VLAN:

```
[edit vlans]
user@switch# set pvlan100 vlan–id 100
```

8. Set the PVLAN trunk interfaces that will connect this VLAN across neighboring switches:

```
[edit vlans]
user@switch# set pvlan100 interface ge-0/0/0.0 pvlan-trunk
user@switch# set pvlan100 interface ge-0/0/5.0 pvlan-trunk
```

9. Set the primary VLAN to have no local switching:

```
[edit vlans]
user@switch# set pvlan100 no-local-switching
```

10. Set the inter-switch isolated ID to create an inter-switch isolated domain that spans the switches:

```
[edit vlans]
user@switch# set pvlan100 isolation-id 50
```

> **NOTE:** To configure an isolated port, include it as one of the members of the primary VLAN but do not configure it as belonging to one of the community VLANs.

Results

Check the results of the configuration:

```
[edit]
user@switch# show
vlans {
    finance-comm {
        vlan-id 300;
        interface {
            ge-0/0/11.0;
            ge-0/0/12.0;
        }
        primary-vlan pvlan100;
    }
    hr-comm {
        vlan-id 400;
        interface {
            ge-0/0/13.0;
            ge-0/0/14.0;
        }
        primary-vlan pvlan100;
    }
    pvlan100 {
        vlan-id 100;
        interface {
            ge-0/0/15.0;
            ge-0/0/16.0;
            ge-0/0/0.0 {
                pvlan-trunk;
            }
            ge-0/0/5.0 {
                pvlan-trunk;
            }
            ge-0/0/17.0;
        }
        no-local-switching;
        isolation-id 50;
    }
}
```

Configuring a PVLAN on Switch 3

CLI Quick Configuration

To quickly configure Switch 3 to function as the distribution switch of this PVLAN, copy the following commands and paste them into the terminal window of Switch 3:

NOTE: Interface ge-0/0/2.0 is a trunk port connecting the PVLAN to a router.

```
[edit]
set vlans finance-comm vlan-id 300
set vlans finance-comm primary-vlan pvlan100
set vlans hr-comm vlan-id 400
set vlans hr-comm primary-vlan pvlan100
set vlans pvlan100 vlan-id 100
set vlans pvlan100 interface ge-0/0/0.0 pvlan-trunk
set vlans pvlan100 interface ge-0/0/1.0 pvlan-trunk
set vlans pvlan100 no-local-switching
set vlans pvlan100 isolation-id 50
```

Step-by-Step Procedure

To configure Switch 3 to function as the distribution switch for this PVLAN, use the following procedure:

1. Set the VLAN ID for the **finance-comm** community VLAN that spans the switches:

   ```
   [edit vlans]
   user@switch#  finance-comm vlan-id 300

   [edit vlans]
   user@switch#  set pvlan100 vlan–id 100
   ```

2. Set the primary VLAN of this secondary community VLAN, **finance-comm**:

   ```
   [edit vlans]
   user@switch#  set vlans finance-comm primary-vlan pvlan100
   ```

3. Set the VLAN ID for the HR community VLAN that spans the switches:

   ```
   [edit vlans]
   user@switch#  hr-comm vlan-id 400
   ```

4. Set the primary VLAN of this secondary community VLAN, **hr-comm**:

[edit vlans]

```
user@switch# set vlans hr-comm primary-vlan pvlan100
```

5. Set the VLAN ID for the primary VLAN:

   [edit vlans]

   ```
   user@switch# set pvlan100 vlan–id 100
   ```

6. Set the PVLAN trunk interfaces that will connect this VLAN across neighboring switches:

   [edit vlans]

   ```
   user@switch# set pvlan100 interface ge-0/0/0.0 pvlan-trunk
   user@switch# set pvlan100 interface ge-0/0/5.0 pvlan-trunk
   ```

7. Set the primary VLAN to have no local switching:

   [edit vlans]

   ```
   user@switch# set pvlan100 no-local-switching
   ```

8. Set the inter-switch isolated ID to create an inter-switch isolated domain that spans the switches:

   [edit vlans]

   ```
   user@switch# set pvlan100 isolation-id 50
   ```

> **NOTE:** To configure an isolated port, include it as one of the members of the primary VLAN but do not configure it as belonging to one of the community VLANs.

Results

Check the results of the configuration:

```
[edit]
user@switch# show
vlans {
    finance-comm {
        vlan-id 300;
        primary-vlan pvlan100;
    }
    hr-comm {
        vlan-id 400;
        primary-vlan pvlan100;
```

```
    }
    pvlan100 {
        vlan-id 100;
        interface {
            ge-0/0/0.0 {
                pvlan-trunk;
            }
            ge-0/0/1.0 {
                pvlan-trunk;
            }
            ge-0/0/2.0;
        }
        no-local-switching;
        isolation-id 50;
    }
}
```

Verification

IN THIS SECTION

To confirm that the configuration is working properly, perform these tasks:

Verifying That the Primary VLAN and Secondary VLANs Were Created on Switch 1

Purpose

Verify that the PVLAN configuration spanning multiple switches is working properly on Switch 1:

Action

Use the **show vlans extensive** command:

```
user@switch> show vlans extensive
```

```
VLAN: __pvlan_pvlan100_ge-0/0/15.0__, Created at: Thu Sep 16 23:15:27 2010
Internal index: 5, Admin State: Enabled, Origin: Static
```

```
Private VLAN Mode: Isolated, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  1 (Active = 1)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/5.0*, tagged, trunk, pvlan-trunk
      ge-0/0/15.0*, untagged, access

VLAN: __pvlan_pvlan100_ge-0/0/16.0__, Created at: Thu Sep 16 23:15:27 2010
Internal index: 6, Admin State: Enabled, Origin: Static
Private VLAN Mode: Isolated, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  1 (Active = 1)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/5.0*, tagged, trunk, pvlan-trunk
      ge-0/0/16.0*, untagged, access

VLAN: __pvlan_pvlan100_isiv__, Created at: Thu Sep 16 23:15:27 2010
802.1Q Tag: 50, Internal index: 7, Admin State: Enabled, Origin: Static
Private VLAN Mode: Inter-switch-isolated, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  0 (Active = 0)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/5.0*, tagged, trunk, pvlan-trunk

VLAN: default, Created at: Thu Sep 16 03:03:18 2010
Internal index: 2, Admin State: Enabled, Origin: Static
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 0 (Active = 0), Untagged  0 (Active = 0)

VLAN: finance-comm, Created at: Thu Sep 16 23:15:27 2010
802.1Q Tag: 300, Internal index: 8, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  2 (Active = 2)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/5.0*, tagged, trunk, pvlan-trunk
      ge-0/0/11.0*, untagged, access
      ge-0/0/12.0*, untagged, access

VLAN: hr-comm, Created at: Thu Sep 16 23:15:27 2010
802.1Q Tag: 400, Internal index: 9, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  2 (Active = 2)
```

```
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/5.0*, tagged, trunk, pvlan-trunk
      ge-0/0/13.0*, untagged, access
      ge-0/0/14.0*, untagged, access

VLAN: pvlan100, Created at: Thu Sep 16 23:15:27 2010
802.1Q Tag: 100, Internal index: 4, Admin State: Enabled, Origin: Static
Private VLAN Mode: Primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  6 (Active = 6)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/5.0*, tagged, trunk, pvlan-trunk
      ge-0/0/11.0*, untagged, access
      ge-0/0/12.0*, untagged, access
      ge-0/0/13.0*, untagged, access
      ge-0/0/14.0*, untagged, access
      ge-0/0/15.0*, untagged, access
      ge-0/0/16.0*, untagged, access
Secondary VLANs: Isolated 2, Community  2, Inter-switch-isolated  1
  Isolated VLANs :
      __pvlan_pvlan100_ge-0/0/15.0__
      __pvlan_pvlan100_ge-0/0/16.0__
  Community VLANs :
      finance-comm
      hr-comm
  Inter-switch-isolated VLAN :
      __pvlan_pvlan100_isiv__
```

Meaning

The output shows that a PVLAN was created on Switch 1 and shows that it includes two isolated VLANs, two community VLANs, and an interswitch isolated VLAN. The presence of the **pvlan-trunk** and **Inter-switch-isolated** fields indicates that this PVLAN is spanning more than one switch.

Verifying That the Primary VLAN and Secondary VLANs Were Created on Switch 2

Purpose

Verify that the PVLAN configuration spanning multiple switches is working properly on Switch 2:

Action

Use the **show vlans extensive** command:

```
user@switch> show vlans extensive
```

```
VLAN: __pvlan_pvlan100_ge-0/0/17.0__, Created at: Thu Sep 16 23:19:22 2010
Internal index: 5, Admin State: Enabled, Origin: Static
Private VLAN Mode: Isolated, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  1 (Active = 1)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/5.0*, tagged, trunk, pvlan-trunk
      ge-0/0/17.0*, untagged, access

VLAN: __pvlan_pvlan100_isiv__, Created at: Thu Sep 16 23:19:22 2010
802.1Q Tag: 50, Internal index: 6, Admin State: Enabled, Origin: Static
Private VLAN Mode: Inter-switch-isolated, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  0 (Active = 0)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/5.0*, tagged, trunk, pvlan-trunk

VLAN: default, Created at: Thu Sep 16 03:03:18 2010
Internal index: 2, Admin State: Enabled, Origin: Static
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 0 (Active = 0), Untagged  0 (Active = 0)

VLAN: finance-comm, Created at: Thu Sep 16 23:19:22 2010
802.1Q Tag: 300, Internal index: 7, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  2 (Active = 2)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/5.0*, tagged, trunk, pvlan-trunk
      ge-0/0/11.0*, untagged, access
      ge-0/0/12.0*, untagged, access

VLAN: hr-comm, Created at: Thu Sep 16 23:19:22 2010
802.1Q Tag: 400, Internal index: 8, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  2 (Active = 2)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/5.0*, tagged, trunk, pvlan-trunk
      ge-0/0/13.0*, untagged, access
      ge-0/0/14.0*, untagged, access

VLAN: pvlan100, Created at: Thu Sep 16 23:19:22 2010
802.1Q Tag: 100, Internal index: 4, Admin State: Enabled, Origin: Static
```

```
Private VLAN Mode: Primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  5 (Active = 5)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/5.0*, tagged, trunk, pvlan-trunk
      ge-0/0/11.0*, untagged, access
      ge-0/0/12.0*, untagged, access
      ge-0/0/13.0*, untagged, access
      ge-0/0/14.0*, untagged, access
      ge-0/0/17.0*, untagged, access
Secondary VLANs: Isolated 1, Community  2, Inter-switch-isolated  1
  Isolated VLANs :
      __pvlan_pvlan100_ge-0/0/17.0__
  Community VLANs :
      finance-comm
      hr-comm
  Inter-switch-isolated VLAN :
      __pvlan_pvlan100_isiv__
```

Meaning

The output shows that a PVLAN was created on Switch 1 and shows that it includes two isolated VLANs, two community VLANs, and an interswitch isolated VLAN. The presence of the **pvlan-trunk** and **Inter-switch-isolated** fields indicates that this is PVLAN spanning more than one switch. When you compare this output to the output of Switch 1, you can see that both switches belong to the same PVLAN (**pvlan100**).

Verifying That the Primary VLAN and Secondary VLANs Were Created on Switch 3

Purpose

Verify that the PVLAN configuration spanning multiple switches is working properly on Switch 3:

Action

Use the **show vlans extensive** command:

```
user@switch> show vlans extensive
```

```
VLAN: __pvlan_pvlan100_isiv__, Created at: Thu Sep 16 23:22:40 2010
802.1Q Tag: 50, Internal index: 5, Admin State: Enabled, Origin: Static
Private VLAN Mode: Inter-switch-isolated, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  0 (Active = 0)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/1.0*, tagged, trunk, pvlan-trunk
```

```
VLAN: default, Created at: Thu Sep 16 03:03:18 2010
Internal index: 2, Admin State: Enabled, Origin: Static
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 0 (Active = 0), Untagged  0 (Active = 0)

VLAN: finance-comm, Created at: Thu Sep 16 23:22:40 2010
802.1Q Tag: 300, Internal index: 6, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  0 (Active = 0)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/1.0*, tagged, trunk, pvlan-trunk

VLAN: hr-comm, Created at: Thu Sep 16 23:22:40 2010
802.1Q Tag: 400, Internal index: 7, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  0 (Active = 0)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/1.0*, tagged, trunk, pvlan-trunk

VLAN: pvlan100, Created at: Thu Sep 16 23:22:40 2010
802.1Q Tag: 100, Internal index: 4, Admin State: Enabled, Origin: Static
Private VLAN Mode: Primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  0 (Active = 0)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/1.0*, tagged, trunk, pvlan-trunk
Secondary VLANs: Isolated 0, Community  2, Inter-switch-isolated  1
  Community VLANs :
      finance-comm
      hr-comm
  Inter-switch-isolated VLAN :
      __pvlan_pvlan100_isiv__
```

Meaning

The output shows that the PVLAN (**pvlan100**) is configured on Switch 3 and that it includes two isolated VLANs, two community VLANs, and an interswitch isolated VLAN. But Switch 3 is functioning as a distribution switch, so the output does not include access interfaces within the PVLAN. It shows only the **pvlan-trunk** interfaces that connect **pvlan100** from Switch 3 to the other switches (Switch 1 and Switch 2) in the same PVLAN.

Example: Configuring PVLANs with Secondary VLAN Trunk Ports and Promiscuous Access Ports on a QFX Series Switch

IN THIS SECTION

This example shows how to configure secondary VLAN trunk ports and promiscuous access ports as part of a private VLAN configuration. Secondary VLAN trunk ports carry secondary VLAN traffic.

For a given private VLAN, a secondary VLAN trunk port can carry traffic for only one secondary VLAN. However, a secondary VLAN trunk port can carry traffic for multiple secondary VLANs as long as each secondary VLAN is a member of a different private (primary) VLAN. For example, a secondary VLAN trunk port can carry traffic for a community VLAN that is part of primary VLAN pvlan100 and also carry traffic for an isolated VLAN that is part of primary VLAN pvlan400.

To configure a trunk port to carry secondary VLAN traffic, use the **isolated** and **interface** statements, as shown in steps 12 and 13 of the example configuration for Switch 1.

> **NOTE:** When traffic egresses from a secondary VLAN trunk port, it normally carries the tag of the primary VLAN that the secondary port is a member of. If you want traffic that egresses from a secondary VLAN trunk port to retain its secondary VLAN tag, use the **extend-secondary-vlan-id** statement.

A promiscuous access port carries untagged traffic and can be a member of only one primary VLAN. Traffic that ingresses on a promiscuous access port is forwarded to the ports of the secondary VLANs that are members of the primary VLAN that the promiscuous access port is a member of. This traffic carries the appropriate secondary VLAN tags when it egresses from the secondary VLAN ports if the secondary VLAN port is a trunk port.

To configure an access port to be promiscuous, use the **promiscuous** statement, as shown in step 12 of the example configuration for Switch 2.

If traffic ingresses on a secondary VLAN port and egresses on a promiscuous access port, the traffic is untagged on egress. If tagged traffic ingresses on a promiscuous access port, the traffic is discarded.

Requirements

This example uses the following hardware and software components:

- Two QFX devices
- Junos OS Release 12.2 or later for the QFX Series

Overview and Topology

Figure 27 on page 567 shows the topology used in this example. Switch 1 includes several primary and secondary private VLANs and also includes two secondary VLAN trunk ports configured to carry secondary VLANs that are members of primary VLANs pvlan100 and pvlan400.

Switch 2 includes the same private VLANs. The figure shows xe-0/0/0 on Switch 2 as configured with promiscuous access ports or promiscuous trunk ports. The example configuration included here configures this port as a promiscuous access port.

The figure also shows how traffic would flow after ingressing on the secondary VLAN trunk ports on Switch 1.

Figure 27: PVLAN Topology with Secondary VLAN Trunk Ports and Promiscuous Access Port

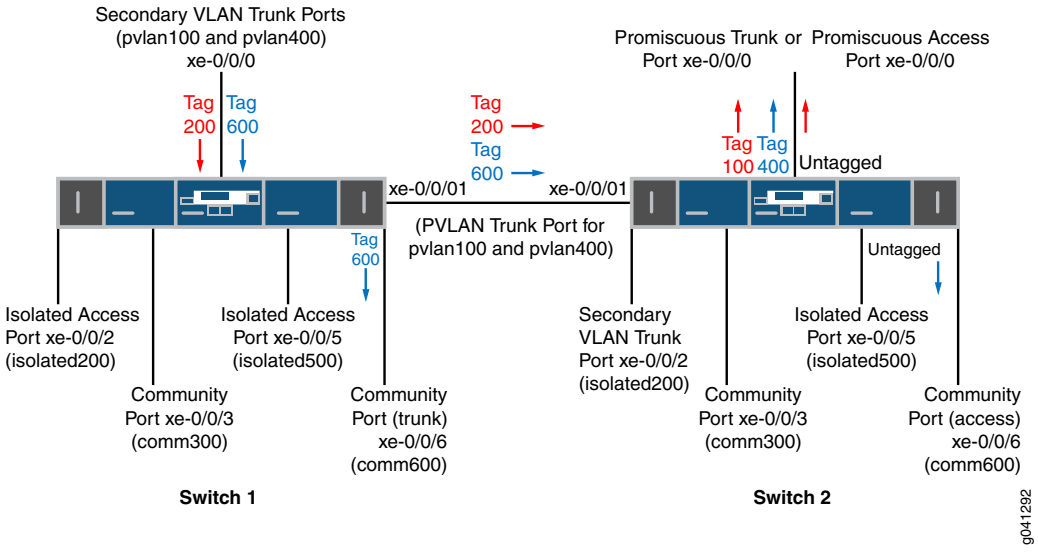


Table 94 on page 568 and Table 95 on page 568 list the settings for the example topology on both switches.

Table 94: Components of the Topology for Configuring a Secondary VLAN Trunk on Switch 1

Component	Description
pvlan100, ID 100	Primary VLAN
pvlan400, ID 400	Primary VLAN
comm300, ID 300	Community VLAN, member of pvlan100
comm600, ID 600	Community VLAN, member of pvlan400
isolation-vlan-id 200	VLAN ID for isolated VLAN, member of pvlan100
isolation–vlan-id 500	VLAN ID for isolated VLAN, member of pvlan400
xe-0/0/0.0	Secondary VLAN trunk port for primary VLANs pvlan100 and pvlan400
xe-0/0/1.0	PVLAN trunk port for primary VLANs pvlan100 and pvlan400
xe-0/0/2.0	Isolated access port for pvlan100
xe-0/0/3.0	Community access port for comm300
xe-0/0/5.0	Isolated access port for pvlan400
xe-0/0/6.0	Community trunk port for comm600

Table 95: Components of the Topology for Configuring a Secondary VLAN Trunk on Switch 2

Component	Description
pvlan100, ID 100	Primary VLAN
pvlan400, ID 400	Primary VLAN
comm300, ID 300	Community VLAN, member of pvlan100
comm600, ID 600	Community VLAN, member of pvlan400
isolation-vlan-id 200	VLAN ID for isolated VLAN, member of pvlan100
isolation–vlan-id 500	VLAN ID for isolated VLAN, member of pvlan400
xe-0/0/0.0	Promiscuous access port for primary VLANs pvlan100

Table 95: Components of the Topology for Configuring a Secondary VLAN Trunk on Switch 2 *(continued)*

Component	Description
xe-0/0/1.0	PVLAN trunk port for primary VLANs pvlan100 and pvlan400
xe-0/0/2.0	Secondary trunk port for isolated VLAN, member of pvlan100
xe-0/0/3.0	Community access port for comm300
xe-0/0/5.0	Isolated access port for pvlan400
xe-0/0/6.0	Community access port for comm600

Configuring the PVLANs on Switch 1

CLI Quick Configuration

To quickly create and configure the PVLANs on Switch 1, copy the following commands and paste them into a switch terminal window:

```
[edit]
```

set interfaces xe-0/0/0 unit 0 family ethernet-switching port-mode trunk

set interfaces xe-0/0/1 unit 0 family ethernet-switching port-mode trunk

set interfaces xe-0/0/1 unit 0 family ethernet-switching vlan members pvlan100

set interfacesxe-0/0/1 unit 0 family ethernet-switching vlan members pvlan400

set interfaces xe-0/0/2 unit 0 family ethernet-switching port-mode access

set interfaces xe-0/0/3 unit 0 family ethernet-switching port-mode access

set interfaces xe-0/0/5 unit 0 family ethernet-switching port-mode access

set interfaces xe-0/0/6 unit 0 family ethernet-switching port-mode trunk

set vlans pvlan100 vlan-id 100

set vlans pvlan400 vlan-id 400

set vlans pvlan100 pvlan

set vlans pvlan400 pvlan

set vlans pvlan100 interface xe-0/0/1.0 pvlan-trunk

set vlans pvlan400 interface xe-0/0/1.0 pvlan-trunk

set vlans comm300 vlan-id 300

set vlans comm300 primary-vlan pvlan100

set vlans comm300 interface xe-0/0/3.0

set vlans comm600 vlan-id 600

set vlans comm600 primary-vlan pvlan400

set vlans comm600 interface xe-0/0/6.0

set vlans pvlan100 pvlan isolation-vlan-id 200

set vlans pvlan400 pvlan isolation-vlan-id 500

set vlans pvlan100 interface xe-0/0/0.0 isolated

set vlans pvlan400 interface xe-0/0/0.0 isolated

set vlans comm600 interface xe-0/0/0.0

set vlans pvlan100 interface xe-0/0/2.0 isolated

set vlans pvlan400 interface xe-0/0/5.0 isolated

Step-by-Step Procedure

To configure the private VLANs and secondary VLAN trunk ports:

1. Configure the interfaces and port modes:

   [edit interfaces]

   ```
   user@switch# set xe-0/0/0 unit 0 family ethernet-switching port-mode trunk
   user@switch# set xe-0/0/1 unit 0 family ethernet-switching port-mode trunk
   user@switch# set xe-0/0/1 unit 0 family ethernet-switching vlan members pvlan100
   user@switch# set xe-0/0/1 unit 0 family ethernet-switching vlan members pvlan400
   user@switch# set xe-0/0/2 unit 0 family ethernet-switching port-mode access
   user@switch# set xe-0/0/3 unit 0 family ethernet-switching port-mode access
   user@switch# set xe-0/0/5 unit 0 family ethernet-switching port-mode access
   user@switch# set xe-0/0/6 unit 0 family ethernet-switching port-mode access
   ```

2. Create the primary VLANs:

   [edit vlans]

```
user@switch#  set pvlan100 vlan-id 100
user@switch#  set pvlan400 vlan-id 400
```

> **NOTE:** Primary VLANs must always be tagged VLANs, even if they exist on only one device.

3. Configure the primary VLANs to be private:

   [edit vlans]

   ```
   user@switch#  set pvlan100 pvlan
   user@switch#  set pvlan400 pvlan
   ```

4. Configure the PVLAN trunk port to carry the private VLAN traffic between the switches:

   [edit vlans]

   ```
   user@switch#  set pvlan100 interface xe-0/0/1.0 pvlan-trunk
   user@switch#  set pvlan400 interface xe-0/0/1.0 pvlan-trunk
   ```

5. Create secondary VLAN comm300 with VLAN ID 300:

   [edit vlans]

   ```
   user@switch#  set comm300 vlan-id 300
   ```

6. Configure the primary VLAN for comm300:

   [edit vlans]

   ```
   user@switch#  set comm300 primary-vlan pvlan100
   ```

7. Configure the interface for comm300:

   [edit vlans]

   ```
   user@switch#  set comm300 interface xe-0/0/3.0
   ```

8. Create secondary VLAN comm600 with VLAN ID 600:

   [edit vlans]

   ```
   user@switch#  set comm600 vlan-id 600
   ```

9. Configure the primary VLAN for comm600:

   [edit vlans]

   ```
   user@switch#  set comm600 primary-vlan pvlan400
   ```

10. Configure the interface for comm600:

    [edit vlans]

    `user@switch#` **set comm600 interface xe-0/0/6.0**

11. Configure the interswitch isolated VLANs:

    [edit vlans]

    `user@switch#` **set pvlan100 pvlan isolation-vlan-id 200**

    `user@switch#` **set pvlan400 pvlan isolation-vlan-id 500**

    > **NOTE:** When you configure a secondary VLAN trunk port to carry an isolated VLAN, you must also configure an **isolation-vlan-id**. This is true even if the isolated VLAN exists only on one switch.

12. Enable trunk port xe-0/0/0 to carry secondary VLANs for the primary VLANs:

    [edit vlans]

    `user@switch#` **set pvlan100 interface xe-0/0/0.0 isolated**

    `user@switch#` **set pvlan400 interface xe-0/0/0.0 isolated**

13. Configure trunk port xe-0/0/0 to carry comm600 (member of pvlan400):

    [edit vlans]

    `user@switch#` **set comm600 interface xe-0/0/0.0**

    > **NOTE:** You do not need to explicitly configure xe-0/0/0 to carry the isolated VLAN traffic (tags 200 and 500) because all the isolated ports in pvlan100 and pvlan400–including xe-0/0/0.0–are automatically included in the isolated VLANs created when you configured **isolation-vlan-id 200** and **isolation-vlan-id 500**.

14. Configure xe-0/0/2 and xe-0/0/6 to be isolated:

    [edit vlans]

    `user@switch#` **set pvlan100 interface xe-0/0/2.0 isolated**

    `user@switch#` **set pvlan400 interface xe-0/0/5.0 isolated**

Results

Check the results of the configuration on Switch 1:

```
[edit]
user@switch#  show
interfaces {
    xe-0/0/0 {
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
                vlan {
                    members pvlan100;
                    members pvlan400;
                }
            }
        }
    }
    xe-0/0/1 {
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
                vlan {
                    members pvlan100;
                    members pvlan400;
                }
            }
        }
    }
    xe-0/0/2 {
        unit 0 {
            family ethernet-switching {
                port-mode access;
            }
        }
    }
    xe-0/0/3 {
        unit 0 {
            family ethernet-switching {
                port-mode access;
            }
        }
    }
    xe-0/0/5 {
        unit 0 {
```

```
            family ethernet-switching {
                port-mode access;
            }
        }
    }
    xe-0/0/6 {
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
            }
        }
    }
}
vlans {
    comm300 {
        vlan-id 300;
            interface {
            xe-0/0/3.0;
        }
        primary-vlan pvlan100;
    }
    comm600 {
        vlan-id 600;
            interface {
            xe-0/0/6.0;
        }
        primary-vlan pvlan400;
    }
    pvlan100 {
        vlan-id 100;
        interface {
            xe-0/0/0.0;
            xe-0/0/2.0;
            xe-0/0/3.0;
            xe-0/0/1.0 {
                pvlan-trunk;
            }
        }
        no-local-switching;
        isolation-id 200;
    }
    pvlan400 {
        vlan-id 400;
        interface {
```

```
            xe-0/0/0.0;
            xe-0/0/5.0;
            xe-0/0/6.0;
            xe-0/0/1.0 {
                pvlan-trunk;
            }
        }
        no-local-switching;
        isolation-id 500;
    }
}
```

Configuring the PVLANs on Switch 2

The configuration for Switch 2 is almost identical to the configuration for Switch 1. The most significant difference is that xe-0/0/0 on Switch 2 is configured as a promiscuous trunk port or a promiscuous access port, as Figure 27 on page 567 shows. In the following configuration, xe-0/0/0 is configured as a promiscuous access port for primary VLAN pvlan100.

If traffic ingresses on VLAN-enabled port and egresses on a promiscuous access port, the VLAN tags are dropped on egress and the traffic is untagged at that point. For example, traffic for comm600 ingresses on the secondary VLAN trunk port configured on xe-0/0/0.0 on Switch 1 and carries tag 600 as it is forwarded through the secondary VLAN. When it egresses from xe-0/0/0.0 on Switch 2, it will be untagged if you configure xe-0/0/0.0 as a promiscuous access port as shown in this example. If you instead configure xe-0/0/0.0 as a promiscuous trunk port (port-mode trunk), the traffic for comm600 carries its primary VLAN tag (400) when it egresses.

CLI Quick Configuration

To quickly create and configure the PVLANs on Switch 2, copy the following commands and paste them into a switch terminal window:

```
[edit]
```

set interfaces xe-0/0/0 unit 0 family ethernet-switching port-mode access

set interfaces xe-0/0/1 unit 0 family ethernet-switching port-mode trunk

set interfaces xe-0/0/1 unit 0 family ethernet-switching vlan members pvlan100

set interfaces xe-0/0/1 unit 0 family ethernet-switching vlan members pvlan400

set interfaces xe-0/0/2 unit 0 family ethernet-switching port-mode trunk

set interfaces xe-0/0/3 unit 0 family ethernet-switching port-mode access

```
set interfaces xe-0/0/5 unit 0 family ethernet-switching port-mode access
set interfaces xe-0/0/6 unit 0 family ethernet-switching port-mode access
set vlans pvlan100 vlan-id 100
set vlans pvlan400 vlan-id 400
set vlans pvlan100 pvlan
set vlans pvlan400 pvlan
set vlans pvlan100 interface xe-0/0/1.0 pvlan-trunk
set vlans pvlan400 interface xe-0/0/1.0 pvlan-trunk
set vlans comm300 vlan-id 300
set vlans comm300 primary-vlan pvlan100
set vlans comm300 interface xe-0/0/3.0
set vlans comm600 vlan-id 600
set vlans comm600 primary-vlan pvlan400
set vlans comm600 interface xe-0/0/6.0
set vlans pvlan100 pvlan isolation-vlan-id 200
set vlans pvlan400 pvlan isolation-vlan-id 500
set vlans pvlan100 interface xe-0/0/0.0 promiscuous
set vlans pvlan100 interface xe-0/0/2.0 isolated
set vlans pvlan400 interface xe-0/0/5.0 isolated
```

Step-by-Step Procedure

To configure the private VLANs and secondary VLAN trunk ports:

1. Configure the interfaces and port modes:

   [edit interfaces]

   ```
   user@switch# set xe-0/0/0 unit 0 family ethernet-switching port-mode access
   user@switch# set xe-0/0/1 unit 0 family ethernet-switching port-mode trunk
   user@switch# set xe-0/0/1 unit 0 family ethernet-switching vlan members pvlan100
   user@switch# set xe-0/0/1 unit 0 family ethernet-switching vlan members pvlan400
   user@switch# set xe-0/0/2 unit 0 family ethernet-switching port-mode trunk
   user@switch# set xe-0/0/3 unit 0 family ethernet-switching port-mode access
   user@switch# set xe-0/0/5 unit 0 family ethernet-switching port-mode access
   user@switch# set xe-0/0/6 unit 0 family ethernet-switching port-mode access
   ```

2. Create the primary VLANs:

   [edit vlans]

   ```
   user@switch# set pvlan100 vlan-id 100
   user@switch# set pvlan400 vlan-id 400
   ```

3. Configure the primary VLANs to be private:

   [edit vlans]

   ```
   user@switch# set pvlan100 pvlan
   user@switch# set pvlan400 pvlan
   ```

4. Configure the PVLAN trunk port to carry the private VLAN traffic between the switches:

   [edit vlans]

   ```
   user@switch# set pvlan100 interface xe-0/0/1.0 pvlan-trunk
   user@switch# set pvlan400 interface xe-0/0/1.0 pvlan-trunk
   ```

5. Create secondary VLAN comm300 with VLAN ID 300:

   [edit vlans]

   ```
   user@switch# set comm300 vlan-id 300
   ```

6. Configure the primary VLAN for comm300:

   [edit vlans]

   ```
   user@switch# set comm300 primary-vlan pvlan100
   ```

7. Configure the interface for comm300:

[edit vlans]

```
user@switch# set comm300 interface xe-0/0/3.0
```

8. Create secondary VLAN comm600 with VLAN ID 600:

   [edit vlans]

   ```
   user@switch# set comm600 vlan-id 600
   ```

9. Configure the primary VLAN for comm600:

   [edit vlans]

   ```
   user@switch# set comm600 primary-vlan pvlan400
   ```

10. Configure the interface for comm600:

    [edit vlans]

    ```
    user@switch# set comm600 interface xe-0/0/6.0
    ```

11. Configure the interswitch isolated VLANs:

    [edit vlans]

    ```
    user@switch# set pvlan100 pvlan isolation-vlan-id 200
    user@switch# set pvlan400 pvlan isolation-vlan-id 500
    ```

12. Configure access port xe-0/0/0 to be promiscuous for pvlan100:

    [edit vlans]

    ```
    user@switch# set pvlan100 interface xe-0/0/0.0 promiscuous
    ```

    NOTE: A promiscuous access port can be a member of only one primary VLAN.

13. Configure xe-0/0/2 and xe-0/0/6 to be isolated:

    [edit vlans]

    ```
    user@switch# set pvlan100 interface xe-0/0/2.0 isolated
    user@switch# set pvlan400 interface xe-0/0/5.0 isolated
    ```

Results

Check the results of the configuration on Switch 2:

```
[edit]
user@switch#  show
interfaces {
    xe-0/0/0 {
        unit 0 {
            family ethernet-switching {
                port-mode access;
                vlan {
                    members pvlan100;
                }
            }
        }
    }
        xe-0/0/1 {
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
                vlan {
                    members pvlan100;
                    members pvlan400;
                }
            }
        }
    }
    xe-0/0/2 {
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
            }
        }
    }
    xe-0/0/3 {
        unit 0 {
            family ethernet-switching {
                port-mode access;
            }
        }
    }
    xe-0/0/5 {
        unit 0 {
            family ethernet-switching {
                port-mode access;
            }
        }
```

```
    }
    xe-0/0/6 {
        unit 0 {
            family ethernet-switching {
                port-mode access;
            }
        }
    }
vlans {
    comm300 {
        vlan-id 300;
            interface {
            xe-0/0/3.0;
        }
        primary-vlan pvlan100;
    }
    comm600 {
        vlan-id 600;
            interface {
            xe-0/0/6.0;
        }
        primary-vlan pvlan400;
    }
    pvlan100 {
        vlan-id 100;
        interface {
            xe-0/0/0.0;
            xe-0/0/2.0;
            xe-0/0/3.0;
            xe-0/0/1.0 {
                pvlan-trunk;
            }
        }
        no-local-switching;
        isolation-id 200;
    }
    pvlan400 {
        vlan-id 400;
        interface {
            xe-0/0/5.0;
            xe-0/0/6.0;
            xe-0/0/1.0 {
                pvlan-trunk;
            }
```

```
        }
        no-local-switching;
        isolation-id 500;
    }
}
```

Verification

IN THIS SECTION

To confirm that the configuration is working properly, perform these tasks:

Verifying That the Private VLAN and Secondary VLANs Were Created

Purpose

Verify that the primary VLAN and secondary VLANs were properly created on Switch 1.

Action

Use the **show vlans** command:

`user@switch>` **show vlans private-vlan**

```
Name               Role        Tag      Interfaces
pvlan100           Primary     100      xe-0/0/0.0, xe-0/0/1.0, xe-0/0/2.0,
xe-0/0/3.0
__iso_pvlan100__   Isolated    200      xe-0/0/2.0
comm300            Community   300      xe-0/0/3.0

pvlan400           Primary     400      xe-0/0/0.0, xe-0/0/1.0, xe-0/0/5.0,
xe-0/0/6.0
__iso_pvlan400__   Isolated    500      xe-0/0/5.0
comm600            Community   600     xe-0/0/6.0
```

Meaning

The output shows that the private VLANs were created and identifies the interfaces and secondary VLANs associated with them.

Verifying The Ethernet Switching Table Entries

Purpose

Verify that the Ethernet switching table entries were created for primary VLAN pvlan100.

Action

Show the Ethernet switching table entries for pvlan100.

```
user@switch> show ethernet-switching table vlan pvlan100 private-vlan
```

```
Ethernet-switching table: 0 unicast entries
  pvlan100             *                 Flood          - All-members
  pvlan100             00:10:94:00:00:02 Learn            xe-0/0/2.0
  __iso_pvlan100__     *                 Flood          - All-members
  __iso_pvlan100__     00:10:94:00:00:02 Replicated      - xe-0/0/2.0
```

SEE ALSO

> *Understanding Egress Firewall Filters with PVLANs*

Verifying That a Private VLAN Is Working on a Switch

Purpose

After creating and configuring private VLANs (PVLANs), verify that they are set up properly.

Action

1. To determine whether you successfully created the primary and secondary VLAN configurations:
   - For a PVLAN on a single switch, use the **show configuration vlans** command:

     ```
     user@switch> show configuration vlans
     ```

     ```
     community1 {
         interface {
             interface a;
             interface b;
         }
         primary-vlan pvlan;
     ```

```
}
community2 {
    interface {
        interface d;
        interface e;
    }
    primary-vlan pvlan;
}
pvlan {
    vlan-id 1000;
    interface {
        isolated1;
        isolated2;
        trunk1;
        trunk2;
    }
    no-local-switching;
}
```

- For a PVLAN spanning multiple switches, use the **show vlans extensive** command:

  user@switch> **show vlans extensive**

```
VLAN: COM1, Created at: Tue May 11 18:16:05 2010
802.1Q Tag: 100, Internal index: 3, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 3 (Active = 3), Untagged  1 (Active = 1)
      ge-0/0/20.0*, tagged, trunk
      ge-0/0/22.0*, tagged, trunk, pvlan-trunk
      ge-0/0/23.0*, tagged, trunk, pvlan-trunk
      ge-0/0/7.0*, untagged, access

VLAN: __pvlan_primary_ge-0/0/0.0__, Created at: Tue May 11 18:16:05 2010
Internal index: 5, Admin State: Enabled, Origin: Static
Private VLAN Mode: Isolated, Primary VLAN: primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 3 (Active = 3), Untagged  1 (Active = 1)
      ge-0/0/20.0*, tagged, trunk
      ge-0/0/22.0*, tagged, trunk, pvlan-trunk
      ge-0/0/23.0*, tagged, trunk, pvlan-trunk
      ge-0/0/0.0*, untagged, access
```

```
VLAN: __pvlan_primary_ge-0/0/2.0__, Created at: Tue May 11 18:16:05 2010
Internal index: 6, Admin State: Enabled, Origin: Static
Private VLAN Mode: Isolated, Primary VLAN: primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 3 (Active = 3), Untagged  1 (Active = 0)
      ge-0/0/20.0*, tagged, trunk
      ge-0/0/22.0*, tagged, trunk, pvlan-trunk
      ge-0/0/23.0*, tagged, trunk, pvlan-trunk
      ge-0/0/2.0, untagged, access


VLAN: __pvlan_primary_isiv__, Created at: Tue May 11 18:16:05 2010
802.1Q Tag: 50, Internal index: 7, Admin State: Enabled, Origin: Static
Private VLAN Mode: Inter-switch-isolated, Primary VLAN: primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 3 (Active = 3), Untagged  0 (Active = 0)
      ge-0/0/20.0*, tagged, trunk
      ge-0/0/22.0*, tagged, trunk, pvlan-trunk
      ge-0/0/23.0*, tagged, trunk, pvlan-trunk


VLAN: community2, Created at: Tue May 11 18:16:05 2010
802.1Q Tag: 20, Internal index: 8, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 3 (Active = 3), Untagged  2 (Active = 2)
      ge-0/0/20.0*, tagged, trunk
      ge-0/0/22.0*, tagged, trunk, pvlan-trunk
      ge-0/0/23.0*, tagged, trunk, pvlan-trunk
      ge-0/0/1.0*, untagged, access
      ge-1/0/6.0*, untagged, access


VLAN: primary, Created at: Tue May 11 18:16:05 2010
802.1Q Tag: 10, Internal index: 2, Admin State: Enabled, Origin: Static
Private VLAN Mode: Primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 3 (Active = 3), Untagged  5 (Active = 4)
      ge-0/0/20.0*, tagged, trunk
      ge-0/0/22.0*, tagged, trunk, pvlan-trunk
      ge-0/0/23.0*, tagged, trunk, pvlan-trunk
      ge-0/0/0.0*, untagged, access
      ge-0/0/1.0*, untagged, access
      ge-0/0/2.0, untagged, access
```

```
      ge-0/0/7.0*, untagged, access
      ge-1/0/6.0*, untagged, access

Secondary VLANs: Isolated 2, Community  2, Inter-switch-isolated  1
  Isolated VLANs :
      __pvlan_primary_ge-0/0/0.0__
      __pvlan_primary_ge-0/0/2.0__
  Community VLANs :
      COM1
      community2
  Inter-switch-isolated VLAN :
      __pvlan_primary_isiv__
```

2. Use the **show vlans extensive** command to view VLAN information and link status for a PVLAN on a single switch or for a PVLAN spanning multiple switches.

   - For a PVLAN on a single switch:

     `user@switch>` **show vlans pvlan extensive**

     ```
     VLAN: pvlan, Created at: time
     802.1Q Tag: vlan-id, Internal index: index-number, Admin State: Enabled, Origin:
      Static
     Private VLAN Mode: Primary
     Protocol: Port Mode
     Number of interfaces: Tagged 2 (Active = 0), Untagged  6 (Active = 0)
           trunk1, tagged, trunk
           interface a, untagged, access
           interface b, untagged, access
           interface c, untagged, access
           interface d, untagged, access
           interface e, untagged, access
           interface f, untagged, access
           trunk2, tagged, trunk
     Secondary VLANs: Isolated 2, Community  2
       Isolated VLANs :
           __pvlan_pvlan_isolated1__
           __pvlan_pvlan_isolated2__
       Community VLANs :
           community1
           community2
     ```

   - For a PVLAN spanning multiple switches:

user@switch> **show vlans extensive**

```
VLAN: COM1, Created at: Tue May 11 18:16:05 2010
802.1Q Tag: 100, Internal index: 3, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 3 (Active = 3), Untagged  1 (Active = 1)
      ge-0/0/20.0*, tagged, trunk
      ge-0/0/22.0*, tagged, trunk, pvlan-trunk
      ge-0/0/23.0*, tagged, trunk, pvlan-trunk
      ge-0/0/7.0*, untagged, access

VLAN: __pvlan_primary_ge-0/0/0.0__, Created at: Tue May 11 18:16:05 2010
Internal index: 5, Admin State: Enabled, Origin: Static
Private VLAN Mode: Isolated, Primary VLAN: primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 3 (Active = 3), Untagged  1 (Active = 1)
      ge-0/0/20.0*, tagged, trunk
      ge-0/0/22.0*, tagged, trunk, pvlan-trunk
      ge-0/0/23.0*, tagged, trunk, pvlan-trunk
      ge-0/0/0.0*, untagged, access


VLAN: __pvlan_primary_ge-0/0/2.0__, Created at: Tue May 11 18:16:05 2010
Internal index: 6, Admin State: Enabled, Origin: Static
Private VLAN Mode: Isolated, Primary VLAN: primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 3 (Active = 3), Untagged  1 (Active = 0)
      ge-0/0/20.0*, tagged, trunk
      ge-0/0/22.0*, tagged, trunk, pvlan-trunk
      ge-0/0/23.0*, tagged, trunk, pvlan-trunk
      ge-0/0/2.0, untagged, access


VLAN: __pvlan_primary_isiv__, Created at: Tue May 11 18:16:05 2010
802.1Q Tag: 50, Internal index: 7, Admin State: Enabled, Origin: Static
Private VLAN Mode: Inter-switch-isolated, Primary VLAN: primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 3 (Active = 3), Untagged  0 (Active = 0)
      ge-0/0/20.0*, tagged, trunk
      ge-0/0/22.0*, tagged, trunk, pvlan-trunk
      ge-0/0/23.0*, tagged, trunk, pvlan-trunk


VLAN: community2, Created at: Tue May 11 18:16:05 2010
```

```
802.1Q Tag: 20, Internal index: 8, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 3 (Active = 3), Untagged  2 (Active = 2)
      ge-0/0/20.0*, tagged, trunk
      ge-0/0/22.0*, tagged, trunk, pvlan-trunk
      ge-0/0/23.0*, tagged, trunk, pvlan-trunk
      ge-0/0/1.0*, untagged, access
      ge-1/0/6.0*, untagged, access


VLAN: primary, Created at: Tue May 11 18:16:05 2010
802.1Q Tag: 10, Internal index: 2, Admin State: Enabled, Origin: Static
Private VLAN Mode: Primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 3 (Active = 3), Untagged  5 (Active = 4)
      ge-0/0/20.0*, tagged, trunk
      ge-0/0/22.0*, tagged, trunk, pvlan-trunk
      ge-0/0/23.0*, tagged, trunk, pvlan-trunk
      ge-0/0/0.0*, untagged, access
      ge-0/0/1.0*, untagged, access
      ge-0/0/2.0, untagged, access
      ge-0/0/7.0*, untagged, access
      ge-1/0/6.0*, untagged, access

Secondary VLANs: Isolated 2, Community  2, Inter-switch-isolated  1
  Isolated VLANs :
      __pvlan_primary_ge-0/0/0.0__
      __pvlan_primary_ge-0/0/2.0__
  Community VLANs :
      COM1
      community2
  Inter-switch-isolated VLAN :
      __pvlan_primary_isiv__
```

3. Use the **show ethernet-switching table** command to view logs for MAC learning on the VLANs:

   `user@switch>` **show ethernet-switching table**

```
Ethernet-switching table: 8 entries, 1 learned

  VLAN              MAC address       Type         Age Interfaces
```

```
default                 *               Flood          - All-members
pvlan                   *               Flood          - All-members
pvlan                   MAC1            Replicated     - interface a
pvlan                   MAC2            Replicated     - interface c
pvlan                   MAC3            Replicated     - isolated2
pvlan                   MAC4            Learn          0 trunk1
__pvlan_pvlan_isolated1__ *             Flood          - All-members
__pvlan_pvlan_isolated1__ MAC4          Replicated     - trunk1
__pvlan_pvlan_isolated2__ *             Flood          - All-members
__pvlan_pvlan_isolated2__ MAC3          Learn          0 isolated2
__pvlan_pvlan_isolated2__ MAC4          Replicated     - trunk1
community1              *               Flood          - All-members
community1              MAC1            Learn          0 interface a
community1              MAC4            Replicated     - trunk1
community2              *               Flood          - All-members
community2              MAC2            Learn          0 interface c
community2              MAC4            Replicated     - trunk1
```

NOTE: If you have configured a PVLAN spanning multiple switches, you can use the same command on all the switches to check the logs for MAC learning on those switches.

Meaning

In the output displays for a PVLAN on a single switch, you can see that the primary VLAN contains two community domains (**community1** and **community2**), two isolated ports, and two trunk ports. The PVLAN on a single switch has only one tag (**1000**), which is for the primary VLAN.

The PVLAN that spans multiple switches contains multiple tags:

- The community domain **COM1** is identified with tag **100**.
- The community domain **community2** is identified with tag **20**.
- The interswitch isolated domain is identified with tag **50**.
- The primary VLAN **primary** is identified with tag **10**.

Also, for the PVLAN that spans multiple switches, the trunk interfaces are identified as **pvlan-trunk**.

Troubleshooting Private VLANs on QFX Switches

IN THIS SECTION

Use the following information to troubleshoot a private VLAN configuration.

Limitations of Private VLANs

The following constraints apply to private VLAN configurations:

- IGMP snooping is not supported with private VLANs.
- Routed VLAN interfaces are not supported on private VLANs
- Routing between secondary VLANs in the same primary VLAN is not supported.
- If you want to change a primary VLAN to be a secondary VLAN, you must first change it to a normal VLAN and commit the change. For example, you would follow this procedure:

1. Change the primary VLAN to be a normal VLAN.

2. Commit the configuration.

3. Change the normal VLAN to be a secondary VLAN.

4. Commit the configuration.

Follow the same sequence of commits if you want to change a secondary VLAN to be a primary VLAN. That is, make the secondary VLAN a normal VLAN and commit that change and then change the normal VLAN to be a primary VLAN.

Forwarding with Private VLANs

Problem
Description:

- When isolated VLAN or community VLAN tagged traffic is received on a PVLAN trunk port, MAC addresses are learned from the primary VLAN. This means that output from the **show ethernet-switching table** command shows that MAC addresses are learned from the primary VLAN and replicated to secondary VLANs. This behavior has no effect on forwarding decisions.
- If a packet with a secondary VLAN tag is received on a promiscuous port, it is accepted and forwarded.
- If a packet is received on a PVLAN trunk port and meets both of the conditions listed below, it is dropped.
  - The packet has a community VLAN tag.
  - The packet is destined to a unicast MAC address or multicast group MAC address that was learned on an isolated VLAN.
- If a packet is received on a PVLAN trunk port and meets both of the conditions listed below, it is dropped.
  - The packet has an isolated VLAN tag.
  - The packet is destined to a unicast MAC address or multicast group MAC address that was learned on a community VLAN.
- If a packet with a primary VLAN tag is received by a secondary (isolated or community) VLAN port, the secondary port forwards the packet.
- If you configure a community VLAN on one device and configure another community VLAN on a second device and both community VLANs use the same VLAN ID, traffic for one of the VLANs can be forwarded to the other VLAN. For example, assume the following configuration:
  - Community VLAN comm1 on switch 1 has VLAN ID 50 and is a member of primary VLAN pvlan100.
  - Community VLAN comm2 on switch 2 also has VLAN ID 50 and is a member of primary VLAN pvlan200.
  - Primary VLAN pvlan100 exists on both switches.

If traffic for comm1 is sent from switch 1 to switch 2, it will be sent to the ports participating in comm2. (The traffic will also be forwarded to the ports in comm1, as you would expect.)

Solution

These are expected behaviors.

Egress Firewall Filters with Private VLANs

Problem

Description: If you apply a firewall filter in the output direction to a primary VLAN, the filter also applies to the secondary VLANs that are members of the primary VLAN when the traffic egresses with the primary VLAN tag or isolated VLAN tag, as listed below:

- Traffic forwarded from a secondary VLAN trunk port to a promiscuous port (trunk or access)
- Traffic forwarded from a secondary VLAN trunk port that carries an isolated VLAN to a PVLAN trunk port.
- Traffic forwarded from a promiscuous port (trunk or access) to a secondary VLAN trunk port
- Traffic forwarded from a PVLAN trunk port. to a secondary VLAN trunk port
- Traffic forwarded from a community port to a promiscuous port (trunk or access)

If you apply a firewall filter in the output direction to a primary VLAN, the filter does *not* apply to traffic that egresses with a community VLAN tag, as listed below:

- Traffic forwarded from a community trunk port to a PVLAN trunk port
- Traffic forwarded from a secondary VLAN trunk port that carries a community VLAN to a PVLAN trunk port
- Traffic forwarded from a promiscuous port (trunk or access) to a community trunk port
- Traffic forwarded from a PVLAN trunk port. to a community trunk port

If you apply a firewall filter in the output direction to a community VLAN, the following behaviors apply:

- The filter is applied to traffic forwarded from a promiscuous port (trunk or access) to a community trunk port (because the traffic egresses with the community VLAN tag).
- The filter is applied to traffic forwarded from a community port to a PVLAN trunk port (because the traffic egresses with the community VLAN tag).
- The filter is *not* applied to traffic forwarded from a community port to a promiscuous port (because the traffic egresses with the primary VLAN tag or untagged).

Solution

These are expected behaviors. They occur only if you apply a firewall filter to a private VLAN in the output direction and do not occur if you apply a firewall filter to a private VLAN in the input direction.

Egress Port Mirroring with Private VLANs

Problem

Description: If you create a port-mirroring configuration that mirrors private VLAN (PVLAN) traffic on egress, the mirrored traffic (the traffic that is sent to the analyzer system) has the VLAN tag of the ingress VLAN instead of the egress VLAN. For example, assume the following PVLAN configuration:

- Promiscuous trunk port that carries primary VLANs pvlan100 and pvlan400.
- Isolated access port that carries secondary VLAN isolated200. This VLAN is a member of primary VLAN pvlan100.
- Community port that carries secondary VLAN comm300. This VLAN is also a member of primary VLAN pvlan100.
- Output interface (monitor interface) that connects to the analyzer system. This interface forwards the mirrored traffic to the analyzer.

If a packet for pvlan100 enters on the promiscuous trunk port and exits on the isolated access port, the original packet is untagged on egress because it is exiting on an access port. However, the mirror copy retains the tag for pvlan100 when it is sent to the analyzer.

Here is another example: If a packet for comm300 ingresses on the community port and egresses on the promiscuous trunk port, the original packet carries the tag for pvlan100 on egress, as expected. However, the mirrored copy retains the tag for comm300 when it is sent to the analyzer.

Solution

This is expected behavior.

Understanding Private VLANs

IN THIS SECTION

VLANs limit broadcasts to specified users. Private VLANs (PVLANs) take this concept a step further by limiting communication within a VLAN. PVLANs accomplish this by restricting traffic flows through their member switch ports (which are called *private ports*) so that these ports communicate only with a specified uplink trunk port or with specified ports within the same VLAN. The uplink trunk port or link aggregation group (LAG) is usually connected to a router, firewall, server, or provider network. Each PVLAN typically contains many private ports that communicate only with a single uplink port, thereby preventing the ports from communicating with each other.

PVLANs provide Layer 2 isolation between ports within a VLAN, splitting a broadcast domain into multiple discrete broadcast subdomains by creating secondary VLANs (community VLANs and an isolated VLAN) inside a primary VLAN. Ports within the same community VLAN can communicate with each other. Ports within an isolated VLAN can communicate *only* with a single uplink port.

Just like regular VLANs, PVLANs are isolated on Layer 2 and require one of the following options to route Layer 3 traffic among the secondary VLANs:

- A promiscuous port connection with a router
- A routed VLAN interface (RVI)

NOTE: To route Layer 3 traffic among secondary VLANs, a PVLAN needs only one of the options mentioned above. If you use an RVI, you can still implement a promiscuous port connection to a router with the promiscuous port set up to handle only traffic that enters and exits the PVLAN.

PVLANs are useful for restricting the flow of broadcast and unknown unicast traffic and for limiting the communication between known hosts. Service providers use PVLANs to keep their customers isolated from each other. Another typical use for a PVLAN is to provide per-room Internet access in a hotel.

NOTE: You can configure a PVLAN to span switches that support PVLANs.

This topic explains the following concepts regarding PVLANs on EX Series switches:

Benefits of PVLANs

The need to segregate a single VLAN is particularly useful in the following deployment scenarios:

- Server farms—A typical Internet service provider uses a server farm to provide Web hosting for numerous customers. Locating the various servers within a single server farm provides ease of management. Security concerns arise if all servers are in the same VLAN because Layer 2 broadcasts go to all servers in the VLAN.
- Metropolitan Ethernet networks—A metro service provider offers Layer 2 Ethernet access to assorted homes, rental communities, and businesses. The traditional solution of deploying one VLAN per customer is not scalable and is difficult to manage, leading to potential waste of IP addresses. PVLANs provide a more secure and more efficient solution.

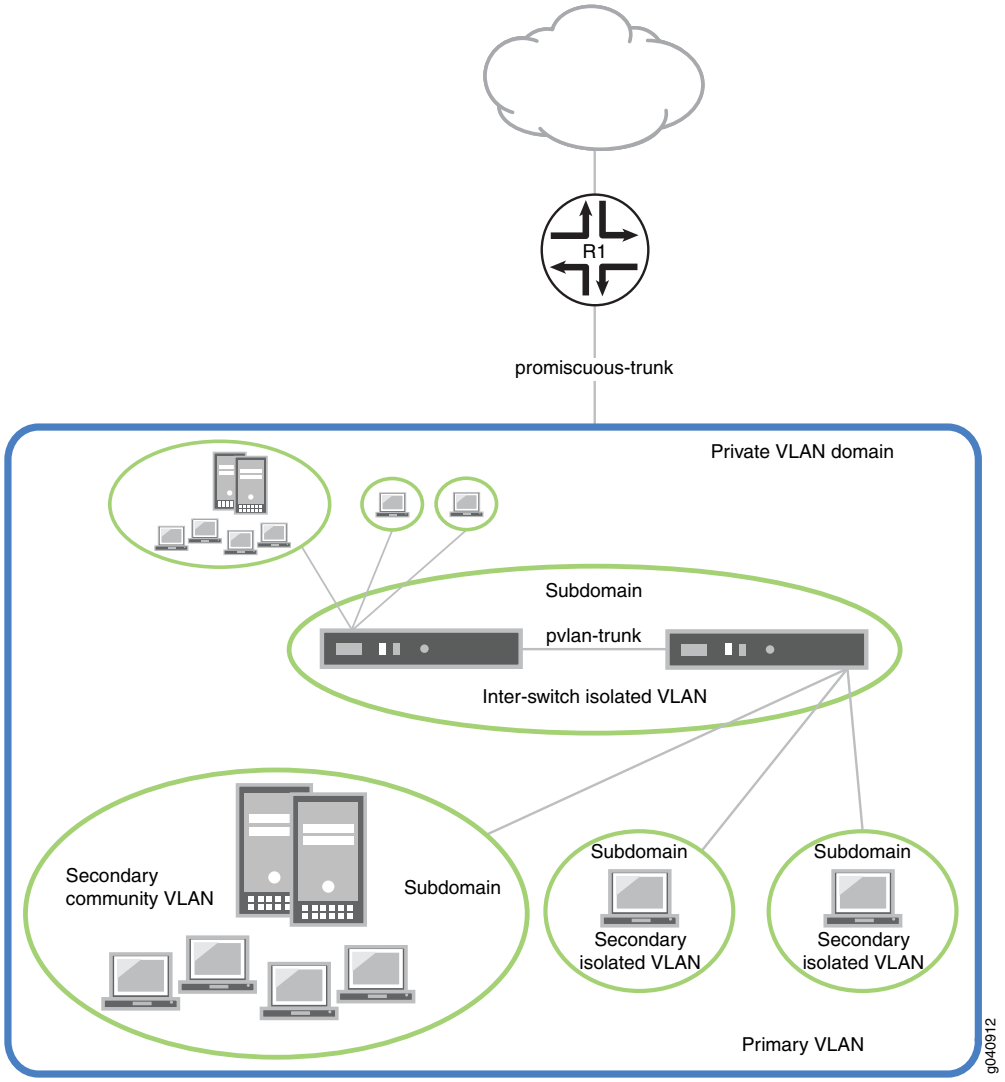
Typical Structure and Primary Application of PVLANs

A PVLAN can be configured on a single switch or can be configured to span multiple switches. The types of domains and ports are:

- Primary VLAN—The primary VLAN of the PVLAN is defined with an 802.1Q tag (VLAN ID) for the complete PVLAN. The primary PVLAN can contain multiple secondary VLANs (one isolated VLAN and multiple community VLANs).
- Isolated VLAN/isolated port—A primary VLAN can contain only one isolated VLAN. An interface within an isolated VLAN can forward packets only to a promiscuous port or the Inter-Switch Link (ISL) port. An isolated interface cannot forward packets to another isolated interface; and an isolated interface cannot receive packets from another isolated interface. If a customer device needs to have access *only* to a gateway router, the device must be attached to an isolated trunk port.
- Community VLAN/community port—You can configure multiple community VLANs within a single PVLAN. An interface within a specific community VLAN can establish Layer 2 communications with any other interface that belongs to the same community VLAN. An interface within a community VLAN can also communicate with a promiscuous port or the ISL port. If you have, for example, two customer devices that you need to isolate from other customer devices but that must be able to communicate with one another, use community ports.
- Promiscuous port—A promiscuous port has Layer 2 communications with all interfaces in the PVLAN, regardless of whether an interface belongs to an isolated VLAN or a community VLAN. A promiscuous port is a member of the primary VLAN but is not included within any secondary subdomain. Layer 3 gateways, DHCP servers, and other trusted devices that need to communicate with endpoint devices are typically connected to a promiscuous port.
- Inter-Switch Link (ISL)—An ISL is a trunk port that connects multiple switches in a PVLAN and contains two or more VLANs. It is required only when a PVLAN spans multiple switches.

The configured PVLAN is the *primary* domain (primary VLAN). Within the PVLAN, you configure *secondary* VLANs, which become subdomains nested within the primary domain. A PVLAN can be configured on a single switch or can be configured to span multiple switches. The PVLAN shown in Figure 8 on page 426 includes two switches, with a primary PVLAN domain and various subdomains.

Figure 28: Subdomains in a PVLAN

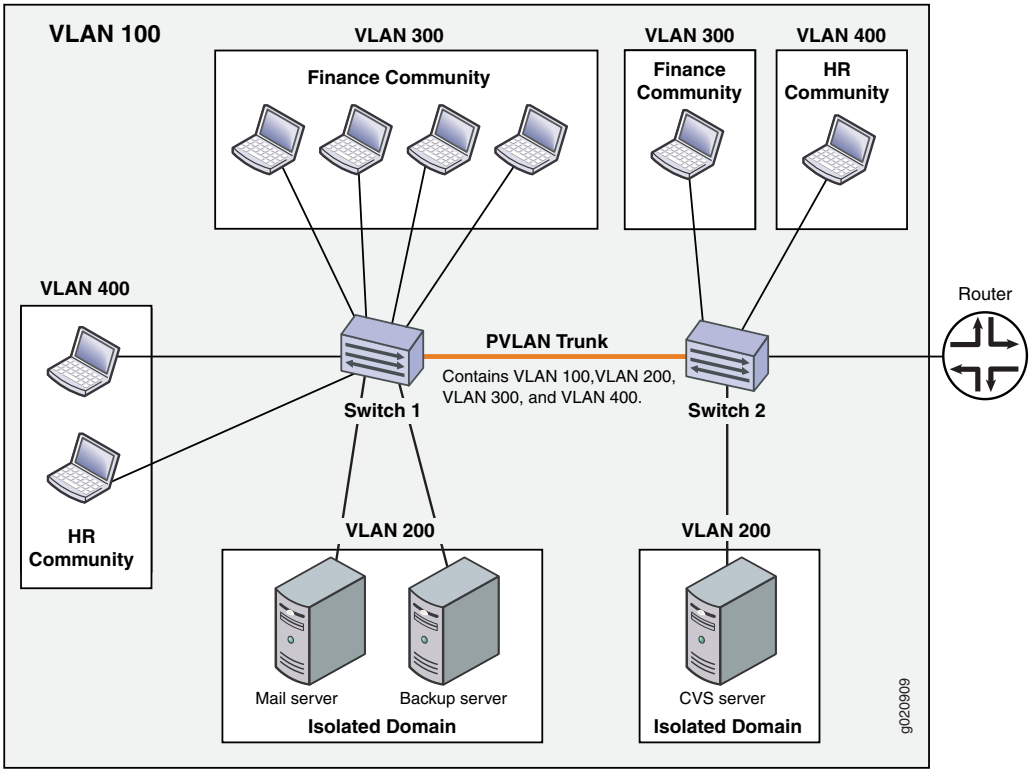


As shown in Figure 10 on page 429, a PVLAN has only one primary domain and multiple secondary domains. The types of domains are:

- Primary VLAN—VLAN used to forward frames downstream to isolated and community VLANs. The primary VLAN of the PVLAN is defined with an 802.1Q tag (VLAN ID) for the complete PVLAN. The primary PVLAN can contain multiple secondary VLANs (one isolated VLAN and multiple community VLANs).
- Secondary isolated VLAN—VLAN that receives packets only from the primary VLAN and forwards frames upstream to the primary VLAN. The isolated VLAN is a secondary VLAN nested within the primary VLAN. A primary VLAN can contain only one isolated VLAN. An interface within an isolated VLAN (isolated interface) can forward packets only to a promiscuous port or the PVLAN trunk port. An isolated interface cannot forward packets to another isolated interface; nor can an isolated interface receive packets from another isolated interface. If a customer device needs to have access *only* to a router, the device must be attached to an isolated trunk port.
- Secondary interswitch isolated VLAN—VLAN used to forward isolated VLAN traffic from one switch to another through PVLAN trunk ports. 802.1Q tags are required for interswitch isolated VLANs because IEEE 802.1Q uses an internal tagging mechanism by which a trunking device inserts a 4-byte VLAN frame identification tab into the packet header. An interswitch isolated VLAN is a secondary VLAN nested within the primary VLAN.
- Secondary community VLAN—VLAN used to transport frames among members of a community (a subset of users within the VLAN) and to forward frames upstream to the primary VLAN. A community VLAN is a secondary VLAN nested within the primary VLAN. You can configure multiple community VLANs within a single PVLAN. An interface within a specific community VLAN can establish Layer 2 communications with any other interface that belongs to the same community VLAN. An interface within a community VLAN can also communicate with a promiscuous port or the PVLAN trunk port.

Figure 9 on page 428 shows a PVLAN spanning multiple switches, where the primary VLAN (**100**) contains two community domains **(300** and **400**) and one interswitch isolated domain.

Figure 29: PVLAN Spanning Multiple Switches

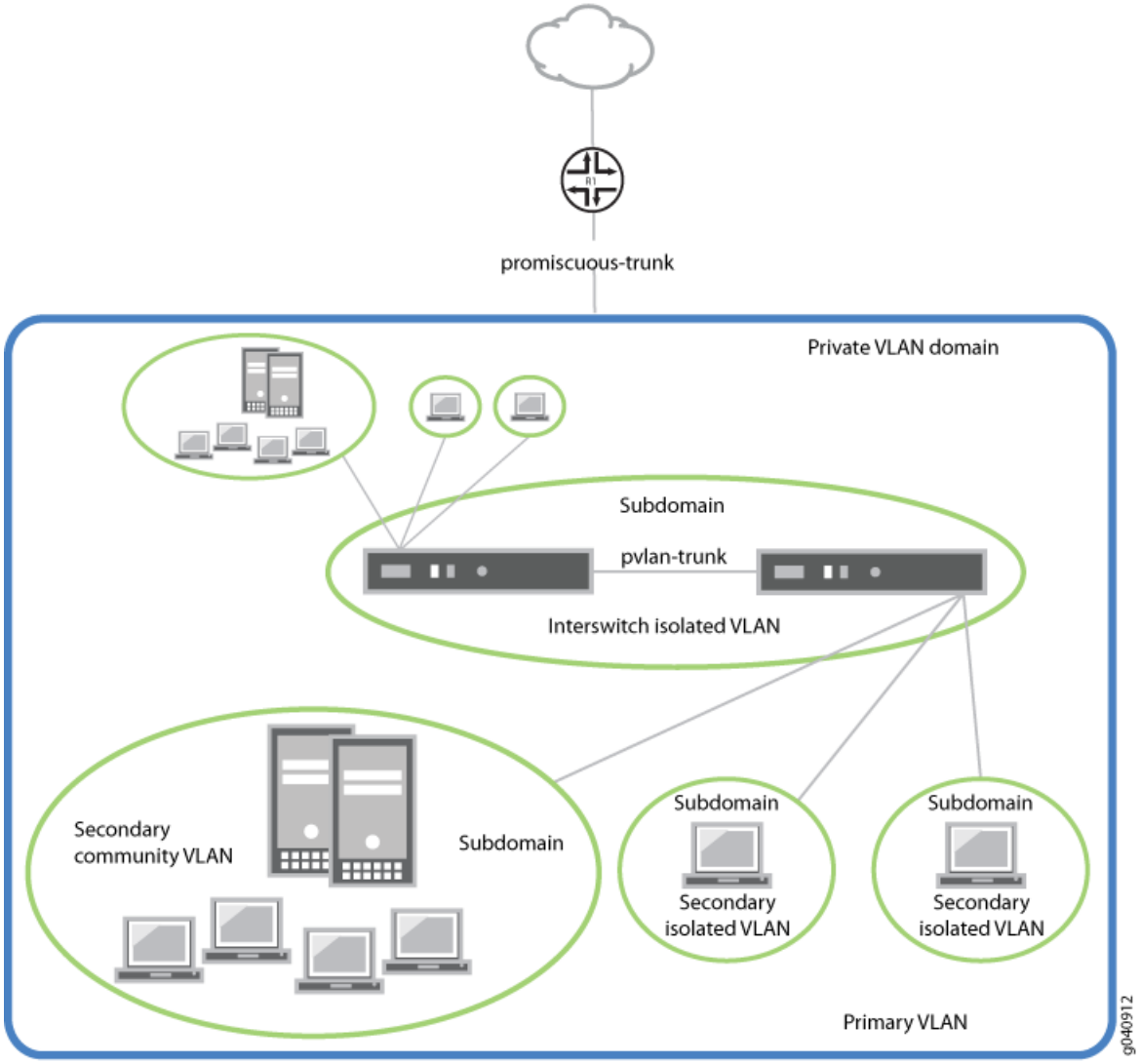


NOTE: Primary and secondary VLANs count against the limit of 4089 VLANs supported on the QFX Series. For example, each VLAN in Figure 9 on page 428 counts against this limit.

Typical Structure and Primary Application of PVLANs on MX Series Routers

The configured PVLAN becomes the primary domain, and secondary VLANs become subdomains that are nested inside the primary domain. A PVLAN can be created on a single router. The PVLAN shown in Figure 10 on page 429 includes one router, with one primary PVLAN domain and multiple secondary subdomains.

Figure 30: Subdomains in a PVLAN With One Router



The types of domains are:

- Primary VLAN—VLAN used to forward frames downstream to isolated and community VLANs.
- Secondary isolated VLAN—VLAN that receives packets only from the primary VLAN and forwards frames upstream to the primary VLAN.
- Secondary interswitch isolated VLAN—VLAN used to forward isolated VLAN traffic from one router to another through PVLAN trunk ports.
- Secondary community VLAN—VLAN used to transport frames among members of a community, which is a subset of users within the VLAN, and to forward frames upstream to the primary VLAN.

NOTE: PVLANs are supported on MX80 routers, on MX240, MX480, and MX960 routers with DPCs in enhanced LAN mode, on MX Series routers with MPC1, MPC2, and Adaptive Services PICs.

Typical Structure and Primary Application of PVLANs on EX Series Switches

NOTE: The primary VLAN of the PVLAN is defined with an 802.1Q tag (VLAN ID) for the complete PVLAN. On EX9200 switches, each secondary VLAN must also be defined with its own separate VLAN ID.

Figure 11 on page 430 shows a PVLAN on a single switch, where the primary VLAN (VLAN **100**) contains two community VLANs (VLAN **300** and VLAN **400**) and one isolated VLAN (VLAN **50**).

Figure 31: Private VLAN on a Single EX Switch

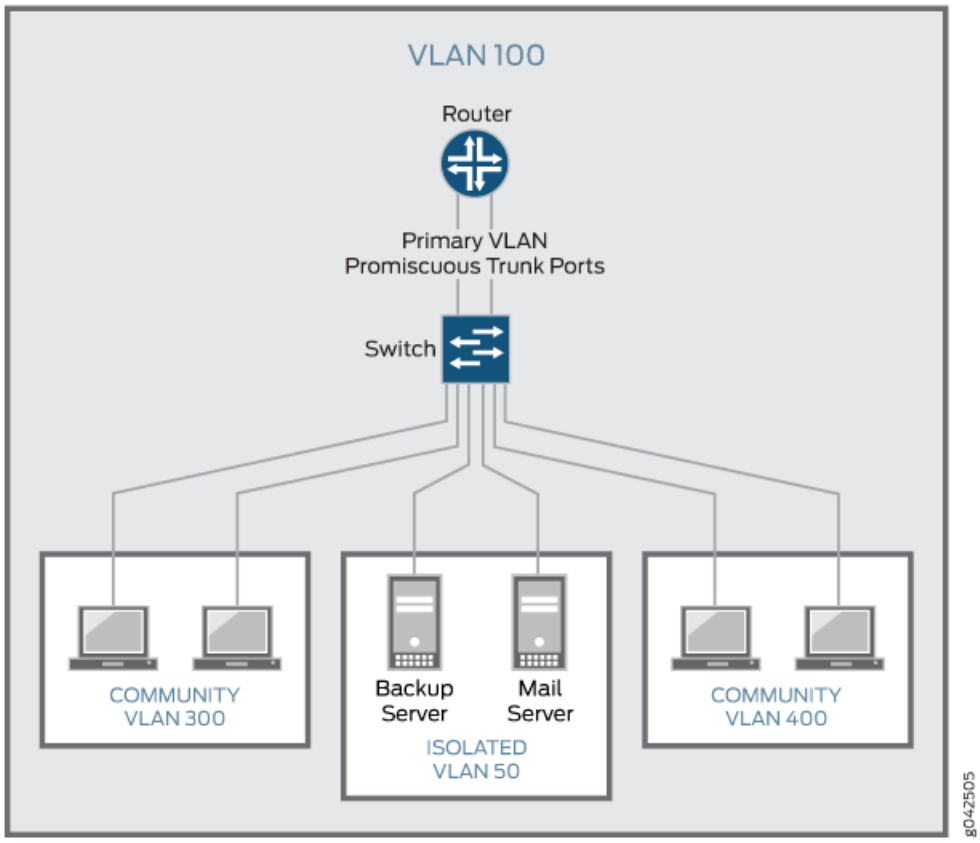
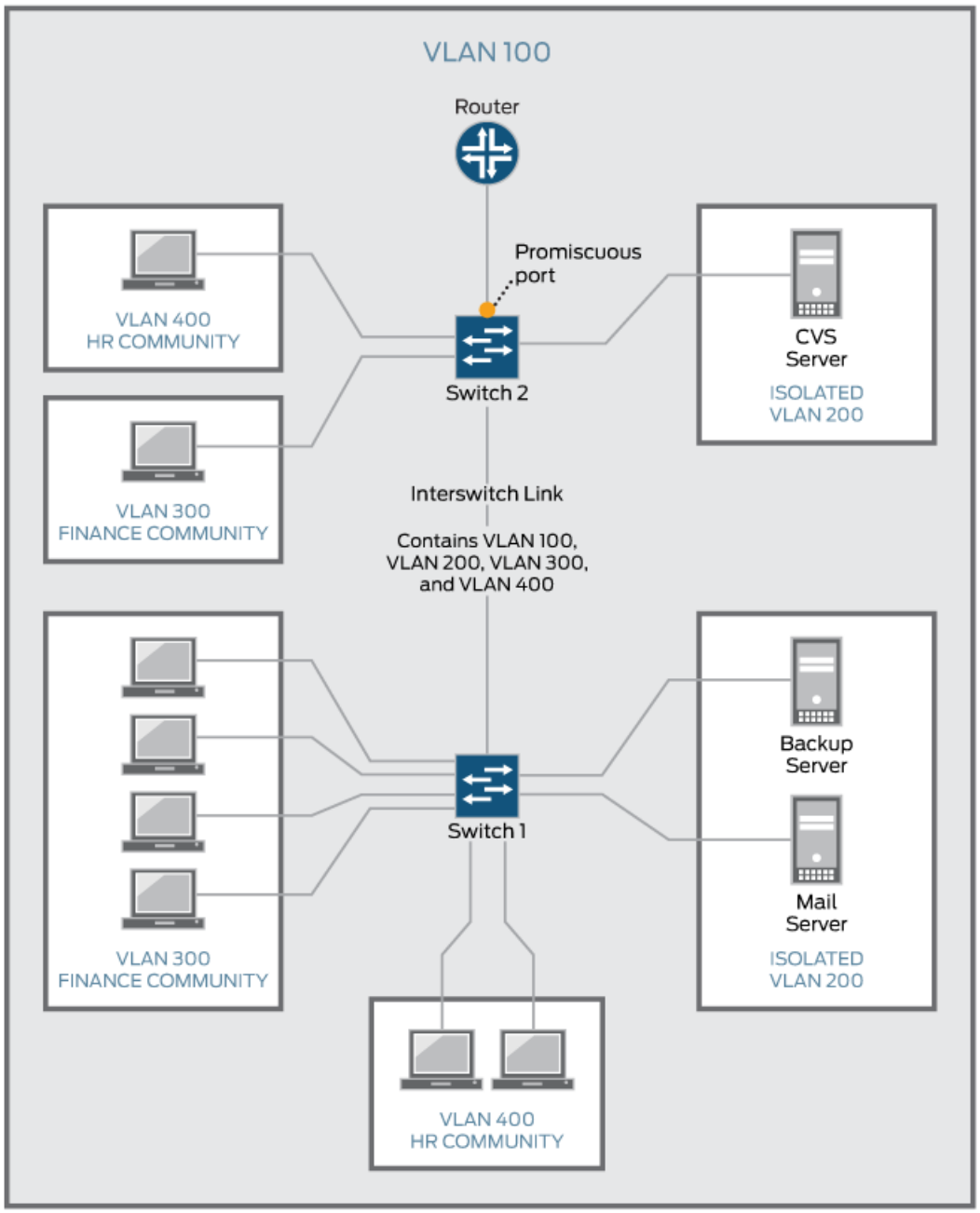


Figure 12 on page 431 shows a PVLAN spanning multiple switches, where the primary VLAN (VLAN **100**) contains two community VLANs (VLAN **300** and VLAN **400**) and one isolated VLAN (VLAN 200). It also shows that Switches 1 and 2 are connected through an interswitch link (PVLAN trunk link).

Figure 32: PVLAN Spanning Multiple EX Series Switches



Also, the PVLANs shown in Figure 11 on page 430 and Figure 12 on page 431 use a promiscuous port connected to a router as the means to route Layer 3 traffic among the community and isolated VLANs. Instead of using the promiscuous port connected to a router, you can configure an RVI on the switch in Figure 11 on page 430 or one of the switches shown in Figure 12 on page 431 (on some EX switches).

To route Layer 3 traffic between isolated and community VLANs, you must either connect a router to a promiscuous port, as shown in Figure 11 on page 430 and Figure 12 on page 431, or configure an RVI.

If you choose the RVI option, you must configure one RVI for the primary VLAN in the PVLAN domain. This RVI serves the entire PVLAN domain regardless of whether the domain includes one or more switches. After you configure the RVI, Layer 3 packets received by the secondary VLAN interfaces are mapped to and routed by the RVI.

When setting up the RVI, you must also enable proxy Address Resolution Protocol (ARP) so that the RVI can handle ARP requests received by the secondary VLAN interfaces.

For information about configuring PVLANs on a single switch and on multiple switches, see "Creating a Private VLAN on a Single EX Series Switch (CLI Procedure)" on page 477. For information about configuring an RVI, see "Configuring a Routed VLAN Interface in a Private VLAN on an EX Series Switch" on page 723.

Routing Between Isolated and Community VLANs

To route Layer 3 traffic between isolated and community VLANs, you must connect an external router or switch to a trunk port of the primary VLAN. The trunk port of the primary VLAN is a *promiscuous* port; therefore, it can communicate with *all* the ports in the PVLAN.

PVLANs Use 802.1Q Tags to Identify Packets

When packets are marked with a customer-specific 802.1Q tag, that tag identifies ownership of the packets for any switch or router in the network. Sometimes, 802.1Q tags are needed within PVLANs to keep track of packets from different subdomains. Table 76 on page 432 indicates when a VLAN 802.1Q tag is needed on the primary VLAN or on secondary VLANs.

Table 96: When VLANs in a PVLAN Need 802.1Q Tags

	On a Single Switch	On Multiple Switches
Primary VLAN	Specify an 802.1Q tag by setting a VLAN ID.	Specify an 802.1Q tag by setting a VLAN ID.

Table 96: When VLANs in a PVLAN Need 802.1Q Tags *(continued)*

	On a Single Switch	On Multiple Switches
Secondary VLAN	No tag needed on VLANs.	VLANs need 802.1Q tags: • Specify an 802.1Q tag for each community VLAN by setting a VLAN ID. • Specify the 802.1Q tag for an isolation VLAN ID by setting an isolation ID.

PVLANs Use IP Addresses Efficiently

PVLANs provide IP address conservation and efficient allocation of IP addresses. In a typical network, VLANs usually correspond to a single IP subnet. In PVLANs, the hosts in all secondary VLANs belong to the same IP subnet because the subnet is allocated to the primary VLAN. Hosts within the secondary VLAN are assigned IP addresses based on IP subnets associated with the primary VLAN, and their IP subnet masking information reflects that of the primary VLAN subnet. However, each secondary VLAN is a separate broadcast domain.

PVLAN Port Types and Forwarding Rules

PVLANs can use up to six different port types. The network depicted inFigure 9 on page 428 uses a promiscuous port to transport information to the router, community ports to connect the finance and HR communities to their respective switches, isolated ports to connect the servers, and a PVLAN trunk port to connect the two switches. PVLAN ports have different restrictions:

- Promiscuous trunk port—A promiscuous port has Layer 2 communications with all the interfaces that are in the PVLAN, regardless of whether the interface belongs to an isolated VLAN or a community VLAN. A promiscuous port is a member of the primary VLAN, but is not included within one of the secondary subdomains. Layer 3 gateways, DHCP servers, and other trusted devices that need to communicate with endpoint devices are typically connected to a promiscuous port.
- PVLAN trunk link—The PVLAN trunk link, which is also known as the interswitch link, is required only when a PVLAN is configured to span multiple switches. The PVLAN trunk link connects the multiple switches that compose the PVLAN.
- PVLAN trunk port—A PVLAN trunk port is required in multiswitch PVLAN configurations to span the switches. The PVLAN trunk port is a member of all VLANs within the PVLAN (that is, the primary VLAN,

the community VLANs, and the interswitch isolated VLAN), and it carries traffic from the primary VLAN and all secondary VLANs. It can communicate with all ports other than the isolated ports.

Communication between a PVLAN trunk port and an isolated port is usually unidirectional. A PVLAN trunk port's membership in the interswitch isolated VLAN is egress-only, meaning that an isolated port can forward packets to a PVLAN trunk port, but a PVLAN trunk port does not forward packets to an isolated port (unless the packets ingressed on a promiscuous access port and are therefore being forwarded to all the secondary VLANs in the same primary VLAN as the promiscuous port).

- Secondary VLAN trunk port (not shown)—Secondary trunk ports carry secondary VLAN traffic. For a given private VLAN, a secondary VLAN trunk port can carry traffic for only one secondary VLAN. However, a secondary VLAN trunk port can carry traffic for multiple secondary VLANs as long as each secondary VLAN is a member of a different primary VLAN. For example, a secondary VLAN trunk port can carry traffic for a community VLAN that is part of primary VLAN pvlan100 and also carry traffic for an isolated VLAN that is part of primary VLAN pvlan400.
- Community port—Community ports communicate among themselves and with their promiscuous ports. Community ports serve only a select group of users. These interfaces are separated at Layer 2 from all other interfaces in other communities or isolated ports within their PVLAN.
- Isolated access port—Isolated ports have Layer 2 connectivity only with promiscuous ports and PVLAN trunk ports—an isolated port cannot communicate with another isolated port even if these two ports are members of the same isolated VLAN (or interswitch isolated VLAN) domain. Typically, a server, such as a mail server or a backup server, is connected on an isolated port. In a hotel, each room would typically be connected on an isolated port, meaning that room-to-room communication is not possible, but each room can access the Internet on the promiscuous port.
- Promiscuous access port (not shown)—These ports carry untagged traffic. Traffic that ingresses on a promiscuous access port is forwarded to all secondary VLAN ports on the device. If traffic ingresses into the device on a VLAN-enabled port and egresses on a promiscuous access port, the traffic is untagged on egress. If tagged traffic ingresses on a promiscuous access port, the traffic is discarded.
- Interswitch link port—An interswitch link (ISL) port is a trunk port that connects two routers when a PVLAN spans those routers. The ISL port is a member of all VLANs within the PVLAN (that is, the primary VLAN, the community VLANs, and the isolated VLAN).

  Communication between an ISL port and an isolated port is unidirectional. An ISL port's membership in the interswitch isolated VLAN is egress-only, meaning that incoming traffic on the ISL port is never assigned to the isolated VLAN. An isolated port can forward packets to a PVLAN trunk port, but a PVLAN trunk port cannot forward packets to an isolated port. Table 78 on page 434 summarizes whether Layer 2 connectivity exists between the different types of ports.

Table 77 on page 434 summarizes Layer 2 connectivity between the different types of ports within a PVLAN on EX Series switches that support ELS.

Table 97: PVLAN Ports and Layer 2 Forwarding on EX Series switches that support ELS

From Port Type	To Isolated Ports?	To Promiscuous Ports?	To Community Ports?	To Inter-Switch Link Port?
Isolated	Deny	Permit	Deny	Permit
Promiscuous	Permit	Permit	Permit	Permit
Community 1	Deny	Permit	Permit	Permit

Table 98: PVLAN Ports and Layer 2 Connectivity

Port Type	Promiscuous Trunk	PVLAN Trunk	Secondary Trunk	Community	Isolated Access	Promiscuous access
Promiscuous trunk	Yes	Yes	Yes	Yes	Yes	Yes
PVLAN trunk	Yes	Yes	Yes	Yes—same community only	Yes	Yes
Secondary Trunk	Yes	Yes	No	Yes	No	Yes
Community	Yes	Yes	Yes	Yes—same community only	No	Yes
Isolated access	Yes	Yes—unidirectional only	No	No	No	Yes
Promiscuous access	Yes	Yes	Yes	Yes	Yes	No

Table 79 on page 435 summarizes whether or not Layer 2 connectivity exists between the different types of ports within a PVLAN.

Table 99: PVLAN Ports and Layer 2 Connectivity on EX Series Switches without ELS Support

Port Type To: → From:↓	Promiscuous	Community	Isolated	PVLAN Trunk	RVI
Promiscuous	Yes	Yes	Yes	Yes	Yes
Community	Yes	Yes—same community only	No	Yes	Yes
Isolated	Yes	No	No	Yes **NOTE:** This communication is unidirectional.	Yes
PVLAN trunk	Yes	Yes—same community only	Yes **NOTE:** This communication is unidirectional.	Yes	Yes
RVI	Yes	Yes	Yes	Yes	Yes

As noted in Table 79 on page 435, Layer 2 communication between an isolated port and a PVLAN trunk port is unidirectional. That is, an isolated port can only send packets to a PVLAN trunk port, and a PVLAN trunk port can only receive packets from an isolated port. Conversely, a PVLAN trunk port cannot send packets to an isolated port, and an isolated port cannot receive packets from a PVLAN trunk port.

NOTE: If you enable **no-mac-learning** on a primary VLAN, all isolated VLANs (or the interswitch isolated VLAN) in the PVLAN inherit that setting. However, if you want to disable MAC address learning on any community VLANs, you must configure **no-mac-learning** on each of those VLANs.

Creating a PVLAN

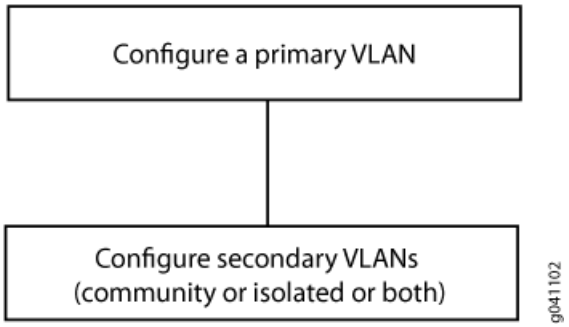
The flowchart shown in Figure 13 on page 437 gives you a general idea of the process for creating PVLANs. If you complete your configuration steps in the order shown, you will not violate these PVLAN rules. (In the PVLAN rules, configuring the PVLAN trunk port applies only to a PVLAN that spans multiple routers.)

- The primary VLAN must be a tagged VLAN.
- If you are going to configure a community VLAN ID, you must first configure the primary VLAN.
- If you are going to configure an isolation VLAN ID, you must first configure the primary VLAN.

> **NOTE:** Configuring a voice over IP (VoIP) VLAN on PVLAN interfaces is not supported.

Configuring a VLAN on a single router is relatively simple, as shown in Figure 13 on page 437.

Figure 33: Configuring a PVLAN on a Single Switch



Configuring a primary VLAN consists of these steps:

1. Configure the primary VLAN name and 802.1Q tag.
2. Set **no-local-switching** on the primary VLAN.
3. Configure the promiscuous trunk port and access ports.
4. Make the promiscuous trunk and access ports members of the primary VLAN.

Within a primary VLAN, you can configure secondary community VLANs or secondary isolated VLANs or both. Configuring a secondary community VLAN consists of these steps:

1. Configure a VLAN using the usual process.
2. Configure access interfaces for the VLAN.
3. Assign a primary VLAN to the community VLAN,

Isolated VLANs are created internally when the isolated VLAN has access interfaces as members and the option **no-local-switching** is enabled on the primary VLAN.

802.1Q tags are required for interswitch isolated VLANs because IEEE 802.1Q uses an internal tagging mechanism by which a trunking device inserts a 4-byte VLAN frame identification tab into the packet header.

Trunk ports are only needed for multirouter PVLAN configurations—the trunk port carries traffic from the primary VLAN and all secondary VLANs.

Limitations of Private VLANs

The following constraints apply to private VLAN configurations:

- An access interface can belong to only one PVLAN domain, that is, it cannot participate in two different primary VLANs.
- A trunk interface can be a member of two secondary VLANs as long as the secondary VLANs are in two *different* primary VLANs. A trunk interface cannot be a member of two secondary VLANs that are in the *same* primary VLAN.
- A single region of Multiple Spanning Tree Protocol (MSTP) must be configured on all VLANs that are included within the PVLAN.
- VLAN Spanning Tree Protocol (VSTP) is not supported.
- IGMP snooping is not supported with private VLANs.
- Routed VLAN interfaces are not supported on private VLANs
- Routing between secondary VLANs in the same primary VLAN is not supported.
- Some configuration statements cannot be specified on a secondary VLAN. You can configure the following statements at the **[edit vlans *vlan-name* switch-options]** hierarchy level *only* on the primary PVLAN.
- If you want to change a primary VLAN to be a secondary VLAN, you must first change it to a normal VLAN and commit the change. For example, you would follow this procedure:

  1. Change the primary VLAN to be a normal VLAN.
  2. Commit the configuration.
  3. Change the normal VLAN to be a secondary VLAN.
  4. Commit the configuration.

  Follow the same sequence of commits if you want to change a secondary VLAN to be a primary VLAN. That is, make the secondary VLAN a normal VLAN and commit that change and then change the normal VLAN to be a primary VLAN.

The following features are *not* supported on PVLANs on Junos switches with support for the ELS configuration style:

- DHCP security features (DHCP snooping, dynamic ARP inspection, IP source guard)
- Egress VLAN firewall filters
- Ethernet ring protection (ERP)
- Flexible VLAN tagging
- **global-mac-statistics**
- Integrated routing and bridging (IRB) interface
- Multicast snooping or IGMP snooping
- Multichassis link aggregation groups (MC-LAGs)
- Port mirroring
- Q-in-Q tunneling
- VLAN Spanning Tree Protocol (VSTP)
- Voice over IP (VoIP)

You can configure the following statements at the **[edit vlans *vlan-name* switch-options]** hierarchy level only on the primary PVLAN:

- **mac-table-size**
- **no-mac-learning**
- **mac-statistics**
- **interface-mac-limit**

RELATED DOCUMENTATION

Understanding Bridging and VLANs on Switches | **168**

Bridge Domains Setup in PVLANs on MX Series Routers

Bridge domain capabilities are used to support PVLANs on MX Series routers. Although this functionality is similar to the PVLAN mechanism on EX Series switches, the difference is that only one isolation VLAN can be configured for all isolated ports on MX routers instead of one isolation VLAN permissible per isolated port on EX Series switches.

Assume a sample deployment in which a primary VLAN named VP contains ports p1, p2, t1, t2, i1, i2, cx1, and cx2. The port types of these configured ports are as follows:

- Promiscuous ports = p1, p2
- ISL ports = t1, t2
- Isolated ports = i1, i2
- Community VLAN = Cx
- Community ports = cx1, cx2

Bridge domains are provisioned for each of the VLANs, namely, Vp, Vi, and Vcx. Assume the bridge domains to be configured as follows:

Vp—BD_primary_Vp (ports contained are p1, t1, i1, i2, cx1, cx2)

Vi—BD_isolate_Vi (ports contained are p1, t1, *i1, *i2)

Vcx—BD_community_Vcx (ports contained are p1, t1, cx1, cx2)

The bridge domains for community, primary, and isolated VLANs are automatically created by the system internally when you configure a bridge domain with a trunk interface, access interface, or interswitch link. The bridge domains contain the same VLAN ID corresponding to the VLANs. To use bridge domains for PVLANs, you must configure the following additional attributes:

- **community-vlans** option—This option is specified on all community vlans and for community BDs created internally.
- **isolated-vlan** option—This option denotes the vlan tag to be used for isolation BD created internally for each PVLAN/BD. This setting is required.
- **inter-switch-link** option with the **interface-mode trunk** statement at the **[edit interfaces interface-name family bridge]** or the **[edit interfaces interface-name unit logical-unit-number family bridge]** hierarchy level—This configuration specifies whether the particular interface assumes the role of interswitch link for the PVLAN domains of which it is a member.

You can use the **vlan-id** configuration statement for PVLAN ports to identify the port role. All the logical interfaces involved in PVLANs must be configured with a VLAN ID and the Layer 2 process uses this VLAN

tag to classify a port role as promiscuous, isolated, or community port by comparing this value with the VLANs configured in the PVLAN bridge domain (using the **bridge-domains** statement at the **[edit]** hierarchy level). The ISL port role is identified by the **inter-switch-link** option. The VLAN ID for ISL port is required and must be set to to the primary VLAN ID. The ISL must be a trunk interface. A list of VLAN IDs is not needed because the Layer 2 process creates such a list internally based on PVLAN bridge domain configuration. For untagged promiscuous, isolated or community, logical interfaces or ports, access mode must be used as the interface mode. For tagged promiscuous, isolated, or community interfaces, trunk mode must be specified as the interface mode.

The bridge domain interface families are enhanced to include ingress-only and egress-only association. The association for the interface family bridge domain (IFBD) is created in the following manner:

- For BD_primary_Vp, IFBD for i1, i2, cx1 and cx2 are egress only.
- BD_isolate_Vi, IFBD for p1 will be egress only and for i1 and i2 are ingress only.
- BD_community_Vcx , IFBD for p1 are egress only. VLAN translation rules ensure the following VLAN mappings to work properly:
  - VLAN mapping on promiscuous ports: On promiscuous ports, the Vlan Vi is mapped to Vlan Vp on egress interfaces. Similarly on promiscuous ports, Vcx is also be mapped to Vp.
  - VLAN mapping on isolation ports: On tagged isolated ports, the VLAN tag, Vp, is mapped to Vi on egress.
  - VLAN mapping on community ports: On tagged community ports, the VLAN tag, Vp, is mapped to Vcx on egress.

A management bridge domain for PVLAN that exists only in the Layer 2 address learning process called PBD to denote bridge domain for VLAN is used by the system. This bridge domain has the same name as the user-configured name. Under this bridge domain, one primary PVLAN bridge domain for the primary vlan, one isolation bridge domain for the isolation vlan, and one community bridge domain for each community vlan are programmed internally. You might find separate bridge domains for the PVLAN ports to be useful if you want to configure a policy for a specific community VLAN or isolation VLAN.

The management bridge domain maintains a list to include all internal bridge domains that belong to this PVLAN bridge domain. Isolation and community bridge domains contain a pointer or a flag to indicate that this bridge domain is for PVLANs and maintain the information about the primary bridge domain index and primary VLAN. All this information is available across the bridge domain interfaces that are mapped to this bridge domain. MAC learning occurs only in the primary bridge domain and the MAC forwarding entry is programmed into the primary bridge domain only. As a result, the isolation bridge domain and all community bridge domains share the same forwarding table as the primary bridge domain.

For the isolation bridge domain, BD_isolate_Vi, isolation port i1 and i2 function as a non-local-switch access port and the flood group for this bridge domain contains only the promiscuous port, p1, and ISL ports, t1 and t2.

Bridging Functions With PVLANs

This topic describes how bridging is implemented on MX Series routers that will help with understanding the unique enhancements involved in implementing PVLAN bridging procedures. Consider two ports in a bridging domain with the respective ports on different FPCs and different Packet Forwarding Engines. When a packet enters a port, the following is the flow, assuming it is a tagged packet:

1. As the starting process, a VLAN lookup is performed to determine which bridging domain the packet forms. The result of the lookup identifies the bridging domain id (bd_id), mesh group id (mg_id). With these parameters, other related information configured for this bridging domain is discovered.
2. A source MAC address (SMAC) lookup is performed to find out whether this MAC addresses is learned or not. If it is not a learned address, an MLP packet (route for flooding traffic to MAC learning chips) is sent to all the other Packet Forwarding Engines that are mapped with this bridging domain. In addition, an MLP packet is also sent to the host.
3. A destination MAC address (DMAC) lookup using the tuple (bridge domain ID, VLAN, and destination MAC address).
4. If a match is observed for the MAC address, the result of the lookup points to the egress next-hop. The egress Packet Forwarding Engine is used to forward the packet.
5. If a miss occurs during the lookup, the flood next-hop is determined using the mesh group ID to flood the packet.

The following two significant conditions are considered in PVLAN bridging: Only a specific port to another port forwarding is permitted. A packet drop occurs on the egress interface after traversing and consuming the fabric bandwidth. To avoid traffic dropping, the decision on whether the packet needs to be dropped arrives before traversing the fabric, thereby saving the fabric bandwidth during DoS attacks. Because multiple overlapping bridge domains exist, which denotes that the same port (promiscuous or interswitch link) appears as a member in multiple bridge domains, the MAC addresses learned in one port must be visible to ports on another bridge domain. For example, a MAC address learned on a promiscuous port must be visible to both an isolated port (isolated bridge domain) and a community port (community bridge domain) on the various community bridge domains.

To resolve this problem, a shared VLAN is used for PVLAN bridging. In the shared VLAN model, all the MACs learned across all the ports are stored in the same bridge domain (primary VLAN BD) and same VLAN (primary VLAN). When the VLAN lookup is done for the packet, the PVLAN port, PVLAN bridge domain, and the PVLAN tag or ID are also used. The following processes occur with a shared VLAN methodology:

- A source MAC address (SMAC) lookup is performed to find out whether this MAC address is learned or not. If it is not a learned address, an MLP packet (route for flooding traffic to MAC learning chips) is sent to all the other Packet Forwarding Engines that are mapped with this bridging domain. In addition, an MLP packet is also sent to the host.
- A destination MAC address (DMAC) lookup using the tuple (bridge domain ID, VLAN, and destination MAC address).
- If a match is observed for the MAC address, the result of the lookup points to the egress next-hop. The egress Packet Forwarding Engine is used to forward the packet.
- If a miss occurs during the lookup, the flood next-hop is determined using the mesh group ID to flood the packet.
- If a match occurs, the group ID is derived from the VLAN lookup table and the following validation is performed to enforce primary VLAN forwarding:

```
Steps  Source  Destination      Action
Step 1     0   {*}       Permit
Step 2     {*}    0       Permit
Step 3     1    1       Drop
Step 4  X <-> Y (X > 1 and Y > 1 and X   Y Drop
```

Here, {*} is a wildcard in regular expression notation referring to any value. Step 1 ensures all forwarding from promiscuous or inter switch link ports to any other port is permitted. Step 2 ensures all forwarding from any port to promiscuous or interswitch link ports is permitted. Step 3 ensures any isolated port to another isolated port is dropped. Step 4 ensures community port forwarding is permitted only within same community(X == Y) and dropped when its across community (X ≠ Y).

RELATED DOCUMENTATION

Flow of Frames on PVLAN Ports Overview

This topic describes the manner in which traffic that enters the different PVLAN ports, such as promiscuous, isolated, and interswitch link VLANs, is processed. Sample configuration scenarios are used to describe the transmission and processing of packets.

Assume a sample deployment in which a primary VLAN named VP contains ports, p1, p2, t1, t2, i1, i2, cx1, and cx2. The port types of these configured ports are as follows:

- Promiscuous ports = p1, p2
- ISL ports = t1, t2
- Isolated ports = i1, i2
- Community VLAN = Cx
- Community ports = cx1, cx2

Bridge domains are provisioned for each of the VLANs, namely, Vp, Vi, and Vcx. Assume the bridge domains to be configured as follows:

Vp—BD_primary_Vp (ports contained are p1, t1, i1, i2, cx1, cx2)

Vi—BD_isolate_Vi (ports contained are p1, t1, *i1, *i2)

Vcx—BD_community_Vcx (ports contained are p1, t1, cx1, cx2)

The bridge domains for community, primary, and isolated VLANs are automatically created by the system internally when you configure a bridge domain with a trunk interface, access interface, or interswitch link. The bridge domains contain the same VLAN ID corresponding to the VLANs. To use bridge domains for PVLANs, you must configure the following additional attributes:

Ingress Traffic on Isolated Ports

Consider an ingress port, i1. i1 is mapped to a bridge domain named BD_isolate_Vi. BD_isolate_Vi does not have any isolated ports as an egress member. Frames can only be sent in the egress direction on p1 and t1. When a frame is sent out on p1, it is tagged with the tag of Primary VLAN Vp. A VLAN translation of Vi to Vp is performed. When a frame is propagated out of t1, it is tagged with the tag Vi.

Ingress Traffic on Community ports

Consider an ingress port as cx1. cx1 is mapped to bridge domain BD_community_Vcx. Because of the VLAN membership with the bridge domain, frames can be sent out of p1, t1, cx1, cx2. When a frame is traversed out on p1, it is tagged with tag of Primary VLAN Vp [VLAN translation]. When a frame goes out of t1, it is tagged with tag Vcx.

Ingress Traffic on Promiscuous Ports

Consider a promisicious port p1 as the ingress port. p1 is mapped to bridge domain BD_primary_Vp. Frames can go out of any member port. When a frame goes out of t1, it is tagged with tag Vp. If another promiscuous port exists, that frame is also sent out with Vp.

Ingress Traffic on Interswitch Links

With the Vlan tag Vp, assume the ingress port as t1 mapped to bridge domain BD_primary_Vp. Frames can go out of any member port. When a frame goes out of p1, it is tagged with tag Vp. With the Vlan tag Vi, t1 mapped to bridge domain BD_isolate_Vi. The frame can not egress isolated ports as they are ingress-only members of BD_isolate_Vi. When a frame goes out on p1, it is tagged with tag of Primary VLAN Vp (VLAN translation). When a frame goes out of any other trunk port, it contains the Vi tag. With the Vlan tag Vcx, t1 is mapped to BD_community_Vcx. Frames can go out of p1, t1, cx1, and cx2. When a frame goes out on p1, it is tagged with the tag of primary VLAN Vp (VLAN translation).

Packet Forwarding in PVLANs

Consider a primary VLAN with the following configuration of ports:

```
Promiscuous  P1 P2
Inter Switch Link  L1 L2
Isolated     I1 I2
Community1    C11 C12
Community2    C21 C22
```

Internally, one global BD called the primary vlan BD is created that consists of all the ports. One isolation bridge domain consisting of all isolation ports in addition the promiscuous and ISL ports and one bridge domain per community is defined consisting of community ports in addition to the promiscuous and ISL ports internally configured in the system. The bridge domains with the PVLAN ports are as follows:

```
Primary Vlan BD P1 P2 L1 L2 I1 I2 C11 C12 C21 C22
Isolated BD           I1 I2 P1 P2 L1 L2
Community1 BD         C11 C12 P1 P2 L1 L2
Community 2 BD          C21 C22 P1 P2 L1 L2
```

The following PVLAN forwarding events take place among these ports with the appropriate VLAN translation as described in the following table:

Port Type To: → From:↓	Isolated	Community	Promiscuous	Inter-switch Link
Isolated	Dropped	Dropped	Primary VLAN tag to Isolation VLAN tag.	If received with the primary VLAN tag, translate to the isolation VLAN Tag; else dropped
Promiscuous	Dropped	No translation if it is the same community; else dropped.	Primary VLAN tag to Community VLAN tag.	If received with primary VLAN tag, translate to community VLAN tag; else no translation if received with same community vlan else dropped.
Community	Isolated VLAN tag to Primary VLAN tag	Community VLAN tag to Primary VLAN tag	No translation	If received with isolation or community VLAN tag, translate to Primary VLAN tag; else no translation
Interswitch Link	No translation	No translation	No translation	No translation

Guidelines for Configuring PVLANs on MX Series Routers

Consider the following guidelines while you configure PVLANs on MX Series routers that function in enhanced LAN mode:

- PVLANs are supported on MX80 routers, on MX240, MX480, and MX960 routers with DPCs in LAN mode, on MX Series routers with MPCs.
- Isolated ports, promiscuous ports, community ports, and interswitch links (ISL) adhere to the following rules of tagging and forwarding:
  - The frames received on the primary VLAN on promiscuous ports can go to any port.
  - The frames received on isolated ports can only go to promiscuous ports and ISL ports.

- The frames received on community ports can only go to ports of the same community, promiscuous ports, and ISL ports.
- The frames received on ISL ports with an isolation VLAN tag or ID can only go to promiscuous ports or ISL ports.
- The frames received on ISL ports with a community VLAN tag can only go to promiscuous ports, ISL ports, or ports belonging to a corresponding community port.
- The frames being sent out of promiscuous ports should have a primary VLAN tag or should be untagged. It is considered untagged if the port is configured as an untagged member of the primary VLAN. The frames going out of isolated or community ports are generally untagged. However, they can also be tagged depending on the port configuration. In any case, the configured VLAN tag must be the same as the related isolated VLAN tag or community VLAN tag.
- The frames going out of ISL ports are tagged with the primary VLAN if they are received on a promiscuous port. An untagged frame cannot exit out of an ISL port in the context of a primary VLAN, isolated VLAN, or community VLAN, but for any other VLAN, it can be untagged depending on the configuration.
- The frames going out of ISL ports are tagged with an isolated VLAN (isolation ID) if received on the isolated port.
- The frames going out of ISL ports are tagged with the community VLAN tag, if it is received on the corresponding community port.

- Graceful Routing Engine switchover (GRES) is supported for PVLANs.
- A virtual switch instance that contains a bridge domain associated with logical interfaces is supported.
- Aggregated Ethernet (ae) interfaces for all types of ports are supported.
- Virtual private LAN service (VPLS) instances is not supported. Integrated routing and bridging (IRB) interfaces in PVLANs are supported.
- MX Series Virtual Chassis configuration is not supported.
- MC-LAG interfaces are not supported. All ports that are associated with PVLAN bridge domains cannot be mc-ae interfaces.
- IGMP snooping is not supported. Q-in-Q tunneling is not supported.

RELATED DOCUMENTATION

Configuring PVLANs on MX Series Routers in Enhanced LAN Mode

You can configure a private VLAN (PVLAN) on a single MX Series router to span multiple MX Series routers. VLANs limit broadcasts to specified users. You need to specify the interswitch link (ISL) for a PVLAN, the PVLAN port types, and secondary VLANs for the PVLAN. You must create a virtual switch routing instance with a bridge domain, and associate the interfaces with the bridge domain. You can specify the secondary VLANs as isolated or community VLANs in the bridge domain.

Before you begin configuring a PVLAN, make sure you have:

- Created and configured the necessary VLANs. See "Configuring VLAN and Extended VLAN Encapsulation" on page 315 and "Enabling VLAN Tagging" on page 298.
- Configured MX240, MX480, and MX960 routers to function in enhanced LAN mode by entering the **network-services lan** statement at the **[edit chassis]** hierarchy level.

  You must reboot the router when you configure or delete the enhanced LAN mode on the router. Configuring the **network-services lan** option implies that the system is running in the enhanced IP mode. When you configure a device to function in MX-LAN mode, only the supported configuration statements and operational show commands that are available for enabling or viewing in this mode are displayed in the CLI interface.

  If your system contains parameters that are not supported in MX-LAN mode in a configuration file, you cannot commit those unsupported attributes. You must remove the settings that are not supported and then commit the configuration. After the successful CLI commit, a system reboot is required for the attributes to become effective. Similarly, if you remove the **network-services lan** statement, the system does not run in MX-LAN mode. Therefore, all of the settings that are supported outside of the MX-LAN mode are displayed and are available for definition in the CLI interface. If your configuration file contains settings that are supported only in MX-LAN mode, you must remove those attributes before you commit the configuration. After the successful CLI commit, a system reboot is required for the CLI parameters to take effect. The Layer 2 Next-Generation CLI configuration settings are supported in MX-LAN mode. As a result, the typical format of CLI configurations might differ in MX-LAN mode.

To configure a PVLAN:

1. Create a promiscuous port for the PVLAN.

   ```
   [edit interfaces]
   user@host# set interface interface-name unit logical-unit-number family bridge interface-mode trunk
   user@host# set interface interface-name unit logical-unit-number family bridge vlan-id vlan-id
   ```

2. Create the interswitch link (ISL) trunk port for the PVLAN.

```
[edit interfaces]
user@host# set interface interface-name unit logical-unit-number family bridge interface-mode trunk
   inter-switch-link
user@host# set interface interface-name unit logical-unit-number family bridge vlan-id vlan-id
```

3. Create the isolated port for the PVLAN. The port is identified as an isolated port or a community port, based on the VLAN ID or the list of VLAN IDs to which the interface corresponds. For example, if you configure a port with a VLAN ID of 50, and if you specify a VLAN ID of 50 as the isolated VLAN or tag in the bridge domain, the port is considered as an isolation port.

```
[edit interfaces]
user@host# set interface interface-name unit logical-unit-number family bridge interface-mode access
user@host# set interface interface-name unit logical-unit-number family bridge vlan-id vlan-id
```

4. Create the community port for the PVLAN. The port is identified as an isolated port or a community port, based on the VLAN ID or the list of VLAN IDs to which the interface corresponds. For example, if you configure a port with a VLAN ID of 50, and if you specify a VLAN ID of 50 as the community VLAN or tag in the bridge domain, the port is considered as a community port.

```
[edit interfaces]
user@host# set interface interface-name unit logical-unit-number family bridge interface-mode access
user@host# set interface interface-name unit logical-unit-number family bridge vlan-id vlan-id
```

5. Create a virtual switch instance with a bridge domain and associate the logical interfaces.

```
[edit routing-instances]
user@host# set routing-instance-name instance-type virtual-switch
user@host# set routing-instance-name interface interface-name unit logical-unit-number
user@host# set routing-instance-name bridge-domains bridge-domain-name
```

6. Specify the primary, isolated, and community VLAN IDs, and associate the VLANs with the bridge domain.

```
[edit routing-instances instance-name bridge-domains bridge-domain-name]
user@host# set vlan-id vlan-id
user@host# set isolated-vlan vlan-id
user@host# set community-vlans [ number number-number ]
```

RELATED DOCUMENTATION

Example: Configuring PVLANs with Secondary VLAN Trunk Ports and Promiscuous Access Ports on a QFX Series Switch

IN THIS SECTION

This example shows how to configure secondary VLAN trunk ports and promiscuous access ports as part of a private VLAN configuration. Secondary VLAN trunk ports carry secondary VLAN traffic.

For a given private VLAN, a secondary VLAN trunk port can carry traffic for only one secondary VLAN. However, a secondary VLAN trunk port can carry traffic for multiple secondary VLANs as long as each secondary VLAN is a member of a different private (primary) VLAN. For example, a secondary VLAN trunk port can carry traffic for a community VLAN that is part of primary VLAN pvlan100 and also carry traffic for an isolated VLAN that is part of primary VLAN pvlan400.

To configure a trunk port to carry secondary VLAN traffic, use the **isolated** and **interface** statements, as shown in steps 12 and 13 of the example configuration for Switch 1.

NOTE: When traffic egresses from a secondary VLAN trunk port, it normally carries the tag of the primary VLAN that the secondary port is a member of. If you want traffic that egresses from a secondary VLAN trunk port to retain its secondary VLAN tag, use the **extend-secondary-vlan-id** statement.

A promiscuous access port carries untagged traffic and can be a member of only one primary VLAN. Traffic that ingresses on a promiscuous access port is forwarded to the ports of the secondary VLANs that are members of the primary VLAN that the promiscuous access port is a member of. This traffic carries the appropriate secondary VLAN tags when it egresses from the secondary VLAN ports if the secondary VLAN port is a trunk port.

To configure an access port to be promiscuous, use the **promiscuous** statement, as shown in step 12 of the example configuration for Switch 2.

If traffic ingresses on a secondary VLAN port and egresses on a promiscuous access port, the traffic is untagged on egress. If tagged traffic ingresses on a promiscuous access port, the traffic is discarded.

Requirements

This example uses the following hardware and software components:

- Two QFX devices
- Junos OS Release 12.2 or later for the QFX Series

Overview and Topology

Figure 27 on page 567 shows the topology used in this example. Switch 1 includes several primary and secondary private VLANs and also includes two secondary VLAN trunk ports configured to carry secondary VLANs that are members of primary VLANs pvlan100 and pvlan400.

Switch 2 includes the same private VLANs. The figure shows xe-0/0/0 on Switch 2 as configured with promiscuous access ports or promiscuous trunk ports. The example configuration included here configures this port as a promiscuous access port.

The figure also shows how traffic would flow after ingressing on the secondary VLAN trunk ports on Switch 1.

Figure 34: PVLAN Topology with Secondary VLAN Trunk Ports and Promiscuous Access Port

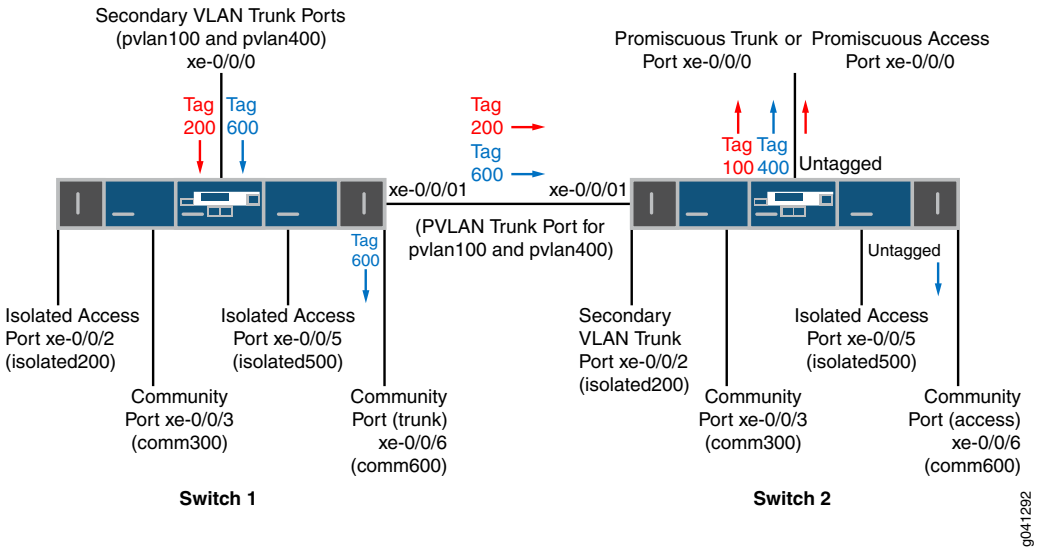


Table 94 on page 568 and Table 95 on page 568 list the settings for the example topology on both switches.

Table 100: Components of the Topology for Configuring a Secondary VLAN Trunk on Switch 1

Component	Description
pvlan100, ID 100	Primary VLAN
pvlan400, ID 400	Primary VLAN
comm300, ID 300	Community VLAN, member of pvlan100
comm600, ID 600	Community VLAN, member of pvlan400
isolation-vlan-id 200	VLAN ID for isolated VLAN, member of pvlan100
isolation–vlan-id 500	VLAN ID for isolated VLAN, member of pvlan400
xe-0/0/0.0	Secondary VLAN trunk port for primary VLANs pvlan100 and pvlan400
xe-0/0/1.0	PVLAN trunk port for primary VLANs pvlan100 and pvlan400
xe-0/0/2.0	Isolated access port for pvlan100
xe-0/0/3.0	Community access port for comm300
xe-0/0/5.0	Isolated access port for pvlan400
xe-0/0/6.0	Community trunk port for comm600

Table 101: Components of the Topology for Configuring a Secondary VLAN Trunk on Switch 2

Component	Description
pvlan100, ID 100	Primary VLAN
pvlan400, ID 400	Primary VLAN
comm300, ID 300	Community VLAN, member of pvlan100
comm600, ID 600	Community VLAN, member of pvlan400
isolation-vlan-id 200	VLAN ID for isolated VLAN, member of pvlan100
isolation–vlan-id 500	VLAN ID for isolated VLAN, member of pvlan400
xe-0/0/0.0	Promiscuous access port for primary VLANs pvlan100
xe-0/0/1.0	PVLAN trunk port for primary VLANs pvlan100 and pvlan400
xe-0/0/2.0	Secondary trunk port for isolated VLAN, member of pvlan100
xe-0/0/3.0	Community access port for comm300
xe-0/0/5.0	Isolated access port for pvlan400
xe-0/0/6.0	Community access port for comm600

Configuring the PVLANs on Switch 1

CLI Quick Configuration

To quickly create and configure the PVLANs on Switch 1, copy the following commands and paste them into a switch terminal window:

```
[edit]
```

set interfaces xe-0/0/0 unit 0 family ethernet-switching port-mode trunk

set interfaces xe-0/0/1 unit 0 family ethernet-switching port-mode trunk

set interfaces xe-0/0/1 unit 0 family ethernet-switching vlan members pvlan100

set interfacesxe-0/0/1 unit 0 family ethernet-switching vlan members pvlan400

```
set interfaces xe-0/0/2 unit 0 family ethernet-switching port-mode access
set interfaces xe-0/0/3 unit 0 family ethernet-switching port-mode access
set interfaces xe-0/0/5 unit 0 family ethernet-switching port-mode access
set interfaces xe-0/0/6 unit 0 family ethernet-switching port-mode trunk
set vlans pvlan100 vlan-id 100
set vlans pvlan400 vlan-id 400
set vlans pvlan100 pvlan
set vlans pvlan400 pvlan
set vlans pvlan100 interface xe-0/0/1.0 pvlan-trunk
set vlans pvlan400 interface xe-0/0/1.0 pvlan-trunk
set vlans comm300 vlan-id 300
set vlans comm300 primary-vlan pvlan100
set vlans comm300 interface xe-0/0/3.0
set vlans comm600 vlan-id 600
set vlans comm600 primary-vlan pvlan400
set vlans comm600 interface xe-0/0/6.0
set vlans pvlan100 pvlan isolation-vlan-id 200
set vlans pvlan400 pvlan isolation-vlan-id 500
set vlans pvlan100 interface xe-0/0/0.0 isolated
set vlans pvlan400 interface xe-0/0/0.0 isolated
set vlans comm600 interface xe-0/0/0.0
set vlans pvlan100 interface xe-0/0/2.0 isolated
set vlans pvlan400 interface xe-0/0/5.0 isolated
```

Step-by-Step Procedure

To configure the private VLANs and secondary VLAN trunk ports:

1. Configure the interfaces and port modes:

   [edit interfaces]

   ```
   user@switch# set xe-0/0/0 unit 0 family ethernet-switching port-mode trunk
   user@switch# set xe-0/0/1 unit 0 family ethernet-switching port-mode trunk
   user@switch# set xe-0/0/1 unit 0 family ethernet-switching vlan members pvlan100
   user@switch# set xe-0/0/1 unit 0 family ethernet-switching vlan members pvlan400
   user@switch# set xe-0/0/2 unit 0 family ethernet-switching port-mode access
   user@switch# set xe-0/0/3 unit 0 family ethernet-switching port-mode access
   user@switch# set xe-0/0/5 unit 0 family ethernet-switching port-mode access
   user@switch# set xe-0/0/6 unit 0 family ethernet-switching port-mode access
   ```

2. Create the primary VLANs:

   [edit vlans]

   ```
   user@switch# set pvlan100 vlan-id 100
   user@switch# set pvlan400 vlan-id 400
   ```

   NOTE: Primary VLANs must always be tagged VLANs, even if they exist on only one device.

3. Configure the primary VLANs to be private:

   [edit vlans]

   ```
   user@switch# set pvlan100 pvlan
   user@switch# set pvlan400 pvlan
   ```

4. Configure the PVLAN trunk port to carry the private VLAN traffic between the switches:

   [edit vlans]

   ```
   user@switch# set pvlan100 interface xe-0/0/1.0 pvlan-trunk
   user@switch# set pvlan400 interface xe-0/0/1.0 pvlan-trunk
   ```

5. Create secondary VLAN comm300 with VLAN ID 300:

   [edit vlans]

   ```
   user@switch# set comm300 vlan-id 300
   ```

6. Configure the primary VLAN for comm300:

[edit vlans]

```
user@switch# set comm300 primary-vlan pvlan100
```

7. Configure the interface for comm300:

   [edit vlans]

   ```
   user@switch# set comm300 interface xe-0/0/3.0
   ```

8. Create secondary VLAN comm600 with VLAN ID 600:

   [edit vlans]

   ```
   user@switch# set comm600 vlan-id 600
   ```

9. Configure the primary VLAN for comm600:

   [edit vlans]

   ```
   user@switch# set comm600 primary-vlan pvlan400
   ```

10. Configure the interface for comm600:

    [edit vlans]

    ```
    user@switch# set comm600 interface xe-0/0/6.0
    ```

11. Configure the interswitch isolated VLANs:

    [edit vlans]

    ```
    user@switch# set pvlan100 pvlan isolation-vlan-id 200
    user@switch# set pvlan400 pvlan isolation-vlan-id 500
    ```

    NOTE: When you configure a secondary VLAN trunk port to carry an isolated VLAN, you must also configure an **isolation-vlan-id**. This is true even if the isolated VLAN exists only on one switch.

12. Enable trunk port xe-0/0/0 to carry secondary VLANs for the primary VLANs:

    [edit vlans]

    ```
    user@switch# set pvlan100 interface xe-0/0/0.0 isolated
    user@switch# set pvlan400 interface xe-0/0/0.0 isolated
    ```

13. Configure trunk port xe-0/0/0 to carry comm600 (member of pvlan400):

[edit vlans]

```
user@switch# set comm600 interface xe-0/0/0.0
```

> **NOTE:** You do not need to explicitly configure xe-0/0/0 to carry the isolated VLAN traffic (tags 200 and 500) because all the isolated ports in pvlan100 and pvlan400–including xe-0/0/0.0–are automatically included in the isolated VLANs created when you configured **isolation-vlan-id 200** and **isolation-vlan-id 500**.

14. Configure xe-0/0/2 and xe-0/0/6 to be isolated:

[edit vlans]

```
user@switch# set pvlan100 interface xe-0/0/2.0 isolated
user@switch# set pvlan400 interface xe-0/0/5.0 isolated
```

Results

Check the results of the configuration on Switch 1:

```
[edit]
user@switch# show
interfaces {
    xe-0/0/0 {
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
                vlan {
                    members pvlan100;
                    members pvlan400;
                }
            }
        }
    }
    xe-0/0/1 {
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
                vlan {
                    members pvlan100;
                    members pvlan400;
                }
```

```
            }
        }
    }
    xe-0/0/2 {
        unit 0 {
            family ethernet-switching {
                port-mode access;
            }
        }
    }
    xe-0/0/3 {
        unit 0 {
            family ethernet-switching {
                port-mode access;
            }
        }
    }
    xe-0/0/5 {
        unit 0 {
            family ethernet-switching {
                port-mode access;
            }
        }
    }
    xe-0/0/6 {
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
            }
        }
    }
}
vlans {
    comm300 {
        vlan-id 300;
            interface {
            xe-0/0/3.0;
        }
        primary-vlan pvlan100;
    }
    comm600 {
        vlan-id 600;
            interface {
            xe-0/0/6.0;
```

```
        }
        primary-vlan pvlan400;
    }
    pvlan100 {
        vlan-id 100;
        interface {
            xe-0/0/0.0;
            xe-0/0/2.0;
            xe-0/0/3.0;
            xe-0/0/1.0 {
                pvlan-trunk;
            }
        }
        no-local-switching;
        isolation-id 200;
    }
    pvlan400 {
        vlan-id 400;
        interface {
            xe-0/0/0.0;
            xe-0/0/5.0;
            xe-0/0/6.0;
            xe-0/0/1.0 {
                pvlan-trunk;
            }
        }
        no-local-switching;
        isolation-id 500;
    }
}
```

Configuring the PVLANs on Switch 2

The configuration for Switch 2 is almost identical to the configuration for Switch 1. The most significant difference is that xe-0/0/0 on Switch 2 is configured as a promiscuous trunk port or a promiscuous access port, as Figure 27 on page 567 shows. In the following configuration, xe-0/0/0 is configured as a promiscuous access port for primary VLAN pvlan100.

If traffic ingresses on VLAN-enabled port and egresses on a promiscuous access port, the VLAN tags are dropped on egress and the traffic is untagged at that point. For example, traffic for comm600 ingresses on the secondary VLAN trunk port configured on xe-0/0/0.0 on Switch 1 and carries tag 600 as it is forwarded through the secondary VLAN. When it egresses from xe-0/0/0.0 on Switch 2, it will be untagged

if you configure xe-0/0/0.0 as a promiscuous access port as shown in this example. If you instead configure xe-0/0/0.0 as a promiscuous trunk port (port-mode trunk), the traffic for comm600 carries its primary VLAN tag (400) when it egresses.

CLI Quick Configuration

To quickly create and configure the PVLANs on Switch 2, copy the following commands and paste them into a switch terminal window:

```
[edit]
set interfaces xe-0/0/0 unit 0 family ethernet-switching port-mode access
set interfaces xe-0/0/1 unit 0 family ethernet-switching port-mode trunk
set interfaces xe-0/0/1 unit 0 family ethernet-switching vlan members pvlan100
set interfaces xe-0/0/1 unit 0 family ethernet-switching vlan members pvlan400
set interfaces xe-0/0/2 unit 0 family ethernet-switching port-mode trunk
set interfaces xe-0/0/3 unit 0 family ethernet-switching port-mode access
set interfaces xe-0/0/5 unit 0 family ethernet-switching port-mode access
set interfaces xe-0/0/6 unit 0 family ethernet-switching port-mode access
set vlans pvlan100 vlan-id 100
set vlans pvlan400 vlan-id 400
set vlans pvlan100 pvlan
set vlans pvlan400 pvlan
set vlans pvlan100 interface xe-0/0/1.0 pvlan-trunk
set vlans pvlan400 interface xe-0/0/1.0 pvlan-trunk
set vlans comm300 vlan-id 300
set vlans comm300 primary-vlan pvlan100
set vlans comm300 interface xe-0/0/3.0
set vlans comm600 vlan-id 600
set vlans comm600 primary-vlan pvlan400
set vlans comm600 interface xe-0/0/6.0
```

set vlans pvlan100 pvlan isolation-vlan-id 200

set vlans pvlan400 pvlan isolation-vlan-id 500

set vlans pvlan100 interface xe-0/0/0.0 promiscuous

set vlans pvlan100 interface xe-0/0/2.0 isolated

set vlans pvlan400 interface xe-0/0/5.0 isolated

Step-by-Step Procedure

To configure the private VLANs and secondary VLAN trunk ports:

1. Configure the interfaces and port modes:

   [edit interfaces]

   ```
   user@switch# set xe-0/0/0 unit 0 family ethernet-switching port-mode access
   user@switch# set xe-0/0/1 unit 0 family ethernet-switching port-mode trunk
   user@switch# set xe-0/0/1 unit 0 family ethernet-switching vlan members pvlan100
   user@switch# set xe-0/0/1 unit 0 family ethernet-switching vlan members pvlan400
   user@switch# set xe-0/0/2 unit 0 family ethernet-switching port-mode trunk
   user@switch# set xe-0/0/3 unit 0 family ethernet-switching port-mode access
   user@switch# set xe-0/0/5 unit 0 family ethernet-switching port-mode access
   user@switch# set xe-0/0/6 unit 0 family ethernet-switching port-mode access
   ```

2. Create the primary VLANs:

   [edit vlans]

   ```
   user@switch# set pvlan100 vlan-id 100
   user@switch# set pvlan400 vlan-id 400
   ```

3. Configure the primary VLANs to be private:

   [edit vlans]

   ```
   user@switch# set pvlan100 pvlan
   user@switch# set pvlan400 pvlan
   ```

4. Configure the PVLAN trunk port to carry the private VLAN traffic between the switches:

   [edit vlans]

   ```
   user@switch# set pvlan100 interface xe-0/0/1.0 pvlan-trunk
   user@switch# set pvlan400 interface xe-0/0/1.0 pvlan-trunk
   ```

5. Create secondary VLAN comm300 with VLAN ID 300:

   [edit vlans]
   `user@switch#` **set comm300 vlan-id 300**

6. Configure the primary VLAN for comm300:

   [edit vlans]
   `user@switch#` **set comm300 primary-vlan pvlan100**

7. Configure the interface for comm300:

   [edit vlans]
   `user@switch#` **set comm300 interface xe-0/0/3.0**

8. Create secondary VLAN comm600 with VLAN ID 600:

   [edit vlans]
   `user@switch#` **set comm600 vlan-id 600**

9. Configure the primary VLAN for comm600:

   [edit vlans]
   `user@switch#` **set comm600 primary-vlan pvlan400**

10. Configure the interface for comm600:

    [edit vlans]
    `user@switch#` **set comm600 interface xe-0/0/6.0**

11. Configure the interswitch isolated VLANs:

    [edit vlans]
    `user@switch#` **set pvlan100 pvlan isolation-vlan-id 200**
    `user@switch#` **set pvlan400 pvlan isolation-vlan-id 500**

12. Configure access port xe-0/0/0 to be promiscuous for pvlan100:

    [edit vlans]
    `user@switch#` **set pvlan100 interface xe-0/0/0.0 promiscuous**

NOTE: A promiscuous access port can be a member of only one primary VLAN.

13. Configure xe-0/0/2 and xe-0/0/6 to be isolated:

```
[edit vlans]
user@switch# set pvlan100 interface xe-0/0/2.0 isolated
user@switch# set pvlan400 interface xe-0/0/5.0 isolated
```

Results

Check the results of the configuration on Switch 2:

```
[edit]
user@switch# show
interfaces {
    xe-0/0/0 {
        unit 0 {
            family ethernet-switching {
                port-mode access;
                vlan {
                    members pvlan100;
                }
            }
        }
    }
        xe-0/0/1 {
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
                vlan {
                    members pvlan100;
                    members pvlan400;
                }
            }
        }
    }
    xe-0/0/2 {
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
```

```
            }
        }
    }
    xe-0/0/3 {
        unit 0 {
            family ethernet-switching {
                port-mode access;
            }
        }
    }
    xe-0/0/5 {
        unit 0 {
            family ethernet-switching {
                port-mode access;
            }
        }
    }
    xe-0/0/6 {
        unit 0 {
            family ethernet-switching {
                port-mode access;
            }
        }
    }
vlans {
    comm300 {
        vlan-id 300;
            interface {
            xe-0/0/3.0;
        }
        primary-vlan pvlan100;
    }
    comm600 {
        vlan-id 600;
            interface {
            xe-0/0/6.0;
        }
        primary-vlan pvlan400;
    }
    pvlan100 {
        vlan-id 100;
        interface {
            xe-0/0/0.0;
            xe-0/0/2.0;
```

```
            xe-0/0/3.0;
            xe-0/0/1.0 {
                pvlan-trunk;
            }
        }
        no-local-switching;
        isolation-id 200;
    }
    pvlan400 {
        vlan-id 400;
        interface {
            xe-0/0/5.0;
            xe-0/0/6.0;
            xe-0/0/1.0 {
                pvlan-trunk;
            }
        }
        no-local-switching;
        isolation-id 500;
    }
}
```

Verification

IN THIS SECTION

To confirm that the configuration is working properly, perform these tasks:

Verifying That the Private VLAN and Secondary VLANs Were Created

Purpose

Verify that the primary VLAN and secondary VLANs were properly created on Switch 1.

Action

Use the **show vlans** command:

```
user@switch>  show vlans private-vlan
```

```
Name               Role        Tag      Interfaces
pvlan100           Primary     100      xe-0/0/0.0, xe-0/0/1.0, xe-0/0/2.0,
xe-0/0/3.0
__iso_pvlan100__   Isolated    200      xe-0/0/2.0
comm300            Community   300      xe-0/0/3.0

pvlan400           Primary     400      xe-0/0/0.0, xe-0/0/1.0, xe-0/0/5.0,
xe-0/0/6.0
__iso_pvlan400__   Isolated    500      xe-0/0/5.0
comm600            Community   600     xe-0/0/6.0
```

Meaning

The output shows that the private VLANs were created and identifies the interfaces and secondary VLANs associated with them.

Verifying The Ethernet Switching Table Entries

Purpose

Verify that the Ethernet switching table entries were created for primary VLAN pvlan100.

Action

Show the Ethernet switching table entries for pvlan100.

```
user@switch> show ethernet-switching table vlan pvlan100 private-vlan
```

```
Ethernet-switching table: 0 unicast entries
  pvlan100             *                 Flood          - All-members
  pvlan100             00:10:94:00:00:02 Learn            xe-0/0/2.0
  __iso_pvlan100__     *                 Flood          - All-members
  __iso_pvlan100__     00:10:94:00:00:02 Replicated      - xe-0/0/2.0
```

RELATED DOCUMENTATION

Understanding Egress Firewall Filters with PVLANs

IRB Interfaces in Private VLANs on MX Series Routers

You can configure integrated routing and bridging (IRB) interfaces in a private VLAN (PVLAN) on a single MX router to span multiple MX routers. PVLANs limit the communication within a VLAN by restricting traffic flows through their member switch ports (which are called “private ports”) so that these ports communicate only with a specified uplink trunk port or with specified ports within the same VLAN. IRB provides simultaneous support for Layer 2 bridging and Layer 3 routing on the same interface. IRB enables you to route packets to another routed interface or to another bridge domain that has an IRB interface configured. You configure a logical routing interface by including the **irb** statement at the **[edit interfaces]** hierarchy level and include that interface in the bridge domain.

PVLANs are supported on MX80 routers, on MX240, MX480, and MX960 routers with DPCs in LAN mode, and on MX Series routers with MPC1, MPC2, and Adaptive Services PICs. This functionality is supported only on MX240, MX480, and MX960 routers that function in enhanced LAN mode (by entering the **network-services lan** statement at the **[edit chassis]** hierarchy level).

IRB in PVLANs replaces the external router used for routing across VLANs. The routing operations in the absence of IRB occur through external router connected to promiscuous port. This behavior takes care of all the routed frames for all the ports defined under the PVLAN domain. In this case, no layer 3 exchange occurs on MX Series routers in enhanced LAN mode for this PVLAN bridge domain. In the case of IRB, the Layer 3 interface is associated with the primary VLAN that is configured and is considered to be a single Layer 3 interface for the entire PVLAN domain. The ingress routed traffic from all ports in the PVLAN domain needs to be mapped to this IRB interface. The egress of the IRB interface take places under the PVLAN. For a PVLAN domain spanning multiple switches, only one IRB interface can be configured in one switch. This IRB interface represent whole PVLAN domain to interact with the Layer 3 domains. An IRB interface only associates with the primary bridge domain and all Layer 3 forwarding occurs only in the primary bridge domain. When a Layer 3 packet is received in an isolated port or a promiscuous port, the device first locates the secondary bridge domain, based on secondary bridge domain to find primary bridge domain identifier. If the destination MAC address is the local IRB MAC address, the microcode transmits the packet to IRB interface associated with primary bridge domain for further processing. The same procedure occurs for receiver Layer 3 packets in an interswitch link (ISL) port with the isolated or community VLAN tag.

For the ingress Layer 3 packet with Layer 3 forwarding logic sent to IRB interfaces associated with a PVLAN bridge domain, the device processes and determines the ARP entry to send packet to the related interface that might be an isolated port or a community port. The microcode appends or translates the packet VLAN ID to the isolation or community vlan ID based on the port type. The VLAN ID is removed if the related port is untagged. A special operational case exists for Layer 3 packets that are forwarded to remote isolated or community port through the ISL link. The Layer 3 packet might contain the primary bridge domain VLAN ID and the remote node performs the translation or pop operation when it sends the packet out on the related port. This method of processing is different from Layer 2 domains. Because all forwarding base on

ARP must be unicast traffic and in the remote node, the port that must be used to forward is known and the transmission of PVLAN ID occurs properly.

An ARP entry carries only the primary bridge domain information. When an ARP response is received from an isolated port or a promiscuous port, the system identifies the secondary bridge domain, and based on the secondary bridge domain, it attempts to retrieve the primary bridge domain identifier. ARP packets eventually reach the IRB interface associated with the primary bridge domain. The kernel considers this ARP packet as a normal bridge domain and creates and maintains the ARP entry only for the primary bridge domain. The same procedure is adopted for ARP request packets that are destined for the local IRB MAC address. The response is transmitted through the IRB interface and appropriate VLAN translation or a pop operation is performed, depending on the received interface.

Guidelines for Configuring IRB Interfaces in PVLANs on MX Series Routers

Keep the following points in mind when you configure IRB interfaces for PVLANs:

- All of the IP applications such as IP multicast, IPv4, IPv6, and VRRP that are compatible with IRB in normal bridge domains function properly when IRB for PVLAN bridge domains is configured.
- MC-LAG interfaces are not supported. All ports that are associated with PVLAN bridge domains cannot be mc-ae interfaces.
- IGMP snooping is not supported.
- A virtual switch instance that contains a bridge domain associated with logical interfaces is supported.
- Q-in-Q tunneling is not supported.
- Logical systems are not supported.
- Virtual private LAN service (VPLS) and Etherent VPN (EVPN) in virtual switch routing instances are not supported. A validation is performed if you attempt to configure Layer 3 interfaces in a secondary VLAN.
- MX Series Virtual Chassis configuration is not supported.

Forwarding of Packets Using IRB Interfaces in PVLANs

This topic describes how PVLAN packet forwarding operates with IRB interfaces on MX Series routers in enhanced LAN mode. The IRB interface operates as a Layer 3 gateway for all members of a bridging domain. All the members of bridging domain are assumed to be in the same subnet as the subnet of the IRB interface, which works as a gateway.

Consider a sample deployment scenario in which two routers, Router1 and Router2, are configured with a PVLAN. On Router1, the promiscuous port is P1, interswitch link is L1, isolated port is I1, and two community ports are C11 and C21. Similarly, on Router2, the promiscuous port is P2, interswitch link is L2, isolated port is I2, and two community ports are C12 and C22. In the example configuration, the two routers are interconnected through an ISL link, L1 with L2. A PVLAN domain is defined across these two routers encompassing a subdomain of isolated ports (I1, I2), and Community1 ports (C11, C12), and Community2 ports (C21, C22). Because all the ports are in the same subnet, without IRB, switching capability works across ports, across routers following the PVLAN rules. When the end-host needs to reach out cross the subnet, you must configure IRB on the bridging domain. From an end-host perspective, to reach out across the bridging domain, it needs to be configured with the IRB IP address as the default gateway address. All Layer 3 connectivity is established by processing ARP request and ARP responses. The following sections describe the different scenarios encountered for Layer 3 traffic support in PVLANs.

Incoming ARP Requests on PVLAN Ports

ARP requests enter a PVLAN port as broadcast packets. All packets that enter in the ingress direction of a PVLAN domain contain their bridge domain ID translated into the primary VLAN bridge domain ID. In this case, the bridge domain ID contained in the ARP packet is also translated be to the bridge domain ID of the primary VLAN. When IRB is configured in a bridging domain, the IRB MAC address is added to the MAC table as an eligible destination MAC address on the primary VLAN bridge domain ID. The ARP request is flooded to all ports of the secondary bridging domain in which it was received and, in addition, a copy is sent to the IRB logical interface.

When an IRB logical interface receives this packet, it sends the packet to the host as an ARP packet with the primary BD and the Layer 2 logical interface on which it is received. The PVLAN domain learns the source MAC address of the ARP packet and the kernel learns the sender IP of the ARP packet, and triggers a next-hop installation. If the ARP request is destined for IRB IP address, then an ARP response is sent. If proxy ARP is enabled on IRB, IRB responds with an ARP reply if the destination IP address is known.

The preceding configuration case describes a scenario the ARP request came on Local PVLAN port. If the ARP request is received on a remote PVLAN port, then it is flooded on all the ports of the remote PVLAN domain. Because IRB is configured only on one router of the PVLAN domain, on the remote PVLAN, the flooding is on all the ports. As part of the flooding in the remote PVLAN domain, a copy of the packet is

sent to the ISL port. The ISL port processes this packet as though it was received on the local isolated port or community port and the aforementioned method of processing takes place

Outgoing ARP Responses on PVLAN Ports

When a ARP request is received in the kernel, both the bridge domain ID and the receiving Layer 2 logical interface are transmitted. A next-hop installation is triggered to create a next-hop to the Layer 2 logical interface for the sender IP address with the IRB MAC Address as the destination MAC address and the sender MAC address as the source MAC address, with both these addresses appearing as Layer 2 rewrite during the next-hop. If the ARP request queries for the IRB IP address, then an ARP response is sent to the receiving Layer 2 logical interface. If the ARP request queries for an IP address other than the IRB IP address, it is processed as though proxy ARP is enabled on IRB or it is discarded. Because all ARP requests are processed as being received on the primary VLAN, the response is also sent with the primary VLAN. However, when it reaches the receiving Layer 2 logical interface, the appropriate VLAN translation takes place.

The preceding scenario describes an ARP response being sent on a local PVLAN port. If the ARP request is received from a remote PVLAN domain, the receiving Layer 2 logical interface is the ISL port. In this case, the ARP response is sent to the ISL port, on the remote PVLAN domain, the ARP response received on the ISL port is forwarded to the same port where the ARP request is received. This behavior is possible because the source MAC address of the ARP request is learned on the shared VLAN.

Outgoing ARP Requests on PVLAN Ports

When IRB has to advertise a ARP request, it uses the kernel flood next-hop for the primary VLAN and floods to all the ports in the local PVLAN domain. The receiving ISL port also floods the packet to the remote PVLAN domain. Although the ARP request is constructed with the primary VLAN, in the egress direction, appropriate VLAN translation or VLAN pop is performed using the specific port.

Incoming ARP Responses on PVLAN Ports

ARP responses are unicast packets with the destination MAC address as the IRB MAC Address. When such a packet is received on the local PVLAN domain where IRB is enabled, it is forwarded to the IRB logical interface. When the packet arrives at the IRB logical interface, it is propagated to the host. The kernel triggers a next-hop installation with the appropriate Layer 2 rewrite. This operation works properly for ARP responses received on the local PVLAN port. If the ARP response is received on a remote PVLAN port, it is forwarded similar to a normal Layer 2 packet because IRB is not enabled in such a scenario. When

the ARP request is sent out from the local PVLAN domain, the receiving ISL port in the remote PVLAN domain might have learned the IRB MAC address on that port, and this address is used to forward the packet to the IRB logical interface.

Receipt of Layer 3 Packets on PVLAN Ports

The packet is received with the IRB MAC address as the destination MAC address and it is processed through the IRB logical interface. The packet is forwarded in the same manner as a regular IP packet.

Configuring IRB Interfaces in PVLAN Bridge Domains on MX Series Routers in Enhanced LAN Mode

You can configure integrated routing and bridging (IRB) interfaces in a private VLAN (PVLAN) on a single MX router to span multiple MX routers. PVLANs limit the communication within a VLAN by restricting traffic flows through their member switch ports (which are called “private ports”) so that these ports communicate only with a specified uplink trunk port or with specified ports within the same VLAN. IRB provides simultaneous support for Layer 2 bridging and Layer 3 routing on the same interface. IRB enables you to route packets to another routed interface or to another bridge domain that has an IRB interface configured. You configure a logical routing interface and include that interface in the virtual switch instance that contains the bridge domain. You can specify the secondary VLANs as isolated or community VLANs in the bridge domain.

Before you begin configuring a PVLAN, make sure you have:

- Created and configured the necessary VLANs. See “Configuring VLAN and Extended VLAN Encapsulation” on page 315 and “Enabling VLAN Tagging” on page 298.
- Configured MX240, MX480, and MX960 routers to function in enhanced LAN mode by entering the **network-services lan** statement at the **[edit chassis]** hierarchy level.

  You must reboot the router when you configure or delete the enhanced LAN mode on the router. Configuring the **network-services lan** option implies that the system is running in the enhanced IP mode. When you configure a device to function in MX-LAN mode, only the supported configuration statements and operational show commands that are available for enabling or viewing in this mode are displayed in the CLI interface.

  If your system contains parameters that are not supported in MX-LAN mode in a configuration file, you cannot commit those unsupported attributes. You must remove the settings that are not supported and then commit the configuration. After the successful CLI commit, a system reboot is required for the attributes to become effective. Similarly, if you remove the **network-services lan** statement, the system

does not run in MX-LAN mode. Therefore, all of the settings that are supported outside of the MX-LAN mode are displayed and are available for definition in the CLI interface. If your configuration file contains settings that are supported only in MX-LAN mode, you must remove those attributes before you commit the configuration. After the successful CLI commit, a system reboot is required for the CLI parameters to take effect. The Layer 2 Next-Generation CLI configuration settings are supported in MX-LAN mode. As a result, the typical format of CLI configurations might differ in MX-LAN mode.

To configure an IRB interface in a PVLAN bridge domain associated with a virtual switch instance:

1. Create a promiscuous port for the PVLAN.

```
[edit interfaces]
user@host# set interface interface-name unit logical-unit-number family bridge interface-mode trunk
user@host# set interface interface-name unit logical-unit-number family bridge vlan-id vlan-id
```

2. Create the interswitch link (ISL) trunk port for the PVLAN.

```
[edit interfaces]
user@host# set interface interface-name unit logical-unit-number family bridge interface-mode trunk
   inter-switch-link
user@host# set interface interface-name unit logical-unit-number family bridge vlan-id vlan-id
```

3. Create the isolated port for the PVLAN. The port is identified as an isolated port or a community port, based on the VLAN ID or the list of VLAN IDs to which the interface corresponds. For example, if you configure a port with a VLAN ID of 50, and if you specify a VLAN ID of 50 as the isolated VLAN or tag in the bridge domain, the port is considered as an isolation port.

```
[edit interfaces]
user@host# set interface interface-name unit logical-unit-number family bridge interface-mode access
user@host# set interface interface-name unit logical-unit-number family bridge vlan-id vlan-id
```

4. Create the community port for the PVLAN. The port is identified as an isolated port or a community port, based on the VLAN ID or the list of VLAN IDs to which the interface corresponds. For example, if you configure a port with a VLAN ID of 50, and if you specify a VLAN ID of 50 as the community VLAN or tag in the bridge domain, the port is considered as a community port.

```
[edit interfaces]
user@host# set interface interface-name unit logical-unit-number family bridge interface-mode access
user@host# set interface interface-name unit logical-unit-number family bridge vlan-id vlan-id
```

5. Create a virtual switch instance with a bridge domain and associate the logical interfaces.

```
[edit routing-instances]
user@host# set routing-instance-name instance-type virtual-switch
user@host# set routing-instance-name interface interface-name unit logical-unit-number
user@host# set routing-instance-name bridge-domains bridge-domain-name
```

6. Create an IRB interface and specify the IRB interface in the bridge domain associated with the virtual switch instance. IRB provides simultaneous support for Layer 2 bridging and Layer 3 IP routing on the same interface. IRB enables you to route local packets to another routed interface or to another bridge domain that has a Layer 3 protocol configured.

```
[edit]
user@host# set interfaces irb unit logical-unit-number family family-name address ip-address
[edit routing-instances instance-name bridge-domains bridge-domain-name]
user@host# set routing-interface irb unit logical-unit-number
```

7. Specify the primary, isolated, and community VLAN IDs, and associate the VLANs with the bridge domain.

```
[edit routing-instances instance-name bridge-domains bridge-domain-name]
user@host# set vlan-id vlan-id
user@host# set isolated-vlan vlan-id
user@host# set community-vlans [ number number-number ]
```

RELATED DOCUMENTATION

Example: Configuring an IRB Interface in a Private VLAN on a Single MX Series Router

IN THIS SECTION

For security reasons, it is often useful to restrict the flow of broadcast and unknown unicast traffic and to even limit the communication between known hosts. The private VLAN (PVLAN) feature on MX Series routers allows an administrator to split a broadcast domain into multiple isolated broadcast subdomains, essentially putting a VLAN inside a VLAN.

This example describes how to create an integrated routing and bridging (IRB) interface in a PVLAN bridge domain associated with a virtual switch instance on a single MX Series router:

> **NOTE:** Configuring a voice over IP (VoIP) VLAN on PVLAN interfaces is not supported.

Requirements

This example uses the following hardware and software components:

- One MX Series router in enhanced LAN mode.
- Junos OS Release 15.1 or later for MX Series routers

Before you begin configuring a PVLAN, make sure you have:

- Created and configured the necessary VLANs. See "Configuring VLAN and Extended VLAN Encapsulation" on page 315 and "Enabling VLAN Tagging" on page 298.
- Configured MX240, MX480, and MX960 routers to function in enhanced LAN mode by entering the **network-services lan** statement at the **[edit chassis]** hierarchy level.

Overview and Topology

In a large office with multiple buildings and VLANs, you might need to isolate some workgroups or other endpoints for security reasons or to partition the broadcast domain. This configuration example shows a simple topology to illustrate how to create a PVLAN with one primary VLAN and four community VLANs, as well as two isolated ports.

Assume a sample deployment in which a primary VLAN named VP contains ports, p1, p2, t1, t2, i1, i2, cx1, and cx2. The port types of these configured ports are as follows:

- Promiscuous ports = p1, p2
- ISL ports = t1, t2
- Isolated ports = i1, i2
- Community VLAN = Cx
- Community ports = cx1, cx2

An IRB interface, irb.0, is configured and mapped to the bridge domain in the virtual switch instance.

Bridge domains are provisioned for each of the VLANs, namely, Vp, Vi, and Vcx. Assume the bridge domains to be configured as follows:

Vp—BD_primary_Vp (ports contained are p1, t1, i1, i2, cx1, cx2)

Vi—BD_isolate_Vi (ports contained are p1, t1, *i1, *i2)

Vcx—BD_community_Vcx (ports contained are p1, t1, cx1, cx2)

The bridge domains for community, primary, and isolated VLANs are automatically created by the system internally when you configure a bridge domain with a trunk interface, access interface, or interswitch link. The bridge domains contain the same VLAN ID corresponding to the VLANs. To use bridge domains for PVLANs, you must configure the following additional attributes:

Configuration

To configure an IRB interface in a PVLAN, perform these tasks:

CLI Quick Configuration

To quickly create and configure a PVLAN and include an IRB interface in a PVLAN bridge domain associated with a virtual switch instance, copy the following commands and paste them into the router terminal window:

Configuring an IRB Interface

```
set interfaces irb unit 0 family inet address 22.22.22.1/24
```

Configuring Promiscuous, ISL, Isolated, and Community Ports

```
set interfaces ge-0/0/9 unit 0 family bridge interface-mode trunk
set interfaces ge-0/0/9 unit 0 family bridge vlan-id 100
set interfaces ge-0/0/13 unit 0 family bridge interface-mode trunk
set interfaces ge-0/0/13 unit 0 family bridge vlan-id 100
set interfaces ge-0/0/10 unit 0 family bridge interface-mode access
set interfaces ge-0/0/10 unit 0 family bridge vlan-id 10
set interfaces ge-0/0/12 unit 0 family bridge interface-mode access
set interfaces ge-0/0/12 unit 0 family bridge vlan-id 10
set interfaces ge-0/0/1 unit 0 family bridge interface-mode access
set interfaces ge-0/0/1 unit 0 family bridge vlan-id 50
set interfaces ge-0/0/2 unit 0 family bridge interface-mode access
set interfaces ge-0/0/2 unit 0 family bridge vlan-id 50
set interfaces ge-0/0/3 unit 0 family bridge interface-mode access
set interfaces ge-0/0/3 unit 0 family bridge vlan-id 60
set interfaces ge-0/0/4 unit 0 family bridge interface-mode access
set interfaces ge-0/0/4 unit 0 family bridge vlan-id 60
```

Configuring a Virtual Switch Instance With Bridge Domain Interfaces

```
set routing-instances vs-1 instance-type virtual-switch
set routing-instances vs-1 interface ge-0/0/1.0
set routing-instances vs-1 interface ge-0/0/2.0
set routing-instances vs-1 interface ge-0/0/3.0
set routing-instances vs-1 interface ge-0/0/4.0
set routing-instances vs-1 interface ge-0/0/9.0
set routing-instances vs-1 interface ge-0/0/10.0
set routing-instances vs-1 interface ge-0/0/12.0
set routing-instances vs-1 interface ge-0/0/13.0
set routing-instances vs-1 bridge-domains bd1
```

Specify the IRB Interface and Primary, Isolated, and Community VLAN IDs in the Bridge Domain

```
set routing-instances vs1 bridge-domains bd1 vlan-id 100
set routing-instances vs1 bridge-domains bd1 isolated-vlan 10
set routing-instances vs1 bridge-domains bd1 community-vlans [50 60]
set routing-instances vs1 bridge-domains bd1 routing-interface irb.0
```

Step-by-Step Procedure

To configure the interswitch link (ISL) for a PVLAN, the PVLAN port types, and secondary VLANs for the PVLAN:

1. Create an IRB interface.

```
[edit interfaces]
user@host# set interfaces irb unit 0 family inet address 22.22.22.1/24
```

2. Create a promiscuous port for the PVLAN.

```
[edit interfaces]
user@host# set ge-0/0/9 unit 0 family bridge interface-mode trunk
user@host# set ge-0/0/9 unit 0 family bridge vlan-id 100
```

3. Create the interswitch link (ISL) trunk port for the PVLAN.

```
[edit interfaces]
user@host# set ge-0/0/13 unit 0 family bridge interface-mode trunk inter-switch-link
user@host# set ge-0/0/13 unit 0 family bridge vlan-id 100
```

4. Create the isolated ports for the PVLAN.

```
[edit interfaces]
user@host# set ge-0/0/10 unit 0 family bridge interface-mode access
user@host# set ge-0/0/10 unit 0 family bridge vlan-id 10
user@host# set ge-0/0/12 unit 0 family bridge interface-mode access
user@host# set ge-0/0/12 unit 0 family bridge vlan-id 10
```

5. Create the community ports for the PVLAN.

```
[edit interfaces]
user@host# set ge-0/0/1 unit 0 family bridge interface-mode access
user@host# set ge-0/0/1 unit 0 family bridge vlan-id 50
user@host# set ge-0/0/2 unit 0 family bridge interface-mode access
user@host# set ge-0/0/2 unit 0 family bridge vlan-id 50
user@host# set ge-0/0/3 unit 0 family bridge interface-mode access
user@host# set ge-0/0/3 unit 0 family bridge vlan-id 60
user@host# set ge-0/0/4 unit 0 family bridge interface-mode access
user@host# set ge-0/0/4 unit 0 family bridge vlan-id 60
```

6. Create a virtual switch instance with a bridge domain and associate the logical interfaces.

```
[edit routing-instances]
user@host# set vs-1 instance-type virtual-switch
user@host# set vs-1 interface ge-0/0/1.0
user@host# set vs-1 interface ge-0/0/2.0
user@host# set vs-1 interface ge-0/0/3.0
user@host# set vs-1 interface ge-0/0/4.0
user@host# set vs-1 interface ge-0/0/9.0
user@host# set vs-1 interface ge-0/0/10.0
user@host# set vs-1 interface ge-0/0/12.0
user@host# set vs-1 interface ge-0/0/13.0
user@host# set vs-1 bridge-domains bd1
```

7. Specify the IRB interface, primary, isolated, and community VLAN IDs, and associate the VLANs with the bridge domain.

```
[edit routing-instances vs1 bridge-domains bd1]
user@host# set vlan-id 100
user@host# set isolated-vlan 10
user@host# set community-vlans [50 60]
user@host# set routing-interface irb.0
```

Results

Check the results of the configuration:

```
[edit]
[interfaces]
     ge-0/0/9 {
         unit 0 {
             family bridge {
  interface-mode trunk;
      vlan-id 100;             Promiscuous port by vlan id
             }
         }
     }

     ge-0/0/13 {
         unit 0 {
             family bridge {
  interface-mode trunk inter-switch-link;   ISL trunk
```

```
vlan-id 100;
           }
       }
   }


   ge-0/0/10 {
       unit 0 {
           family bridge {
                 interface-mode access;
                 vlan-id 10;              isolated port by vlan ID
            }
       }
   }

   ge-0/0/12 {
       unit 0 {
           family bridge {
                 interface-mode access;
                 vlan-id 10;              isolated port by vlan ID
            }
       }
   }
   ge-0/0/1 {
       unit 0 {
           family bridge {
                 interface-mode access;
                 vlan-id 50;              community port by vlan ID
            }
       }
   }

   ge-0/0/2 {
       unit 0 {
           family bridge {
                 interface-mode access;
                 vlan-id 50;              community port by vlan ID
            }

       }
   }

   ge-0/0/3 {
       unit 0 {
```

```
            family bridge {
                  interface-mode access;
                  vlan-id 60;              community port by vlan ID
                }
          }
     }

     ge-0/0/4 {
         unit 0 {
             family bridge {
                  interface-mode access;
                  vlan-id 60;              community port by vlan ID
                }
          }
     }

     irb {
        unit 0 {
           family inet {
             address 22.22.22.1/24;
           }
     }
}
```

```
[edit]
routing-instances {
    vs-1 {
        instance-type virtual-switch;
        interface ge-0/0/1.0;
        interface ge-0/0/2.0;
        interface ge-0/0/3.0;
        interface ge-0/0/4.0;
        interface ge-0/0/9.0;
        interface ge-0/0/10.0;
        interface ge-0/0/12.0;
        interface ge-0/0/13.0;

 bridge-domains {
     bd1 {
             vlan-id 100;               /* primary vlan */
                  isolated-vlan 10;
                  community-vlans [50 60]
                  routing-interface irb.0 /* IRB interface */
```

```
      }
}
}
```

Verification

IN THIS SECTION

- Verifying That the Private VLAN and Secondary VLANs Were Created | **651**

To confirm that the configuration is working properly, perform these tasks:

Verifying That the Private VLAN and Secondary VLANs Were Created

Purpose

Verify that the primary VLAN and secondary VLANs were properly created on the switch.

Action

Use the **show bridge domain** command:

`user@host>` **show bridge domain**

```
Routing instance          Bridge domain            VLAN ID     Interfaces
default-switch                  bd1-primary-100          100
                                                              ge-0/0/9.0
                                        ge-0/0/10.0
                                                             ge-0/0/12.0
                                                             ge-0/0/13.0
                                                              ge-0/0/1.0
                                                              ge-0/0/2.0
                                                              ge-0/0/3.0

                                                              ge-0/0/4.0
default-switch                bd1-isolation-10          10
                                                                ge-0/0/9.0
                                                                 ge-0/0/10.0
```

```
                                                          ge-0/0/12.0
                                                          ge-0/0/13.0

default-switch                  bd1-comunity-50     50
                                                            ge-0/0/9.0
                                                             ge-0/0/13.0
                                                             ge-0/0/1.0
                                                            ge-0/0/2.0

default-switch                   bd1-comunity-60     60
                                                            ge-0/0/9.0
                                                             ge-0/0/13.0
                                                         ge-0/0/3.0
                                                          ge-0/0/4.0
```

Meaning

The output shows that the primary VLAN was created and identifies the interfaces and secondary VLANs associated with it.

RELATED DOCUMENTATION

CHAPTER 18

Configuring Layer 2 Bridging Interfaces

Layer 2 Bridging Interfaces Overview

Bridging operates at Layer 2 of the OSI reference model while routing operates at Layer 3. A set of logical ports configured for bridging can be said to constitute a bridging domain.

A bridging domain can be created by configuring a routing instance and specifying the instance-type as **bridge**.

Integrated routing and bridging (IRB) is the ability to:

- Route a packet if the destination MAC address is the MAC address of the router and the packet **ethertype** is IPv4, IPv6, or MPLS.
- Switch all multicast and broadcast packets within a bridging domain at layer 2.
- Route a copy of the packet if the destination MAC address is a multicast address and the **ethertype** is IPv4 or IPv6.
- Switch all other unicast packets at Layer 2.
- Handle supported Layer 2 control packets such as STP and LACP.
- Handle supported Layer 3 control packets such as OSPF and RIP.

RELATED DOCUMENTATION

Configuring Layer 2 Bridging Interfaces | **655**

Ethernet Interfaces User Guide for Routing Devices

Configuring Layer 2 Bridging Interfaces

Integrated routing and bridging interfaces are logical Layer 3 VLAN interfaces that route traffic between bridge domains (VLANs). So, an IRB logical interface is usually associated with a bridge domain or VLAN. The IRB logical interface also functions as the gateway IP address for the other devices on the same sub-network that are associated with the same VLAN. IRB interfaces support Layer 2 bridging and Layer 3 routing on the same interface. As a result, IRB interfaces enable the router to act both as a router and as a Layer 2 switch at the same time.

> **NOTE:** If the status of all Layer 2 logical interfaces in the bridge domain is down, the status of the **irb** logical interface is also down.

To configure an IRB logical interface:

1. In configuration mode, at the [**edit bridge-domains**] hierarchy level, configure the bridge domain by specifying the name of the bridge and the VLAN ID.

   ```
   [edit bridge-domains]
   user@host# set bridge-domain-name vlan-id vlan-id
   ```

2. Configure an interface in trunk mode and include the interface in the appropriate bridge domain using the **vlan-id-list** command at the [**edit interfaces**] hierarchy level.

   ```
   [edit interfaces]
   user@host# set interfacetype-fpc/pic/port vlan-tagging
   user@host# set interfacetype-fpc/pic/port unit logical-unit-number family bridge interface-mode trunk
   user@host# set interfacetype-fpc/pic/port unit logical-unit-number family bridge vlan-id-list vlan-id
   ```

3. Configure the IRB interface at the [**edit interfaces**] hierarchy level and specify the associated IP address.

   ```
   [edit interfaces]
   user@host# set interfaces irb unit logical-unit-number family inet address address
   ```

4. Configure the IRB interface as the routing interface for the bridge domain at the [**edit bridge-domains**] hierarchy level.

   ```
   [edit bridge-domains]
   ```

```
user@host# set bridge-domain- name vlan-id vlan-id routing-interface irb.logical-interface-number
```

RELATED DOCUMENTATION

Layer 2 Bridging Interfaces Overview | 654
Example: Configuring the MAC Address of an IRB Interface | 656

Example: Configuring the MAC Address of an IRB Interface

IN THIS SECTION

This example shows how to configure the media access control (MAC) address of an integrated routing and bridging (IRB) interface for devices with Modular Port Concentrator (MPC) cards . An IRB interface is a Layer 3 routing interface that is used in a bridge domain or virtual private LAN service (VPLS) routing.

Requirements

This example requires the following hardware and software components:

- MX Series routers with MPC cards.
- Junos OS Release 13.2 or later running on all devices.

Overview

Junos OS Release 13.2 and later support the assignment of MAC addresses to IRB logical interfaces. The IRB logical interfaces provide support for simultaneous Layer 2 bridging and Layer 3 routing within the same bridge domain. Packets that arrive on an interface of the bridge domain are either switched or routed, based on the destination MAC address of the packet. The packets with the router's Layer 2 virtual MAC address, which is manually configured, are switched to Layer 2 interfaces.

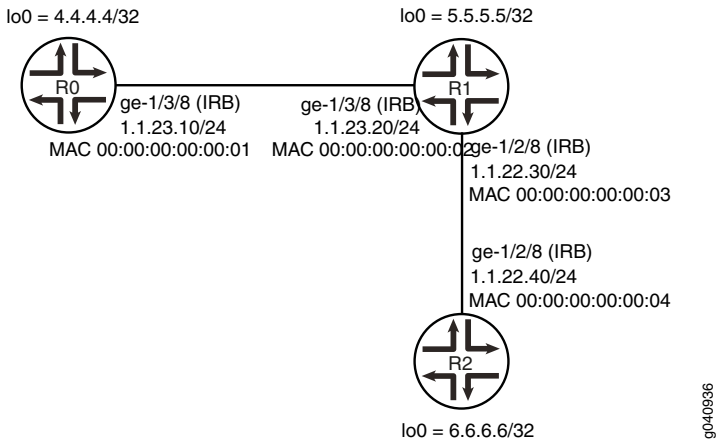
Configuring a MAC address of an IRB logical interface allows the use of a transparent firewall between two VLANs on the same switch. When both VLANs are on the same subnet and traffic from one VLAN needs to go through the firewall to the host on the other VLAN, then the VLAN tag is changed to communicate with the host on the other VLAN.

Before the introduction of this feature, if the MAC address of an IRB logical interface was the same for both VLANs, the firewall dropped the traffic. This new feature allows you to configure distinct MAC addresses for different VLANs, which facilitates the exchange of traffic between two VLANs on the same switch.

In case of VPLS multihoming, if there is a failover of the primary provider edge (PE) router to a secondary PE router, the MAC address of an IRB changes. The hosts connected to the customer edge (CE) router must change their Address Resolution Protocol (ARP) for IRB's IP and MAC address. This feature allows you to configure the same MAC address for IRB interfaces in both the primary and secondary PE routers and eliminates the need for changing the ARP binding of the IRB logical interface in CE routers, in case of a failover.

Figure 35 on page 658 shows the sample topology.

Figure 35: Configuring the MAC Address of an IRB Interface



In this example you configure MAC address of IRB logical interfaces.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Router R0

```
set interfaces ge-1/3/8 vlan-tagging
set interfaces ge-1/3/8 encapsulation flexible-ethernet-services
set interfaces ge-1/3/8 unit 10 encapsulation vlan-bridge
set interfaces ge-1/3/8 unit 10 vlan-id 10
set interfaces irb unit 10 family inet address 1.1.23.1/24
set interfaces irb unit 10 family mpls
set interfaces irb unit 10 mac 00:00:00:00:00:01
set interfaces lo0 unit 10 family inet address 4.4.4.4/32
```

```
set protocols rsvp interface irb.10
set protocols mpls label-switched-path R0-1-R2 to 6.6.6.6
set protocols mpls label-switched-path R0-1-R2 install 6.6.6.6/32 active
set protocols mpls label-switched-path R0-1-R2 no-cspf
set protocols mpls interface irb.10
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 4.4.4.4
set protocols bgp group ibgp neighbor 6.6.6.6
set protocols ospf area 0.0.0.0 interface irb.10
set protocols ospf area 0.0.0.0 interface lo0.10 passive
set protocols ldp interface irb.10
set protocols ldp interface lo0.10
set routing-options autonomous-system 400
set bridge-domains lsbd1 vlan-id 10
set bridge-domains lsbd1 interface ge-1/3/8.10
set bridge-domains lsbd1 routing-interface irb.10
```

Router R1

```
set interfaces ge-1/3/8 vlan-tagging
set interfaces ge-1/3/8 encapsulation flexible-ethernet-services
set interfaces ge-1/3/8 unit 10 encapsulation vlan-bridge
set interfaces ge-1/3/8 unit 10 vlan-id 10
set interfaces ge-1/2/8 vlan-tagging
set interfaces ge-1/2/8 encapsulation flexible-ethernet-services
set interfaces ge-1/2/8 unit 40 encapsulation vlan-bridge
set interfaces ge-1/2/8 unit 40 vlan-id 40
set interfaces irb unit 20 family inet address 1.1.23.2/24
set interfaces irb unit 20 family mpls
set interfaces irb unit 20 mac 00:00:00:00:00:02
set interfaces irb unit 30 family inet address 1.1.22.2/24
set interfaces irb unit 30 family mpls
set interfaces irb unit 30 mac 00:00:00:00:00:03
set interfaces lo0 unit 20 family inet address 5.5.5.5/32
set protocols rsvp interface irb.20
set protocols rsvp interface irb.30
set protocols mpls interface irb.30
set protocols mpls interface irb.20
set protocols ospf area 0.0.0.0 interface irb.20
set protocols ospf area 0.0.0.0 interface irb.30
```

```
set protocols ospf area 0.0.0.0 interface lo0.20 passive
set protocols ldp interface irb.20
set protocols ldp interface irb.30
set protocols ldp interface lo0.20
set routing-options autonomous-system 400
set bridge-domains lsbd2 vlan-id 10
set bridge-domains lsbd2 interface ge-1/3/8.10
set bridge-domains lsbd2 routing-interface irb.20
set bridge-domains lsbd3 vlan-id 40
set bridge-domains lsbd3 interface ge-1/2/8.40
set bridge-domains lsbd3 routing-interface irb.30
```

Router R2

```
set interfaces ge-1/2/8 vlan-tagging
set interfaces ge-1/2/8 encapsulation flexible-ethernet-services
set interfaces ge-1/2/8 unit 40 encapsulation vlan-bridge
set interfaces ge-1/2/8 unit 40 vlan-id 40
set interfaces irb unit 40 family inet address 1.1.22.1/24
set interfaces irb unit 40 family mpls
set interfaces irb unit 40 mac 00:00:00:00:00:04
set interfaces lo0 unit 30 family inet address 6.6.6.6/32
set protocols rsvp interface irb.40
set protocols mpls label-switched-path R2-1-R0 to 4.4.4.4
set protocols mpls label-switched-path R2-1-R0 no-cspf
set protocols mpls interface irb.40
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 6.6.6.6
set protocols bgp group ibgp neighbor 4.4.4.4
set protocols ospf area 0.0.0.0 interface irb.40
set protocols ospf area 0.0.0.0 interface lo0.30 passive
set protocols ldp interface irb.40
set protocols ldp interface lo0.30
set routing-options autonomous-system 400
set bridge-domains lsbd4 vlan-id 40
set bridge-domains lsbd4 interface ge-1/2/8.40
set bridge-domains lsbd4 routing-interface irb.40
```

Configuring the MAC Address of an IRB Interface

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

> **NOTE:** Repeat this procedure for Juniper Networks Routers R1 and R2, modifying the appropriate interface names, addresses, and any other parameters for each router.

To configure the MAC address of an IRB interface on Router R0:

1. Configure the physical interfaces.

```
[edit interfaces ge-1/3/8]
user@R0# set vlan-tagging
user@R0# set encapsulation flexible-ethernet-services
user@R0# set unit 10 encapsulation vlan-bridge
user@R0# set unit 10 vlan-id 10
```

2. Configure the IRB logical interface.

```
[edit interfaces irb]
user@R0# set unit 10 family inet address 1.1.23.1/24
user@R0# set unit 10 family mpls
user@R0# set unit 10 mac 00:00:00:00:00:01

[edit interfaces]
user@R0# set lo0 unit 10 family inet address 4.4.4.4/32
```

3. Configure the RSVP protocol.

```
[edit protocols rsvp]
user@R0# set interface irb.10
```

4. Configure the MPLS protocol.

```
[edit protocols mpls]
user@R0# set label-switched-path R0-1-R2 to 6.6.6.6
```

```
user@R0# set label-switched-path R0-1-R2 install 6.6.6.6/32 active
user@R0# set label-switched-path R0-1-R2 no-cspf
user@R0# set interface irb.10
user@R0# set interface irb.10
```

5. Configure the BGP protocol.

```
[edit protocols BGP]
user@R0# set group ibgp type internal
user@R0# set group ibgp local-address 4.4.4.4
user@R0# set group ibgp neighbor 6.6.6.6
```

6. Configure the OSPF protocol.

```
[edit protocols ospf]
user@R0# set area 0.0.0.0 interface irb.10
user@R0# set area 0.0.0.0 interface lo0.10 passive
```

7. Configure the LDP protocol.

```
[edit protocols ldp]
user@R0# set interface irb.10
user@R0# set interface lo0.10
```

8. Configure the autonomous system (AS) number.

```
[edit routing-options]
user@R0# set autonomous-system 400
```

9. Configure the bridge domains.

```
[edit]
user@R0# set bridge-domains lsbd1 vlan-id 10
user@R0# set bridge-domains lsbd1 interface ge-1/3/8.10
user@R0# set bridge-domains lsbd1 routing-interface irb.10
```

Results

From configuration mode, enter the **show interfaces**, **show protocols** and **show bridge-domains**, commands and confirm your configuration. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R0# show interfaces
ge-1/3/8 {
    unit 10 {
        encapsulation vlan-bridge;
        vlan-id 10;
    }
}
irb {
    unit 10 {
        family inet {
            mtu 1450;
            address 1.1.1.1/24;
            address 1.1.23.1/24;
        }
        family mpls;
        mac 00:00:00:00:00:01;
    }
}
lo0 {
    unit 10 {
        family inet {
            address 4.4.4.4/32;
        }
    }
}
user@R0# show protocols
rsvp {
    interface irb.10;
}
    mpls {
        label-switched-path R0-1-R2 {
            to 6.6.6.6;
            install 6.6.6.6/32 active;
            no-cspf;
        }
        interface irb.10;
    }
    bgp {
        group ibgp {
```

```
            type internal;
            local-address 4.4.4.4;
            neighbor 6.6.6.6;
        }
    }
    ospf {
        area 0.0.0.0 {
            interface irb.10;
            interface lo0.10 {
                passive;
            }
        }
    }
    ldp {
        interface irb.10;
        interface lo0.10;
    }
user@R0# show bridge-domains
lsbd1 {
    vlan-id 10;
    interface ge-1/3/8.10;
    routing-interface irb.10;
}
```

If you are done configuring the devices, commit the configuration.

```
user@host# commit
```

Verification

IN THIS SECTION

- Verifying the MAC Address of the IRB Interface | 665

Confirm that the configuration is working properly.

Verifying the MAC Address of the IRB Interface

Purpose

Verify that the specified MAC address is assigned to the IRB interface.

Action

From operational mode, run the **show interfaces irb** command on the device.

user@host# **show interfaces irb**

```
Physical interface: irb, Enabled, Physical link is Up
  Interface index: 132, SNMP ifIndex: 505
  Type: Ethernet, Link-level type: Ethernet, MTU: 1514
  Device flags   : Present Running
  Interface flags: SNMP-Traps
  Link type      : Full-Duplex
  Link flags     : None
  Current address: 80:71:1f:c2:58:f0, Hardware address: 80:71:1f:c2:58:f0
  Last flapped   : Never
    Input packets : 0
    Output packets: 0

  Logical interface irb.10 (Index 326) (SNMP ifIndex 634)
    Flags: SNMP-Traps 0x0 Encapsulation: ENET2
    MAC: 00:00:00:00:00:01
    Bandwidth: 1000mbps
    Routing Instance: LS1/default Bridging Domain: lsbd1+10
    Input packets : 55202
    Output packets: 69286
    Protocol inet, MTU: 1450
      Flags: Sendbcast-pkt-to-re, Is-Primary, User-MTU
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 1.1.1/24, Local: 1.1.1.1, Broadcast: 1.1.1.255
      Addresses, Flags: Is-Preferred
        Destination: 1.1.23/24, Local: 1.1.23.1, Broadcast: 1.1.23.255
    Protocol mpls, MTU: 1500, Maximum labels: 3
      Flags: Is-Primary
    Protocol multiservice, MTU: 1500

  Logical interface irb.20 (Index 358) (SNMP ifIndex 635)
    Flags: SNMP-Traps 0x0 Encapsulation: ENET2
    MAC: 00:00:00:00:00:02
    Bandwidth: 1000mbps
    Routing Instance: LS2/default Bridging Domain: lsbd2+10
```

```
    Input packets : 66044
    Output packets: 68464
    Protocol inet, MTU: 1450
      Flags: Sendbcast-pkt-to-re, Is-Primary, User-MTU
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 1.1.1/24, Local: 1.1.1.2, Broadcast: 1.1.1.255
      Addresses, Flags: Is-Preferred
        Destination: 1.1.23/24, Local: 1.1.23.2, Broadcast: 1.1.23.255
    Protocol mpls, MTU: 1500, Maximum labels: 3
      Flags: Is-Primary
    Protocol multiservice, MTU: 1500

  Logical interface irb.30 (Index 360) (SNMP ifIndex 636)
    Flags: SNMP-Traps 0x0 Encapsulation: ENET2
    MAC: 00:00:00:00:00:03
    Bandwidth: 1000mbps
    Routing Instance: LS2/default Bridging Domain: lsbd3+40
    Input packets : 26948
    Output packets: 53605
    Protocol inet, MTU: 1500
      Flags: Sendbcast-pkt-to-re
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 1.1.22/24, Local: 1.1.22.2, Broadcast: 1.1.22.255
      Addresses, Flags: Is-Preferred
        Destination: 2.2.2/24, Local: 2.2.2.1, Broadcast: 2.2.2.255
    Protocol mpls, MTU: 1500, Maximum labels: 3
    Protocol multiservice, MTU: 1500

  Logical interface irb.40 (Index 355) (SNMP ifIndex 632)
    Flags: SNMP-Traps 0x0 Encapsulation: ENET2
    MAC: 00:00:00:00:00:04
    Bandwidth: 1000mbps
    Routing Instance: LS3/default Bridging Domain: lsbd4+40
    Input packets : 40575
    Output packets: 31128
    Protocol inet, MTU: 1500
      Flags: Sendbcast-pkt-to-re, Is-Primary
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 1.1.22/24, Local: 1.1.22.1, Broadcast: 1.1.22.255
    Protocol mpls, MTU: 1500, Maximum labels: 3
      Flags: Is-Primary
    Protocol multiservice, MTU: 1500
```

Meaning

The output shows the manually configured MAC address in the MAC field.

> **NOTE:** If you did not configure the MAC address for a logical interface, the output does not include this value. However, the device uses the MAC address of the physical interface during data transmission.

RELATED DOCUMENTATION

mac

Active-Active Bridging and VRRP over IRB Functionality Overview

19
CHAPTER

Configuring Layer 2 Virtual Switch Instances

Layer 2 Virtual Switch Instances

IN THIS SECTION

Understanding Layer 2 Virtual Switches Instances

Benefit of Using Layer 2 Virtual Switch Instances:

- Splitting Layer 2 traffic using virtual switch instances allows you to more logically organize your Layer 2 traffic into multiple “virtual” Layer 2 networks.

At Layer 2, you can group one or more VLANs into a single routing instance to form a virtual switch instance. A virtual switch instance is composed of VLANs. The virtual switch instance isolates a LAN segment and contains most Layer 2 functions, such as spanning-tree protocol instances and VLAN ID spaces, into it’s own smaller, logical network. Splitting Layer 2 traffic using virtual switch instances allows you to more logically organize your Layer 2 traffic into multiple “virtual” Layer 2 networks.

A default virtual switch, called default-switch, is automatically created when a virtual switch is configured. All Layer 2 traffic not assigned to a VLAN in a virtual switch automatically becomes part of the default virtual switch.

You can configure a virtual switch to participate only in Layer 2 bridging and optionally to perform Layer 3 routing. In addition, you can configure spanning-tree protocols (STPs) within the virtual switch to prevent forwarding loops. For more information about how to configure Layer 2 logical ports on an interface, see the *Junos OS Network Interfaces Library for Routing Devices*.

You can associate one or more logical interfaces configured as trunk interfaces with a virtual switch. A trunk interface, or Layer 2 trunk port, enables you to configure a logical interface to represent multiple VLANs on the physical interface. For more information about how to configure trunk interfaces, see the *Junos OS Network Interfaces Library for Routing Devices*.

You can also configure Layer 2 forwarding and learning properties for the virtual switch.

Configuring a Layer 2 Virtual Switch on an EX Series Switch

A Layer 2 virtual switch, which isolates a LAN segment with its spanning-tree protocol instance and separates its VLAN ID space, filters and forwards traffic only at the data link layer. Each VLAN consists of a set of logical ports that participate in Layer 2 learning and forwarding. A virtual switch represents a Layer 2 network.

Two main types of interfaces are used in virtual switch hierarchies:

- Layer 2 logical interface—This type of interface uses the VLAN-ID as a virtual circuit identifier and the scope of the VLAN-ID is local to the interface port. This type of interface is often used in service-provider-centric applications.
- Access or trunk interface—This type of interface uses a VLAN-ID with global significance. The access or trunk interface is implicitly associated with VLANs based on VLAN membership. Access or trunk interfaces are typically used in enterprise-centric applications.

> **NOTE:** The difference between access interfaces and trunk interfaces is that access interfaces can be part of one VLAN only and the interface is normally attached to an end-user device (packets are implicitly associated with the configured VLAN). In contrast, trunk interfaces multiplex traffic from multiple VLANs and usually interconnect switches.

To configure a Layer 2 virtual switch, include the following statements:

```
[edit]
routing-instances {
   routing-instance-name (
      instance-type virtual-switch;
      vlans vlan-name{
         vlan-id (all | none | number);
            [...configure optional VLAN parameters]
      }
   }
}
```

To enable a virtual switch, you must specify **virtual-switch** as the **instance-type**.

The VLANs that are specified with the **vlan-id** statement are included in the virtual switch.

You can configure other optional VLAN parameters in the virtual switch.

Configuring a Layer 2 Virtual Switch with a Layer 2 Trunk Port

You can associate one or more Layer 2 trunk interfaces with a virtual switch.

A virtual switch configured with a Layer 2 trunk port also supports IRB within a VLAN. IRB provides simultaneous support for Layer 2 bridging and Layer 3 IP routing on the same interface. Only an interface configured with the **interface-mode (access | trunk)** statement can be associated with a virtual switch. An access interface enables you to accept packets with no VLAN identifier.

In addition, you can configure Layer 2 learning and forwarding properties for the virtual switch.

To configure a virtual switch with a Layer 2 trunk interface, include the following statements:

```
[edit]
routing-instances {
   routing-instance-name {
      instance-type virtual-switch;
      interface interface-name;
      vlans name{
         vlan-id (all | none | number);
            [...configure optional VLAN parameters]
      }
   }
}
```

CHAPTER 20

Configuring Link Layer Discovery Protocol

LLDP Overview

The Link Layer Discovery Protocol (LLDP) is an industry-standard, vendor-neutral method to allow networked devices to advertise capabilities, identity, and other information onto a LAN. The Layer 2 protocol, detailed in IEEE 802.1AB-2005, replaces several proprietary protocols implemented by individual vendors for their equipment.

LLDP allows network devices that operate at the lower layers of a protocol stack (such as Layer 2 bridges and switches) to learn some of the capabilities and characteristics of LAN devices available to higher layer protocols, such as IP addresses. The information gathered through LLDP operation is stored in a network device and is queried with SNMP. Topology information can also be gathered from this database.

Some of the information that can be gathered by LLDP (only minimal information is mandatory) is:

- System name and description
- Port name and description
- VLAN name and identifier
- IP network management address
- Capabilities of the device (for example, switch, router, or server)
- MAC address and physical layer information
- Power information
- Link aggregation information

LLDP frames are sent at fixed intervals on each port that runs LLDP. LLDP protocol data units (LLDP PDUs) are sent inside Ethernet frames and identified by their destination Media Access Control (MAC) address (**01:80:C2:00:00:0E**) and Ethertype (**0x88CC**). Mandatory information supplied by LLDP is chassis ID, port ID, and a time-to-live value for this information.

RELATED DOCUMENTATION

Configuring LLDP | **674**

Tracing LLDP Operations | **681**

Example: Configuring LLDP | **679**

LLDP Operational Mode Commands | **680**

Configuring LLDP

You configure LLDP by including the **lldp** statement and associated parameters at the **[edit protocols]** hierarchy level. The complete set of LLDP statements follows:

```
lldp {
    advertisement-interval seconds;
    (disable | enable);
    hold-multiplier number;
    interface (all | [interface-name]) {
        (disable | enable);
        power-negotiation <(disable | enable)>;
        tlv-filter;
        tlv-select;
        trap-notification (disable | enable);
    }
    lldp-configuration-notification-interval seconds;
    management-address ip-management-address;;
    mau-type;
    netbios-snooping;
    no-tagging;
    neighbour-port-info-display (port-description | port-id);
    port-description-type (interface-alias | interface-description);
    port-id-subtype (interface-name | locally-assigned);
    ptopo-configuration-maximum-hold-time seconds;
    ptopo-configuration-trap-interval seconds;
    tlv-filter;
    tlv-select;
    traceoptions {
        file filename <files number> <size maximum-file-size> <world-readable | no-world-readable>;
        flag flag <disable>;
    }
    transmit-delay seconds;
    vlan-name-tlv-option (name | vlan-id);
}
```

The following statements have default values:

- **advertisement-interval**—The default value is 30 seconds. The allowable range is from 5 through 32768 seconds.
- **hold-multiplier**—The default values is 4. The allowable range is from 2 through 10.

- **ptopo-configuration-maximum-hold-time**—The default value is 300 seconds. The allowable range is from 1 through 2147483647 seconds.
- **transmit-delay**—The default values is 2 seconds. The allowable range is from 1 through 8192 seconds.

The following statements must be explicitly configured:

- **lldp-configuration-notification-interval**—The allowable range is from 0 through 3600 seconds. There is no default value.
- **ptopo-configuration-trap-interval**—The allowable range is from 1 through 2147483647 seconds. There is no default value.

By default, LLDP is disabled, and user must configure it using **[set protocols lldp interface (all | *interface-name*)]** to use the LLDP services. If it is enabled for all interfaces, you can disable LLDP on specific interfaces.

> **NOTE:** The **interface-name** must be the physical interface (for example, **ge-1/0/0**) and not a logical interface (unit).

Starting in Junos OS Release 19.4R2, you can configure the LLDP on redundant Ethernet (reth) interfaces. Use the **set protocol lldp interface *<reth-interface>*** command to configure LLDP on reth interface.

- To configure LLDP on all interfaces:

```
[edit protocols lldp]
user@switch# set interface all
```

- To configure LLDP on a specific interface:

```
[edit protocols lldp]
user@switch# set interface interface-name
```

To disable LLDP, include the **disable** option:

- To disable LLDP on all interfaces:

```
[edit protocols lldp]
user@switch# set interface all disable
```

- To disable LLDP on a specific interface:

```
[edit protocols lldp]
user@switch# set interface interface-name disable
```

Starting with Junos OS Release 14.2, you can configure management interfaces, such as fxp0 or me0, on MX Series routers to send LLDP frames to and receive LLDP frames from neighboring LLDP interfaces. To configure the management interfaces, include the **interface *interface-name*** statement at the **[edit protocols lldp]** and **[edit routing-instances *routing-instance-name* protocols lldp]** hierarchy levels. By default, the functionality to send LLDP frames is enabled. You can also specify a management interface with the **show lldp neighbors interface *interface-name*** command to view configuration details about LLDP neighbors for the corresponding management interface.

To configure LLDP on a T Series router within a TX Matrix, you must specify the interface name in the LLDP configuration for the TX Matrix. For information about interface names for TX Matrix routers, see *TX Matrix Router Chassis and Interface Names*. For information about FPC numbering, see *Routing Matrix with a TX Matrix Router FPC Numbering*.

Starting with Junos OS Release 14.2, LLDP is supported on extended ports in the Junos Fusion technology.For information about interface names in the Junos Fusion technology, see *Understanding Junos Fusion Ports* .

The advertisement interval determines the frequency that an LLDP interface sends LLDP advertisement frames. The default value is 30 seconds. The allowable range is from 5 through 32768 seconds. You adjust this parameter by including the **advertisement-interval** statement at the **[edit protocols lldp]** hierarchy level.

The hold multiplier determines the multiplier to apply to the advertisement interval. The resulting value in seconds is used to cache learned LLDP information before discard. The default value is 4. When used with the default advertisement interval value of 30 seconds, this makes the default cache lifetime 120 seconds. The allowable range of the hold multiplier is from 2 through 10. You adjust this parameter by including the **hold-multiplier** statement at the **[edit protocols lldp]** hierarchy level.

The transmit delay determines the delay between any two consecutive LLDP advertisement frames. The default value is 2 seconds. The allowable range is from 1 through 8192 seconds. You adjust this parameter by including the **transmit-delay** statement at the **[edit protocols lldp]** hierarchy level.

The physical topology configuration maximum hold time determines the time interval for which an agent device maintains physical topology database entries. The default value is 300 seconds. The allowable range is from 1 through 2147483647 seconds. You adjust this parameter by including the **ptopo-configuration-maximum-hold-time** statement at the **[edit protocols lldp]** hierarchy level.

The LLDP configuration notification interval determines the period for which trap notifications are sent to the SNMP Master Agent when changes occur in the database of LLDP information. This capability is disabled by default. The allowable range is from 0 (disabled) through 3600 seconds. You adjust this parameter by including the **lldp-configuration-notification-interval** statement at the **[edit protocols lldp]** hierarchy level.

The physical topology configuration trap interval determines the period for which trap notifications are sent to the SNMP Master Agent when changes occur in the global physical topology statistics. This capability is disabled by default. The allowable range is from 0 (disabled) through 3600 seconds. The LLDP agent

sends traps to the SNMP Master Agent if this interval has a value greater than 0 and there is any change during the **lldp-configuration-notification-interval** trap interval. You adjust this parameter by including the **ptopo-configuration-trap-interval** statement at the **[edit protocols lldp]** hierarchy level.

Starting in Junos OS Release 15.1R7, you can enable or disable the Link Layer Discovery Protocol (LLDP) and Physical Topology (PTOPO) MIB traps for a specific interface or for all interfaces on EX3300, EX4200, EX4500, EX4550, EX6200, and EX8200 switches by configuring the trap-notification statement at the **[edit protocols lldp interface interface-name]** hierarchy level.

By default, LLDP generates the SNMP index of the interface for the port ID Type, Length, and Value (TLV). Starting with Junos OS Release 12.3R1, you can generate the interface name as the port ID TLV. To do so, include the **interface-name** statement at the **[edit protocols lldp port-id-subtype**] hierarchy level. When the **interface-name** statement is configured on the remote LLDP neighbor, the **show lldp neighbors** command output displays the interface name in the **Port ID** field rather than the SNMP index of the interface, which is displayed by default. If you change the default behavior of generating the SNMP index of the interface as the Port ID TLV, you can reenable the default behavior by including the **locally-assigned** statement at the **[edit protocols lldp port-id-subtype**] hierarchy level.

> **NOTE:** Starting with Junos OS Release 12.3, the value of the MIB variable **lldpLocPortId** depends on the SNMP MIB object entity that is used to generate the port ID TLV. If the port ID TLV generation is configured to use the interface name in the **set port-id-subtype interface-name** command, then the value of the MIB variable **lldpLocPortId** is the interface name and not the SNMP index.

Release History Table

Release	Description
19.4R2	Starting in Junos OS Release 19.4R2, you can configure the LLDP on redundant Ethernet (reth) interfaces. Use the **set protocol lldp interface *<reth-interface>*** command to configure LLDP on reth interface.
15.1R7	Starting in Junos OS Release 15.1R7, you can enable or disable the Link Layer Discovery Protocol (LLDP) and Physical Topology (PTOPO) MIB traps for a specific interface or for all interfaces on EX3300, EX4200, EX4500, EX4550, EX6200, and EX8200 switches by configuring the trap-notification statement at the **[edit protocols lldp interface interface-name]** hierarchy level.
14.2	Starting with Junos OS Release 14.2, you can configure management interfaces, such as fxp0 or me0, on MX Series routers to send LLDP frames to and receive LLDP frames from neighboring LLDP interfaces.
14.2	Starting with Junos OS Release 14.2, LLDP is supported on extended ports in the Junos Fusion technology.
12.3	Starting with Junos OS Release 12.3R1, you can generate the interface name as the port ID TLV.
12.3	Starting with Junos OS Release 12.3, the value of the MIB variable **lldpLocPortId** depends on the SNMP MIB object entity that is used to generate the port ID TLV.

RELATED DOCUMENTATION

LLDP Overview | **673**

Tracing LLDP Operations | **681**

Example: Configuring LLDP | **679**

TX Matrix Router Chassis and Interface Names

Monitoring a Routing Matrix with a TX Matrix Router

Example: Configuring LLDP

The following example configures LLDP on interface **ge-1/1/1** but disables LLDP on all other interfaces, explicitly configures the default values for all automatically enabled features, and configures a value of 30 seconds for the LLDP configuration notification interval and a value of 30 seconds for the physical topology trap interval.

```
[edit]
protocols {
    lldp {
        advertisement-interval 30;
        hold-multiplier 4;
        interface all {
            disable;
        }
        interface ge-1/1/1;
        lldp-configuration-notification-interval 30;
        ptopo-configuration-maximum-hold-time 300;
        ptopo-configuration-trap-interval 30;
        transmit-delay 2;
    }
}
```

You verify operation of LLDP with several show commands:

- **show lldp <detail>**
- **show lldp neighbors** ***interface-name***
- **show lldp statistics** ***interface-name***
- **show lldp local-information**
- **show lldp remote-global-statisitics**

You can clear LLDP neighbor information or statistics globally or on an interface:

- **clear lldp neighbors** ***interface-name***
- **clear lldp statistics** ***interface-name***

You can display basic information about LLDP with the **show lldp detail** command:

```
user@host> show lldp detail
```

```
LLDP                    : Enabled
Advertisement interval  : 30 Second(s)
Transmit delay          : 2 Second(s)
Hold timer              : 4 Second(s)
Notification interval   : 30 Second(s)
Config Trap Interval    : 300 Second(s)
Connection Hold timer   : 60 Second(s)


Interface      LLDP       Neighbor count
ge-1/1/1       Enabled    0

LLDP basic TLVs supported:
Chassis identifier, Port identifier, Port description, System name, System
description, System capabilities, Management address.

LLDP 802 TLVs supported:
Link aggregation, Maximum frame size, MAC/PHY Configuration/Status, Port VLAN ID,
 Port VLAN name.
```

For more details about the output of these commands, see the CLI Explorer.

RELATED DOCUMENTATION

- LLDP Overview | 673
- Configuring LLDP | 674
- Tracing LLDP Operations | 681

LLDP Operational Mode Commands

Table 102 on page 680 summarizes the command-line interface (CLI) commands you can use to monitor and troubleshoot the Link Layer Discovery Protocol (LLDP) protocol. Commands are listed in alphabetical order.

Table 102: LLDP Operational Mode Commands

Task	Command
Clear LLDP neighbor information.	clear lldp neighbors

Table 102: LLDP Operational Mode Commands *(continued)*

Task	Command
Clear LLDP statistics.	clear lldp statistics
Display basic LLDP information.	show lldp
Display LLDP local information.	show lldp local-information
Display LLDP neighbor information.	show lldp neighbors
Display LLDP remote global statistics.	show lldp remote-global-statistics
Display LLDP statistics.	show lldp statistics

RELATED DOCUMENTATION

- LLDP Overview | **673**
- Configuring LLDP | **674**
- Tracing LLDP Operations | **681**
- Example: Configuring LLDP | **679**

Tracing LLDP Operations

To trace LLDP operational traffic, you can specify options in the global **traceoptions** statement included at the **[edit routing-options]** hierarchy level, and you can specify LLDP-specific options by including the **traceoptions** statement:

```
traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
}
```

You can include this statement at the following hierarchy levels:

- **[edit protocols lldp]**
- **[edit routing-instances *routing-instance-name* protocols lldp]**

You can specify the following LLDP-specific options in the LLDP **traceoptions** statement:

- **all**—Trace all operations.
- **config**—Log configuration events.
- **interface**—Trace interface update events.
- **protocol**—Trace protocol information.
- **rtsock**—Trace real-time socket events.
- **vlan**—Trace VLAN update events.

> **NOTE:** Use the trace flag **all** with caution. This flag may cause the CPU to become very busy.

For general information about tracing and global tracing options, see the statement summary for the global **traceoptions** statement in the *Junos OS Routing Protocols Library*.

RELATED DOCUMENTATION

LLDP Overview | **673**

Configuring LLDP | **674**

Example: Configuring LLDP | **679**

21

CHAPTER

Configuring Layer 2 Protocol Tunneling

Layer 2 Protocol Tunneling

IN THIS SECTION

Understanding Layer 2 Protocol Tunneling

IN THIS SECTION

Juniper Networks Ethernet switches and routers use Layer 2 protocol tunneling (L2PT) to send Layer 2 protocol data units (PDUs) across the network and deliver them to devices that are not part of the local broadcast domain. This feature is useful when you want to run Layer 2 protocols on a network that includes switches located at remote sites that are connected across a service provider network.

You can also use L2PT to tunnel protocols between two locally-connected user-to-network interfaces (UNIs) in the same broadcast domain, but in that case, the device floods protocol packets in the VLAN instead of rewriting the packets with the tunnel MAC address.

See Feature Explorer for the list of devices that support L2PT.

Benefits of Layer 2 Protocol Tunneling

- Enables you to run supported Layer 2 protocols in a tunnel across a service provider network to remote sites.
- Provides a single spanning-tree protocol domain for subscribers across a service provider network.

How Layer 2 Protocol Tunneling Works

L2PT works by encapsulating Layer 2 PDUs, tunneling them across a service provider network, and decapsulating them for delivery to their destination switches. The ingress service provider edge (PE) device encapsulates Layer 2 PDUs by rewriting the PDUs' destination media access control (MAC) addresses before forwarding them onto the service provider network. The devices in the service provider network treat these encapsulated PDUs as multicast Ethernet packets. Upon receipt of these PDUs, the egress PE devices decapsulate them by replacing the destination MAC addresses with the address of the Layer 2 protocol that is being tunneled before forwarding the PDUs to their destination devices.

When a PE port configured for Layer 2 protocol tunneling receives a control packet for a supported Layer 2 protocol, the PE device rewrites the multicast destination MAC address with the predefined multicast tunnel MAC address 01:00:0C:CD:CD:D0. The PE device then sends the modified packet onto the provider network. The packet travels across the provider network transparently across the service provider network with the tunnel MAC address. All devices on the provider network treat these packets as multicast Ethernet packets and deliver them to all PE devices for the customer. The egress PE devices receive all the control PDUs with the tunnel MAC address, identify the packet type by doing deeper packet inspection, and replace the destination MAC address with the appropriate destination MAC address. The egress PE devices send out the modified PDUs to the customer PE devices, and the original MAC address is restored when the packets reach the destination ports.

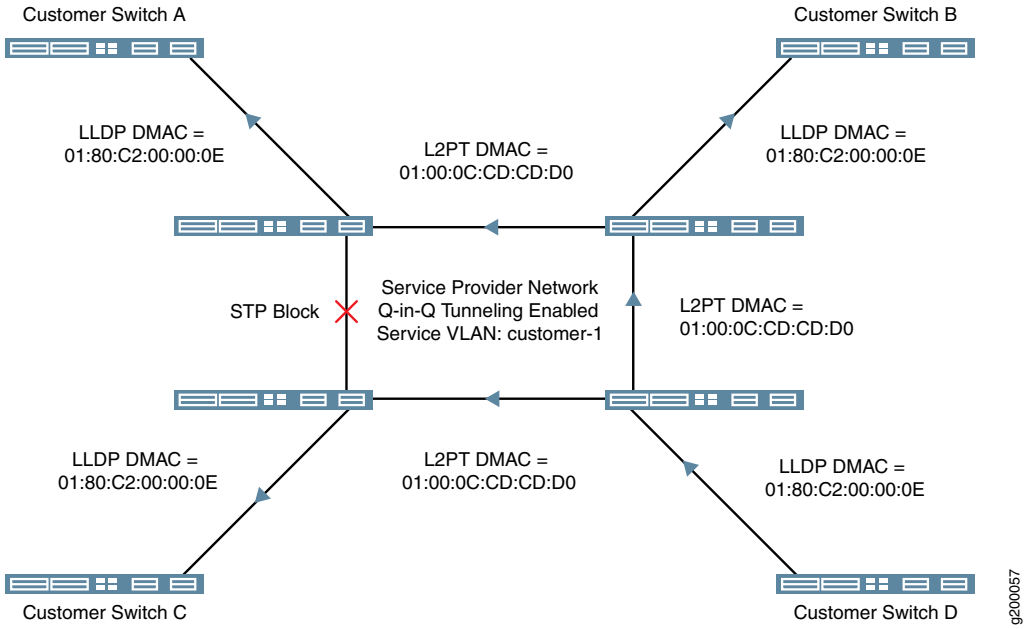
The L2PT protocol is valid for all types of packets, such as untagged, tagged, and Q-in-Q tagged packets.

If a PE device receives a packet on a tunnel interface that already has a destination MAC address of 01:00:0C:CD:CD:D0, the device puts the port into an error state and shuts down the port. You can clear this error condition on an interface using the CLI by entering the **clear error mac-rewrite interface *interface-name*** command on the following devices that support L2PT:

- MX Series and ACX Series routers
- EX Series switches that use Enhanced Layer 2 Software (ELS)—EX2300, EX3400, EX4300, EX4600, EX4650, and EX9200 switches
- QFX Series switches

Figure 36 on page 686 illustrates an example of the L2PT process with EX Series switches in a service provider network that are configured to tunnel LLDP packets on a service VLAN with Q-in-Q tunneling enabled.

Figure 36: L2PT LLDP Example



1. Customer Switch D sends an LLDP PDU to the service provider network that is ultimately intended for the other switches in the customer network.

2. The receiving provider switch rewrites the LLDP destination MAC address with the L2PT destination MAC address, and sends the frame with the encapsulated LLDP PDU to the other switches in the service provider network.

3. When the other service provider switches receive the frame, they detect the L2PT destination MAC address, restore the LLDP destination MAC address, and forward it to Customer Switches A, B, and C.

MX Series Router Support for Layer 2 Protocol Tunneling

MX Series routers support tunneling the following Layer 2 PDUs:

- Cisco Discovery Protocol (CDP)—MAC address 01:00:0C:CC:CC:CC
- Per-VLAN Spanning Tree Protocol (PVSTP)—MAC address 01:00:0C:CC:CC:CD

- Spanning Tree Protocol (STP), Rapid Spanning Tree Protocol (RSTP), and Multiple Spanning Tree Protocol (MSTP)—MAC address 01:80:C2:00:00:00
- VLAN Trunking Protocol (VTP)—MAC address 01:00:0C:CC:CC:CC

You can configure L2PT on an interface using the **mac-rewrite** CLI command at the **[edit protocols layer2-control]** hierarchy level.

Layer 2 protocol tunneling is supported on MX Series routers with Enhanced (Dense Port Concentrators) DPCs and Enhanced Queuing DPCs. See Table 104 on page 687 for a list of the supported DPCs. Layer 2 protocol tunneling is supported on all Modular Port Concentrators (MPCs).

> **NOTE:** Layer 2 protocol tunneling is not supported on Rev-A DPCs on MX Series routers because of microcode space limitations.

Layer 2 protocol tunneling and MAC rewrite are supported in VPLS, but only certain hardware configurations are supported.

Table 103 on page 687 shows the MPCs and Enhanced DPCs supported when configuring Layer 2 protocol tunneling and VPLS.

Table 103: MAC Rewrite and VPLS Configurations

CE-Facing Interface	PE-Core Facing Interface	Layer 2 Protocol Tunneling
MPC	MPC	Yes
MPC	Enhanced DPC	Yes
Enhanced DPC	MPC	Yes
Enhanced DPC	Enhanced DPC	No

Table 104 on page 687 lists the DPCs that support the Layer 2 tunneling protocol.

Table 104: DPCs Supported for Layer 2 Protocol Tunneling

DPC Name	DPC Model Number
Gigabit Ethernet	
Gigabit Ethernet Enhanced DPC with SFP	DPCE-R-40GE-SFP

Table 104: DPCs Supported for Layer 2 Protocol Tunneling *(continued)*

DPC Name	DPC Model Number
Gigabit Ethernet Enhanced Ethernet Services DPC with SFP	DPCE-X-40GE-SFP
Gigabit Ethernet Enhanced Queuing Ethernet Services DPC with SFP	DPCE-X-Q-40GE-SFP
Gigabit Ethernet Enhanced Queuing IP Services DPCs with SFP	DPCE-R-Q-20GE-SFP
Gigabit Ethernet Enhanced Queuing IP Services DPCs with SFP	DPCE-R-Q-40GE-SFP
10-Gigabit Ethernet	
10-Gigabit Ethernet Enhanced DPCs with XFP	DPCE-R-2XGE-XFP
10-Gigabit Ethernet Enhanced DPCs with XFP	DPCE-R-4XGE-XFP
10-Gigabit Ethernet Enhanced Ethernet Services DPC with XFP	DPCE-X-4XGE-XFP
10-Gigabit Ethernet Enhanced Queuing Ethernet Services DPC with XFP	DPCE-X-Q-4XGE-XFP
10-Gigabit Ethernet Enhanced Queuing IP Services DPC with XFP	DPCE-R-Q-4XGE-XFP
Multi-Rate Ethernet	
Multi-Rate Ethernet Enhanced DPC with SFP and XFP	DPCE-R-20GE-2XGE
Multi-Rate Ethernet Enhanced Ethernet Services DPC with SFP and XFP	DPCE-X-20GE-2XGE
Multi-Rate Ethernet Enhanced Queuing IP Services DPC with SFP and XFP	DPCE-R-Q-20GE-2XGE
Tri-Rate Ethernet	
Tri-Rate Enhanced DPC	DPCE-R-40GE-TX

Table 104: DPCs Supported for Layer 2 Protocol Tunneling *(continued)*

DPC Name	DPC Model Number
Tri-Rate Enhanced Ethernet Services DPC	DPCE-X-40GE-TX

> **NOTE:** When a device sends a RADIUS access request, the **Chargeable-User-Identity** parameter is an empty field. For more information about configuring RADIUS, see the *Junos Subscriber Access Configuration Guide*.

ACX Series Router Support for Layer 2 Protocol Tunneling

On ACX Series routers, you can configure L2PT on an interface using the **mac-rewrite** CLI command at the**[edit protocols layer2-control]** hierarchy level.

L2PT on ACX Series routers supports tunneling the Layer 2 PDUs listed in Table 105 on page 689, with the indicated Ethernet encapsulation type and MAC address:

Table 105: Layer 2 Protocol Tunneling Support on ACX Series Routers

Protocol	Ethernet Encapsulation	MAC Address
802.1X (IEEE 802.1X authentication)	Ether (0x888E)	01:80:C2:00:00:03
802.3ah (IEEE 802.3ah Operation, Administration, and Maintenance (OAM) link fault management (LFM))	Ether (0x8809)	01:80:C2:00:00:02
Cisco Discovery Protocol (CDP)	LLC (0xAAAA03)	01:00:0C:CC:CC:CC
Ethernet local management interface (E-LMI)	Ether (0x88EE)	01:80:C2:00:00:07
Link Aggregation Control Protocol (LACP)	Ether (0x8809)	01:80:C2:00:00:02
Link Layer Discovery Protocol (LLDP)	Ether (0x88CC)	01:80:C2:00:00:0E
Multiple MAC Registration Protocol (MMRP)	Ether (0x88F5)	01:80:C2:00:00:20
MVRP VLAN Registration Protocol (MVRP)	Ether (0x88F6)	01:80:c2:00:00:21

Table 105: Layer 2 Protocol Tunneling Support on ACX Series Routers *(continued)*

Protocol	Ethernet Encapsulation	MAC Address
Spanning Tree Protocol (STP), Rapid Spanning Tree Protocol (RSTP), and Multiple Spanning Tree Protocol (MSTP)	LLC (0x424203)	01:80:C2:00:00:00
VLAN Trunking Protocol (VTP)	LLC (0xAAAA03)	01:00:0C:CC:CC:CC

EX Series and QFX Series Switch Support for Layer 2 Protocol Tunneling

Table 106 on page 690 lists the Layer 2 protocols that can be tunneled on QFX Series and EX Series switches. QFX Series and EX Series switches that use the Enhanced Layer 2 Software (ELS) configuration style share the same configuration hierarchy to set up L2PT. The configuration hierarchy is different for EX Series switches that do not support ELS. For details on the configuration options to enable tunneling the supported protocols on each type of switch, and the releases in which those options are supported, see either of the following configuration statements:

- QFX Series switches and EX Series ELS switches (EX2300, EX3400, EX4300, EX4600, EX4650, and EX9200): **protocol** statement in the **[edit protocols layer2-control mac-rewrite interface *interface-name*]** hierarchy.
- Non-ELS switches (EX2200, EX3300, EX4200, EX4500, and EX4450): **layer2-protocol-tunneling** statement in the **[edit vlans *vlan-name* dot1q-tunneling]** hierarchy.

All switches that support L2PT can tunnel the listed protocols unless otherwise noted in the second column.

Table 106: L2PT Protocols Supported on EX Series and QFX Series Switches

Layer 2 Protocol That Can Be Tunneled	Support Notes and Exceptions
802.1X authentication	Not supported on EX2300 multigigabit model switches.
802.3ah Operation, Administration, and Maintenance (OAM) link fault management (LFM)	If you enable L2PT for untagged OAM LFM packets, do not configure LFM on the corresponding access interface.
Cisco Discovery Protocol (CDP)	You can't configure CDP on EX Series and QFX Series switches. However, L2PT can tunnel CDP PDUs.
Ethernet local management interface (E-LMI)	Not supported on EX2300 multigigabit model switches.
Generic Attribute Registration Protocol (GARP) VLAN Registration Protocol (GVRP)	

Table 106: L2PT Protocols Supported on EX Series and QFX Series Switches *(continued)*

Layer 2 Protocol That Can Be Tunneled	Support Notes and Exceptions
Link Aggregation Control Protocol (LACP)	If you enable L2PT for untagged LACP packets, do not configure Link Aggregation Control Protocol (LACP) on the corresponding access interface.
Link Layer Discovery Protocol (LLDP)	
Multiple MAC Registration Protocol (MMRP)	Not supported on EX2300 multigigabit model switches.
MVRP VLAN Registration Protocol (MVRP)	
Per-VLAN Spanning Tree and Per-VLAN Spanning Tree Plus (PVST+) Protocols	Only supported on EX9200 switches. Use this option to enable tunneling VSTP instead of the **vstp** option.
Spanning Tree Protocol (STP), Rapid Spanning Tree Protocol (RSTP), and Multiple Spanning Tree Protocol (MSTP)	
Unidirectional Link Detection (UDLD)	Not supported on EX2300 multigigabit model switches. You can't configure UDLD on EX Series and QFX Series switches. However, L2PT can tunnel UDLD PDUs.
VLAN Spanning Tree Protocol (VSTP)	EX9200 switches support tunneling VSTP packets but do not have a separate option to enable tunneling VSTP. The option that enables tunneling PVST and PVST+ (**pvstp**) also enables tunneling VSTP.
VLAN Trunking Protocol (VTP)	You can't configure VTP on EX Series and QFX Series switches. However, L2PT can tunnel VTP PDUs.

The egress PE switches use the encapsulated MAC address to identify the tunneled Layer 2 control protocol and do the destination MAC address rewrite. Table 107 on page 691 lists the supported protocols and their corresponding encapsulation types and MAC addresses on EX Series and QFX Series switches:

Table 107: Protocol Destination MAC Addresses

Protocol	Ethernet Encapsulation	MAC Address
802.1X	Ether-II	01:80:C2:00:00:03

Table 107: Protocol Destination MAC Addresses *(continued)*

Protocol	Ethernet Encapsulation	MAC Address
802.3ah	Ether-II	01:80:C2:00:00:02
CDP	LLC/SNAP	01:00:0C:CC:CC:CC
E-LMI	Ether-II	01:80:C2:00:00:07
GVRP	LLC/SNAP	01:80:C2:00:00:21
LACP	Ether-II	01:80:C2:00:00:02
LLDP	Ether-II	01:80:C2:00:00:0E
MMRP	Ether-II	01:80:C2:00:00:20
MVRP	Ether-II	01:80:C2:00:00:21
PVSTP	LLC/SNAP	01:00:0C:CC:CC:CD
STP, RSTP, MSTP	LLC/SNAP	01:80:C2:00:00:00
UDLD	LLC/SNAP	01:00:0C:CC:CC:CC
VSTP	LLC/SNAP	01:00:0C:CC:CC:CD
VTP	LLC/SNAP	01:00:0C:CC:CC:CC

VLAN and Q-in-Q Tunneling Configuration Requirements for Configuring L2PT on Switches

On switches, you enable L2PT on a per-VLAN basis. When you enable L2PT for a particular Layer 2 protocol on a VLAN, all access interfaces are considered to be customer-facing interfaces and all trunk interfaces are considered to be service provider network-facing interfaces. You cannot configure the specified protocol on the access interfaces. L2PT only acts on logical interfaces with family **ethernet-switching**. The switch floods L2PT PDUs to all trunk and access ports within a given S-VLAN.

NOTE: Access interfaces in an L2PT-enabled VLAN should not receive L2PT-tunneled PDUs. If an access interface does receive L2PT-tunneled PDUs, there might be a loop in the network, and the device will shut down the interface.

You must configure and enable Q-in-Q tunneling (802.1Q VLAN encapsulation) before you can configure L2PT. For information about Q-in-Q tunneling on EX9200 switches, see "Configuring VLAN Encapsulation" on page 291 and related topics, or for other EX Series and QFX Series switches, see "Understanding Q-in-Q Tunneling and VLAN Translation" on page 887.

For QFX Series and ELS EX Series switches, you configure L2PT using statements in the **[edit layer2-control mac-rewrite interface *interface-name*]** hierarchy to enable MAC address rewriting for Layer 2 protocol tunneling for a configured Q-in-Q interface. For details, see "Configuring Layer 2 Protocol Tunneling" on page 694.

For non-ELS EX Series switches, you configure L2PT using statements in the **[edit vlans *vlan-name* dot1q-tunneling]** hierarchy, which means Q-in-Q tunneling is (and must be) enabled. For details on configuring L2PT on non-ELS EX Series switches, see "Configuring Layer 2 Protocol Tunneling on EX Series Switches Without ELS Support" on page 699.

> **NOTE:** If the switch receives untagged or priority-tagged Layer 2 control PDUs to be tunneled, then you must configure the switch to map untagged and priority-tagged packets to an L2PT-enabled VLAN. For more information on assigning untagged and priority-tagged packets to VLANs, see "Understanding Q-in-Q Tunneling and VLAN Translation" on page 887 and "Configuring Q-in-Q Tunneling on EX Series Switches" on page 910.

SEE ALSO

Configuring VLAN Encapsulation | **291**

Configuring Layer 2 Protocol Tunneling

> **NOTE:** This topic applies to Junos OS for routers, QFX Series switches, and EX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style. To configure Layer 2 protocol tunneling (L2PT) on EX Series switches that do not use ELS, see "Configuring Layer 2 Protocol Tunneling on EX Series Switches Without ELS Support" on page 699. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50.

With Layer 2 protocol tunneling (L2PT) enabled, Juniper Networks Ethernet routers and switches can send Layer 2 protocol data units (PDUs) across the network and deliver them to devices that are not part of the local broadcast domain. This feature is useful when you want to run Layer 2 protocols on a network that includes switches located at remote sites that are connected across a service provider network.

You can also use L2PT to tunnel protocols between two locally-connected user-to-network interfaces (UNIs) in the same broadcast domain, but in that case, the protocol packets are simply flooded in the VLAN instead of being rewritten with the tunnel MAC address.

To configure L2PT, you enable MAC address rewriting for Layer 2 protocol tunneling, which installs the destination multicast tunnel MAC address 01:00:0C:CD:CD:D0 in the MAC table. At the same time, you select the Layer 2 protocol to be tunneled from the list of available options for the type of switch you are configuring (see **protocol**).

Use the following guidelines when you configure L2PT:

- Layer 2 protocol tunneling must be configured on the interfaces at both ends of the tunnel.
- You can enable Layer 2 protocol tunneling for untagged interfaces and single-identifier tagged interfaces only, not for double-identifier tagged interfaces.

  For single-identifier tagged ports, configure a logical interface with the native VLAN identifier. This configuration associates the untagged control packets with a logical interface.
- MX Series routers must have enhanced queuing Dense Port Concentrators (DPCs) to support Layer 2 protocol tunneling.
- To configure L2PT on a QFX Series switch or an EX Series switch, you must first configure a Q-in-Q interface or group of interfaces, and configure L2PT on a specified Q-in-Q interface.
  - For information on configuring Q-in-Q tunneling on EX9200 switches, see "Configuring VLAN Encapsulation" on page 291, "Configuring Inner and Outer TPIDs and VLAN IDs" on page 374, and "Stacking a VLAN Tag" on page 367.
  - For information on configuring Q-in-Q tunneling on other EX Series switches that use the Enhanced Layer 2 Software (ELS) configuration style, see "Configuring Q-in-Q Tunneling on EX Series Switches with ELS Support" on page 900.

- For information on configuring Q-in-Q tunneling on EX Series switches that do not use the ELS configuration style, see "Configuring Q-in-Q Tunneling on EX Series Switches" on page 910.
- For information on configuring Q-in-Q tunneling on QFX Series switches, see "Configuring Q-in-Q Tunneling on QFX Series Switches" on page 899.

> **NOTE:** When you enable L2PT tunneling for a protocol on one user-to-network interface (UNI) in a bridge domain or VLAN, you should also configure all UNIs in the bridge domain or VLAN to tunnel the same protocol for consistent behavior. In that case, those UNIs can receive non-tunneled packets, and tunneled packets are forwarded through the network-to-network interfaces (NNIs).

1. To configure L2PT on a specified interface:

```
[edit protocols]
user@device# set layer2-control mac-rewrite interface interface-name protocol protocol-name
```

> **NOTE:** You can select only one Layer 2 protocol at a time. If you want an interface to support tunneling more than one Layer 2 protocol, you must enter the **mac-rewrite** statement separately to select each of the protocols you want to tunnel.

For example, on an EX9200 switch, the following commands configure a UNI (**xe-1/1/3**) for Q-in-Q tunneling and MAC address rewriting for STP:

```
set interfaces xe-1/1/3 flexible-vlan-tagging
set interfaces xe-1/1/3 encapsulation extended-vlan-bridge
set interfaces xe-1/1/3 unit 10 encapsulation vlan-bridge
set interfaces xe-1/1/3 unit 10 vlan-id 10
set interfaces xe-1/1/3 native-vlan-id 10
set interfaces xe-1/1/3 unit 10 input-vlan-map push
set interfaces xe-1/1/3 unit 10 input-vlan-map vlan-id 100
set interfaces xe-1/1/3 unit 10 output-vlan-map pop
set protocols layer2-control mac-rewrite interface xe-1/1/3 protocol stp
set vlans v10 interface xe-1/1/3.10
```

On an ELS EX Series switch or a QFX Series switch, the following commands configure a UNI (**ge-0/0/0**) for Q-in-Q tunneling and MAC address rewriting for STP and LLDP:

```
set interfaces ge-0/0/0 flexible-vlan-tagging
set interfaces ge-0/0/0 encapsulation extended-vlan-bridge
set interfaces ge-0/0/0 unit 10 vlan-id 10
```

```
set interfaces ge-0/0/0 native-vlan-id 10
set interfaces ge-0/0/0 unit 10 input-vlan-map push
set interfaces ge-0/0/0 unit 10 output-vlan-map pop
set protocols layer2-control mac-rewrite interface ge-0/0/0 protocol stp
set protocols layer2-control mac-rewrite interface ge-0/0/0 protocol lldp
set vlans v10 interface ge-0/0/0.10
```

When configuring L2PT on switches in the case where you want to tunnel protocols to or from two locally-connected UNIs on the *same* switch, although you still configure the **mac-rewrite** statement to specify the protocol being tunneled, the switch simply floods the protocol packets within the VLAN instead of rewriting the MAC address. You use the same configuration for both interfaces, and you don't need to use a loopback cable.

For example, the following commands configure two UNIs (**ge-0/0/0** and **ge-0/0/1**) in VLAN v20 for Q-in-Q tunneling on a switch, and the two ports on the switch exchange LACP and LLDP packets:

```
set vlans v20 vlan-id 20
set interfaces ge-0/0/0 unit 20 vlan-id 20
set interfaces ge-0/0/0 unit 20 family ethernet-switching vlan members v20
set interfaces ge-0/0/0 flexible-vlan-tagging
set interfaces ge-0/0/0 native-vlan-id 20
set interfaces ge-0/0/0 encapsulation extended-vlan-bridge
set interfaces ge-0/0/0 unit 20 input-vlan-map push
set interfaces ge-0/0/0 unit 20 output-vlan-map pop
set interfaces ge-0/0/1 unit 20 vlan-id 20
set interfaces ge-0/0/1 unit 20 family ethernet-switching vlan members v20
set interfaces ge-0/0/1 flexible-vlan-tagging
set interfaces ge-0/0/1 native-vlan-id 20
set interfaces ge-0/0/1 encapsulation extended-vlan-bridge
set interfaces ge-0/0/1 unit 20 input-vlan-map push
set interfaces ge-0/0/1 unit 20 output-vlan-map pop
set protocols layer2-control mac-rewrite interface ge-0/0/0 protocol lacp
set protocols layer2-control mac-rewrite interface ge-0/0/0 protocol lldp
set protocols layer2-control mac-rewrite interface ge-0/0/1 protocol lacp
set protocols layer2-control mac-rewrite interface ge-0/0/1 protocol lldp
set vlans v20 interface ge-0/0/0.20
set vlans v20 interface ge-0/0/1.20
```

2. To check the protocols configured for L2PT on an interface, enter the **show mac-rewrite interface** CLI command with the interface name.

   For example:

   ```
   user@device> show mac-rewrite interface ge-0/0/0
   ```

```
Interface       Protocols
 ge-0/0/0        LLDP STP
```

If you don't specify an interface name, the **show mac-rewrite interface** command displays all interfaces with L2PT configured.

For example:

```
user@switch> show mac-rewrite interface
```

```
Interface       Protocols
 ge-0/0/0        LACP LLDP
 ge-0/0/1        LACP LLDP
```

3. To detect and clear an interface configured with L2PT that appears to be blocked due to a MAC rewrite error, see "Clearing a MAC Rewrite Error on an Interface with Layer 2 Protocol Tunneling" on page 697.

SEE ALSO

Configuring VLAN Encapsulation | 291

Stacking a VLAN Tag | 367

Clearing a MAC Rewrite Error on an Interface with Layer 2 Protocol Tunneling

On devices with Layer 2 protocol tunneling (L2PT) configured, customer-facing ports should not receive packets with the L2PT MAC address as the destination address unless you have a network topology or configuration error. Under these conditions, when an interface with L2PT enabled receives an L2PT packet, the interface state becomes disabled due to a MAC rewrite error, and you must subsequently re-enable it to continue operation.

1. To check whether an interface with L2PT enabled has become disabled due to a MAC rewrite error condition, use the **show interfaces** operational command:

   user@switch> **show interfaces *interface-name***

   If the interface status includes **Disabled, Physical link is Down** or **Enabled, Physical link is Down** and the **MAC-REWRITE Error** field is **Detected**, then the device detected a MAC rewrite error that

contributed to the interface being down. When the device did not detect any MAC rewrite errors, the **MAC-REWRITE Error** field is **None**.

For example, the following output shows the device detected a MAC rewrite error on the given interface:

```
user@switch> show interfaces ge-0/0/2
Physical interface: ge-0/0/2, Disabled, Physical link is Down
  Interface index: 150, SNMP ifIndex: 531
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Speed: 1000mbps, BPDU
Error: None,
  Loop Detect PDU Error: None, Ethernet-Switching Error: None, Source filtering:
 Disabled
  Ethernet-Switching Error: None, MAC-REWRITE Error: Detected, Loopback: Disabled,

  Flow control: Enabled, Auto-negotiation: Enabled, Remote fault: Online, Media
 type: Fiber
  Device flags   : Present Running
```

2. On routers, QFX Series switches, and EX Series switches that use the Enhanced Layer 2 Software configuration style, you can clear a MAC rewrite error from the Junos CLI.

   To clear a MAC rewrite error from an interface that has L2PT enabled, use the **clear error mac-rewrite** operational command:

   ```
   user@switch> clear error mac-rewrite interface-name
   ```

Configuring Layer 2 Protocol Tunneling on EX Series Switches Without ELS Support

NOTE: This task applies only to switches that do not support the Enhanced Layer 2 Software (ELS) configuration style.

An EX Series switch can use Layer 2 protocol tunneling (L2PT) to send Layer 2 protocol data units (PDUs) across a service provider network and deliver them to EX Series switches at a remote location. This feature is useful when you have a network that includes remote sites that are connected across a service provider network and you want to run Layer 2 protocols on switches connected across the service provider network.

Tunneled Layer 2 PDUs do not normally arrive at high rate. If the tunneled Layer 2 PDUs do arrive at high rate, there might be a problem in the network. Typically, you would want to shut down the interface that is receiving a high rate of tunneled Layer 2 PDUs to isolate the problem. You can use the **shutdown-threshold** statement to do so. However, if you do not want to completely shut down the interface, you can use the **drop-threshold** statement to configure the switch to drop tunneled Layer 2 PDUs that exceed a certain threshold.

There are no default settings for **drop-threshold** and **shutdown-threshold**, so unless you explicitly configure these values, the switch doesn't enforce any thresholds. As a result, the switch tunnels all Layer 2 PDUs regardless of the speed at which they are received, although the number of packets tunneled per second might be limited by other factors.

You can specify a drop threshold value without specifying a shutdown threshold value, and you can specify a shutdown threshold value without specifying a drop threshold value. If you specify both threshold values, then the drop threshold value must be less than or equal to the shutdown threshold value. If the drop threshold value is greater than the shutdown threshold value and you try to commit the configuration, the commit will fail.

NOTE: You can't configure L2PT and VLAN translation with the **mapping** statement on the same switch.

NOTE: If the switch receives untagged Layer 2 control PDUs to be tunnelled, then you must configure the switch to map untagged (native) packets to an L2PT-enabled VLAN. Otherwise, the switch discards untagged Layer 2 control PDU packets. For more information, see "Understanding Q-in-Q Tunneling and VLAN Translation" on page 887 and "Configuring Q-in-Q Tunneling on EX Series Switches" on page 910.

To configure L2PT on an EX Series switch:

1. Because L2PT operates under the Q-in-Q tunneling configuration, you must enable Q-in-Q tunneling before you can configure L2PT. Enable Q-in-Q tunneling on VLAN customer-1:

   ```
   [edit]
   user@switch# set vlans customer-1 dot1q-tunneling
   ```

2. Enable L2PT for the Layer 2 protocol you want to tunnel, on the VLAN:

   - To enable L2PT for a specific protocol (here, STP):

     ```
     [edit]
     user@switch# set vlans customer-1 dot1q-tunneling layer2-protocol-tunneling stp
     ```

   - To enable L2PT for all supported protocols:

     ```
     [edit]
     user@switch# set vlans customer-1 dot1q-tunneling layer2-protocol-tunneling all
     ```

3. (Optional) Configure the drop threshold:

   > **NOTE:** If you also configure the shutdown threshold, ensure that you configure the drop threshold value to be less than or equal to the shutdown threshold value. If the drop threshold value is greater than the shutdown threshold value and you to try to commit the configuration changes, the commit will fail.

   ```
   [edit]
   user@switch# set vlans customer-1 dot1q-tunneling layer2–protocol-tunneling stp drop-threshold
   50
   ```

4. (Optional) Configure the shutdown threshold:

   > **NOTE:** If you also configure the drop threshold, ensure that you configure the shutdown threshold value to be greater than or equal to the drop threshold value. If the shutdown threshold value is less than the drop threshold value and you to try to commit the configuration changes, the commit will fail.

   ```
   [edit]
   ```

```
user@switch# set vlans customer-1 dot1q-tunneling layer2–protocol-tunneling stp
shutdown-threshold 100
```

> **NOTE:** After an interface becomes disabled, you must explicitly reenable it using the **clear ethernet-switching layer2-protocol-tunneling error** command. Otherwise, the interface remains disabled.

Example: Configuring Layer 2 Protocol Tunneling on EX Series Switches Without ELS Support

IN THIS SECTION

> **NOTE:** This example uses Junos OS for EX Series switches that does not support the Enhanced Layer 2 Software (ELS) configuration style.

Layer 2 protocol tunneling (L2PT) enables service providers to send Layer 2 protocol data units (PDUs) across the provider's cloud and deliver them to EX Series switches that are not part of the local broadcast domain. This feature is useful when you want to run Layer 2 protocols on a network that includes switches located at remote sites that are connected across a service provider network.

> **NOTE:** You can't configure both L2PT and VLAN translation configured with the **mapping** statement on the same VLAN. However, you can configure L2PT on one VLAN on a switch and VLAN translation on a different VLAN that doesn't have L2PT configured.

This example describes how to configure L2PT:

Requirements

This example uses the following hardware and software components:

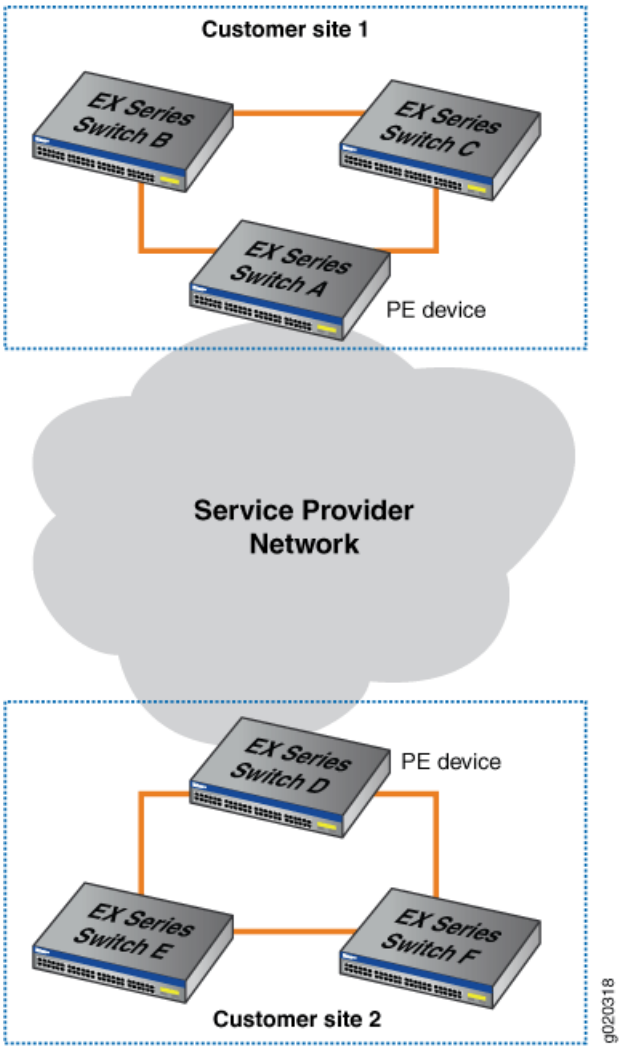
- Six EX Series switches, with three each at two customer sites, with one of the switches at each site designated as the provider edge (PE) device
- Junos OS Release 10.0 or later for EX Series switches

Overview and Topology

L2PT enables you to send Layer 2 PDUs across a service provider network and deliver them to EX Series switches that are not part of the local broadcast domain.

Figure 37 on page 703 shows a customer network that includes two sites that are connected across a service provider network. Site 1 contains three switches connected in a Layer 2 network, with Switch A designated as a provider edge (PE) device in the service provider network. Site 2 contains a Layer 2 network with a similar topology to that of Site 1, with Switch D designated as a PE device.

Figure 37: L2PT Topology



When you enable L2PT on a VLAN, you also must enable Q-in-Q tunneling. Q-in-Q tunneling ensures that Switches A, B, C, D, E, and F are part of the same broadcast domain.

This example uses STP as the Layer 2 protocol being tunneled, but you could substitute any of the supported protocols for STP. You can also use the **all** keyword to enable L2PT for all supported Layer 2 protocols.

Tunneled Layer 2 PDUs do not normally arrive at a high rate. If the tunneled Layer 2 PDUs do arrive at a high rate, you might have a problem in the network. Typically, you would want to shut down the interface that is receiving a high rate of tunneled Layer 2 PDUs so that the problem can be isolated. Alternately, if you do not want to completely shut down the interface, you can configure the switch to drop tunneled Layer 2 PDUs that exceed a certain threshold.

The **drop-theshold** configuration statement enables you to specify the maximum number of Layer 2 PDUs of the specified protocol that can be received per second on the interfaces in a specified VLAN before the switch begins dropping the Layer 2 PDUs. The drop threshold must be less than or equal to the shutdown

threshold. If the drop threshold is greater than the shutdown threshold and you try to commit the configuration, the commit will fail.

The **shutdown-threshold** configuration statement enables you to specify the maximum number of Layer 2 PDUs of the specified protocol that can be received per second on the interfaces in a specified VLAN before the specified interface is disabled. The shutdown threshold must be greater than or equal to the drop threshold. You can specify a drop threshold without specifying a shutdown threshold, and you can specify a shutdown threshold without specifying a drop threshold. If you do not specify these thresholds, then no thresholds are enforced. As a result, the switch tunnels all Layer 2 PDUs regardless of the speed at which they are received, although the number of packets tunneled per second might be limited by other factors.

In this example, we will configure both a drop threshold and a shutdown threshold to show how this is done.

If L2PT-encapsulated packets are received on an access interface, the switch reacts as it does when there is a loop between the service provider network and the customer network and shuts down (disables) the access interface.

Once an interface is disabled, you must explicitly reenable it using the **clear ethernet-switching layer2-protocol-tunneling error** command or else the interface will remain disabled.

Configuration

To configure L2PT, perform these tasks:

CLI Quick Configuration

To quickly configure L2PT, copy the following commands and paste them into the switch terminal window of each PE device (in Figure 37 on page 703, Switch A and Switch D are the PE devices):

```
[edit]
set vlans customer-1 dot1q-tunneling
set vlans customer-1 dot1q-tunneling layer2-protocol-tunneling stp
set vlans customer-1 dot1q-tunneling layer2-protocol-tunneling stp drop-threshold 50
set vlans customer-1 dot1q-tunneling layer2-protocol-tunneling stp shutdown-threshold 100
```

Step-by-Step Procedure

To configure L2PT, perform these tasks on each PE device (in Figure 37 on page 703, Switch A and Switch D are the PE devices):

1. Enable Q-in-Q tunneling on VLAN **customer-1**:

   ```
   [edit]
   user@switch# set vlans customer-1 dot1q-tunneling
   ```

2. Enable L2PT for STP on VLAN **customer-1**:

   ```
   [edit]
   user@switch# set vlans customer-1 dot1q-tunneling layer2-protocol-tunneling stp
   ```

3. Configure the drop threshold as 50:

   ```
   [edit]
   user@switch# set vlans customer-1 dot1q-tunneling layer2–protocol-tunneling stp drop-threshold
   50
   ```

4. Configure the shutdown threshold as 100:

   ```
   [edit]
   user@switch# set vlans customer-1 dot1q-tunneling layer2–protocol-tunneling stp
   shutdown-threshold 100
   ```

Results

Check the results of the configuration:

```
[edit]
user@switch# show vlans customer-1 dot1q-tunneling
layer2-protocol-tunneling {
    stp {
        drop-threshold 50;
        shutdown-threshold 100;
    }
}
```

Verification

IN THIS SECTION

- Verify That L2PT Is Working Correctly | 706

To verify that L2PT is working correctly, perform this task:

Verify That L2PT Is Working Correctly

Purpose

Verify that Q-in-Q tunneling and L2PT are enabled.

Action

Check to see that Q-in-Q tunneling and L2PT are enabled on each PE device (Switch A and Switch D are the PE devices):

```
user@switchA> show vlans extensive customer-1
```

```
VLAN: customer–1, Created at: Thu Jun 25 05:07:38 2009
802.1Q Tag: 100, Internal index: 4, Admin State: Enabled, Origin: Static
Dot1q Tunneling status: Enabled
Layer2 Protocol Tunneling status: Enabled
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 0 (Active = 0), Untagged  3 (Active = 0)
      ge-0/0/7.0, untagged, access
      ge-0/0/8.0, untagged, access
      ge-0/0/9.0, untagged, access
```

Check to see that L2PT is tunneling STP on VLAN **customer-1** and that **drop-threshold** and **shutdown-threshold** have been configured:

```
user@switchA> show ethernet-switching layer2-protocol-tunneling vlan customer-1
```

```
Layer2 Protocol Tunneling VLAN information:
VLAN          Protocol      Drop       Shutdown
                            Threshold  Threshold
customer–1    stp           50         100
```

Check the state of the interfaces on which L2PT has been enabled, including what kind of operation (encapsulation or decapsulation) they are performing:

```
user@switchA> show ethernet-switching layer2-protocol-tunneling interface
```

```
Layer2 Protocol Tunneling information:
Interface     Operation       State      Description
ge-0/0/0.0    Encapsulation   Shutdown   Shutdown threshold exceeded
```

```
ge-0/0/1.0    Decapsulation   Shutdown   Loop detected
ge-0/0/2.0    Decapsulation   Active
```

Meaning

The **show vlans extensive customer-1** command shows that Q-in-Q tunneling and L2PT have been enabled. The **show ethernet-switching layer2-protocol-tunneling vlan customer-1** command shows that L2PT is tunneling STP on VLAN **customer-1**,the drop threshold is set to 50, and the shutdown threshold is set to 100. The **show ethernet-switching layer2-protocol-tunneling interface** command shows the type of operation being performed on each interface, the state of each interface and, if the state is **Shutdown**, the reason why the interface is shut down.

22

CHAPTER

Configuring Virtual Routing Instances

Virtual Routing Instances

IN THIS SECTION

Understanding Virtual Routing Instances on EX Series Switches

Virtual routing instances allow administrators to divide a Juniper Networks EX Series Ethernet Switch into multiple independent virtual routers, each with its own routing table. Splitting a device into many virtual routing instances isolates traffic traveling across the network without requiring multiple devices to segment the network.

You can use virtual routing instances to isolate customer traffic on your network and to bind customer-specific instances to customer-owned interfaces.

Virtual routing and forwarding (VRF) is often used in conjunction with Layer 3 subinterfaces, allowing traffic on a single physical interface to be differentiated and associated with multiple virtual routers. Each logical Layer 3 subinterface can belong to only one routing instance.

EX Series switches support IPv4 and IPv6 unicast and multicast VRF traffic. See Feature Explorer for details on VRF support by switch per Junos OS release.

SEE ALSO

Understanding Layer 3 Subinterfaces

Configuring Virtual Routing Instances on EX Series Switches

Use virtual routing and forwarding (VRF) to divide an EX Series switch into multiple virtual routing instances. VRF allows you to isolate traffic traversing the network without using multiple devices to segment your network. VRF is supported on all Layer 3 interfaces.

Before you begin, make sure to set up your VLANs. See "Configuring VLANs for EX Series Switches" on page 183, *Configuring VLANs for EX Series Switches with ELS Support (CLI Procedure)*, or *Configuring VLANs for EX Series Switches (J-Web Procedure)*.

To configure virtual routing instances:

1. Create a routing instance:

   [edit routing-instances]`user@switch#` **set *routing-instance-name* instance-type virtual-router**

   > **NOTE:** EX Series switches only support the virtual-router instance type.

2. Bind each routing instance to the corresponding physical interfaces:

   [edit routing-instances]`user@switch#` **set *routing-instance-name* interface *interface-name.logical-unit-number***

3. Create the logical interfaces that are bound to the routing instance.
   - To create a logical interface with an IPv4 address:

     [edit interfaces]`user@switch#` **set *interface-name* unit *logical-unit-number* family inet address *ip-address***

   - To create a logical interface with an IPv6 address:

     [edit interfaces]`user@switch#` **set *interface-name* unit *logical-unit-number* family inet6 address *ipv6–address***

     > **NOTE:** Do not create a logical interface using the **family ethernet-switching** option in this step. Binding an interface using the **family ethernet-switching** option to a routing instance can cause the interface to shutdown.

4. Enable VLAN tagging on each physical interface that was bound to the routing instance:

```
[edit interfaces]user@switch# set interface-name vlan-tagging
```

Example: Using Virtual Routing Instances to Route Among VLANs on EX Series Switches

IN THIS SECTION

Virtual routing instances allow each EX Series switch to have multiple routing tables on a device. With virtual routing instances, you can segment your network to isolate traffic without setting up additional devices.

This example describes how to create virtual routing instances:

Requirements

This example uses the following hardware and software components:

- One EX Series switch
- Junos OS Release 9.2 or later for EX Series switches

Before you create the virtual routing instances, make sure you have:

- Configured the necessary VLANs. See "Configuring VLANs for EX Series Switches" on page 183, *Configuring VLANs for EX Series Switches with ELS Support (CLI Procedure)*, or *Configuring VLANs for EX Series Switches (J-Web Procedure)*.

Overview and Topology

In a large office, you may need multiple VLANs to properly manage your traffic. This configuration example shows a simple topology wherein a LAN is segmented into two VLANs, each of which is associated with

an interface and a virtual routing instance, on the EX Series switch. This example also shows how to use policy statements to import routes from one of the virtual routing instances to the other.

Configuration

CLI Quick Configuration

To quickly create and configure virtual routing instances, copy the following commands and paste them into the switch terminal window:

```
[edit]
set interfaces ge-0/0/3 vlan-tagging
set interfaces ge-0/0/3 unit 0 vlan-id 1030 family inet address 10.1.1.1/24
set interfaces ge-0/0/3 unit 1 vlan-id 1031 family inet address 10.1.1.1/24
set interfaces ge-0/0/1 unit 0 family inet address 10.11.1.1/24
set interfaces ge-0/0/2 unit 0 family inet address 10.12.1.1/24
set routing-instances r1 instance-type virtual-router
set routing-instances r1 interface ge-0/0/1.0
set routing-instances r1 interface ge-0/0/3.0
set routing-instances r1 routing-options instance-import import-from-r2
set routing-instances r2 instance-type virtual-router
set routing-instances r2 interface ge-0/0/2.0
set routing-instances r2 interface ge-0/0/3.1
set routing-instances r2 routing-options instance-import import-from-r1
set policy-options policy-statement import-from-r1 term 1 from instance r1
set policy-options policy-statement import-from-r1 term 1 then accept
set policy-options policy-statement import-from-r2 term 1 from instance r2
set policy-options policy-statement import-from-r2 term 1 then accept
```

Step-by-Step Procedure

To configure virtual routing instances:

1. Create a VLAN-tagged interface:

```
[edit]user@switch# set interfaces ge-0/0/3 vlan-tagging
```

2. Create one or more subinterfaces on the interfaces to be included in each routing instance:

```
[edit]user@switch# set interfaces ge-0/0/3 unit 0 vlan-id 1030 family inet address 10.1.1.1/24
user@switch# set interfaces ge-0/0/3 unit 1 vlan-id 1031 family inet address 10.1.1.1/24
user@switch# set interfaces ge-0/0/1 unit 0 family inet address 10.11.1.1/24
user@switch# set interfaces ge-0/0/2 unit 0 family inet address 10.12.1.1/24
```

3. Create two virtual routing instances:

```
[edit]user@switch# set routing-instances r1 instance-type virtual-router
user@switch# set routing-instances r2 instance-type virtual-router
```

4. Set the interfaces for the virtual routing instances:

```
[edit]user@switch# set routing-instances r1 interface ge-0/0/1.0
user@switch# set routing-instances r1 interface ge-0/0/3.0
user@switch# set routing-instances r2 interface ge-0/0/2.0
user@switch# set routing-instances r2 interface ge-0/0/3.1
```

5. Apply a policy to routes being imported into each of the virtual routing instances:

```
[edit]user@switch# set routing-instances r1 routing-options instance-import import-from-r2
user@switch# set routing-instances r2 routing-options instance-import import-from-r1
```

6. Create a policy that imports routes from routing instances r1 to r2 and another policy that imports routes from routing instances r2 to r1:

```
[edit]user@switch# set policy-options policy-statement import-from-r1 term 1 from instance
r1
user@switch# set policy-options policy-statement import-from-r1 term 1 then accept
user@switch# set policy-options policy-statement import-from-r2 term 1 from instance r2
user@switch# set policy-options policy-statement import-from-r2 term 1 then accept
```

Results

Check the results of the configuration:

```
user@switch> show configuration
```

```
interfaces {
            ge-0/0/1 {
                unit 0 {
                    family inet {
                        address 10.11.1.1/24;
                    }
                }
            }
            ge-0/0/2 {
                unit 0 {
                    family inet {
                        address 10.12.1.1/24;
                    }
                }
            }
            ge-1/0/3 {
                vlan-tagging;
                unit 0 {
                    vlan-id 1030;
                    family inet {
                        address 10.1.1.1/24;
                    }
                }
                unit 1 {
                    vlan-id 1031;
                    family inet {
                        address 10.1.1.1/24;
                    }
                }
            }
        }
        policy-options {
            policy-statement import-from-r1 {
                term 1 {
                    from instance r1;
                    then accept;
                }
            }
            policy-statement import-from-r2 {
                term 1 {
                    from instance r2;
                    then accept;
                }
            }
```

```
        }
        routing-instances {
            r1 {
                instance-type virtual-router;
                interface ge-0/0/1.0;
                interface ge-0/0/3.0;
                routing-options {
                    instance-import import-from-r2;
                }
            }
            r2 {
                instance-type virtual-router;
                interface ge-0/0/2.0;
                interface ge-0/0/3.1;
                routing-options {
                    instance-import import-from-r1;
                }
            }
        }
    }
}
```

Verification

IN THIS SECTION

- Verifying That the Routing Instances Were Created | **715**

To confirm that the configuration is working properly, perform these tasks:

Verifying That the Routing Instances Were Created

Purpose

Verify that the virtual routing instances were properly created on the switch.

Action

Use the **show route instance** command:

```
user@switch> show route instance
```

```
Instance             Type
         Primary RIB                                    Active/holddown/hidden
master               forwarding
         inet.0                                         6/0/0
         iso.0                                          1/0/0
         inet6.0                                        2/0/0

...

r1                   virtual-router
         r1.inet.0                                      7/0/0

r2                   virtual-router
         r2.inet.0                                      7/0/0
```

Meaning

Each routing instance created is displayed, along with its type, information about whether it is active or not, and its primary routing table.

Verifying That Virtual Routing Instances Are Working on EX Series Switches

Purpose

After creating a virtual routing instance, make sure it is set up properly.

Action

1. Use the **show route instance** command to list all of the routing instances and their properties:

   `user@switch>` **show route instance**

```
Instance             Type
         Primary RIB                                    Active/holddown/hidden
master               forwarding
         inet.0                                          3/0/0

__juniper_private1__ forwarding
         __juniper_private1__.inet.0                     1/0/3

__juniper_private2__ forwarding

instance1            forwarding
```

```
r1                  virtual-router
         r1.inet.0                                        1/0/0

r2                  virtual-router
         r2.inet.0                                        1/0/0
```

2. Use the **show route forwarding-table** command to view the forwarding table information for each routing instance:

   `user@switch>` **show route forwarding-table**

```
Routing table: r1.inet
Internet:
Destination        Type RtRef Next hop           Type Index NhRef Netif
default            perm     0                    rjct   539     2
0.0.0.0/32         perm     0                    dscd   537     1
10.1.1.0/24       ifdn      0                    rslv   579     1 ge-0/0/3.0
10.1.1.0/32       iddn      0 10.1.1.0           recv   577     1 ge-0/0/3.0
10.1.1.1/32       user      0                    rjct   539     2
10.1.1.1/32       intf      0 10.1.1.1           locl   578     2
10.1.1.1/32       iddn      0 10.1.1.1           locl   578     2
10.1.1.255/32     iddn      0 10.1.1.255         bcst   576     1 ge-0/0/3.0
233.252.0.1/32       perm     0 233.252.0.1          mcst   534     1
255.255.255.255/32 perm     0                    bcst   535     1
```

Meaning

The output confirms that the virtual routing instances are created and the links are up and displays the routing table information.

CHAPTER 23

Configuring Layer 3 Logical Interfaces

Layer 3 Logical Interfaces

IN THIS SECTION

Understanding Layer 3 Logical Interfaces

A Layer 3 logical interface is a logical division of a physical interface that operates at the network level and therefore can receive and forward 802.1Q VLAN tags. You can use Layer 3 logical interfaces to route traffic among multiple VLANs along a single trunk line that connects a Juniper Networks switch to a Layer 2 switch. Only one physical connection is required between the switches. .

> **NOTE:** You can also use Layer 3 logical interfaces to provide alternative gateway addresses for smart DHCP relay. The logical tunnel (lt) and virtual loopback tunnel (vt) interfaces are not supported in logical interfaces.

To create Layer 3 logical interfaces on a switch, enable VLAN tagging, partition the physical interface into logical partitions, and bind the VLAN ID to the logical interface.

We recommend that you use the VLAN ID as the logical interface number when you configure the logical interface. QFX Series and EX4600 switches support a maximum of 4089 VLANs, which includes the default VLAN. You can, however, assign a VLAN ID in the range of 1 to 4094, but five of these VLAN IDs are reserved for internal use.

VLAN tagging places the VLAN ID in the frame header, allowing each physical interface to handle multiple VLANs. When you configure multiple VLANs on an interface, you must also enable tagging on that interface. Junos OS on switches supports a subset of the 802.1Q standard for receiving and forwarding routed or bridged Ethernet frames with single VLAN tags and running Virtual Router Redundancy Protocol (VRRP) over 802.1Q-tagged interfaces.

Configuring a Layer 3 Logical Interface

Devices use Layer 3 logical interfaces to divide a physical interface into multiple logical interfaces, each corresponding to a VLAN. Layer 3 logical interfaces route traffic between subnets.

To configure Layer 3 logical interfaces, enable VLAN tagging and partition one or more physical ports into multiple logical interfaces, each corresponding to a VLAN ID.

Before you begin, make sure you set up your VLANs. See "Configuring VLANs on Switches" on page 182.

To configure Layer 3 logical interfaces:

1. Enable VLAN tagging:

   ```
   [edit interfaces interface-name]
   user@switch# set vlan-tagging
   ```

2. Bind each VLAN ID to a logical interface:

   ```
   [edit interfaces interface-name]
   user@switch# set unit logical-unit-number vlan-id vlan-id-number
   ```

Verifying That Layer 3 Logical Interfaces Are Working

Purpose

After configuring Layer 3 logical interfaces, verify that they are set up properly and transmitting data.

Action

1. To determine if you have successfully created the logical interfaces and the links are up:

   ```
   [edit interfaces]
   user@switch> show interfaces interface-name terse
   ```

   ```
   Interface               Admin Link Proto    Local                 Remote
   ge-0/0/0                up    up
   ge-0/0/0.0              up    up   inet     10.0.0.1/8
   ge-0/0/0.1              up    up   inet     10.0.0.2/8
   ge-0/0/0.2              up    up   inet     10.0.0.3/8
   ge-0/0/0.3              up    up   inet     10.0.0.4/8
   ge-0/0/0.4              up    up   inet     10.0.0.5/8
   ge-0/0/0.32767          up    up
   ```

2. Use the **ping** command from a device on one subnet to an address on another subnet to determine if packets were transmitted correctly on the logical interface VLANs:

   `user@switch>` **ping** ***ip-address***

```
PING 10.1.1.1 (1.1.1.1): 56 data bytes
64 bytes from 10.1.1.1: icmp_seq=0 ttl=64 time=0.157 ms
64 bytes from 10.1.1.1: icmp_seq=1 ttl=64 time=0.238 ms
64 bytes from 10.1.1.1: icmp_seq=2 ttl=64 time=0.255 ms
64 bytes from 10.1.1.1: icmp_seq=3 ttl=64 time=0.128 ms
--- 10.1.1.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
```

Meaning

The output confirms that the logical interfaces have been created and the links are up.

CHAPTER 24

Configuring Routed VLAN Interfaces

Routed VLAN Interfaces

IN THIS SECTION

Configuring a Routed VLAN Interface in a Private VLAN on an EX Series Switch

Starting with Junos OS Release 14.1X53-D10, you can configure a routed VLAN interface (RVI) for a private VLAN (PVLAN) on an EX8200 switch or EX8200 Virtual Chassis. Instead of a router connected to a promiscuous port routing Layer 3 traffic between isolated and community members, you can alternatively use an RVI.

To set up routing within a PVLAN, one RVI must be configured for the primary VLAN on one EX8200 switch or EX8200 Virtual Chassis in the PVLAN domain. This RVI serves the entire PVLAN domain regardless of whether the domain consists of one or more switches. After you configure the RVI, Layer 3 packets received by the secondary VLAN interfaces are mapped to and routed by the RVI.

When setting up the RVI, you must also enable proxy Address Resolution Protocol (ARP) so that the RVI can handle ARP requests received by the secondary VLAN interfaces.

This topic describes how to configure an RVI for a PVLAN.

Before you begin, configure the PVLAN as described in "Creating a Private VLAN on a Single EX Series Switch (CLI Procedure)" on page 477 or "Creating a Private VLAN Spanning Multiple EX Series Switches (CLI Procedure)" on page 484.

To configure an RVI for a PVLAN:

1. Create a logical Layer 3 RVI on a subnet for the primary VLAN's broadcast domain:

   [edit interfaces]

   `user@switch#` **set vlan unit *logical-unit-number* family inet address *inet-address***

2. Enable unrestricted proxy ARP on the RVI:

   [edit interfaces]

```
user@switch# set vlan unit logical-unit-number proxy-arp unrestricted
```

3. Disable sending protocol redirect messages on the RVI:

   [edit interfaces]

   ```
   user@switch# set vlan unit logical-unit-number family inet no-redirects
   ```

4. Link the primary VLAN to the RVI:

   [edit vlans]

   ```
   user@switch# set vlan-name l3-interface vlan.logical-unit-number
   ```

   The value of *logical-unit-number* is the same value that you supplied for *logical-unit-number* in the previous steps.

Verifying Routed VLAN Interface Status and Statistics on EX Series Switches

Purpose

Determine status information and traffic statistics for routed VLAN interfaces (RVIs) by using the following commands:

Action

Display RVI interfaces and their current states:

```
user@switch> show interfaces vlan terse
```

```
Interface               Admin Link Proto    Local                 Remote
vlan                    up    up
vlan.111                up    up   inet     111.111.111.1/24
```

Display Layer 2 VLANs, including any tags assigned to the VLANs and the interfaces associated with the VLANs:

```
user@switch> show vlans
```

```
Name           Tag     Interfaces
default
                       None
employee-vlan  20
                       ge-1/0/0.0, ge-1/0/1.0, ge-1/0/2.0
```

```
marketing       40
                        ge-1/0/10.0, ge-1/0/20.0, ge-1/0/30.0
support         111
                        ge-0/0/18.0
mgmt
                        bme0.32769, bme0.32771*
```

Display Ethernet switching table entries for the VLAN that is attached to the RVI:

```
user@switch> show ethernet-switching table
```

```
Ethernet-switching table: 1 entries, 0 learned
  VLAN              MAC address       Type         Age Interfaces
  support           00:19:e2:50:95:a0 Static         - Router
```

Display an RVI's ingress-counting statistics with either the **show interfaces vlan detail** command or the **show interfaces vlan extensive** command. Ingress counting is displayed as **Input bytes** and **Input packets** under **Transit Statistics**.

```
user@switch> show interfaces vlan.100 detail
```

```
Logical interface vlan.100 (Index 65) (SNMP ifIndex 503) (HW Token 100) (Generation
 131)
Flags: SNMP-Traps 0x4000  Encapsulation: ENET2
Traffic statistics:
 Input bytes:    17516756
 Output bytes:   411764
 Input packets:  271745
 Output packets: 8256
Local statistics:
 Input bytes:    3240
 Output bytes:   411764
 Input packets:  54
 Output packets: 8256
Transit statistics:
 Input bytes:    17513516   0 bps
 Output bytes:   0    0 bps
 Input packets:  271745  0 pps
 Output packets: 0    0 pps
Protocol inet, Generation: 148, Route table: 0
```

```
Flags: None
Addresses, Flags: iS-Preferred  Is-Primary
 Destination: 50.1.1/24, Local: 50.1.1.1, Broadcast: 50.1.1.255, Generation: 136
```

Meaning

- **show interfaces vlan** displays a list of interfaces, including RVI interfaces, and their current states (up, down).
- **show vlans** displays a list of VLANs, including any tags assigned to the VLANs and the interfaces associated with the VLANs.
- **show ethernet-switching table** displays the Ethernet switching table entries, including VLANs attached to the RVI.
- **show interfaces vlan detail** displays RVI ingress counting as Input Bytes and Input Packets under Transit Statistics.

Release History Table

Release	Description
14.1X53-D10	Starting with Junos OS Release 14.1X53-D10, you can configure a routed VLAN interface (RVI) for a private VLAN (PVLAN) on an EX8200 switch or EX8200 Virtual Chassis.

25

CHAPTER

Configuring Integrated Routing and Bridging

Integrated Routing and Bridging

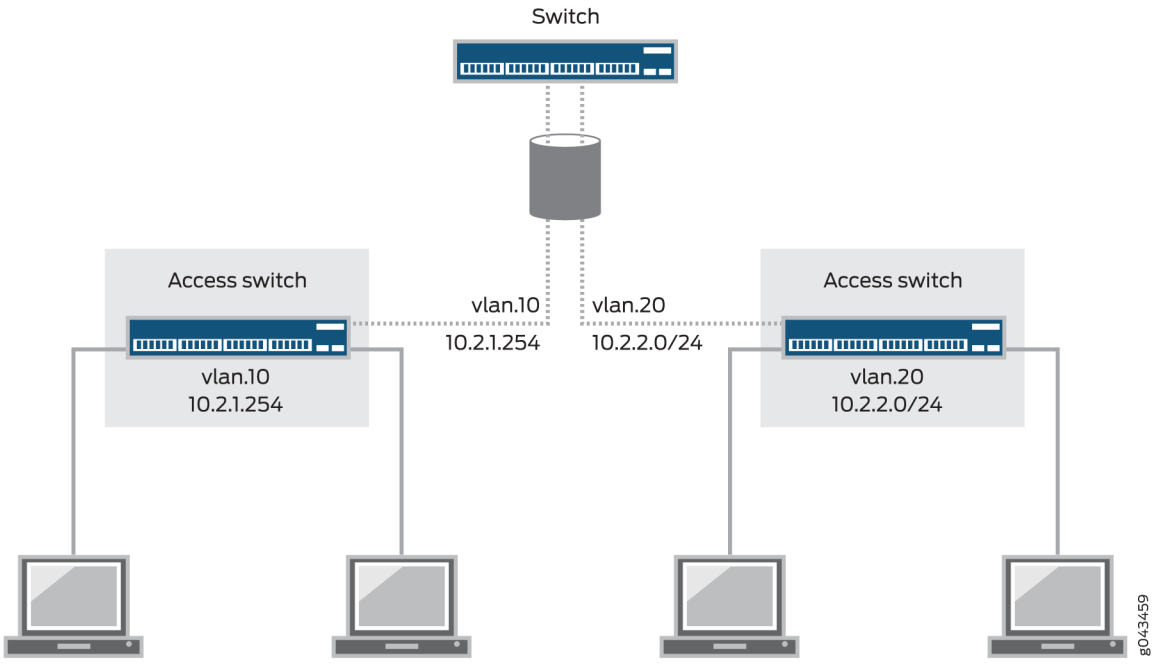
IN THIS SECTION

Understanding Integrated Routing and Bridging

To segment traffic on a LAN into separate broadcast domains, you create separate virtual LANs (VLANs). VLANs limit the amount of traffic flowing across the entire LAN, reducing the possible number of collisions and packet retransmissions within the LAN. For example, you might want to create a VLAN that includes the employees in a department and the resources that they use often, such as printers, servers, and so on.

Figure 38 on page 729 illustrates a switch routing VLAN traffic between two access layer switches using one of these interfaces.

Figure 38: An IRB Interface or RVI on a Switch Providing Routing Between Two Access Switches



Of course, you also want to allow these employees to communicate with people and resources in other VLANs. To forward packets between VLANs, you normally need a router that connects the VLANs. However, you can accomplish this forwarding on a switch without using a router by configuring an integrated routing and bridging (IRB) interface. (These interfaces are also called routed VLAN interfaces, or RVIs). Using this approach reduces complexity and avoids the costs associated with purchasing, installing, managing, powering, and cooling another device.

An IRB is a special type of Layer 3 virtual interface named **vlan**. Like normal Layer 3 interfaces, the **vlan** interface needs a logical unit number with an IP address. In fact, to be useful an IRB needs at least two logical units and two IP addresses—you must create units with addresses in each of the subnets associated with the VLANs between which you want traffic to be routed. That is, if you have two VLANs (for example, VLAN **red** and VLAN **blue**) with corresponding subnets, your IRB must have a logical unit with an address in the subnet for **red** and a logical unit with an address in the subnet for **blue**. The switch automatically creates direct routes to these subnets and uses these routes to forward traffic between VLANs. Packets arriving on a Layer 2 interface that are destined for the device's MAC address are classified as Layer 3 traffic while packets that are not destined for the device's MAC address are classified as Layer 2 traffic. Packets destined for the device's MAC address are sent to the IRB interface. Packets from the device's routing engine are sent out the IRB interface.

NOTE: If you specify a VLAN identifier list in the VLAN configuration, you cannot configure an IRB interface for the VLAN.

NOTE: If you are using a version of Junos OS that supports Enhanced Layer 2 Software (ELS), you can also create a Layer 3 virtual interface named **irb** instead of **vlan**—that is, both statements are supported by ELS

IRB interfaces supporting the Enhanced Layer 2 Software (ELS) configuration style and RVIs that support non-ELS switches provide the same functionality. Where the functionality for both features is the same, this topic uses the term *these interfaces* to refer collectively to both IRB interfaces and RVIs. Where differences exist between the two features, this topic calls out the IRB interfaces and RVIs separately.

Table 108 on page 730 shows values you might use when configuring an IRB:

Table 108: Sample IRB Values

Property	Settings
VLAN names and tags (IDs)	**blue**, ID **100** **red**, ID **200**
Subnets associated with VLANs	**blue**: **192.0.2.0/25** (addresses **192.0.2.1** through **192.0.2.126**) **red**: **192.0.2.128/25** (addresses **192.0.2.129** through **192.0.2.254**)
IRB name	interface **irb**
IRB units and addresses	logical unit 100: **192.0.2.1/25** logical unit 200: **192.0.2.129/25**

For the sake of consistency and to avoid confusion, Table 108 on page 730 shows IRB logical unit numbers that match the IDs of the corresponding VLANs. However, you do not have to assign logical unit numbers that match the VLAN IDs—you can use any values for the units. To bind the logical units of the IRB to the appropriate VLANs, you use the **l3-interface** statement.

Because IRBs operate at Layer 3, you can use Layer 3 services such as firewall filters or CoS rewriting with them.

Table 109 on page 730 shows the number of IRBs/RVIs that each QFX platform supports.

Table 109: Number of Supported IRBs/RVIs by Platform

Platform	Number of Supported IRBs/RVIs
QFX3500	1200

Table 109: Number of Supported IRBs/RVIs by Platform *(continued)*

QFX3000-G	1024
QFX3000-M	1024

IRB Interfaces on SRX Series Devices

On SRX1400, SRX1500, SRX3400, SRX3600, SRX4100, SRX4200, SRX4600, SRX5600, and SRX5800 devices, Juniper supports an IRB interface that allows you to terminate management connections in transparent mode. However, you cannot route traffic on that interface or terminate IPsec VPNs. (Platform support depends on the Junos OS release in your installation.)

> **NOTE:** You can configure only one IRB logical interface for each VLAN.

On SRX300, SRX320, SRX340, SRX345 devices, and SRX550M on the IRB interface, the following features are not supported:

- IS-IS (family ISO)
- Encapsulations (Ether CCC, VLAN CCC, VPLS, PPPoE, and so on) on VLAN interfaces
- CLNS
- DVMRP
- VLAN interface MAC change
- G-ARP
- Change VLAN-Id for VLAN interface

> **NOTE:** Starting with Junos OS Release 15.1X49-D60 and Junos OS Release 17.3R1, interface statistics are supported on the IRB logical interface for SRX300, SRX320, SRX340, SRX345, and SRX550M devices.
>
> To verify the IRB logical interface statistics, enter the **show interfaces irb.<index> extensive** and **show interfaces irb.<index>statistics** commands.

When Should I Use an IRB Interface or RVI?

Configure an IRB interface or an RVI for a VLAN if you need to:

- Allow traffic to be routed between VLANs.
- Provide Layer 3 IP connectivity to the switch.
- Monitor individual VLANs for billing purposes. Service providers often need to monitor traffic for this purpose, but this capability can be useful for enterprises where various groups share the cost of the network.

How Does an IRB Interface or RVI Work?

For an IRB interface, the switch provides the name irb, and for an RVI, the switch provides the name vlan. Like all Layer 3 interfaces, these interfaces require a logical unit number with an IP address assigned to it. In fact, to be useful, the implementation of these interfaces in an enterprise with multiple VLANs requires at least two logical units and two IP addresses—you must create units with addresses in each of the subnets associated with the VLANs between which you want traffic to be routed. That is, if you have two VLANs (for example, VLAN **red** and VLAN **blue**) with corresponding subnets, your interfaces must have a logical unit with an address in the subnet for **red** and a logical unit with an address in the subnet for **blue**. The switch automatically creates direct routes to these subnets and uses these routes to forward traffic between VLANs.

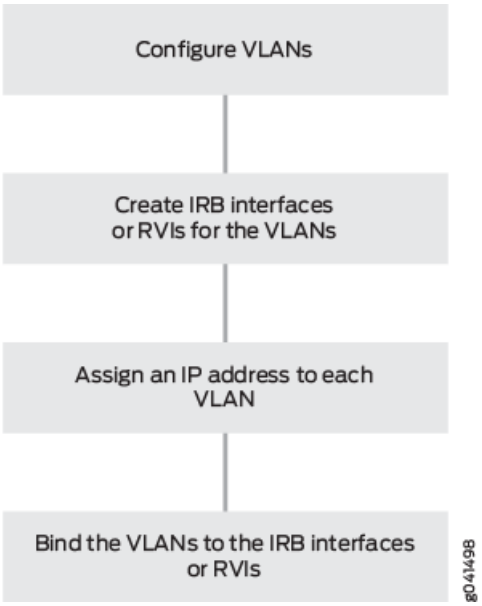
The interface on the switch detects both MAC addresses and IP addresses, then routes data to other Layer 3 interfaces on routers or other switches. These interfaces detect both IPv4 and IPv6 unicast and multicast virtual routing and forwarding (VRF) traffic. Each logical interface can belong to only one routing instance and is further subdivided into logical interfaces, each with a logical interface number appended as a suffix to the names irb and vlan—for example, irb.10 and vlan.10.

Creating an IRB Interface or RVI

You create an IRB logical interface in a similar manner as a Layer 3 interface, but the IRB interface does not support traffic forwarding or routing. The IRB interface cannot be assigned to a security zone; however, you can configure certain services on a per-zone basis to allow host-inbound traffic for management of the device. This allows you to control the type of traffic that can reach the device from interfaces bound to a specific zone.

There are four basic steps in creating an IRB interface or RVI as shown in Figure 39 on page 733.

Figure 39: Creating an IRB Interface or RVI



The following explanations correspond to the four steps for creating a VLAN, as depicted in Figure 39 on page 733.

- Configure VLANs—Virtual LANs are groups of hosts that communicate as if they were attached to the same broadcast stream. VLANs are created with software and do not require a physical router to forward traffic. VLANs are Layer 2 constructs.
- Create IRB interfaces or RVIs for the VLANs—The switch's IRB interfaces and RVIs use Layer 3 logical interfaces (unlike routers, which can use either physical or logical interfaces).
- Assign an IP address to each VLAN—An IRB interface or RVI cannot be activated unless it is associated with a physical interface.
- Bind the VLANs to the logical interfaces—There is a one-to-one mapping between a VLAN and an IRB interface or RVI, which means that only one of these interfaces can be mapped to a VLAN.

For specific instructions for creating an IRB interface, see "Configuring Integrated Routing and Bridging Interfaces on Switches (CLI Procedure)" on page 739, and for an RVI, see *Configuring Routed VLAN Interfaces on Switches (CLI Procedure)*.

Viewing IRB Interface and RVI Statistics

Some switches automatically track IRB interface and RVI traffic statistics. Other switches allow you to configure tracking. Table 110 on page 734 illustrates the IRB interface- and RVI-tracking capability on various switches.

Table 110: Tracking IRB Interface and RVI Usage

Switch	Input (ingress)	Output (Egress)
EX4300	Automatic	Automatic
EX3200, EX4200	Automatic	–
EX8200	Configurable	Automatic
EX2200, EX3300, EX4500, EX6200	–	–

You can view input (ingress) and output (egress) totals with the following commands:

- For IRB interfaces, use the **show interfaces irb extensive** command. Look at the input and output values in the Transit Statistics field for IRB interface activity values.
- For RVI, use the **show interfaces vlan extensive** command. Look at the input and output values in the Logical Interface Transit Statistics field for RVI activity values.

IRB Interfaces and RVI Functions and Other Technologies

IRB interfaces and RVIs are similar to switch virtual interfaces (SVIs) and bridge-group virtual interfaces (BVIs), which are supported on other vendors' devices. They can also be combined with other functions:

- VRF is often used in conjunction with Layer 3 subinterfaces, allowing traffic on a single physical interface to be differentiated and associated with multiple virtual routers. For more information about VRF, see "Understanding Virtual Routing Instances on EX Series Switches" on page 709.
- For redundancy, you can combine an IRB interface or RVI with implementations of the Virtual Router Redundancy Protocol (VRRP) in both bridging and virtual private LAN service (VPLS) environments. For more information about VRRP, see *Understanding VRRP*.

SEE ALSO

Ethernet Switching and Layer 2 Transparent Mode Overview | **41**

Example: Configuring VLANs on Security Devices | **187**

Configuring IRB Interfaces on Switches

Integrated routing and bridging (IRB) interfaces enable a switch to recognize which packets are being sent to local addresses so that they are bridged whenever possible and are routed only when needed. Whenever packets can be switched instead of routed, several layers of processing are eliminated. Switching also reduces the number of address look-ups.

> **NOTE:** In versions of Junos OS that do not support Enhanced Layer 2 Software (ELS), this type of interface is called a routed VLAN interface (RVI).

> **NOTE:** When you upgrade from Junos OS Release 15.1X53 to Junos OS Release 17.3R1, you must define an IRB interface at both the **[edit vlans l3-interface]** and **[edit interfaces irb]** hierarchies, otherwise there will be a commit error.

To configure the routed VLAN interface:

1. Create the VLAN by assigning it a name and a VLAN ID:

   ```
   [edit]
   user@switch# set vlans support vlan-id 111
   ```

2. Assign an interface to the VLAN by specifying the logical interface (with the **unit** statement) and specifying the VLAN name as the member:

   ```
   [edit]
   user@switch# set interfaces ge-0/0/18 unit 0 family ethernet-switching vlan members support
   ```

3. Create the subnet for the VLAN's broadcast domain:

   ```
   [edit]
   user@switch# set interfaces irb unit 111 family inet address 10.0.0.X/8
   ```

   Where the value of X can be any number between the range 1 to 254.

4. Bind a Layer 3 interface with the VLAN:

   ```
   [edit]
   user@switch# set vlans support l3-interface irb.111
   ```

NOTE: If you are using a version of Junos OS that does not support ELS, you create a Layer 3 virtual interface named **vlan**

NOTE: Layer 3 interfaces on trunk ports allow the interface to transfer traffic between multiple VLANs. Within a VLAN, traffic is bridged, while across VLANs, traffic is routed.

You can display the configuration settings:

```
user@switch> show interfaces irb terse
```

```
Interface               Admin Link Proto    Local                 Remote
vlan                    up    up
irb.111                 up    up   inet     10.0.0.0/8
```

```
user@switch> show vlans
```

```
Name            Tag     Interfaces
default
                        None
employee-vlan   20
                        ge-1/0/0.0, ge-1/0/1.0, ge-1/0/2.0
marketing       40
                        ge-1/0/10.0, ge-1/0/20.0, ge-1/0/30.0
support         111
                        ge-0/0/18.0
mgmt
                        bme0.32769, bme0.32771*
```

```
user@switch> show ethernet-switching table
```

```
Ethernet-switching table: 1 entries, 0 learned
  VLAN              MAC address       Type         Age Interfaces
  support           00:19:e2:50:95:a0 Static         - Router
```

Configuring Integrated Routing and Bridging for VLANs

Integrated routing and bridging (IRB) provides simultaneous support for Layer 2 bridging and Layer 3 routing on the same interface. IRB enables you to route packets to another routed interface or to another VLAN that has an IRB interface configured. You configure a logical routing interface by specifying **irb** as an interface name at the **[edit interfaces]** hierarchy level and including that interface in the VLAN.

> **NOTE:** You can include only one Layer 3 interface in a VLAN.

To configure a VLAN with IRB support, include the following statements:

```
[edit]
vlans {
    vlan-name {
        domain-type bridge;
        interface interface-name;
        l3-interface (VLAN) interface-name;
        vlan-id (none | number);
        vlan-tags outer number inner number;
    }
}
```

For each VLAN that you configure, specify a ***vlan-name***. You must also specify the value **bridge** for the **domain-type** statement.

For the **vlan-id** statement, you can specify either a valid VLAN identifier or the **none** option.

> **NOTE:** If you configure a Layer 3 interface to support IRB in a VLAN, you cannot use the **all** option for the **vlan-id** statement.

The **vlan-tags** statement enables you to specify a pair of VLAN identifiers; an **outer** tag and an **inner** tag.

> **NOTE:** For a single VLAN, you can include either the **vlan-id** statement or the **vlan-tags** statement, but not both.

To include one or more logical interfaces in the VLAN, specify the ***interface-name*** for each Ethernet interface to include that you configured at the **[edit interfaces]** hierarchy level.

NOTE: A maximum of 4096 active logical interfaces are supported for a VLAN or on each mesh group in a VPLS routing instance configured for Layer 2 bridging.

To associate a Layer 3 interface with a VLAN, include the **l3-interface *interface-name*** statement and specify an ***interface-name*** you configured at the **[edit interfaces irb]** hierarchy level. You can configure only one Layer 3 interface for each VLAN.

IRB interfaces are supported for multicast snooping.

In multihomed VPLS configurations, you can configure VPLS to keep a VPLS connection up if only an IRB interface is available by configuring the **irb** option for the **connectivity-type** statement at the **[edit routing-instances *routing-instance-name* protocols vpls]** hierarchy level. The **connectivity-type** statement has the **ce** and **irb** options. The **ce** option is the default and specifies that a CE interface is required to maintain the VPLS connection. By default, if only an IRB interface is available, the VPLS connection is brought down.

NOTE: When you configure IRB interfaces in more than one logical system on a device, all of the IRB logical interfaces share the same MAC address.

Configuring Integrated Routing and Bridging Interfaces on Switches (CLI Procedure)

Integrated routing and bridging (IRB) interfaces allow a switch to recognize packets that are being sent to local addresses so that they are bridged (switched) whenever possible and are routed only when necessary. Whenever packets can be switched instead of routed, several layers of processing are eliminated.

An interface named irb functions as a logical router on which you can configure a Layer 3 logical interface for each virtual LAN (VLAN). For redundancy, you can combine an IRB interface with implementations of the Virtual Router Redundancy Protocol (VRRP) in both bridging and virtual private LAN service (VPLS) environments.

Jumbo frames of up to 9216 bytes are supported on an IRB interface. To route jumbo data packets on the IRB interface, you must configure the jumbo MTU size on the member physical interfaces of the VLAN that you have associated with the IRB interface, as well as on the IRB interface itself (the interface named irb).

CAUTION: Setting or deleting the jumbo MTU size on the IRB interface (the interface named irb) while the switch is transmitting packets might result in dropped packets.

To configure the IRB interface:

1. Create a Layer 2 VLAN by assigning it a name and a VLAN ID:

   ```
   [edit]
   user@switch# set vlans vlan-name vlan-id vlan-id
   ```

2. Assign an interface to the VLAN by naming the VLAN as a trunk member on the logical interface, thereby making the interface part of the VLAN's broadcast domain:

   ```
   [edit]
   user@switch# set interfaces interface-name unit logical-unit-number family ethernet-switching vlan members vlan-name
   ```

3. Create a logical Layer 3 IRB interface (its name will be irb.*logical-interface-number*, where the value for *logical-interface-number* is the value you supplied for *vlan-id* in Step 1; in the following command, it is the *logical-unit-number*) on a subnet for the VLAN's broadcast domain:

   ```
   [edit]
   user@switch# set interfaces irb unit logical-unit-number family inet address inet-address
   ```

4. Link the Layer 2 VLAN to the logical Layer 3 IRB interface:

```
[edit]
user@switch# set vlans vlan-name l3-interface irb.logical-interface-number
```

> **NOTE:** Layer 3 interfaces on trunk ports allow the interface to transfer traffic between multiple Layer 2 VLANs. Within a VLAN, traffic is switched, while across VLANs, traffic is routed.

Using an IRB Interface in a Private VLAN on a Switch

IN THIS SECTION

VLANs limit broadcasts to specified users. Private VLANs (PVLANs) take this concept a step further by splitting the broadcast domain into multiple isolated broadcast subdomains and essentially putting secondary VLANs inside a primary VLAN. PVLANs restrict traffic flows through their member switch ports (called “private ports”) so that these ports communicate only with a specified uplink trunk port or with specified ports within the same VLAN. PVLANs are useful for restricting the flow of broadcast and unknown unicast traffic and for limiting the communication between known hosts. Service providers use PVLANs to keep their customers isolated from one another.

Just like regular VLANs, PVLANs are isolated at Layer 2 and normally require that a Layer 3 device be used if you want to route traffic. Starting with Junos OS 14.1X53-D30, you can use an integrated routing and bridging (IRB) interface to route Layer 3 traffic between devices connected to a PVLAN. Using an IRB interface in this way can also allow the devices in the PVLAN to communicate at Layer 3 with devices outside the PVLAN.

Configuring an IRB Interface in a Private VLAN

Use the following guidelines when configuring an IRB interface in a PVLAN:

- You can create only one IRB interface in a PVLAN, regardless of how many switches participate in the PVLAN.
- The IRB interface must be a member of the primary VLAN in the PVLAN.

- Each host device that you want to connect at Layer 3 must use the IP address of the IRB as its default gateway address.
- • Because the host devices are isolated at Layer 2, you must configure the following statement for the IRB interface to allow ARP resolution to occur:

  set interfaces irb unit *unit-number* proxy-arp unrestricted

IRB Interface Limitation in a PVLAN

If your PVLAN includes multiple switches, an issue can occur if the Ethernet switching table is cleared on a switch that does not have an IRB interface. If a Layer 3 packet transits the switch before its destination MAC address is learned again, it is broadcast to all the Layer 3 hosts connected to the PVLAN.

RELATED DOCUMENTATION

Understanding Private VLANs | 423

Creating a Private VLAN on a Single QFX Switch | 475

Example: Configuring Routing Between VLANs on One Switch Using an IRB Interface

IN THIS SECTION

To segment traffic on a LAN into separate broadcast domains, you create separate virtual LANs (VLANs). For example, you might want to create a VLAN that includes the employees in a department and the resources that they use often, such as printers, servers, and so on.

Of course, you also want to allow these employees to communicate with people and resources in other VLANs. To forward packets between VLANs you normally you need a router that connects the VLANs.

However, you can accomplish this on a Juniper Networks switch without using a router by configuring an integrated routing and bridging (IRB) interface (also known as a routed VLAN interface—or RVI—in versions of Junos OS that do not support Enhanced Layer 2 Software). Using this approach reduces complexity and avoids the costs associated with purchasing, installing, managing, powering, and cooling another device.

Requirements

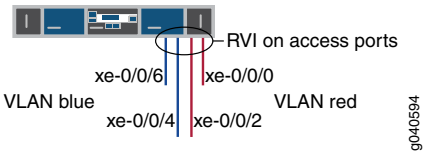
This example uses the following hardware and software components:

- One switch
- Junos OS Release 11.1 or later

Overview and Topology

This example uses an IRB to route traffic between two VLANs on the same switch. The topology is shown in Figure 40 on page 742.

Figure 40: IRB with One Switch



This example shows a simple configuration to illustrate the basic steps for creating two VLANs on a single switch and configuring an IRB to enable routing between the VLANs. One VLAN, called **blue**, is for the sales and marketing group, and a second, called **red**, is for the customer support team. The sales and support groups each have their own file servers and wireless access points. Each VLAN must have a unique name, tag (VLAN ID), and distinct IP subnet. Table 111 on page 742 lists the components of the sample topology.

Table 111: Components of the Multiple VLAN Topology

Property	Settings
VLAN names and tag IDs	**blue**, ID **100** **red**, ID **200**
Subnets associated with VLANs	**blue**: **192.0.2.0/25** (addresses **192.0.2.1** through **192.0.2.126**) **red**: **192.0.2.128/25** (addresses **192.0.2.129** through **192.0.2.254**)
Interfaces in VLAN **blue**	Sales server port: **xe-0/0/4** Sales wireless access points: **xe-0/0/6**

Table 111: Components of the Multiple VLAN Topology *(continued)*

Property	Settings
Interfaces in VLAN **red**	Support server port: **xe-0/0/0** Support wireless access points: **xe-0/0/2**
IRB name	interface **irb**
IRB units and addresses	logical unit 100: **192.0.2.1/25** logical unit 200: **192.0.2.129/25**

This configuration example creates two IP subnets, one for the blue VLAN and the second for the red VLAN. The switch bridges traffic within the VLANs. For traffic passing between two VLANs, the switch routes the traffic using an IRB on which you have configured addresses in each IP subnet.

To keep the example simple, the configuration steps show only a few interfaces and VLANs. Use the same configuration procedure to add more interfaces and VLANs. By default, all interfaces are in access mode, so you do not have to configure the port mode.

Configure Layer 2 switching for two VLANs

CLI Quick Configuration

To quickly configure Layer 2 switching for the two VLANs (**blue** and **red**) and to quickly configure Layer 3 routing of traffic between the two VLANs, copy the following commands and paste them into the switch terminal window:

> **NOTE:** The following example uses a version of Junos OS that supports Enhanced Layer 2 Software (ELS). When you use ELS, you create a Layer 3 virtual interface named *irb*. If you are using a version of Junos OS that does not support ELS, you create a Layer 3 virtual interface named **vlan**.

```
[edit]
```

set interfaces xe-0/0/4 unit 0 description “Sales server port”

set interfaces xe-0/0/4 unit 0 family ethernet-switching vlan members blue

set interfaces xe-0/0/6 unit 0 description “Sales wireless access point port”

set interfaces xe-0/0/6 unit 0 family ethernet-switching vlan members blue

set interfaces xe-0/0/0 unit 0 description “Support servers”

set interfaces xe-0/0/0 unit 0 family ethernet-switching vlan members red

set interfaces xe-0/0/2 unit 0 description “Support wireless access point port”

set interfaces xe-0/0/2 unit 0 family ethernet-switching vlan members red

set interfaces irb unit 100 family inet address 192.0.2.1/25

set interfaces irb unit 200 family inet address 192.0.2.129/25

set vlans blue l3-interface irb.100

set vlans blue vlan-id 100

set vlans red vlan-id 200

set vlans red l3-interface irb.200

Step-by-Step Procedure

To configure the switch interfaces and the VLANs to which they belong:

1. Configure the interface for the sales server in the blue VLAN:

```
[edit interfaces xe-0/0/4 unit 0]
user@switch# set description “Sales server port”
user@switch# set family ethernet-switching vlan members blue
```

2. Configure the interface for the wireless access point in the blue VLAN:

```
[edit interfaces xe-0/0/6 unit 0]
user@switch# set description “Sales wireless access point port”
user@switch# set family ethernet-switching vlan members blue
```

3. Configure the interface for the support server in the red VLAN:

```
[edit interfaces xe-0/0/0 unit 0]
user@switch# set description “Support server port”
user@switch# set family ethernet-switching vlan members red
```

4. Configure the interface for the wireless access point in the red VLAN:

```
[edit interfaces xe-0/0/2 unit 0]
user@switch# set description “Support wireless access point port”
user@switch# set family ethernet-switching vlan members red
```

Step-by-Step Procedure

Now create the VLANs and the IRB. The IRB will have logical units in the broadcast domains of both VLANs.

1. Create the red and blue VLANs by configuring the VLAN IDs for them:

```
[edit vlans]
user@switch# set blue vlan-id 100
user@switch# set red vlan-id 200
```

2. Create the interface named **irb** with a logical unit in the sales broadcast domain (blue VLAN):

```
[edit interfaces]
user@switch# set irb unit 100 family inet address 192.0.2.1/25
```

   The unit number is arbitrary and does not have to match the VLAN tag ID. However, configuring the unit number to match the VLAN ID can help avoid confusion.

3. Add a logical unit in the support broadcast domain (red VLAN) to the **irb** interface:

```
[edit interfaces]
user@switch# set irb unit 200 family inet address 192.0.2.129/25
```

4. Complete the IRB configuration by binding the red and blue VLANs (Layer 2) with the appropriate logical units of the **irb** interface (Layer 3):

```
[edit vlans]
user@switch# set blue l3-interface irb.100
user@switch# set red l3-interface irb.200
```

Configuration Results

Display the results of the configuration:

```
user@switch> show configuration
interfaces {
    xe-0/0/4 {
        unit 0 {
            description “Sales server port”;
            family ethernet-switching {
                vlan members blue;
            }
        }
    }
    xe-0/0/6 {
        unit 0 {
```

```
            description “Sales wireless access point port”;
            family ethernet-switching {
               vlan members blue;
            }
         }
      }
      xe-0/0/0 {
         unit 0 {
            description “Support server port”;
            family ethernet-switching {
               vlan members red;
            }
         }
      }
      xe-0/0/2 {
         unit 0 {
            description “Support wireless access point port”;
            family ethernet-switching {
               vlan members red;
            }
         }
      }
      irb {
         unit 100 {
            family inet address 192.0.2.1/25;
         }
         unit 200 {
            family inet address 192.0.2.129/25;
            }
         }
      }
   }
   vlans {
      blue {
         vlan-id 100;
         interface xe-0/0/4.0:
         interface xe-0/0/6.0;
         l3-interface irb 100;
      }
      red {
         vlan-id 200;
         interface xe-0/0/0.0:
         interface xe-0/0/2.0;
         l3-interface irb 200;
```

```
    }
}
```

TIP: To quickly configure the blue and red VLAN interfaces, issue the **load merge terminal** command, copy the hierarchy, and paste it into the switch terminal window.

Verification

IN THIS SECTION

To verify that the **blue** and **red**VLANs have been created and are operating properly, perform these tasks:

Verifying That the VLANs Have Been Created and Associated with the Correct Interfaces

Purpose

Verify that the VLANs **blue** and **red** have been created on the switch and that all connected interfaces on the switch are members of the correct VLAN.

Action

List all VLANs configured on the switch:

```
user@switch> show vlans
```

```
Name          Tag      Interfaces
default               xe-0/0/0.0, xe-0/0/2.0, xe-0/0/4.0, xe-0/0/6.0,
blue          100      xe-0/0/4.0, xe-0/0/6,
red           200      xe-0/0/0.0, xe-0/0/2.0, *
mgmt                   me0.0*
```

Meaning

The **show vlans** command lists all VLANs configured on the switch and which interfaces are members of each VLAN. This command output shows that the **blue** and **red** VLANs have been created. The **blue** VLAN

has a tag ID of 100 and is associated with interfaces **xe-0/0/4.0** and **xe-0/0/6.0**. VLAN **red** has a tag ID of 200 and is associated with interfaces **xe-0/0/0.0** and **xe-0/0/2.0**.

Verifying That Traffic Can Be Routed Between the Two VLANs

Purpose

Verify routing between the two VLANs.

Action

Verify that the IRB logical units are up:

```
user@switch> show interfaces terse
```

```
irb.100                 up    up   inet     192.0.2.1/25
```

```
irb.200                 up    up   inet     192.0.2.129/25
```

> **NOTE:** At least one port (access or trunk) with an appropriate VLAN assigned to it must be up for the **irb** interface to be up.

Verify that switch has created routes that use the IRB logical units:

```
user@switch> show route
```

```
192.0.2.0/25        *[Direct/0] 1d 03:26:45
                     > via irb.100
192.0.2.1/32        *[Local/0] 1d 03:26:45
                       Local via irb.100
192.0.2.128/25      *[Direct/0] 1d 03:26:45
                     > via irb.200
192.0.2.129/32      *[Local/0] 1d 03:26:45
                       Local via irb.200
```

List the Layer 3 routes in the switch's Address Resolution Protocol (ARP) table:

```
user@switch> show arp
```

```
MAC Address             Address          Name                    Flags
```

```
00:00:0c:06:2c:0d       192.0.2.7       irb.100                 None
00:13:e2:50:62:e0       192.0.2.132     irb.200                 None
```

Meaning

The output of the **show interfaces** and **show route** commands show that the Layer 3 IRB logical units are working and the switch has used them to create direct routes that it will use to forward traffic between the VLAN subnets. The **show arp** command displays the mappings between the IP addresses and MAC addresses for devices on both **irb.100** (associated with VLAN **blue**) and **irb.200** (associated with VLAN **red**).These two devices can communicate.

Example: Configuring an IRB Interface on a Security Device

IN THIS SECTION

This example shows how to configure an IRB interface so it can act as a Layer 3 routing interface for a VLAN.

Requirements

Before you begin, configure a VLAN with a single VLAN identifier. See “Example: Configuring VLANs on Security Devices” on page 187.

Overview

In this example, you configure the IRB logical interface unit 0 with the family type inet and IP address 10.1.1.1/24, and then reference the IRB interface irb.10 in the vlan10 configuration. Then you enable Web authentication on the IRB interface and activate the webserver on the device.

NOTE: To complete the Web authentication configuration, you must perform the following tasks:

- Define the access profile and password for a Web authentication client.
- Define the security policy that enables Web authentication for the client.

Either a local database or an external authentication server can be used as the Web authentication server.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set interfaces ge-1/0/0 unit 0 family ethernet-switching interface-mode trunk
set interfaces ge-1/0/0 unit 0 family ethernet-switching vlan members 10
set interface irb unit 10 family inet address 10.1.1.1/24 web-authentication http
set vlans vlan10 vlan-id 10
set vlans vlan10 l3-interface irb.10
set system services web-management http
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure an IRB interface:

1. Create a Layer 2 trunk interface.

```
[edit]
user@host# set interfaces ge-1/0/0 unit 0 family ethernet-switching interface-mode trunk
user@host# set interfaces ge-1/0/0 unit 0 family ethernet-switching vlan members 10
```

2. Create an IRB logical interface.

```
[edit]
user@host# set interface irb unit 10 family inet address 10.1.1.1/24 web-authentication http
```

3. Create a Layer 2 VLAN.

```
[edit]
user@host# set vlans vlan10 vlan-id 10
```

4. Associate the IRB interface with the VLAN.

```
[edit]
user@host# set vlans vlan10 l3-interface irb.10
```

5. Activate the webserver.

```
[edit]
user@host# set system services web-management http
```

6. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show interface irb** , and **show vlans** commands.

SEE ALSO

Example: Configuring Layer 2 Security Zones | 980

Example: Configuring VLAN with Members Across Two Nodes on a Security Device

IN THIS SECTION

Requirements

This example uses the following hardware and software components:

- configure a switching fabric interface on both nodes to configure Ethernet switching-related features on the nodes. See *Example: Configuring Switch Fabric Interfaces to Enable Switching in Chassis Cluster Mode on a Security Device*
- SRX240 security device
- Junos OS 12.3X48-D90
- interface-mode is supported in 15.1X49 release.
- port-mode is supported in 12.1 and 12.3X48 releases.

Overview

This example shows the configuration of a VLAN with members across node 0 and node 1.

Configuration

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set interfaces ge-0/0/3 unit 0 family ethernet-switching port-mode access
set interfaces ge-0/0/3 unit 0 family ethernet-switching vlan members vlan100
set interfaces ge0/0/4 unit 0 family ethernrt-switching port-mode access
set interfaces ge-0/0/4 unit 0 family ethernet-switching vlan members vlan100
set interfaces ge-7/0/5 unit 0 family ethernet-switching port-mode trunk
set interfaces ge-7/0/5 unit 0 family ethernet-switching vlan members vlan100
set interfaces vlan unit 100 family inet address 11.1.1.1/24
set vlans vlan100 vlan-id 100
set vlans vlan100 l3-interface vlan.100
```

Step-by-Step Procedure

To configure VLAN:

1. Configure Ethernet switching on the node0 interface.

```
{primary:node0} [edit]
user@host# set interfaces ge-0/0/3 unit 0 family ethernet-switching port-mode access
user@host# set interfaces ge0/0/4 unit 0 family ethernet-switching port-mode access
```

2. Configure Ethernet switching on the node1 interface.

```
{primary:node0} [edit]
user@host# set interfaces ge-7/0/5 unit 0 family ethernet-switching port-mode trunk
```

3. Create VLAN vlan100 with vlan-id 100.

```
{primary:node0} [edit]
user@host# set vlans vlan100 vlan-id 100
```

4. Add interfaces from both nodes to the VLAN.

```
{primary:node0} [edit]
user@host# set interfaces ge-0/0/3 unit 0 family ethernet-switching vlan members vlan100
user@host# set interfaces ge-0/0/4 unit 0 family ethernet-switching vlan members vlan100
user@host# set interfaces ge-7/0/5 unit 0 family ethernet-switching vlan members vlan100
```

5. Create a VLAN interface.

```
user@host# set interfaces vlan unit 100 family inet address 11.1.1.1/24
```

6. Associate an VLAN interface with the VLAN.

```
user@host# set vlans vlan100 l3-interface vlan.100
```

7. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Results

From configuration mode, confirm your configuration by entering the **show vlans** and **show interfaces** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct the configuration.

```
[edit]
user@host# show vlans
vlan100 {
    vlan-id 100;
    l3-interface vlan.100;
}
[edit]
user@host# show interfaces
ge-0/0/3 {
    unit 0 {
        family ethernet-switching {
            port-mode access;
            vlan {
                members vlan100;
            }
        }
    }
}
ge-0/0/4 {
    unit 0 {
        family ethernet-switching {
            port-mode access;
            vlan {
                members vlan100;
            }
        }
    }
}
```

```
ge-7/0/5 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan {
                members vlan100;
            }
        }
    }
}
```

Verification

Verifying VLAN

Purpose

Verify that the configuration of VLAN is working properly.

Action

From operational mode, enter the **show interfaces terse ge-0/0/3** command to view the node 0 interface.

```
user@host> show interfaces terse ge-0/0/3
Interface               Admin Link Proto    Local                 Remote
ge-0/0/3                up    up
ge-0/0/3.0              up    up   eth-switch
```

From operational mode, enter the **show interfaces terse ge-0/0/4** command to view the node 0 interface.

```
user@host> show interfaces terse ge-0/0/4
Interface               Admin Link Proto    Local                 Remote
ge-0/0/4                up    up
ge-0/0/4.0              up    up   eth-switch
```

From operational mode, enter the **show interfaces terse ge-7/0/5** command to view the node1 interface.

```
user@host> show interfaces terse ge-7/0/5
Interface               Admin Link Proto    Local                 Remote
ge-7/0/5                up    up
ge-7/0/5.0              up    up   eth-switch
```

From operational mode, enter the **show vlans** command to view the VLAN interface.

```
user@host> show vlans
Routing instance      VLAN name     Tag      Interfaces
default-switch        default        1
default-switch        vlan100        100     ge-0/0/3.0*
                                             ge-0/0/4.0*
                                             ge-7/0/5.0*
```

From operational mode, enter the **show ethernet-switching interface** command to view the information about Ethernet switching interfaces.

```
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down,
                         MMAS - Mac-move action shutdown,  AS - Autostate-exclude
 enabled,
                         SCTL - shutdown by Storm-control )

Logical         Vlan                  TAG   MAC    STP         Logical
Tagging
interface       members                     limit  state       interface flags
ge-0/0/3.0                                  16383              DN
untagged
                vlan100               100   1024   Discarding
untagged
ge-0/0/4.0                                  16383              DN
untagged
                vlan100               100   1024   Discarding
untagged
ge-7/0/5.0                                  16383              DN
tagged
                vlan100               100   1024   Discarding
tagged
```

Meaning

The output shows the VLANs are configured and working fine.

SEE ALSO

Example: Configuring an IRB Interface

Example: Configuring IRB Interfaces on QFX5100 Switches over an MPLS Core Network

IN THIS SECTION

Starting with Junos OS Release 14.1X53-D40 and Junos OS Release 17.1R1, QFX5100 switches support integrated routing and bridging (IRB) interfaces over an MPLS core network. An IRB interface is a logical Layer 3 VLAN interface used to route traffic between VLANs.

By definition, VLANs divide a LAN's broadcast environment into isolated virtual broadcast domains, thereby limiting the amount of traffic flowing across the entire LAN and reducing the possible number of collisions and packet retransmissions within the LAN. To forward packets between different VLANs, you traditionally needed a router that connects the VLANs. However, using the Junos OS you can accomplish this inter-VLAN forwarding without using a router by simply configuring an IRB interface on the switch.

The IRB interface functions as a logical switch on which you can configure a Layer 3 logical interface for each VLAN. The switch relies on its Layer 3 capabilities to provide this basic routing between VLANs. With an IRB interface, you can configure label-switched paths (LSPs) to enable the switch to recognize which packets are being sent to local addresses, so that they are bridged (switched) whenever possible and are routed only when necessary. Whenever packets can be switched instead of routed, several layers of processing are eliminated.

This example shows how to configure an IRB interface over an MPLS core network using QFX5100 switches.

Requirements

This example uses the following hardware and software components:

- Three QFX5100 switches
- Junos OS Release 14.1X53-D40 or later

Before you begin, be sure you have:

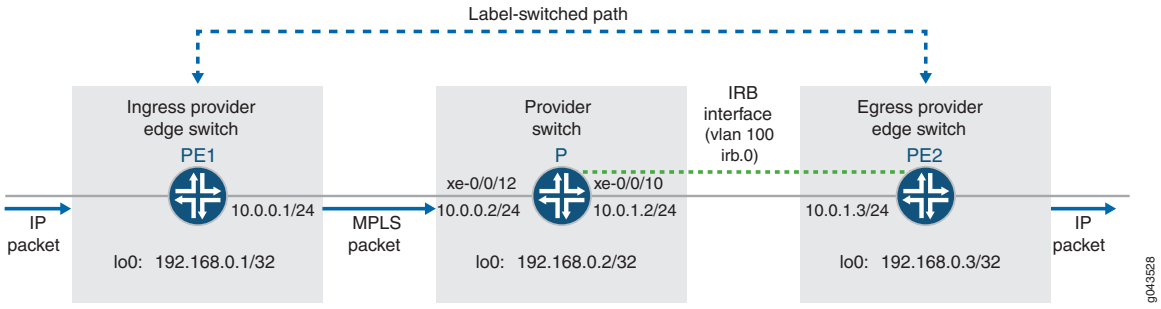
- An understanding of IRB concepts. See "Understanding Integrated Routing and Bridging" on page 728 for an overview of IRB.

- The required ternary content addressable memory (TCAM) space available on the switch. TCAM rules must be observed while configuring and implementing IRBs. For detailed information, see *MPLS Limitations on QFX Series and EX4600 Switches*.

Overview and Topology

Figure 41 on page 758 illustrates a sample topology for configuring IRB over an MPLS core network. In this example, an LSP is established between the ingress provider edge switch (PE1) and the provider edge egress switch (PE2). An IRB Layer 3 interface (irb.0) is configured on switches P and PE2, and associated to VLAN 100. In this configuration, the P switch replaces (swaps) the label at the top of the label stack with a new label, adds the VLAN identifier 100 to the MPLS packet, and then sends the packet out the IRB interface. PE2 receives this vlan-tagged MPLS packet, removes (pops) the label from the top of the label stack, performs a regular IP route lookup, and then forwards the packet with its IP header to the next-hop address.

Figure 41: IRB Topology over an MPLS Core Network



Configuration

IN THIS SECTION

To configure the topology in this example, perform these tasks:

Configuring the Local Ingress PE Switch

CLI Quick Configuration

To quickly configure the local ingress PE switch (PE1), copy and paste the following commands into the switch terminal window of switch PE1:

```
set interfaces xe-0/0/12 unit 0 family inet address 10.0.0.1/24
set interfaces xe-0/0/12 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.0.1/32
set routing-options router-id 192.168.0.1
set routing-options autonomous-system 65550
set policy-options policy-statement pplb then load-balance per-packet
set routing-options forwarding-table export pplb
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface em0.0 disable
set protocols mpls interface all
set protocols ldp interface xe-0/0/12.0
set protocols ldp interface lo0.0
```

Step-by-Step Procedure

To configure the ingress PE switch (PE1):

1. Configure the interfaces.

```
[edit interfaces]
user@switchPE1# set xe-0/0/12 unit 0 family inet address 10.0.0.1/24
user@switchPE1# set xe-0/0/12 unit 0 family mpls
user@switchPE1# set lo0 unit 0 family inet address 192.168.0.1/32
```

2. Configure the router ID and autonomous system (AS) number.

> **NOTE:** We recommend that you explicitly configure the router identifier under the **[edit routing-options]** hierarchy level to prevent unpredictable behavior if the interface address on a loopback interface changes.

```
[edit routing-options]
user@switchPE1# set router-id 192.168.0.1/32
user@switchPE1# set autonomous-system 65550
```

3. Configure and apply an export routing policy to the forwarding table for per-packet load balancing.

```
[edit policy-options]
user@switchPE1# set policy-statement pplb then load-balance per-packet
[edit routing-options]
user@switchPE1# set forwarding-table export pplb
```

4. Create an OSPF area and set the loopback address to be passive.

```
[edit protocols ospf]
user@switchPE1# set area 0.0.0.0 interface all
user@switchPE1# set area 0.0.0.0 interface lo0.0 passive
user@switchPE1# set area 0.0.0.0 interface em0.0 disable
```

5. Enable MPLS on all interfaces.

```
[edit protocols mpls]
user@switchPE1# set interface all
```

6. Configure LDP on the provider-facing and loopback interfaces.

```
[edit protocols ldp]
user@switchPE1# set interface xe-0/0/12.0
user@switchPE1# set interface lo0.0
```

Results

Display the results of the PE1 switch configuration:

```
user@switchPE1# show
interfaces {
    xe-0/0/12 {
        unit 0 {
            family inet {
                address 10.0.0.1/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
```

```
            address 192.168.0.1/32;
            }
          }
        }
    }
}
routing-options {
    router-id 192.168.0.1;
    autonomous-system 65550;
    forwarding-table {
        export pplb;
    }
}
protocols {
    mpls {
        interface all;
    }
    ospf {
        area 0.0.0.0 {
            interface all;
            interface lo0.0 {
                passive;
            }
            interface em0.0 {
                disable;
            }
        }
    }
    ldp {
        interface xe-0/0/12.0
        interface lo0.0;
    }
}
policy-options {
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
    }
}
```

Configuring the Provider Switch

CLI Quick Configuration

To quickly configure the provider switch (P), copy and paste the following commands into the switch terminal window of the P switch:

```
set interfaces xe-0/0/12 unit 0 family inet address 10.0.0.2/24
set interfaces xe-0/0/12 unit 0 family mpls
set interfaces xe-0/0/10 unit 0 family ethernet-switching interface-mode trunk
set interfaces xe-0/0/10 unit 0 family ethernet-switching vlan members v100
set interfaces lo0 unit 0 family inet address 192.168.0.2/32
set interfaces irb unit 0 family inet address 10.0.1.2/24
set interfaces irb unit 0 family mpls
set routing-options router-id 192.168.0.2
set routing-options autonomous-system 65550
set policy-options policy-statement pplb then load-balance per-packet
set routing-options forwarding-table export pplb
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface em0.0 disable
set protocols mpls interface all
set protocols ldp interface all
set vlans v100 vlan-id 100
set vlans v100 l3-interface irb.0
```

Step-by-Step Procedure

To configure the provider switch (P):

1. Configure the physical and loopback interfaces.

```
[edit interfaces]
user@switchP#  set xe-0/0/12 unit 0 family inet address 10.0.0.2/24
user@switchP#  set xe-0/0/12 unit 0 family mpls
user@switchP# set xe-0/0/10 unit 0 family ethernet-switching interface-mode trunk
user@switchP# set xe-0/0/10 unit 0 family ethernet-switching vlan members v100
user@switchP# set lo0 unit 0 family inet address 192.168.0.2/32
```

2. Configure an IRB interface.

```
[edit interfaces]
user@switchP# set irb unit 0 family inet address 10.0.1.2/24
user@switchP# set irb unit 0 family mpls
```

3. Configure the router ID and AS number.

NOTE: We recommend that you explicitly configure the router identifier under the **[edit routing-options]** hierarchy level to avoid unpredictable behavior if the interface address on a loopback interface changes.

```
[edit routing-options]
user@switchP# router-id 192.168.0.2
user@switchP# set autonomous-system 65550
```

4. Configure and apply an export routing policy to the forwarding table for per-packet load balancing.

```
[edit policy-options]
user@switchP# set policy-statement pplb then load-balance per-packet
[edit routing-options]
user@switchP# set forwarding-table export pplb
```

5. Enable OSPF and set the loopback address to passive.

```
[edit protocols ospf]
user@switchP# set area 0.0.0.0 interface all
user@switchP# set area 0.0.0.0 interface lo0.0 passive
user@switchP# set area 0.0.0.0 interface em0.0 disable
```

6. Enable MPLS on all interfaces.

```
[edit protocols mpls]
user@switchP# set interface all
```

7. Configure LDP to include all interfaces.

```
[edit protocols ldp]
user@switchP# set interface all
```

8. Create the VLAN and associate the IRB interface to it.

```
[edit vlans]
user@switchP# set v100 vlan-id 100
```

```
user@switchP# set v100 l3-interface irb.0
```

NOTE: Layer 3 interfaces on trunk ports allow the interface to transfer traffic between multiple VLANs. Within a VLAN, traffic is switched, while across VLANs, traffic is routed.

Results

Display the results of the provider switch configuration:

```
user@switchP# show
interfaces {
    xe-0/0/10 {
        unit 0 {
            family ethernet-switching {
                interface-mode trunk;
                vlan {
                    members v100;
                }
            }
        }
    }
    xe-0/0/12 {
        unit 0
        family inet {
            address 10.0.0.2/24;
        }
        family mpls;
    }
    irb {
        unit 0 {
            family inet {
                address 10.0.1.2/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 192.168.0.2/32;
            }
```

```
        }
    }
}
```

```
routing-options {
    router-id 192.168.0.2;
    autonomous-system 65550;
    forwarding-table {
        export pplb;
    }
}
```

```
protocols {
    mpls {
        interface all;
    }
    ospf {
        area 0.0.0.0 {
            interface all;
            interface lo0.0 {
                passive;
            }
            interface em0.0 {
                disable;
            }
        }
    }
    ldp {
        interface all;
    }
}
```

```
policy-options {
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
    }
}
```

```
vlans {
```

```
    v100 {
        vlan-id 100;
        l3-interface irb.0;
    }
}
```

Configuring the Remote Egress PE Switch

CLI Quick Configuration

To quickly configure the remote egress PE switch (PE2), copy and paste the following commands into the switch terminal window of PE2:

```
set interfaces xe-0/0/10 unit 0 family ethernet-switching interface-mode trunk
set interfaces xe-0/0/10 unit 0 family ethernet-switching vlan members v100
set interfaces irb unit 0 family inet address 10.0.1.3/24
set interfaces lo0 unit 0 family inet address 192.168.0.3/32
set interfaces irb unit 0 family mpls
set routing-options router-id 192.168.0.3
set routing-options autonomous-system 65550
set policy-options policy-statement pplb then load-balance per-packet
set routing-options forwarding-table export pplb
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface em0.0 disable
set protocols mpls interface all
set protocols ldp interface all
set vlans v100 vlan-id 100
set vlans v100 l3-interface irb.0
```

Step-by-Step Procedure

To configure the remote PE switch (PE2):

1. Configure the physical and loopback interfaces.

```
[edit interfaces]
user@switchPE2# set xe-0/0/10 unit 0 family ethernet-switching interface-mode trunk
user@switchPE2# set xe-0/0/10 unit 0 family ethernet-switching vlan members v100
user@switchPE2# set lo0 unit 0 family inet address 192.168.0.3/32
```

2. Configure an IRB interface.

```
[edit interfaces]
```

```
user@switchPE2# set irb unit 0 family inet address 10.0.1.3/24
user@switchPE2# set irb unit 0 family mpls
```

3. Configure the the router ID and AS number.

```
[edit routing-options]
user@switchPE2# set router-id 192.168.0.3/32
user@switchPE2# set autonomous-system 65550
```

4. Configure and apply an export routing policy to the forwarding table for per-packet load balancing.

```
[edit policy-options]
user@switchPE2# set policy-statement pplb then load-balance per-packet
[edit routing-options]
user@switchPE2# set forwarding-table export pplb
```

5. Enable OSPF.

```
[edit protocols ospf]
user@switchPE2# set area 0.0.0.0 interface all
user@switchPE2# set area 0.0.0.0 interface lo0.0 passive
user@switchPE2# set area 0.0.0.0 interface em0.0 disable
```

6. Enable MPLS on all interfaces.

```
[edit protocols mpls]
user@switchPE2# set interface all
```

7. Configure LDP to include all interfaces.

```
[edit protocols ldp]
user@switchPE2# set interface all
```

8. Create the VLAN and associate the IRB interface to it.

```
[edit vlans]
user@switchPE2# set v100 vlan-id 100
user@switchPE2# set v100 l3-interface irb.0
```

Results

Display the results of the PE2 switch configuration:

```
user@switchPE2# show
interfaces {
    xe-0/0/10 {
        unit 0 {
            family ethernet-switching {
                interface-mode trunk;
                vlan {
                    members v100;
                }
            }
        }
    irb {
        unit 0 {
            family inet {
                address 10.0.1.3/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 192.168.0.3;
            }
        }
    }
}
```

```
routing-options {
    router-id 192.168.0.3;
    autonomous-system 65550;
    forwarding-table {
        export pplb;
    }
}
```

```
protocols {
    mpls {
        interface all;
    }
```

```
    ospf {
        area 0.0.0.0 {
            interface all;
            interface lo0.0 {
                passive;
            }
            interface em0.0 {
                disable;
            }
        }
    }
    ldp {
        interface all;
    }
}
```

```
policy-options {
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
    }
}
```

```
vlans {
    v100 {
        vlan-id 100;
        l3-interface irb.0;
    }
}
```

Example: Configuring a Large Delay Buffer on a Security Device IRB Interface

IN THIS SECTION

This example shows how to configure a large delay buffer on an IRB interface to help slower interfaces avoid congestion and packet dropping when they receive large bursts of traffic.

Requirements

Before you begin, enable the large buffer feature on the IRB interface and then configure a buffer size for each queue in the CoS scheduler. See *Scheduler Buffer Size Overview*.

Overview

On devices, you can configure large delay buffers on an irb interfaces.

In this example, you configure scheduler map to associate schedulers to a defined forwarding class **be-class**, **ef-class** , **af-class**, and **nc-class** using scheduler map **large-buf-sched-map**. You apply scheduler maps to irb interface, and define per-unit scheduler for the IRB interface.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from the configuration mode.

```
set class-of-service scheduler-maps large-buf-sched-map forwarding-class be-class scheduler be-scheduler
set class-of-service scheduler-maps large-buf-sched-map forwarding-class ef-class scheduler ef-scheduler
set class-of-service scheduler-maps large-buf-sched-map forwarding-class af-class scheduler af-scheduler
set class-of-service scheduler-maps large-buf-sched-map forwarding-class nc-class scheduler nc-scheduler
set class-of-service interfaces irb unit 0 scheduler-map large-buf-sched-map
set interfaces irb per-unit-scheduler
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure a large delay buffer on a channelized T1 interface:

1. Configure the scheduler map to associate schedulers with defined forwarding classes.

```
[edit class-of-service]
set scheduler-maps large-buf-sched-map forwarding-class be-class scheduler be-scheduler
set scheduler-maps large-buf-sched-map forwarding-class ef-class scheduler ef-scheduler
set scheduler-maps large-buf-sched-map forwarding-class af-class scheduler af-scheduler
set scheduler-maps large-buf-sched-map forwarding-class nc-class scheduler nc-scheduler
```

2. Apply the scheduler map to the IRB interface.

```
[edit ]
user@host# set interfaces irb unit 0 scheduler-map large-buf-sched-map
```

3. Define the per-unit scheduler for the irb interface.

```
[edit ]
user@host# set interfaces irb per-unit-scheduler
```

Results

From configuration mode, confirm your configuration by entering the **show class-of-service** and **show chassis** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
   user@host# show class-of-service
   interfaces {
   irb {
      unit 0 {
         scheduler-map large-buf-sched-map;
      }
   }
}
scheduler-maps {
   large-buf-sched-map {
      forwarding-class be-class scheduler be-scheduler;
      forwarding-class ef-class scheduler ef-scheduler;
```

```
        forwarding-class af-class scheduler af-scheduler;
        forwarding-class nc-class scheduler nc-scheduler;
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Verifying Large Delay Buffers Configuration

Purpose

Verify that the large delay buffers are configured properly.

Action

From configuration mode, enter the **show class-of-service interface irb** command.

`user@host>` **show class-of-service interface irb**

```
Physical interface: irb, Index: 132
Maximum usable queues: 8, Queues in use: 4Code point type: dscp
Scheduler map: <default>, Index :2
Congestion-notification: Disabled
Logical interface: irb.10, Index: 73
Object                  Name                    Type                    Index
Classifier              ipprec-compatibility    ip                         13
```

Meaning

The large delay buffers are configured on IRB interface as expected.

SEE ALSO

- *Schedulers Overview*
- *Default Scheduler Settings*
- *Example: Configuring and Applying Scheduler Maps*
- *Transmission Scheduling Overview*

Configuring a Set of VLANs to Act as a Switch for a Layer 2 Trunk Port

You can configure a set of VLANs that are associated with a Layer 2 trunk port. The set of VLANs function as a switch. Packets received on a trunk interface are forwarded within a VLAN that has the same VLAN identifier. A trunk interface also provides support for IRB, which provides support for Layer 2 bridging and Layer 3 IP routing on the same interface.

To configure a Layer 2 trunk port and set of VLANs, include the following statements:

```
[edit interfaces]
interface-name {
    unit number {
        family ethernet-switching {
            interface-mode access;
            vlan-members (vlan-name | vlan-tag);
        }
    }
}
interface-name {
    native-vlan-id number;
    unit number {
        family ethernet-switching {
            interface-mode trunk;
            vlan-members (vlan-name | vlan-tag);
        }
    }
}
[edit vlans ]
vlan-name {
    vlan-id number;
    vlan-id-list [ vlan-id-numbers ];
    . . . .
}
```

You must configure a VLAN and VLAN identifier for each VLAN associated with the trunk interface. You can configure one or more trunk or access interfaces at the **[edit interfaces]** hierarchy level. An access interface enables you to accept packets with no VLAN identifier.

Excluding an IRB Interface from State Calculations on a QFX Series Switch

IRB interfaces are used to bind specific VLANs to Layer 3 interfaces, enabling a switch to forward packets between those VLANs— without having to configure another device, such as a router, to connect VLANs. Because an IRB interface often has multiple ports in a single VLAN, the state calculation for a VLAN member might include a port that is down, possibly resulting in traffic loss.

Starting with Junos OS Release 14.1X53-D40 and Junos OS Release 17.3R1 on QFX5100 switches, this feature enables you to exclude a trunk or access interface from the state calculation, which means that as soon as the port assigned to a member VLAN goes down, the IRB interface for the VLAN is also marked as down. In a typical scenario, one port on the interface is assigned to a single VLAN, while a second port on that interface is assigned to a trunk interface that carries traffic between multiple VLANs. A third port is often also assigned to an access interface to connect the VLAN to network devices.

Before you begin:

- Configure VLANs
- Configure IRB interfaces for the VLANs.

  For more information about configuring IRB interfaces, see "Example: Configuring Routing Between VLANs on One Switch Using an IRB Interface" on page 741.

To exclude an access or 802.1Q trunk interface from the state calculations for an IRB interface:

1. Configure a trunk or access interface.

   ```
   [edit interfaces interface-name]
   user@switch# set unit logical-unit-number family ethernet-switching port-mode (access | trunk)
   ```

   For example, configure interface xe-0/1/0.0 as a trunk interface:

   ```
   [edit interfaces xe-0/1/0]
   user@swtich# set unit 0 family ethernet-switching port-mode trunk
   ```

2. Assign VLAN members to the access or trunk interface.

   ```
   [edit interfaces interface-name unit logical-unit-number ethernet-switching]
   user@switch# set vlan members [ (all | names | vlan-ids) ]
   ```

   For example, assign all VLAN members configured on the device to the trunk interface xe-0/1/0:

   ```
   [edit interfaces xe-0/1/0 unit 0 ethernet-switchg]
   user@switch# set vlan members all
   ```

3. Exclude an access or trunk interface from state calculations for the IRB interfaces for member VLANs.

```
[edit interfaces interface-name ether-options]
user@switch# set autostate-exclude
```

For example, exclude the trunk interface xe-0/1/0 from state calculations for the IRB interfaces for member VLANs:

```
[edit interfaces xe-0/1/0]
user@switch# set autostate-exclude
```

4. To confirm your configuration, from configuration mode, enter the **show interfaces xe-0/1/0** command. If your output does not display the intended configuration, repeat steps 1 through 4 to correct the configuration.

```
user@switch# show interfaces xe-0/1/0
ether-options {
   autostate-exclude;
}
unit 0 {
   family ethernet-switching {
      port-mode trunk;
      vlan {
         members all;
      }
   }
}
```

5. After you commit the configuration, issue the **show ethernet-switching interface xe-0/1/0.0** to verify that the logical interface is enabled with **autostate-exclude**.

```
user@switch> show ethernet-switching interface xe-0/1/0.0
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down,
                         SCTL - shutdown by Storm-control,
                        MMAS - Mac-move action shutdown, AS - Autostate-exclude
 enabled)

Logical          Vlan          TAG     MAC         STP         Logical
   Tagging
```

```
interface        members              limit       state       interface flags

xe-0/1/0.0                            294912                   AS
   untagged
                 vlan_100      100     294912      Forwarding
   untagged
```

The **AS** in the **Logical interface flags** field indicates that **autostate-exclude** is enabled and that this interface will be excluded from the state calculations for the IRB interfaces for the member VLANs.

Verifying Integrated Routing and Bridging Interface Status and Statistics on EX Series Switches

Purpose

Determine status information and traffic statistics for integrated routing and bridging (IRB) interfaces.

Action

Display IRB interfaces and their current states:

```
user@switch> show interfaces irb  terse
```

```
Interface               Admin Link Proto    Local                 Remote
irb                     up    up
irb.111                 up    up   inet     10.111.111.1/24
...
```

Display Layer 2 VLANs, including any tags assigned to the VLANs and the interfaces associated with the VLANs:

```
user@switch> show vlans
```

```
Routing instance        VLAN name             Tag         Interfaces
default-switch          irb                     101
default-switch          support                 111
                                                            ge-0/0/18.0
...
```

Display Ethernet switching table entries for the VLAN that is attached to the IRB interface:

```
user@switch> show ethernet-switching table
```

```
MAC flags (S -static MAC, D -dynamic MAC, L -locally learned
           SE -Statistics enabled, NM -Non configured MAC, R -Remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    Name                address             flags               interface
    support             00:01:02:03:04:05   S             -   ge-0/0/18.0
...
```

Display the ingress-counting statistics of an IRB interface with either the **show interfaces irb detail** command or the **show interfaces irb extensive** command. Ingress counting is displayed as **Input bytes** and **Input packets** and egress counting is displayed as **Output bytes** and **Output packets** under **Transit Statistics**.

```
user@switch> show interfaces irb .111 detail
```

```
Logical interface irb.111 (Index 65) (SNMP ifIndex 503) (HW Token 100) (Generation
 131)
Flags: SNMP-Traps 0x4000  Encapsulation: ENET2
Bandwidth: 1000mbps
Routing Instance: default-switch Bridging Domain: irb+111
Traffic statistics:
 Input bytes:    17516756
 Output bytes:   411764
 Input packets:  271745
 Output packets: 8256
Local statistics:
 Input bytes:    3240
 Output bytes:   411764
 Input packets:  54
 Output packets: 8256
Transit statistics:
 Input bytes:    17513516   0 bps
 Output bytes:   0    0 bps
 Input packets:  271745  0 pps
 Output packets: 0    0 pps
Protocol inet, MTU: 1514, Generation: 148, Route table: 0
 Flags: None
```

```
Addresses, Flags: iS-Preferred  Is-Primary
 Destination: 10.1.1/24, Local: 10.1.1.1, Broadcast: 10.1.1.255, Generation: 136
```

Meaning

- **show interfaces irb terse** displays a list of interfaces, including IRB interfaces, and their current states (up, down).
- **show vlans** displays a list of VLANs, including any tags assigned to the VLANs and the interfaces associated with the VLANs.
- **show ethernet-switching table** displays the Ethernet switching table entries, including VLANs attached to the IRB interface.
- **show interfaces irb detail** displays IRB interface ingress counting as **Input Bytes** and **Input Packets** under **Transit Statistics**.

CHAPTER 26

Configuring VLANS and VPLS Routing Instances

VLANs and VPLS Routing Instances

IN THIS SECTION

Guidelines for Configuring VLAN Identifiers for VLANs and VPLS Routing Instances

For a VLAN that is performing Layer 2 switching only, you do not have to specify a VLAN identifier.

For a VLAN that is performing Layer 3 IP routing, you must specify either a VLAN identifier or dual VLAN identifier tags.

For a VPLS routing instance, you must specify either a VLAN identifier or dual VLAN identifier tags.

SEE ALSO

Layer 2 Learning and Forwarding for VLANs Overview | 78

Configuring VLAN Identifiers for VLANs and VPLS Routing Instances

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

- 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 VLAN or a VPLS routing instance.
- By using the **input-vlan-map** and the **output-vlan-map** statements 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 to configure VLAN mapping.

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

- Translate, or normalize, the VLAN tags of 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 VLAN or VPLS routing instance using the **vlan-id** or **vlan-tags** statements.

To configure a VLAN identifier for a VLAN, include either the **vlan-id** or the **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 VLAN configuration.

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.

> **NOTE:** ACX Series routers do not support the **[edit logical-systems]** hierarchy.

> **NOTE:** For a single VLAN or VPLS routing instance, you can include either the **vlan-id** or the **vlan-tags** statement, but not both. If you do not configure a **vlan-id**, **vlan-tags**, or **vlan-id-list [*vlan-id-numbers*]** for the VLAN or the VPLS routing instance, the Layer 2 packets received are forwarded to the outbound Layer 2 interface without having the VLAN tag modified unless an **output-vlan-map** is configured on the Layer 2 interface. This results in a frame being forwarded to a Layer 2 interface with a VLAN tag that is different from what is configured for the Layer 2 interface. Note that a frame received from the Layer 2 interface is still required to match the VLAN tag(s) specified in the interface configuration. The invalid configuration may cause a Layer 2 loop to occur.

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 112 on page 784. 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 VLAN 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 VLAN within a routing instance. Each VLAN 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 113 on page 785.

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

We recommend that you do not use customer VLAN IDs in a VPLS routing instance because customer VLAN IDs are used for learning only.

You should use the service VLAN ID in a VPLS routing instance, as in the following configuration:

```
[edit]
interface ge-1/1/1 {
   vlan-tagging;
   unit 1 {
      vlan-id s1; /* Service vlan */
      encapsulation vlan-vpls;
      input-vlan-map pop; /* Pop the service vlan on input */
      output-vlan-map push; /* Push the service vlan on output */
   }
}
interface ge-1/1/2 {
```

```
    encapsulation ethernet-vpls;
    unit 0;
}
routing-instances {
    V1 {
        instance-type vpls;
        vlan-id all;
        interface ge-1/1/1.1;
        interface ge-1/1/2.0;
    }
}
```

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 double-tagged frames on scaling. You cannot use the native **vlan- id** statement when the **vlan-id all** statement is included in the configuration.

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 VLAN 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 112 on page 784.

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 VLAN 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 VLAN or VPLS routing instance, the VLAN tags are rewritten as described in Table 113 on page 785.

The tables below show how VLAN tags are applied for traffic sent to and from the VLAN, depending on how the **vlan-id** and **vlan-tags** statements are configured for the VLAN and on how identifiers are configured for the logical interfaces in a VLAN 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 112 on page 784 shows specific examples of how the VLAN tags for packets sent to the VLAN 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 112: Statement Usage and Input Rewrite Operations for VLAN Identifiers for a VLAN

	VLAN Configurations for a VLAN			
VLAN Identifier of Logical Interface	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

Table 112: Statement Usage and Input Rewrite Operations for VLAN Identifiers for a VLAN *(continued)*

VLAN Identifier of Logical Interface	VLAN Configurations for a VLAN			
	vlan-id none	vlan-id 200	vlan-id all	vlan tags outer 100 inner 300
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 113 on page 785 shows specific examples of how the VLAN tags for packets sent from the VLAN 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 113: Statement Usage and Output Rewrite Operations for VLAN Identifiers for a VLAN

VLAN Identifier of Logical Interface	VLAN Configurations for a VLAN			
	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
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	–

CHAPTER 27

Configuring Multiple VLAN Registration Protocol (MVRP)

Multiple VLAN Registration Protocol

IN THIS SECTION

Understanding Multiple VLAN Registration Protocol (MVRP)

IN THIS SECTION

Multiple VLAN Registration Protocol (MVRP) is a Layer 2 messaging protocol that automates the creation and management of virtual LANs, thereby reducing the time you have to spend on these tasks. Use MVRP on Juniper Networks switches to dynamically register and unregister active VLANs on trunk interfaces. Using MVRP means that you do not have to manually register VLANs on all connections—that is, you do not need to explicitly bind a VLAN to each trunk interface. With MVRP, you configure a VLAN on one switch interface and the VLAN configuration is distributed through all active switches in the domain.

NOTE: MVRP is an application protocol of the Multiple Registration Protocol (MRP) and is defined in the IEEE 802.1ak standard. MRP and MVRP replace Generic Attribute Registration Protocol (GARP) and GARP VLAN Registration Protocol (GVRP) and overcome GARP and GVRP limitations.

NOTE: MVRP on QFabric systems does not support private VLANs.

If your QFabric system connects to servers that host many virtual machines that require their own VLANs, using MVRP can save you the time and effort that would be required to manually create and administer the VLANs on the ports that connect to the servers. For example, if a virtual machine moves between servers—and therefore connects to a different redundant server Node group interface—MVRP can configure the appropriate VLAN membership on the new server Node group interface.

When using MVRP on a QFabric system, you must manually create on the QFabric the VLANs that exist on the attached servers because the QFabric implementation of MVRP does not allow VLANs to be created dynamically. However, you do not need to manually assign VLAN membership to the QFabric ports that connect to the servers. MVRP automatically assigns VLAN membership to server-facing QFabric ports when it learns about a VLAN from an attached server.

MVRP Operations

MVRP stays synchronized by using MVRP protocol data units (PDUs). These PDUs specify which QFabric systems and switches are members of which VLANs, and which switch interfaces are in each VLAN. The MVRP PDUs are sent to other switches in the QFabric system when an MVRP state change occurs, and the receiving switches update their MVRP states accordingly. MVRP timers dictate when PDUs can be sent and when switches receiving MVRP PDUs can update their MVRP information.

In addition to this behavior, QFX switches include a mode—called passive mode—in which an MVRP-configured interface does not announce its membership in a VLAN or send any VLAN declarations (updates) unless it receives registration for that VLAN from a peer (server) on that interface. By default MVRP-configured interfaces behave in the standard manner and automatically send PDU updates to announce any VLAN changes. (This is called active mode.)

To enable passive mode on an interface, enter and commit this statement:

set protocols mvrp interface *interface-name* passive

To keep VLAN membership information current, MVRP removes switches and interfaces when they become unavailable. Pruning VLAN information has these benefits:

- Limits the network VLAN configuration to active participants, thereby reducing network overhead.
- Limits broadcast, unknown unicast, and multicast (BUM) traffic to interested devices.

MVRP is disabled by default and is valid only for trunk interfaces.

How MVRP Updates, Creates, and Deletes VLANs on Switches

When any MVRP-member VLAN is changed, that VLAN sends a protocol data unit (PDU) to all other MVRP-member active VLANs. The PDU informs the other VLANs which switches and interfaces currently belong to the sending VLAN. This way, all MVRP-member VLANs are always updated with the current VLAN state of all other MVRP-member VLANs. Timers dictate when PDUs can be sent and when switches receiving MVRP PDUs can update their MVRP VLAN information.

In addition to sending PDU updates, MVRP dynamically creates VLANs on member interfaces when a new VLAN is added to any one interface. This way, VLANs created on one member switch are propagated to other member switches as part of the MVRP message exchange process.

To keep VLAN membership information current, MVRP removes switches and interfaces when they become unavailable. Pruning VLAN information has these benefits:

- Limits the network VLAN configuration to active participants, thereby reducing network overhead.
- Limits broadcast, unknown unicast, and multicast (BUM) traffic to interested devices.

MVRP Is Disabled by Default on Switches

MVRP is disabled by default on the switches and, when enabled, affects only trunk interfaces. Once you enable MVRP, all VLAN interfaces on the switch belong to MVRP (the default **normal** registration mode) and those interfaces accept PDU messages and send their own PDU messages. To prevent one or more interfaces from participating in MVRP, you can specifically configure an interface to **forbidden** registration mode instead of the default **normal** mode.

VLAN updating, dynamic VLAN configuration through MVRP, and VLAN pruning are all active on trunk interfaces when MVRP is enabled.

MRP Timers Control MVRP Updates

MVRP registration and updates are controlled by timers that are part of the MRP. The timers define when MVRP PDUs can be sent and when MVRP information can be updated on a switch.

The timers are set on a per-interface basis, and on EX Series switches that use Juniper Networks Junos operating system (Junos OS) with support for the Enhanced Layer 2 Software (ELS) configuration style, the timers are also set on a per-switch basis.

On an EX Series switch that uses Junos OS with support for ELS, if the timer value set on an interface level is different from the value set on a switch level, the value on the interface level takes precedence.

The following MRP timers are used to control the operation of MVRP:

- Join timer—Controls the interval for the next MVRP PDU transmit opportunity.
- Leave timer—Controls the period of time that an interface on the switch waits in the leave state before changing to the unregistered state.
- LeaveAll timer—Controls the frequency with which the interface generates LeaveAll messages.

> **BEST PRACTICE:** Unless there is a compelling reason to change the timer settings, leave the default settings in place. Modifying timers to inappropriate values can cause an imbalance in the operation of MVRP.

MVRP Uses MRP Messages to Transmit Switch and VLAN States

MVRP uses MRP messages to register and declare MVRP states for a switch or VLAN and to inform the switching network that a switch or VLAN is leaving MVRP. These messages are communicated as part of the PDU sent by any switch interface to the other switches in the network.

The following MRP messages are communicated for MVRP:

- Empty—MVRP information is not declared and no VLAN is registered.
- In—MVRP information is not declared but a VLAN is registered.
- JoinEmpty—MVRP information is declared but no VLAN is registered.
- JoinIn—MVRP information is declared and a VLAN is registered.
- Leave—MVRP information that was previously declared is withdrawn.
- LeaveAll—Unregister all VLANs on the switch. VLANs must re-register to participate in MVRP.
- New—The MVRP information is new and a VLAN might not be registered yet.

Compatibility Issues with Junos OS Releases of MVRP

Except in Junos OS Releases 11.2 and earlier, MVRP has conformed with IEEE standard 802.1ak and IEEE Draft 802.1Q regarding the inclusion of an extra byte in the protocol data units (PDUs) sent and received by MVRP. Table 114 on page 791 outlines the MVRP versions and whether or not each version includes the extra byte in the PDU. Table 114 on page 791 also labels each MVRP version with a scenario number, which is used throughout the remainder of this discussion for brevity.

Table 114: Junos OS MVRP Versions and Inclusion of Extra Byte in PDU

MVRP in Junos OS Releases 11.2 and Earlier For EX Series Switches That Do Not Support Enhanced Layer 2 Software (ELS) Configuration Style Scenario 1	MVRP in Junos OS Releases 11.3 and Later For EX Series Switches That Do Not Support ELS Scenario 2	MVRP in Junos OS Releases 13.2 and Later For EX Series Switches With Support For ELS Scenario 3
Includes extra byte in the PDU	By default, does not include extra byte in the PDU	By default, includes extra byte in the PDU

As a result of the non-conformance of Releases 11.2 and earlier and changes in the standards with regard to the extra byte, a compatibility issue exists between some of the Junos OS versions of MVRP. This issue can result in some versions of MVRP not recognizing PDUs without the extra byte.

To address this compatibility issue, the MVRP versions described in scenarios 2 and 3 include the ability to control whether or not the PDU includes the extra byte. Before using these controls, however, you must understand each scenario that applies to your environment and plan carefully so that you do not inadvertently create an additional compatibility issue between the MVRP versions in scenarios 2 and 3.

Table 115 on page 791 provides a summary of environments that include the various MVRP scenarios and whether or not a particular environment requires you to take action.

Table 115: MVRP Environments and Description of Required Actions

Environment	Action Required?	Action Description
Includes MVRP versions in scenario 1 only	No	—
Includes MVRP versions in scenario 2 only	No	—
Includes MVRP versions in scenario 3 only	No	—

Table 115: MVRP Environments and Description of Required Actions *(continued)*

Environment	Action Required?	Action Description
Includes MVRP versions in scenarios 1 and 2. MVRP version in scenario 2 is in its default state.	Yes	On switches running MVRP version in scenario 2, use the **add-attribute-length-in-pdu** statement. For more information, see "Configuring Multiple VLAN Registration Protocol (MVRP) on Switches" on page 797.
Includes MVRP versions in scenarios 1 and 3. MVRP version in scenario 3 is in its default state.	No	—
Includes MVRP versions in scenarios 2 and 3, and both versions are in their respective default states	Yes	Do one of the following: • On switches running MVRP version in scenario 2, use the **add-attribute-length-in-pdu** statement. For more information, see "Configuring Multiple VLAN Registration Protocol (MVRP) on Switches" on page 797. • On switches running MVRP version in scenario 3, use the **no-attribute-length-in-pdu** statement. For more information, see "Configuring Multiple VLAN Registration Protocol (MVRP) on Switches" on page 797.

QFabric Requirements

When configuring MVRP on a QFabric system, you can enable it globally or enable it only on the trunk ports that need to carry VLAN traffic from the attached servers. You also must manually create the expected VLANs, but you do not have to assign VLAN membership to the server-facing redundant server Node ports (as mentioned previously). However, you *do* have to manually assign VLAN membership to the uplink ports on the redundant server Node group and network Node group devices that will carry the VLAN traffic. Table 116 on page 792 summarizes the VLAN requirements for redundant server Node groups and network Node groups:

Table 116: MVRP VLAN Requirements for Node Devices

Node Group Type	Interface	Assign VLAN Membership to Trunk Ports?
Redundant server Node group	Server-facing trunk	No
Redundant server Node group	Uplink trunk (to interconnect device)	Yes
Network Node groups	Uplink trunk (to interconnect device)	Yes

Determining Whether MVRP is Working

You can determine whether the switches in your network are running incompatible versions of MVRP by issuing the **show mvrp statistics** command. For more information on diagnosing and correcting this MVRP compatibility situation, see "Configuring Multiple VLAN Registration Protocol (MVRP) on Switches" on page 797.

SEE ALSO

- Understanding Bridging and VLANs on Switches | **168**

Understanding Multiple VLAN Registration Protocol (MVRP) for Dynamic VLAN Registration

IN THIS SECTION

Multiple VLAN Registration Protocol (MVRP) is a Layer 2 messaging protocol that manages the addition, deletion, and renaming of active virtual LANs, thereby reducing network administrators' time spent on these tasks. Use MVRP on Juniper Networks MX Series routers, EX Series switches and SRX devices to dynamically register and unregister active VLANs on trunk interfaces. Using MVRP means that you do not have to manually register VLANs on all connections—that is, you do not need to explicitly bind a VLAN to each trunk interface. With MVRP, you configure a VLAN on one interface and the VLAN configuration is distributed through all active interfaces in the domain.

The primary purpose of MVRP is to manage dynamic VLAN registration in Layer 2 networks. In managing dynamic VLAN registration, MVRP also prunes VLAN information.

MVRP is an Layer 2 application protocol of the Multiple Registration Protocol (MRP) and is defined in the IEEE 802.1ak standard. MRP and MVRP were designed by IEEE to perform the same functions as Generic Attribute Registration Protocol (GARP) and GARP VLAN Registration Protocol (GVRP) while overcoming some GARP and GVRP limitations, in particular, limitations involving bandwidth usage and convergence time in large networks with large numbers of VLANs.

MVRP was created by IEEE as a replacement application for GVRP. MVRP and GVRP cannot be run concurrently to share VLAN information in a Layer 2 network.

This topic describes:

How MVRP Works

When any MVRP-member VLAN is changed, that VLAN sends a protocol data unit (PDU) to all other MVRP-member active VLANs. The PDU informs the other VLANs which devices and interfaces currently belong to the sending VLAN. This way, all MVRP-member VLANs are always updated with the current VLAN state of all other MVRP-member VLANs. Timers dictate when PDUs can be sent and when devices receiving MVRP PDUs can update their MVRP VLAN information.

The VLAN registration information sent by MVRP protocol data units (PDUs) includes the current VLANs membership—that is, which routers are members of which VLANs—and which router interfaces are in which VLAN. MVRP shares all information in the PDU with all routers participating in MVRP in the Layer 2 network.

MVRP stays synchronized using these PDUs. The routers in the network participating in MVRP receive these PDUs during state changes and update their MVRP states accordingly. MVRP timers dictate when PDUs can be sent and when routers receiving MVRP PDUs can update their MVRP information.

In addition to sending PDU updates, MVRP dynamically creates VLANs on member interfaces when a new VLAN is added to any one interface. This way, VLANs created on one member device are propagated to other member devices as part of the MVRP message exchange process.

VLAN information is distributed as part of the MVRP message exchange process and can be used to dynamically create VLANs, which are VLANs created on one switch and propagated to other routers as

part of the MVRP message exchange process. Dynamic VLAN creation using MVRP is enabled by default, but can be disabled.

As part of ensuring that VLAN membership information is current, MVRP removes routers and interfaces from the VLAN information when they become unavailable. Pruning VLAN information has these benefits:

- Limits the network VLAN configuration to active participants only, reducing network overhead.
- Targets the scope of broadcast, unknown unicast, and multicast (BUM) traffic to interested devices only.

Using MVRP

MVRP is disabled by default on the devices and, when enabled, affects only trunk interfaces. Once you enable MVRP, all VLAN interfaces on the device belong to MVRP (the default **normal** registration mode) and those interfaces accept PDU messages and send their own PDU messages. To prevent one or more interfaces from participating in MVRP, you can specifically configure an interface to **forbidden** registration mode instead of the default **normal** mode.

VLAN updating, dynamic VLAN configuration through MVRP, and VLAN pruning are all active on trunk interfaces when MVRP is enabled.

MVRP Registration Modes

The MVRP registration mode defines whether an interface does or does not participate in MVRP.

The following MVRP registration modes are configurable:

- forbidden—The interface does not register or declare VLANS (except statically configured VLANs).
- normal—The interface accepts MVRP messages and participates in MVRP. This is the default registration mode setting.
- restricted—The interface ignores all MVRP JOIN messages received for VLANs that are not statically configured on the interface.

MRP Timers Control MVRP Updates

MVRP registration and updates are controlled by timers that are part of the MRP protocol. These timers are set on a per-interface basis and define when MVRP PDUs can be sent and when MVRP information can be updated on a switch.

The following timers are used to control the operation of MVRP:

- Join timer—Controls the interval for the next MVRP PDU transmit opportunity.
- Leave timer—Controls the period of time that an interface on the switch waits in the Leave state before changing to the unregistered state.

- LeaveAll timer—Controls the frequency with which the interface generates LeaveAll messages.

> **BEST PRACTICE:** Maintain default timer settings unless there is a compelling reason to change the settings. Modifying timers to inappropriate values might cause an imbalance in the operation of MVRP.

MVRP Uses MRP Messages to Transmit Device and VLAN States

MVRP uses MRP messages to register and declare MVRP states for a switch and to inform the Layer 2 network that a switch is leaving MVRP. These messages are communicated as part of the PDU to communicate the state of a particular switch interface on the Layer 2 network to the other switches in the network.

The following messages are communicated for MVRP:

- Empty—VLAN information is not being declared and is not registered.
- In—VLAN information is not being declared but is registered.
- JoinEmpty—VLAN information is being declared but not registered.
- JoinIn—VLAN information is being declared and is registered.
- Leave—VLAN information that was previously registered is being withdrawn.
- LeaveAll—All registrations will be de-registered. Participants that want to participate in MVRP will need to re-register.
- New—VLAN information is new and possibly not previously registered.

MVRP Limitations

The following limitations apply when configuring MVRP:

- MVRP works with Rapid Spanning Tree Protocol (RSTP) and Multiple Spanning Tree Protocol (MSTP), but not with VLAN Spanning Tree Protocol (VSTP).
- MVRP is allowed only on single tagged trunk ports.
- MVRP is not allowed if a physical interface has more than one logical interface.
- MVRP is only allowed if a logical has one trunk interface (unit 0).

Configuring Multiple VLAN Registration Protocol (MVRP) on Switches

IN THIS SECTION

Multiple VLAN Registration Protocol (MVRP) is used to manage dynamic VLAN registration in a LAN. You can use MVRP on QFX switches and on EX Series switches that support or do not support ELS.

MVRP is disabled by default.

To enable MVRP or set MVRP options, follow these instructions:

Enabling MVRP on Switches With ELS Support

This example uses Junos OS for EX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style. MVRP can only be enabled on trunk interfaces.

> **NOTE:** For ELS details, see “Using the Enhanced Layer 2 Software CLI” on page 50.

MVRP can only be enabled on trunk interfaces. To enable MVRP on a trunk interface:

```
[edit protocols mvrp]
user@switch# set interface interface-name
```

Enabling MVRP on Switches Without ELS Support

This example uses Junos OS for EX Series switches that do not support the Enhanced Layer 2 Software (ELS) configuration style. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50.

Multiple VLAN Registration Protocol (MVRP) is used to manage dynamic VLAN registration in a LAN. You can use MVRP on EX Series switches.

MVRP is disabled by default on EX Series switches.

MVRP can only be enabled on trunk interfaces. To enable MVRP on a trunk interface:

```
[edit protocols mvrp]
user@switch# set interface all
```

To enable MVRP on a specific trunk interface:

```
[edit protocols mvrp]
user@switch# set interface xe-0/0/1.0
```

Enabling MVRP on Switches With QFX Support

Multiple VLAN Registration Protocol (MVRP) automates the creation and management of VLANs. When using MVRP on a QFabric system, you must manually create on the QFabric the VLANs that exist on the attached servers because the QFabric implementation of MVRP does not allow VLANs to be created dynamically. However, you do not need to manually assign VLAN membership to the QFabric ports that connect to the servers. MVRP automatically assigns VLAN membership to server-facing QFabric ports when it learns about a VLAN from an attached server. .

MVRP is disabled by default. To enable MVRP or set MVRP options, follow these instructions:

This example uses Junos OS for EX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style. MVRP can only be enabled on trunk interfaces.

NOTE: For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50.

MVRP can only be enabled on trunk interfaces. To enable MVRP on a trunk interface:

```
[edit protocols mvrp]
user@qfabric# set interface interface-name
```

NOTE: On QFX Series switches, you must configure specific interfaces—you cannot specify **interface all**. You can enable MVRP on an interface range.

Disabling MVRP

MVRP is disabled by default. Perform this procedure only if you have previously enabled MVRP.

You can disable MVRP globally only. To disable MVRP on all trunk interfaces on a switch with ELS support, use one of the following commands:

```
user@switch# deactivate protocols mvrp
user@switch# delete protocols mvrp
```

To disable MVRP on all trunk interfaces of a QFX switch, an EX switch without ELS Support or an entire QFabric system:

```
[edit protocols mvrp]
user@switch# set disable
```

To disable MVRP on a specific trunk QFX switch or an EX switch without interface support:

```
[edit protocols mvrp]
user@qfabric# set disable interface interface-name
```

```
[edit protocols mvrp]
user@switch# set disable interface xe-0/0/1.0
```

SEE ALSO

disable | 1117

add-attribute-length-in-pdu | 1086

no-attribute-length-in-pdu | 1280

Disabling Dynamic VLANs on EX Series Switches

By default, dynamic VLANs can be created on interfaces participating in MVRP. Dynamic VLANs are VLANs created on one switch that are propagated to other switches dynamically, in this case, using MVRP.

Dynamic VLAN creation through MVRP cannot be disabled per switch interface. To disable dynamic VLAN creation for interfaces participating in MVRP, you must disable it for all interfaces on the switch.

To disable dynamic VLAN creation:

```
[edit protocols mvrp]
user@switch# set no-dynamic-vlan
```

SEE ALSO

no-dynamic-vlan | 1281

add-attribute-length-in-pdu | 1086

no-attribute-length-in-pdu | 1280

Configuring Timer Values

The timers in MVRP define the amount of time all interfaces on a switch or a specific interface wait to join or leave MVRP, or to send or process the MVRP information for the switch after receiving an MVRP PDU. The join timer controls the amount of time the switch waits to accept a registration request, the leave timer controls the period of time that the switch waits in the Leave state before changing to the unregistered state, and the leaveall timer controls the frequency with which the LeaveAll messages are communicated.

The default MVRP timer values are 200 ms for the join timer, 1000 ms for the leave timer, and 10 seconds for the leaveall timer.

BEST PRACTICE: Maintain default timer settings unless there is a compelling reason to change the settings. Modifying timers to inappropriate values might cause an imbalance in the operation of MVRP.

On an EX Series switch that uses Junos OS with support for ELS, if the timer value set on an interface level is different from the value set on a switch level, then the value on the interface level takes precedence.

To set the join timer for all interfaces on the switch:

```
[edit protocols mvrp]
user@switch# set join-timer milliseconds
```

```
[edit protocols mvrp]
user@switch# set interface all join-timer 300
```

To set the join timer for a specific interface:

```
[edit protocols mvrp]
user@switch# set interface interface-name join-timer milliseconds
```

```
[edit protocols mvrp]
user@qfabric# set interface interface-name 300
```

```
[edit protocols mvrp]
user@switch# set interface xe-0/0/1.0 300
```

To set the leave timer for all interfaces on the switch:

```
[edit protocols mvrp]
user@switch# set leave-timer milliseconds
```

```
[edit protocols mvrp]
user@switch# set interface all leave-timer 1200
```

To set the leave timer for a specific interface:

```
[edit protocols mvrp]
user@switch# set interface interface-name leave-timer milliseconds
```

```
[edit protocols mvrp]
user@qfabric# set interface interface-name leave-timer 1200
```

To set the leaveall timer for all interfaces on the switch:

```
[edit protocols mvrp]
user@switch# set leaveall-timer seconds
```

```
[edit protocols mvrp]
user@qfabric# set interface interface-name leaveall-timer 12000
```

```
[edit protocols mvrp]
user@switch# set interface all leaveall-timer 12000
```

To set the leaveall timer for a specific interface:

```
[edit protocols mvrp]
user@switch# set interface interface-name leaveall-timer seconds
```

```
[edit protocols mvrp]
user@switch# set interface xe-0/0/1.0 leaveall-timer 12000
```

SEE ALSO

- join-timer (MVRP) | 1227
- leave-timer (MVRP) | 1238
- leaveall-timer (MVRP) | 1240

Configuring Passive Mode on QFX Switches

QFX switches include a mode—called passive mode—in which an MVRP-configured interface does not announce its membership in a VLAN or send any VLAN declarations (updates) unless it receives registration for that VLAN from a peer (server).

To configure an interface to operate in passive mode:

```
[edit protocols mvrp]
user@qfabric# set interface interface-name passive
```

Configuring MVRP Registration Mode on EX Switches

NOTE: Not supported in QFabric.

The default MVRP registration mode for any interface participating in MVRP is normal. An interface in normal registration mode participates in MVRP when MVRP is enabled on the switch.

You can change the registration mode of a specific interface to **forbidden**. An interface in forbidden registration mode does not participate in MVRP even if MVRP is enabled on the switch.

To set an interface to forbidden registration mode:

```
[edit protocols mvrp]
user@switch# set interface xe-0/0/1.0 registration forbidden
```

```
[edit protocols mvrp]
user@switch# set interface all registration forbidden
```

To set an interface to normal registration mode:

```
[edit protocols mvrp]
user@switch# set interface xe-0/0/1.0 registration normal
```

```
[edit protocols mvrp]
user@switch# set interface xe-0/0/1.0 registration normal
```

To set all interfaces to normal registration mode:

```
[edit protocols mvrp]
user@switch# set interface all registration normal
```

SEE ALSO

registration | 1340

Using MVRP in a Mixed-Release EX Series Switching Network

Except in Junos OS Releases 11.2 and earlier, MVRP has conformed with IEEE standard 802.1ak and IEEE Draft 802.1Q regarding the inclusion of an extra byte in the protocol data units (PDUs) sent and received by MVRP.

As a result of the non-comformance of releases 11.2 and earlier and changes in the standards regarding the extra byte, the following mixed environments can arise in EX Series switches without ELS support:

- Mixed environment A: MVRP in Junos OS Releases 11.2 and earlier includes the extra byte, while MVRP in Junos OS Releases 11.3 and later for EX Series switches that do not support the Enhanced Layer 2 Software (ELS) configuration style does not include the extra byte.
- Mixed environment B: MVRP in Junos OS Releases 13.2 and later for EX Series switches with support for ELS includes the extra byte, while MVRP in Junos OS Releases 11.3 and later for EX Series switches that do not support ELS does not include the extra byte.

As a result of changes in the standards with regard to the extra byte, MVRP in Junos OS Releases 13.2 and later for EX Series switches with support for the Enhanced Layer 2 Software (ELS) includes the extra byte, while MVRP in Junos OS Releases 11.3 and later for EX Series switches that do not support ELS does not include the extra byte. A compatibility issue arises, wherein the ELS version of MVRP does not recognize PDUs without the extra byte sent by the non-ELS version of MVRP.

A compatibility issue arises in mixed environments A and B, wherein the versions of MVRP that include the extra byte do not recognize PDUs that do not include the extra byte.

If your network has a mix of MVRP versions, you can alter MVRP on the switches running Release 11.3 and later on switches that do not support ELS so they include the extra byte in the PDU and are therefore, compatible with the other MVRP versions.

A compatibility issue arises in mixed environments A and B, wherein the versions of MVRP that include the extra byte do not recognize PDUs that do not include the extra byte.

For more information about these issues, see "Understanding Multiple VLAN Registration Protocol (MVRP)" on page 787.

To make MVRP on on switches that do not support ELS (Release 11.3 or later) compatible with MVRP in the other releases:

```
[edit protocols mvrp]
user@switch# set add-attribute-length-in-pdu
```

If your network includes a mix of EX Series switches running ELS and non-ELS versions of MVRP, you can eliminate the compatibility issue by entering the following command on the switches running the ELS version of MVRP:

```
[edit protocols mvrp]
user@switch# set no-attribute-length-in-pdu
```

The **no-attribute-length-in-pdu** statement prevents the ELS version of MVRP from sending PDUs with the extra byte, thereby eliminating the compatibility issue with the non-ELS version of MVRP.

You can recognize an MVRP version compatibility issue by observing the switch running the ELS version of MVRP. Because a switch running the ELS version of MVRP cannot interpret an unmodified PDU from a switch running the non-ELS version of MVRP, the switch will not add VLANs from the non-ELS version

of MVRP. When you use the **show mvrp statistics** command in the ELS version of MVRP, the values for **Received Join Empty** and **Received Join In** will incorrectly display zero, even though the value for the **Received MVRP PDUs without error** has been increased. Another indication that MVRP is having a version compatibility issue is that unexpected VLAN activity, such as multiple VLAN creation, takes place on the switch running the ELS version of MVRP.

Configuring Multiple VLAN Registration Protocol (MVRP) to Manage Dynamic VLAN Registration on Security Devices

IN THIS SECTION

Starting in Junos OS Release 15.1X49-D80, Multiple VLAN Registration Protocol (MVRP) to manage dynamic VLAN registration is supported on SRX1500 devices. Multiple VLAN Registration Protocol (MVRP) is used to manage dynamic VLAN registration in a Layer 2 network. You can configure MVRP on SRX Series devices.

MVRP is disabled by default on SRX Series devices.

To enable MVRP and to set MVRP options, follow these instructions:

Enabling MVRP

MVRP can be enabled only on trunk interfaces.

To enable MVRP on a specific trunk interface (here, interface ge-0/0/1):

```
[edit protocols mvrp]
user@host# set interface ge-0/0/1
```

Changing the Registration Mode to Disable Dynamic VLANs

When the registration mode for an interface is set to **normal** (the default), dynamic VLANs are created on interfaces participating in MVRP. The dynamic VLANs created on one SRX Series device are then propagated by means of MVRP to other SRX Series devices in the topology.

However, dynamic VLAN creation through MVRP can be disabled for all trunk interfaces or for individual trunk interfaces.

Configuring Timer Values

The timers in MVRP define the amount of time an interface waits to join or leave MVRP or to send or process the MVRP information for the router or switch after receiving an MVRP PDU:

- The join timer controls the amount of time the router or switch waits to accept a registration request.
- The leave timer controls the period of time that the router or switch waits in the Leave state before changing to the unregistered state.
- The leaveall timer controls the frequency with which the LeaveAll messages are communicated.

The default MVRP timer values are 200 ms for the join timer, 1000 ms for the leave timer, and 60 seconds for the leaveall timer.

> **BEST PRACTICE:** Maintain default timer settings unless there is a compelling reason to change the settings. Modifying timers to inappropriate values might cause an imbalance in the operation of MVRP.

To set the join timer at 300 ms for a specific interface (here, interface ge-0/0/1):

```
[edit protocols mvrp]
user@host# set interface ge-0/0/1 join-timer (MVRP) 300
```

To set the leave timer at 400 ms for a specific interface (here, interface ge-0/0/1):

```
[edit protocols mvrp]
user@host# set interface ge-0/0/1 leave-timer 400
```

To set the leaveall timer at 20 seconds for a specific interface (here, interface ge-0/0/1):

```
[edit protocols mvrp]
user@host# set interface ge-0/0/1 leaveall-timer 20
```

SEE ALSO

- join-timer (MVRP) | 1227
- leave-timer (MVRP) | 1238
- leaveall-timer (MVRP) | 1240

Configuring the Multicast MAC Address for MVRP

MVRP uses the customer MVRP multicast MAC address when MVRP is enabled. However, you can configure MVRP to use the provider MVRP multicast MAC address instead.

To configure MVRP to use the provider MVRP multicast MAC address:

```
[edit protocols mvrp]
user@host# set bpdu-destination-mac-address provider-bridge-group;
```

SEE ALSO

- bpdu-destination-mac-address | 1094

Configuring an MVRP Interface as a Point-to-Point Interface

Specify that a configured interface is connected point-to-point. If specified, a point-to-point subset of the MRP state machine provides a simpler and more efficient method to accelerate convergence on the network.

To specify that an MVRP interface is point-to-point (here, interface ge-0/0/1):

```
[edit protocols mvrp]
user@host# set interface ge-0/0/1 point-to-point (MVRP);
```

SEE ALSO

- point-to-point (MVRP) | 1301

Configuring MVRP Tracing Options

Set MVRP protocol-level tracing options.

To specify MVRP protocol tracing (here, the file is **/var/log/mvrp-log**, size is **2m**, number of files is **28**, the option **world-readable** indicates the log can be read by user, and MVRP is flagging **events**):

```
[edit protocols mvrp]
user@host# edit traceoptions file /var/log/mvrp-log size 2m files 28 world-readable flag events
```

Disabling MVRP

MVRP is disabled by default. You need to perform this procedure only if MVRP is previously enabled.

To disable MVRP on all trunk interfaces, use one of the following commands:

```
[edit]
user@host# deactivate protocols mvrp
user@host# delete protocols mvrp
```

SEE ALSO

Understanding VLANs

Example: Configuring Automatic VLAN Administration on QFX Switches Using MVRP

IN THIS SECTION

As the numbers of servers and VLANs attached to a QFabric systems increase, VLAN administration becomes complex and the task of efficiently configuring VLANs on multiple redundant server Node group devices becomes increasingly difficult. To partially automate VLAN administration, you can enable Multiple

VLAN Registration Protocol (MVRP) on your QFabric system. If your QFabric system connects to servers that host many virtual machines that require their own VLANs, using MVRP can save you the time and effort that would be required to manually configure and administer the VLANs on the interfaces that connect to the servers. For example, if a virtual machine moves between servers—and therefore connects to a different redundant server Node group interface—MVRP can configure the appropriate VLAN membership on the new server Node group interface.

> **NOTE:** Only trunk interfaces can be enabled for MVRP.

This example describes how to configure MVRP on a QFabric system.

Requirements

This example uses the following hardware and software components:

- One QFabric system
- Junos OS Release 13.1 for the QFX Series

Overview and Topology

MVRP ensures that the VLAN membership information on the trunk interface is updated as the switch's access interfaces become active or inactive in the configured VLANs in a static or dynamic VLAN creation setup.

You do not need to explicitly bind a VLAN to the trunk interface. When MVRP is enabled, the trunk interface advertises all the VLANs that are active (bound to access interfaces) on that switch. An MVRP-enabled trunk interface does not advertise VLANs that have been configured on the switch but that are not currently bound to an access interface. Thus, MVRP provides the benefit of reducing network overhead—by limiting the scope of broadcast, unknown unicast, and multicast (BUM) traffic to interested devices only.

When VLAN access interfaces become active or inactive, MVRP ensures that the updated information is advertised on the trunk interface. Thus, in this example, distribution Switch C does not forward traffic to inactive VLANs.

A redundant server Node group device is connected to a server that hosts virtual machines for three customers, each of which requires its own VLAN.

- **customer-1**: VLAN ID 100
- **customer-2**: VLAN ID 200
- **customer-3**: VLAN ID 300

Table 117 on page 810 explains the components of the example topology.

Table 117: Components of the Example Topology

Settings	Settings
Hardware	• Redundant server Node group device • Network Node group device
VLAN names and IDs	• **customer-1**, VLAN ID (tag)**100** • **customer-2**, VLAN ID (tag)**200** • **customer-3**, VLAN ID (tag)**300**
Interfaces	Redundant server Node group device interfaces: • **RSNG:xe-0/1/1**—Uplink to interconnect device • **RSNG:xe-0/0/1**—Server-facing interface Network Node group device interface: • **NNG:xe-0/0/1**—Uplink to interconnect device

Configuring VLANs and Network Node Group Interfaces

To configure VLANs, bind the VLANs to the server-facing trunk interface, and enable MVRP on the trunk interface of the network Node group device, perform these tasks:

CLI Quick Configuration

To quickly configure VLANs on the QFabric system, assign VLAN membership to the uplink port on the network Node group device, and configure the uplink port to be trunk:

```
[edit]
set vlans customer-1 vlan-id 100

set vlans customer-2 vlan-id 200

set vlans customer-3 vlan-id 300

set interfaces NNG:xe-0/0/1 unit 0 family ethernet-switching port-mode trunk

set interfaces NNG:xe-0/0/1 unit 0 family ethernet-switching vlan members [customer-1 customer-2
customer-3]
```

NOTE: As recommended as a best practice, default MVRP timers are used in this example, so they are not configured. The default values associated with each MVRP timer are: 200 ms for the join timer, 1000 ms for the leave timer, and 10000 ms for the leaveall timer. Modifying timers to inappropriate values might cause an imbalance in the operation of MVRP.

Step-by-Step Procedure

To create the VLANs and configure the network Node group device for MVRP, follow these steps. Note that you are creating VLANs for the entire QFabric system, so you do not need to create them on specific QFabric devices.

1. Configure the VLAN for customer 1:

```
[edit]
user@qfabric# set vlans customer-1 vlan-id 100
```

2. Configure the VLAN for customer 2:

```
[edit]
user@qfabric# set vlans customer-2 vlan–id 200
```

3. Configure the VLAN for customer 3:

```
[edit]
user@qfabric# set vlans customer-3 vlan–id 300
```

4. Configure an uplink interface (one that connects to an interconnect device) to be a trunk:

```
[edit]
user@qfabric# set interfaces NNG:xe-0/0/1 unit 0 family ethernet-switching port-mode trunk
```

5. Configure the uplink interface to be a member of all three VLANs:

```
[edit]
user@qfabric# set interfaces NNG:xe-0/0/1 unit 1 family ethernet-switching vlan members
[customer-1 customer-2 customer-3]
```

NOTE: If you want the uplink interface to be a member of all the VLANs in the QFabric system, you can enter **all** instead of specifying the individual VLANs.

Results

Check the results of the configuration on the network Node group device:

```
[edit]
user@qfabric# show interfaces NNG:xe-0/0/1.0
family ethernet-switching {
    port-mode trunk;
    vlan {
        members customer-1 customer-2 customer-3;
    }
}
```

```
[edit]
user@qfabric# show vlans
customer-1 {
    vlan-id 100;
}
customer-2 {
    vlan-id 200;
}
customer-3 {
    vlan-id 300;
}
```

Configuring the Redundant Server Node Group

CLI Quick Configuration

To quickly configure the redundant server Node group device for MVRP:

```
[edit]
```

set interfaces RSNG:xe-0/1/1 unit 0 family ethernet-switching port-mode trunk
set interfaces RSNG:xe-0/1/1 unit 0 family ethernet-switching vlan members [customer-1 customer-2 customer-3]
set interfaces RSNG:xe-0/0/1 unit 0 family ethernet-switching port-mode trunk
set protocols mvrp interface RSNG:xe-0/0/1.0 passive

Step-by-Step Procedure

To configure the redundant server Node group device, follow these steps. Note that you do not need to configure the VLANs on the server-facing interface (RSNG:xe-0/0/1), but you do need to configure the VLANs on the uplink interface. Also notice that in this example you configure the server-facing interface to be passive, which means that it will not announce its membership in a VLAN or send any VLAN declarations (updates) unless it receives registration for that VLAN from the server.

1. Configure an uplink interface (one that connects to the interconnect device) to be a trunk:

   ```
   [edit]
   user@qfabric# set interfaces RSNG:xe-0/1/1 unit 0 family ethernet-switching port-mode trunk
   ```

2. Configure the uplink interface to be a member of all three VLANs:

   ```
   [edit]
   user@qfabric# set interfaces NNG:xe-0/1/1 unit 0 family ethernet-switching vlan members
   [customer-1 customer-2 customer-3]
   ```

3. Configure an interface that connects to the server that hosts multiple virtual machines to be a trunk:

   ```
   [edit]
   user@qfabric# set interfaces RSNG:xe-0/0/1 unit 0 family ethernet-switching port-mode trunk
   ```

4. Enable MVRP on the server-facing trunk interface and configure it to be passive:

   ```
   [edit]
   user@qfabric# set protocols mvrp interface RSNG:xe-0/0/1.0 passive
   ```

Results

Check the results of the configuration for the redundant server Node group:

```
[edit]
user@qfabric# show interfaces RSNG:xe-0/0/1.0
family ethernet-switching {
   port-mode trunk;
}
```

```
[edit]
user@qfabric# show interfaces RSNG:xe-0/1/1.0
family ethernet-switching {
   port-mode trunk;
   }
passive
}
```

```
[edit]
user@qfabric# show protocols mvrp
interface RSNG:xe-0/0/1.0;
```

Verification

IN THIS SECTION

- Verifying That MVRP Is Enabled On The QFabric System | **814**

To confirm that the configuration is updating VLAN membership, perform these tasks:

Verifying That MVRP Is Enabled On The QFabric System

Purpose

Verify that MVRP is enabled on the appropriate interfaces

Action

Show the MVRP configuration:

```
user@qfabric> show mvrp
```

```
MVRP configuration
MVRP status                 : Enabled

MVRP timers (ms):
Interface          Join   Leave       LeaveAll
--------------     -----  --------    -----------
NNG:xe-0/0/1.0     200     1000       10000
RSNG:xe-0/0/1.0    200     1000       10000
RSNG:xe-0/1/1.0    200     1000       10000


Interface           Status        Registration Mode
--------------  --------      -----------------
NNG:xe-0/0/1.0      Enabled       Normal
RSNG:xe-0/1/1.0     Enabled       Normal
RSNG:xe-0/0/1.0     Enabled       Passive
```

Meaning

The results show that MVRP is enabled on the appropriate network Node group and redundant server Node group interfaces and that the default timers are used.

Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches with ELS Support

IN THIS SECTION

> **NOTE:** This example uses Junos OS for EX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does not support ELS, see "Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches" on page 832. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50.

As a network expands and the number of clients and VLANs increases, VLAN administration becomes complex and the task of efficiently configuring VLANs on multiple EX Series switches becomes increasingly difficult. However, you can automate VLAN administration by enabling Multiple VLAN Registration Protocol (MVRP) on the network.

MVRP also dynamically creates VLANs, further simplifying the network overhead required to statically configure VLANs.

> **NOTE:** Only trunk interfaces can be enabled for MVRP.

This example describes how to use MVRP to automate administration of VLAN membership changes within your network and how to use MVRP to dynamically create VLANs:

Requirements

This example uses the following hardware and software components:

- Two EX Series access switches
- One EX Series distribution switch
- Junos OS Release 13.2X50-D10 or later for EX Series switches

Before you configure MVRP on an interface, you must enable one of the following spanning tree protocols on that interface:

- Rapid Spanning-Tree Protocol (RSTP). For more information about RSTP, see *Understanding RSTP* .
- Multiple Spanning-Tree Protocol (MSTP). For more information about MSTP, see *Understanding MSTP*.

Overview and Topology

MVRP is used to manage dynamic VLAN registration in a LAN. It can also be used to dynamically create VLANs.

This example uses MVRP to dynamically create VLANs on the switching network. Alternatively, you can disable dynamic VLAN creation and create VLANs statically. Enabling MVRP on the trunk interface of each switch in your switching network ensures that the active VLAN information for the switches in the network is propagated to each switch through the trunk interfaces, assuming dynamic VLAN creation is enabled for MVRP.

MVRP ensures that the VLAN membership information on the trunk interface is updated as the switch's access interfaces become active or inactive in the configured VLANs in a static or dynamic VLAN creation setup.

You do not need to explicitly bind a VLAN to the trunk interface. When MVRP is enabled, the trunk interface advertises all the VLANs that are active (bound to access interfaces) on that switch. An MVRP-enabled trunk interface does not advertise VLANs that are configured on the switch but are not currently bound to an access interface. Thus, MVRP provides the benefit of reducing network overhead—by limiting the scope of broadcast, unknown unicast, and multicast (BUM) traffic to interested devices only.

When VLAN access interfaces become active or inactive, MVRP ensures that the updated information is advertised on the trunk interface. Thus, in this example, distribution Switch C does not forward traffic to inactive VLANs.

NOTE: This example shows a network with three VLANs: **finance**, **sales**, and **lab**. All three VLANs are running the same version of Junos OS. If switches in this network were running a mix of Junos OS releases that included Release 11.3, additional configuration would be necessary—see "Configuring Multiple VLAN Registration Protocol (MVRP) on Switches" on page 797 for details.

Figure 42 on page 817 shows MVRP configured on two access switches and one distribution switch.

Figure 42: MVRP Configured on Two Access Switches and One Distribution Switch for Automatic VLAN Administration

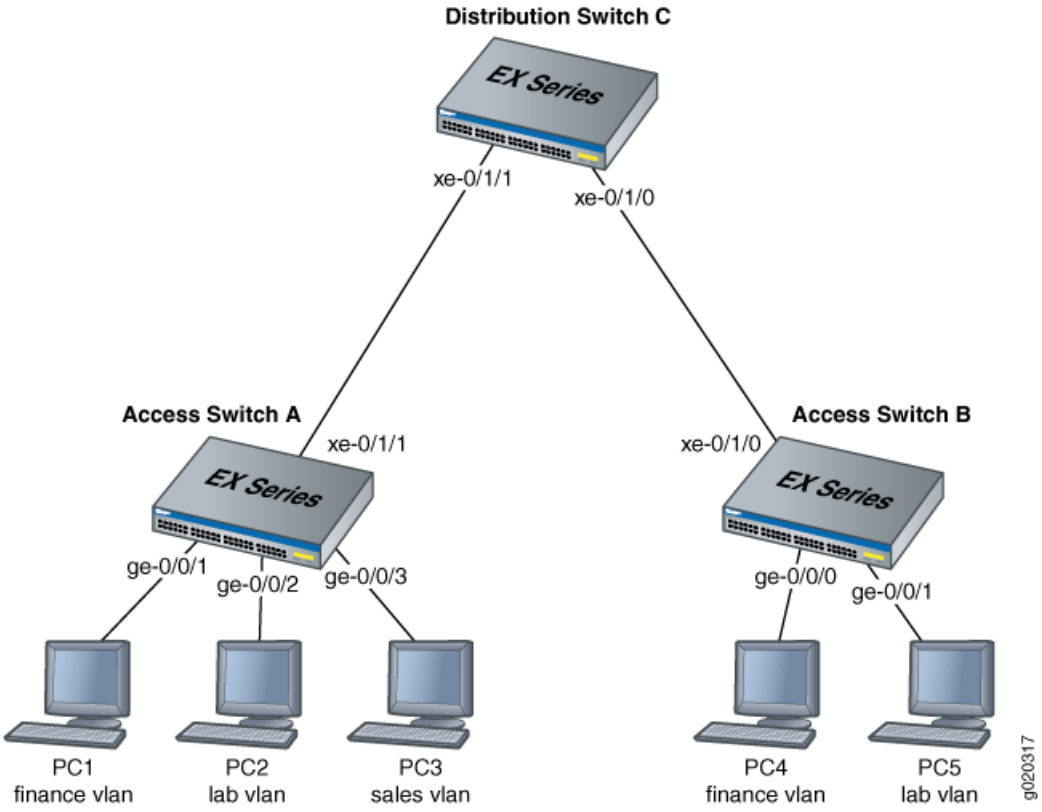


Table 118 on page 817 explains the components of the example topology.

Table 118: Components of the Network Topology

Settings	Settings
Switch hardware	• Access Switch A • Access Switch B • Distribution Switch C

Table 118: Components of the Network Topology *(continued)*

Settings	Settings
VLAN names and tag IDs	**finance**, tag **100** **lab**, tag **200** **sales**, tag **300**
Interfaces	Access Switch A interfaces: • ge-0/0/1—Connects PC1 to access Switch A. • ge-0/0/2—Connects PC2 to access Switch A. • ge-0/0/3—Connects PC3 to access Switch A. • xe-0/1/1—Connects access Switch A to distribution Switch C (trunk). Access Switch B interfaces: • ge-0/0/0—Connects PC4 to access Switch B. • ge-0/0/1—Connects PC5 to access Switch B. • ge-0/0/2—Reserved for future use, • xe-0/1/0—Connects access Switch B to distribution Switch C. (trunk) Distribution Switch C interfaces: • xe-0/1/1—Connects distribution Switch C to access Switch A. (trunk) • xe-0/1/0—Connects distribution Switch C to access Switch B. (trunk)

Configuring VLANs and MVRP on Access Switch A

To configure VLANs on the switch, bind access interfaces to the VLANs, and enable MVRP on the trunk interface of access Switch A, perform these tasks:

CLI Quick Configuration

To quickly configure access Switch A for MVRP, copy the following commands and paste them into the switch terminal window of Switch A:

```
[edit]
set vlans finance vlan-id 100

set vlans lab vlan-id 200

set vlans sales vlan-id 300
```

set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members finance

set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan members lab

set interfaces ge-0/0/3 unit 0 family ethernet-switching vlan members sales

set interfaces xe-0/1/1 unit 0 family ethernet-switching interface-mode trunk

set protocols mvrp interface xe-0/1/1

> **NOTE:** This example uses default MVRP timers. The default values associated with each MVRP timer are: 200 ms for the join timer, 1000 ms for the leave timer, and 10000 ms (10 seconds) for the leaveall timer. We recommend retaining the use of default timer values as modifying timers to inappropriate values might cause an imbalance in the operation of MVRP. However, if you choose to change the default settings, keep in mind that on an EX Series switch that uses Junos OS with support for ELS, if the timer value set on an interface level is different from the value set on a switch level, then the value on the interface level takes precedence.

Step-by-Step Procedure

To configure access Switch A for MVRP:

1. Configure the finance VLAN:

   ```
   [edit]
   user@Access-Switch-A# set vlans finance vlan-id 100
   ```

2. Configure the lab VLAN:

   ```
   [edit]
   user@Access-Switch-A# set vlans lab vlan–id 200
   ```

3. Configure the sales VLAN:

   ```
   [edit]
   user@Access-Switch-A# set vlans sales vlan–id 300
   ```

4. Configure an Ethernet interface as a member of the finance VLAN:

   ```
   [edit]
   user@Access-Switch-A# set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members
   finance
   ```

5. Configure an Ethernet interface as a member of the lab VLAN:

```
[edit]
user@Access-Switch-A# set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan members lab
```

6. Configure an Ethernet interface as a member of the sales VLAN:

```
[edit]
user@Access-Switch-A# set interfaces ge-0/0/3 unit 0 family ethernet-switching vlan members sales
```

7. Configure a trunk interface:

```
[edit]
user@Access-Switch-A# set interfaces xe-0/1/1 unit 0 family ethernet-switching interface-mode trunk
```

8. Enable MVRP on the trunk interface:

```
[edit]
user@Access-Switch-A# set protocols mvrp interface xe-0/1/1
```

Results

Check the results of the configuration on Switch A:

```
[edit]
user@Access-Switch-A# show
interfaces {
    ge-0/0/1 {
        unit 0 {
            family ethernet-switching {
                vlan {
                    members finance;
                }
            }
        }
    }
    ge-0/0/2 {
        unit 0 {
            family ethernet-switching {
                vlan {
                    members lab;
                }
            }
```

```
        }
    }
    ge-0/0/3 {
        unit 0 {
            family ethernet-switching {
                vlan {
                    members sales;
                }
            }
        }
    }
    xe-0/1/1 {
        unit 0 {
            family ethernet-switching {
                interface-mode trunk;
            }
        }
    }
}
protocols {
    mvrp {
        interface xe-0/1/1;
    }
}
vlans {
    finance {
        vlan-id 100;
    }
    lab {
        vlan-id 200;
    }
    sales {
        vlan-id 300;
    }
}
```

Configuring VLANs and MVRP on Access Switch B

To configure three VLANs on the switch, bind access interfaces for PC4 and PC5 to the VLANs, and enable MVRP on the trunk interface of access Switch B, perform these tasks:

CLI Quick Configuration

To quickly configure Access Switch B for MVRP, copy the following commands and paste them into the switch terminal window of Switch B:

```
[edit]
set vlans finance vlan-id 100

set vlans lab vlan-id 200

set vlans sales vlan-id 300

set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members finance

set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members lab

set interfaces xe-0/1/0 unit 0 family ethernet-switching interface-mode trunk

set protocols mvrp interface xe-0/1/0
```

Step-by-Step Procedure

To configure access Switch B for MVRP:

1. Configure the finance VLAN:

   ```
   [edit]
   user@Access-Switch-B# set vlans finance vlan-id 100
   ```

2. Configure the lab VLAN:

   ```
   [edit]
   user@Access-Switch-B# set vlans lab vlan–id 200
   ```

3. Configure the sales VLAN:

   ```
   [edit]
   user@Access-Switch-B# set vlans sales vlan–id 300
   ```

4. Configure an Ethernet interface as a member of the finance VLAN:

   ```
   [edit]
   user@Access-Switch-B# set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members
   finance
   ```

5. Configure an Ethernet interface as a member of the lab VLAN:

   ```
   [edit]
   ```

```
user@Access-Switch-B# set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members lab
```

6. Configure a trunk interface:

```
user@Access-Switch-B# set interfaces xe-0/1/0 unit 0 family ethernet-switching interface-mode trunk
```

7. Enable MVRP on the trunk interface:

```
[edit]
user@Access-Switch-B# set protocols mvrp xe-0/1/0
```

> **NOTE:** This example uses default MVRP timers. The default values associated with each MVRP timer are: 200 ms for the join timer, 1000 ms for the leave timer, and 10000 ms (10 seconds) for the leaveall timer. We recommend retaining the use of default timer values as modifying timers to inappropriate values might cause an imbalance in the operation of MVRP. However, if you choose to change the default values, keep in mind that on an EX Series switch that uses Junos OS with support for ELS, if the timer value set on an interface level is different from the value set on a switch level, then the value on the interface level takes precedence.

Results

Check the results of the configuration for Switch B:

```
[edit]
user@Access-Switch-B# show
interfaces {
    ge-0/0/0 {
        unit 0 {
            family ethernet-switching {
                vlan {
                    members finance;
                }
            }
        }
    }
    ge-0/0/1 {
        unit 0 {
            family ethernet-switching {
                vlan {
                    members lab;
```

```
            }
          }
        }
    }
    xe-0/1/0 {
        unit 0 {
            family ethernet-switching {
                interface-mode trunk;
            }
        }
    }
}
```

```
protocols {
    mvrp {
        interface xe-0/1/0;
    }
}
vlans {
    finance {
        vlan-id 100;
    }
    lab {
        vlan-id 200;
    }
    sales {
        vlan-id 300;
    }
}
```

Configuring VLANS and MVRP on Distribution Switch C

CLI Quick Configuration

To quickly configure distribution Switch C for MVRP, copy the following commands and paste them into the switch terminal window of distribution Switch C:

```
[edit]
```

set interfaces xe-0/1/1 unit 0 family ethernet-switching interface-mode trunk

set interfaces xe-0/1/0 unit 0 family ethernet-switching interface-mode trunk

set protocols mvrp interface xe-0/1/1

set protocols mvrp interface xe-0/1/0

Step-by-Step Procedure

To configure distribution Switch C for MVRP:

1. Configure the trunk interface to access Switch A:

```
[edit]
user@Distribution-Switch-C# set interfaces xe-0/1/1 unit 0 family ethernet-switching
interface-mode trunk
```

2. Configure the trunk interface to access Switch B:

```
[edit]
user@Distribution-Switch-C# set interfaces xe-0/1/0 unit 0 family ethernet-switching
interface-mode trunk
```

3. Enable MVRP on the trunk interface for **xe-0/1/1** :

```
[edit]
user@Distribution-Switch-C# set protocols mvrp interface xe-0/1/1
```

4. Enable MVRP on the trunk interface for **xe-0/1/0** :

```
[edit]
user@Distribution-Switch-C# set protocols mvrp interface xe-0/1/0
```

Results

Check the results of the configuration for Switch C:

```
[edit]
user@Distribution Switch-C# show
interfaces {
    xe-0/1/0 {
        unit 0 {
            family ethernet-switching {
                interface-mode trunk;
            }
        }
    }
    xe-0/1/1 {
        unit 0 {
            family ethernet-switching {
                interface-mode trunk;
            }
        }
```

```
    }
}
protocols {
    mvrp {
        interface xe-0/1/0;
        interface xe-0/1/1;
    }
```

Verification

IN THIS SECTION

To confirm that the configuration is updating VLAN membership, perform these tasks:

Verifying That MVRP Is Enabled on Access Switch A

Purpose

Verify that MVRP is enabled on the switch.

Action

Show the MVRP configuration:

```
user@Access-Switch-A> show mvrp
```

```
MVRP configuration for routing instance 'default-switch'
MVRP dynamic VLAN creation : Enabled
MVRP BPDU MAC address      : Customer bridge group (01-80-C2-00-00-21)
MVRP timers (ms)
  Interface       Join   Leave  LeaveAll
  xe-0/1/1         200    1000     10000
```

Meaning

The results show that MVRP is enabled on the trunk interface of Switch A and that the default timers are used.

Verifying That MVRP Is Updating VLAN Membership on Access Switch A

Purpose

Verify that MVRP is updating VLAN membership by displaying the Ethernet switching interfaces and associated VLANs that are active on Switch A.

Action

List Ethernet switching interfaces on the switch:

```
user@Access-Switch-A> show ethernet-switching interface
```

```
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
ge-0/0/1.0                    65535                                 tagged
             finance    100
                               65535    Forwarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
ge-0/0/2.0                    65535                                 tagged
             lab        200
                               65535    Forwarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
ge-0/0/3.0                    65535                                   tagged
             sales      300
                               65535    Forwarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/1/1.0                    65535                                   tagged
```

```
              finance    100
                                  65535    Forwarding
              lab        200
                                  65535    Forwarding
```

Meaning

MVRP has automatically added **finance** and **lab** as VLAN members on the trunk interface because they are being advertised by access Switch B.

Verifying That MVRP Is Enabled on Access Switch B

Purpose

Verify that MVRP is enabled on the switch.

Action

Show the MVRP configuration:

```
user@Access-Switch-B> show mvrp
```

```
MVRP configuration for routing instance 'default-switch'
MVRP dynamic VLAN creation : Enabled
MVRP BPDU MAC address      : Customer bridge group (01-80-C2-00-00-21)
MVRP timers (ms)
  Interface       Join   Leave  LeaveAll
  xe-0/1/0         200    1000     10000
```

Meaning

The results show that MVRP is enabled on the trunk interface of Switch B and that the default timers are used.

Verifying That MVRP Is Updating VLAN Membership on Access Switch B

Purpose

Verify that MVRP is updating VLAN membership by displaying the Ethernet switching interfaces and associated VLANs that are active on Switch B.

Action

List Ethernet switching interfaces on the switch:

```
user@Access-Switch-B> show ethernet-switching interface
```

```
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
ge-0/0/0.0                    65535                                 tagged
             finance    100
                               65535    Forwarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
ge-0/0/1.0                    65535                                 tagged
             lab        200
                               65535    Forwarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/1/0.0                    65535                                   tagged
             finance    100
                               65535    Forwarding
             lab        200
                               65535    Forwarding
             sales      300
                               65535    Forwarding
```

Meaning

MVRP has automatically added **finance**, **lab**, and **sales** as VLAN members on the trunk interface because they are being advertised by access Switch A.

Verifying That MVRP Is Enabled on Distribution Switch C

Purpose

Verify that MVRP is enabled on the switch.

Action

Show the MVRP configuration:

```
user@Distribution-Switch-C> show mvrp
```

```
MVRP configuration for routing instance 'default-switch'
MVRP dynamic VLAN creation : Enabled
MVRP BPDU MAC address      : Customer bridge group (01-80-C2-00-00-21)
MVRP timers (ms)
  Interface       Join   Leave  LeaveAll
  xe-0/1/1         200    1000     10000
  xe-0/1/0         200    1000     10000
```

Meaning

The results show that MVRP is enabled on the trunk interfaces of Switch C and that the default timers are used.

Verifying That MVRP Is Updating VLAN Membership on Distribution Switch C

Purpose

Verify that MVRP is updating VLAN membership on distribution Switch C by displaying the Ethernet switching interfaces and associated VLANs on distribution Switch C.

Action

List the Ethernet switching interfaces on the switch:

```
user@Distribution-Switch-C> show ethernet-switching interface
```

```
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/1/1.0                    65535                                 tagged
             mvrp_100
                               65535    Forwarding
             mvrp_200
                               65535    Forwarding
             mvrp_300
                               65535    Forwarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/1/0.0                    65535                                   tagged
             mvrp_100
                               65535    Forwarding
```

```
            mvrp_200
                                65535    Forwarding
```

List the VLANs that were created dynamically using MVRP on the switch:

```
user@Distribution-Switch-C> show mvrp dynamic-vlan-memberships
```

```
MVRP dynamic vlans for routing instance 'default-switch'
 (s) static vlan, (f) fixed registration

VLAN ID          Interfaces
100              xe-0/1/1.0
                 xe-0/1/0.0
200              xe-0/1/1.0
                 xe-0/1/0.0
300              xe-0/1/1.0
```

Note that this scenario does not have any fixed registration, which is typical when MVRP is enabled.

Meaning

Distribution Switch C has two trunk interfaces. Interface **xe-0/1/1.0** connects Distribution Switch C to Access Switch A and is, therefore, updated to show that it is a member of all the VLANs that are active on Switch A. Any traffic for those VLANs will be passed on from Switch C to Switch A, through interface **xe-0/1/1.0**. Interface **xe-0/1/0.0** connects Switch C to Switch B and is updated to show that it is a member of the two VLANs that are active on Switch B. Thus, Switch C sends traffic for **finance** and **lab** to both Switch A and Switch B. But Switch C sends traffic for **sales** only to Switch A.

Switch C also has three dynamic VLANs created using MVRP: **mvrp_100**, **mvrp_200**, and **mvrp_300**. The dynamically created VLANs **mvrp_100** and **mvrp_200** are active on interfaces **xe-0/1/1.0** and **xe-0/1/0.0**, and dynamically created VLAN **mvrp_300** is active on interface **xe-0/1/1.0**.

Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches

IN THIS SECTION

NOTE: This example uses Junos OS for EX Series switches that does not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see "Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches with ELS Support" on page 815. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50.

As a network expands and the number of clients and VLANs increases, VLAN administration becomes complex and the task of efficiently configuring VLANs on multiple EX Series switches becomes increasingly difficult. To automate VLAN administration, you can enable Multiple VLAN Registration Protocol (MVRP) on the network.

MVRP also dynamically creates VLANs, further simplifying the network overhead required to statically configure VLANs.

NOTE: Only trunk interfaces can be enabled for MVRP.

This example describes how to use MVRP to automate administration of VLAN membership changes within your network and how to use MVRP to dynamically create VLANs:

Requirements

This example uses the following hardware and software components:

- Two EX Series access switches
- One EX Series distribution switch
- Junos OS Release 10.0 or later for EX Series switches

Overview and Topology

MVRP is used to manage dynamic VLAN registration in a LAN. It can also be used to dynamically create VLANs.

This example uses MVRP to dynamically create VLANs on the switching network. You can disable dynamic VLAN creation and create VLANs statically, if desired. Enabling MVRP on the trunk interface of each switch in your switching network ensures that the active VLAN information for the switches in the network is propagated to each switch through the trunk interfaces, assuming dynamic VLAN creation is enabled for MVRP.

MVRP ensures that the VLAN membership information on the trunk interface is updated as the switch's access interfaces become active or inactive in the configured VLANs in a static or dynamic VLAN creation setup.

You do not need to explicitly bind a VLAN to the trunk interface. When MVRP is enabled, the trunk interface advertises all the VLANs that are active (bound to access interfaces) on that switch. An MVRP-enabled trunk interface does not advertise VLANs that have been configured on the switch but that are not currently bound to an access interface. Thus, MVRP provides the benefit of reducing network overhead—by limiting the scope of broadcast, unknown unicast, and multicast (BUM) traffic to interested devices only.

When VLAN access interfaces become active or inactive, MVRP ensures that the updated information is advertised on the trunk interface. Thus, in this example, distribution Switch C does not forward traffic to inactive VLANs.

> **NOTE:** This example shows a network with three VLANs: **finance**, **sales**, and **lab**. All three VLANs are running the same version of Junos OS. If switches in this network were running a mix of Junos OS releases that included Release 11.3, additional configuration would be necessary—see "Configuring Multiple VLAN Registration Protocol (MVRP) on Switches" on page 797 for details.

Access Switch A has been configured to support all three VLANS and all three VLANS are active, bound to interfaces that are connected to personal computers:

- **ge-0/0/1**—Connects PC1 as a member of **finance**, VLAN ID 100
- **ge-0/0/2**—Connects PC2 as a member of **lab**, VLAN ID 200
- **ge-0/0/3**—Connects PC3 as a member of **sales**, VLAN ID 300

Access Switch B has also been configured to support three VLANS. However, currently only two VLANs are active, bound to interfaces that are connected to personal computers:

- **ge-0/0/0**—Connects PC4 as a member of **finance**, VLAN ID 100
- **ge-0/0/1**—Connects PC5 as a member of **lab**, VLAN ID 200

Distribution Switch C learns the VLANs dynamically using MVRP through the connection to the access switches. Distribution Switch C has two trunk interfaces:

- **xe-0/1/1**—Connects the switch to access Switch A.
- **xe-0/1/0**—Connects the switch to access Switch B.

Figure 43 on page 834 shows MVRP configured on two access switches and one distribution switch.

Figure 43: MVRP Configured on Two Access Switches and One Distribution Switch for Automatic VLAN Administration

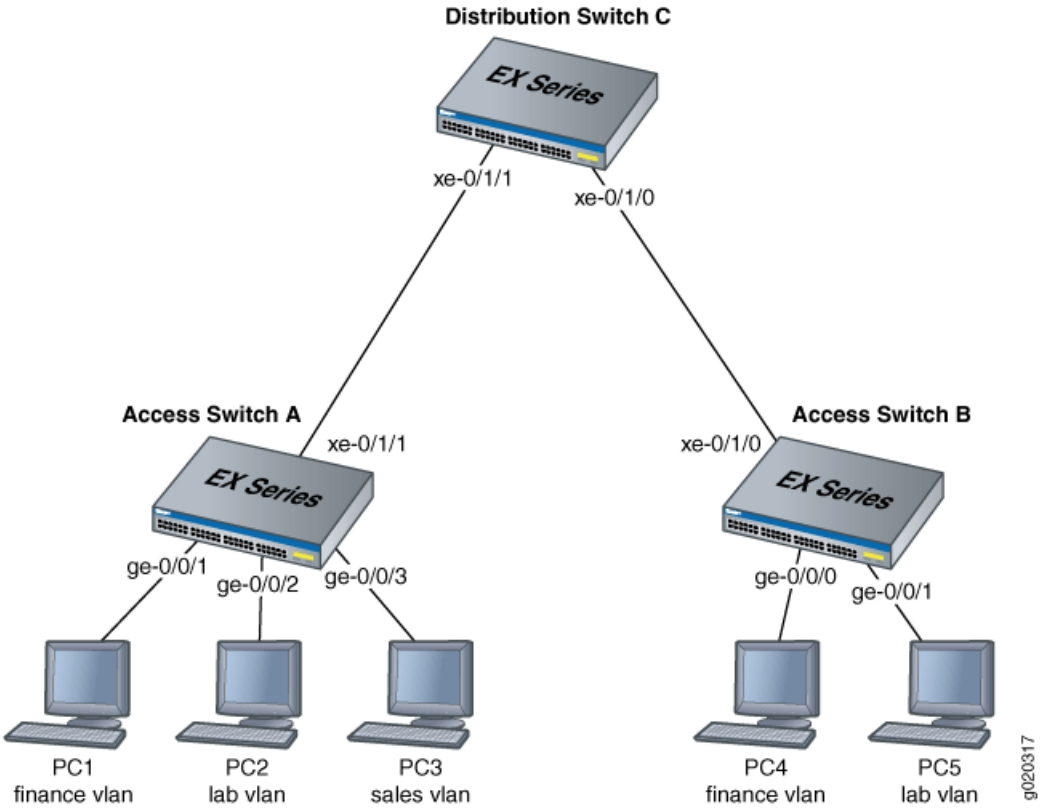


Table 119 on page 835 explains the components of the example topology.

Table 119: Components of the Network Topology

Settings	Settings
Switch hardware	• Access Switch A • Access Switch B • Distribution Switch C
VLAN names and tag IDs	**finance**, tag **100** **lab**, tag **200** **sales**, tag **300**
Interfaces	Access Switch A interfaces: • **ge-0/0/1**—Connects PC1 to access Switch A. • **ge-0/0/2**—Connects PC2 to access Switch A. • **ge-0/0/3**—Connects PC3 to access Switch A. • **xe-0/1/1**—Connects access Switch A to distribution Switch C (trunk). Access Switch B interfaces: • **ge-0/0/0**—Connects PC4 to access Switch B. • **ge-0/0/1**—Connects PC5 to access Switch B. • **xe-0/1/0**—Connects access Switch B to distribution Switch C. (trunk) Distribution Switch C interfaces: • **xe-0/1/1**—Connects distribution Switch C to access Switch A. (trunk) • **xe-0/1/0**—Connects distribution Switch C to access Switch B. (trunk)

Configuring VLANs and MVRP on Access Switch A

To configure VLANs on the switch, bind access interfaces to the VLANs, and enable MVRP on the trunk interface of access Switch A, perform these tasks:

CLI Quick Configuration

To quickly configure access Switch A for MVRP, copy the following commands and paste them into the switch terminal window of Switch A:

```
[edit]
set vlans finance vlan-id 100
```

set vlans lab vlan-id 200

set vlans sales vlan-id 300

set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members finance

set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan members lab

set interfaces ge-0/0/3 unit 0 family ethernet-switching vlan members sales

set interfaces xe-0/1/1 unit 0 family ethernet-switching port-mode trunk

set protocols mvrp interface xe-0/1/1.0

> **NOTE:** As recommended as a best practice, default MVRP timers are used in this example. The default values associated with each MVRP timer are: 200 ms for the join timer, 1000 ms for the leave timer, and 10000 ms for the leaveall timer. Modifying timers to inappropriate values might cause an imbalance in the operation of MVRP.

Step-by-Step Procedure

To configure access Switch A for MVRP:

1. Configure the finance VLAN:

   ```
   [edit]
   user@Access-Switch-A# set vlans finance vlan-id 100
   ```

2. Configure the lab VLAN:

   ```
   [edit]
   user@Access-Switch-A# set vlans lab vlan–id 200
   ```

3. Configure the sales VLAN:

   ```
   [edit]
   user@Access-Switch-A# set vlans sales vlan–id 300
   ```

4. Configure an Ethernet interface as a member of the finance VLAN:

   ```
   [edit]
   user@Access-Switch-A# set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members
   finance
   ```

5. Configure an Ethernet interface as a member of the lab VLAN:

```
[edit]
user@Access-Switch-A# set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan members lab
```

6. Configure an Ethernet interface as a member of the sales VLAN:

```
[edit]
user@Access-Switch-A# set interfaces ge-0/0/3 unit 0 family ethernet-switching vlan members sales
```

7. Configure a trunk interface:

```
[edit]
user@Access-Switch-A# set interfaces xe-0/1/1 unit 0 family ethernet-switching port-mode trunk
```

8. Enable MVRP on the trunk interface:

```
[edit]
user@Access-Switch-A# set protocols mvrp interface xe-0/1/1.0
```

Results

Check the results of the configuration on Switch A:

```
[edit]
user@Access-Switch-A# show
interfaces {
    ge-0/0/1 {
        unit 0 {
            family ethernet-switching {
                vlan {
                    members finance;
                }
            }
        }
    }
    ge-0/0/2 {
        unit 0 {
            family ethernet-switching {
                vlan {
                    members lab;
```

```
            }
          }
        }
      }
      ge-0/0/3 {
        unit 0 {
          family ethernet-switching {
            members sales;
            }
          }
        }
      }
      xe-0/1/1 {
        unit 0 {
          family ethernet-switching {
            port-mode trunk;
          }
        }
      }
  }
  protocols {
      mvrp {
        interface xe-0/1/1.0;
      }
  }
  vlans {
      finance {
        vlan-id 100;
      }
      lab {
        vlan-id 200;
      }
      sales {
        vlan-id 300;
      }
  }
```

Configuring VLANs and MVRP on Access Switch B

To configure three VLANs on the switch, bind access interfaces for PC4 and PC5 to the VLANs, and enable MVRP on the trunk interface of access Switch B, perform these tasks:

CLI Quick Configuration

To quickly configure Access Switch B for MVRP, copy the following commands and paste them into the switch terminal window of Switch B:

```
[edit]
set vlans finance vlan-id 100

set vlans lab vlan-id 200

set vlans sales vlan-id 300

set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members finance

set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members lab

set interfaces xe-0/1/0 unit 0 family ethernet-switching port-mode trunk

set protocols mvrp interface xe-0/1/0.0
```

Step-by-Step Procedure

To configure access Switch B for MVRP:

1. Configure the finance VLAN:

   ```
   [edit]
   user@Access-Switch-B# set vlans finance vlan-id 100
   ```

2. Configure the lab VLAN:

   ```
   [edit]
   user@Access-Switch-B# set vlans lab vlan–id 200
   ```

3. Configure the sales VLAN:

   ```
   [edit]
   user@Access-Switch-B# set vlans sales vlan–id 300
   ```

4. Configure an Ethernet interface as a member of the finance VLAN:

   ```
   [edit]
   user@Access-Switch-B# set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members
   finance
   ```

5. Configure an Ethernet interface as a member of the lab VLAN:

   ```
   [edit]
   ```

```
user@Access-Switch-B# set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members
lab
```

6. Configure a trunk interface:

```
user@Access-Switch-B# set interfaces xe-0/1/0 unit 0 family ethernet-switching port-mode
trunk
```

7. Enable MVRP on the trunk interface:

```
[edit]
user@Access-Switch-B# set protocols mvrp xe-0/1/0.0
```

> **NOTE:** As we recommend as a best practice, default MVRP timers are used in this example. The default values associated with each MVRP timer are: 200 ms for the join timer, 1000 ms for the leave timer, and 10000 ms for the leaveall timer. Modifying timers to inappropriate values might cause an imbalance in the operation of MVRP.

Results

Check the results of the configuration for Switch B:

```
[edit]
user@Access-Switch-B# show
interfaces {
    ge-0/0/0 {
        unit 0 {
            family ethernet-switching {
                vlan {
                    members finance;
                }
            }
        }
    }
    ge-0/0/1 {
        unit 0 {
            family ethernet-switching {
                vlan {
                    members lab;
                }
            }
        }
```

```
    }
    xe-0/1/0 {
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
            }
        }
    }
}
```

```
protocols {
    mvrp {
        interface xe-0/1/0.0;
    }
}
vlans {
    finance {
        vlan-id 100;
    }
    lab {
        vlan-id 200;
    }
    sales {
    vlan-id 300;
    }
}
```

Configuring VLANS and MVRP on Distribution Switch C

CLI Quick Configuration

To quickly configure distribution Switch C for MVRP, copy the following commands and paste them into the switch terminal window of distribution Switch C:

```
[edit]
```

set interfaces xe-0/1/1 unit 0 family ethernet-switching port-mode trunk

set interfaces xe-0/1/0 unit 0 family ethernet-switching port-mode trunk

set protocols mvrp interface xe-0/1/1.0

set protocols mvrp interface xe-0/1/0.0

Step-by-Step Procedure

To configure distribution Switch C for MVRP:

1. Configure the trunk interface to access Switch A:

   ```
   [edit]
   user@Distribution-Switch-C# set interfaces xe-0/1/1 unit 0 family ethernet-switching
   port-mode trunk
   ```

2. Configure the trunk interface to access Switch B:

   ```
   [edit]
   user@Distribution-Switch-C# set interfaces xe-0/1/0 unit 0 family ethernet-switching
   port-mode trunk
   ```

3. Enable MVRP on the trunk interface for **xe-0/1/1** :

   ```
   [edit]
   user@Distribution-Switch-C# set protocols mvrp interface xe-0/1/1.0
   ```

4. Enable MVRP on the trunk interface for **xe-0/1/0** :

   ```
   [edit]
   user@Distribution-Switch-C# set protocols mvrp interface xe-0/1/0.0
   ```

Results

Check the results of the configuration for Switch C:

```
[edit]
user@Distribution Switch-C# show
interfaces {
    xe-0/1/0 {
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
            }
        }
    }
    xe-0/1/1 {
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
            }
        }
    }
```

```
}
protocols {
    mvrp {
        interface xe-0/1/0.0;
        interface xe-0/1/1.0;
    }
```

Verification

IN THIS SECTION

To confirm that the configuration is updating VLAN membership, perform these tasks:

Verifying That MVRP Is Enabled on Access Switch A

Purpose

Verify that MVRP is enabled on the switch.

Action

Show the MVRP configuration:

```
user@Access-Switch-A> show mvrp
```

```
MVRP configuration
MVRP status                 : Enabled
MVRP dynamic VLAN creation  : Enabled

MVRP timers (ms):
Interface            Join   Leave       LeaveAll
--------------       -----  --------    -----------
all                   200   1000        10000
```

```
xe-0/1/1.0            200   1000        10000

Interface       Status        Registration Mode
--------------  --------      -----------------
all             Disabled      Normal
xe-0/1/1.0      Enabled       Normal
```

Meaning

The results show that MVRP is enabled on the trunk interface of Switch A and that the default timers are used.

Verifying That MVRP Is Updating VLAN Membership on Access Switch A

Purpose

Verify that MVRP is updating VLAN membership by displaying the Ethernet switching interfaces and associated VLANs that are active on Switch A.

Action

List Ethernet switching interfaces on the switch:

```
user@Access-Switch-A> show ethernet-switching interfaces
```

```
Interface   State    VLAN members   Tag  Tagging       Blocking
ge-0/0/1.0  up       finance        100  untagged      unblocked
ge-0/0/2.0  up       lab            200  untagged      unblocked
ge-0/0/3.0  up       sales          300  untagged      unblocked
xe-0/1/1.0  up       finance        100  untagged      unblocked
                     lab            200  untagged      unblocked
```

Meaning

MVRP has automatically added **finance** and **lab** as VLAN members on the trunk interface because they are being advertised by access Switch B.

Verifying That MVRP Is Enabled on Access Switch B

Purpose

Verify that MVRP is enabled on the switch.

Action

Show the MVRP configuration:

```
user@Access-Switch-B> show mvrp
```

```
MVRP configuration
MVRP status                 : Enabled
MVRP dynamic VLAN creation  : Enabled

MVRP timers (ms):
Interface         Join   Leave       LeaveAll
--------------    -----  --------    -----------
all                200   1000       10000
xe-0/1/0.0         200   1000       10000

Interface       Status        Registration Mode
--------------  --------      -----------------
all             Disabled      Normal
xe-0/1/0.0      Enabled       Normal
```

Meaning

The results show that MVRP is enabled on the trunk interface of Switch B and that the default timers are used.

Verifying That MVRP Is Updating VLAN Membership on Access Switch B

Purpose

Verify that MVRP is updating VLAN membership by displaying the Ethernet switching interfaces and associated VLANs that are active on Switch B.

Action

List Ethernet switching interfaces on the switch:

```
user@Access-Switch-B> show ethernet-switching interfaces
```

```
Interface   State    VLAN members   Tag  Tagging       Blocking
ge-0/0/0.0  up       finance        100  untagged      unblocked
ge-0/0/1.0  up       lab            200  untagged      unblocked
xe-0/1/1.0  up       finance        100  untagged      unblocked
                     lab            200  untagged      unblocked
                     sales          300  untagged      unblocked
```

Meaning

MVRP has automatically added **finance**, **lab**, and **sales** as VLAN members on the trunk interface because they are being advertised by access Switch A.

Verifying That MVRP Is Enabled on Distribution Switch C

Purpose

Verify that MVRP is enabled on the switch.

Action

Show the MVRP configuration:

```
user@Distribution-Switch-C> show mvrp
```

```
MVRP configuration
MVRP status                 : Enabled
MVRP dynamic VLAN creation  : Enabled

MVRP timers (ms):
Interface            Join   Leave       LeaveAll
--------------       -----  --------    -----------
all                   200   1000        10000
xe-0/0/1.0            200   1000        10000
xe-0/1/1.0            200   1000        10000

Interface       Status        Registration Mode
--------------  --------      -----------------
all             Disabled      Normal
xe-0/0/1.0      Enabled       Normal
xe-0/1/1.0      Enabled       Normal
```

Verifying That MVRP Is Updating VLAN Membership on Distribution Switch C

Purpose

Verify that MVRP is updating VLAN membership on distribution Switch C by displaying the Ethernet switching interfaces and associated VLANs on distribution Switch C.

Action

List the Ethernet switching interfaces on the switch:

```
user@Distribution-Switch-C> show ethernet-switching interfaces
```

```
Interface   State    VLAN members   Tag  Tagging       Blocking
xe-0/1/1.0  up       __mvrp_100__               unblocked
                     __mvrp_200__               unblocked
                     __mvrp_300__               unblocked
```

```
xe-0/1/0.0  up       __mvrp_100__                unblocked
                     __mvrp_200__                unblocked
```

List the VLANs that were created dynamically using MVRP on the switch:

```
user@Distribution-Switch-C> show mvrp dynamic-vlan-memberships
```

```
MVRP dynamic vlans for routing instance 'default-switch'
 (s) static vlan, (f) fixed registration

VLAN ID          Interfaces
100              xe-0/1/1.0
                 xe-0/1/0.0
200              xe-0/1/1.0
                 xe-0/1/0.0
300              xe-0/1/1.0
```

Note that this scenario does not have any fixed registration, which is typical when MVRP is enabled.

Meaning

Distribution Switch C has two trunk interfaces. Interface **xe-0/1/1.0** connects distribution Switch C to Access Switch A and is therefore updated to show that it is a member of all the VLANs that are active on Switch A. Any traffic for those VLANs will be passed on from distribution Switch C to Switch A, through interface **xe-0/1/1.0**. Interface **xe-0/1/0.0** connects distribution Switch C to Switch B and is updated to show that it is a member of the two VLANs that are active on Switch B. Thus, distribution Switch C sends traffic for **finance** and **lab** to both Switch A and Switch B. But distribution Switch C sends traffic for **sales** only to Switch A.

Distribution Switch C also has three dynamic VLANs created using MVRP: **mvrp_100**, **mvrp_200**, and **mvrp_300**. The dynamically created VLANs **mvrp_100** and **mvrp_200** are active on interfaces **xe-0/1/1.0** and **xe-0/1/1.0**, and dynamically created VLAN **mvrp_300** is active on interface **xe-0/1/1.0**.

Verifying That MVRP Is Working Correctly on Switches

Purpose

After configuring your switch to participate in MVRP, verify that the configuration is properly set and that MVRP messages are being sent and received on your switch.

Action

1. Confirm that MVRP is enabled on your switch.

   user@switch> **show mvrp**

```
Global MVRP configuration
  MVRP status               : Enabled
  MVRP dynamic vlan creation: Enabled
  MVRP Timers (ms):
   Interface        Join   Leave   LeaveAll
   --------------   ----   -----   --------
   all              200    600     10000
   xe-0/1/1.0       200    600     10000

  Interface based configuration:
   Interface        Status     Registration   Dynamic VLAN Creation
   --------------   --------   ------------   --------------------
   all          Disabled   Fixed          Enabled
   xe-0/1/1.0       Enabled    Normal         Enabled
```

2. Confirm that MVRP messages are being sent and received on your switch.

   user@switch> **show mvrp statistics interface xe-0/1/1.0**

```
MVRP statistics
  MRPDU received               : 3342
  Invalid PDU received         : 0
  New received                 : 2
  Join Empty received          : 1116
  Join In received             : 2219
  Empty received               : 2
  In received                  : 2
  Leave received               : 1
  LeaveAll received            : 1117
  MRPDU transmitted            : 3280
  MRPDU transmit failures      : 0
  New transmitted              : 0
  Join Empty transmitted       : 1114
  Join In transmitted          : 2163
  Empty transmitted            : 1
  In transmitted               : 1
  Leave transmitted            : 1
  LeaveAll transmitted         : 1111
```

Meaning

The output of **show mvrp** shows that interface **xe-0/1/1.0** is enabled for MVRP participation as shown in the status in the **Interface based configuration** field.

The output for **show mvrp statistics interface xe-0/1/1.0** confirms that MVRP messages are being transmitted and received on the interface.

> **NOTE:** You can identify an MVRP compatibility issue on EX Series switches by looking at the output from this command. If *Join Empty received* and *Join In received* incorrectly display zero, even though the value for *MRPDU received* has been increased, you are probably running different versions of Junos OS, including Release 11.3, on the switches in this network. Another indication that MVRP is having a version problem is that unexpected VLAN activity, such as multiple VLAN creation, takes place on the switch running the earlier release version. To remedy these problems, see "Configuring Multiple VLAN Registration Protocol (MVRP) on Switches" on page 797.

Verifying That MVRP Is Working Correctly on EX Series Switches with ELS Support

Purpose

> **NOTE:** This task uses Junos OS for EX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does not support ELS, see "Verifying That MVRP Is Working Correctly on Switches" on page 847. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50.

After configuring your EX Series switch to participate in MVRP, verify that the configuration is properly set and that MVRP messages are being sent and received on your switch.

Action

1. Confirm that MVRP is enabled on your switch.

```
user@switch> show mvrp
```

```
MVRP configuration for routing instance 'default-switch'
MVRP dynamic VLAN creation : Enabled
MVRP BPDU MAC address      : Customer bridge group (01-80-C2-00-00-21)
MVRP timers (ms)
```

```
  Interface       Join   Leave  LeaveAll
  xe-0/1/1         200    1000     10000
```

2. Confirm that MVRP messages are being sent and received on your switch.

   `user@switch>` **show mvrp statistics**

```
MVRP statistics for routing instance 'default-switch'

Interface name                  : xe-0/1/1
VLAN IDs registered             : 117
Sent MVRP PDUs                  : 118824
Received MVRP PDUs without error: 118848
Received MVRP PDUs with error   : 0
Transmitted Join Empty          : 5229
Transmitted Leave All           : 2
Recieved Join In                : 11884924
Transmitted Join In             : 1835
Transmitted Empty               : 93606408
Transmitted Leave               : 888
Transmitted In                  : 13780024
Transmitted New                 : 2692
Received Leave All              : 118761
Received Leave                  : 97
Received In                     : 3869
Received Empty                  : 828
Received Join Empty             : 2020152
Received New                    : 224
...
```

Meaning

The output of **show mvrp** shows that interface xe-0/1/1 is enabled for MVRP participation.

The output for **show mvrp statistics** confirms that MVRP messages are being transmitted and received on interface xe-0/1/1.

NOTE: You can identify an MVRP compatibility issue by observing the output from this command. If **Received Join Empty** and **Received Join In** incorrectly display zero, even though the value for **Received MVRP PDUs without error** has been increased, you are probably running different versions of Junos OS on the switches in this network. Another indication that MVRP is having a version problem is that unexpected VLAN activity, such as multiple VLAN creation, takes place on the switch running the earlier release version. To remedy these problems, see "Configuring Multiple VLAN Registration Protocol (MVRP) on Switches" on page 797.

Verifying That MVRP Is Working Correctly

Purpose

After configuring your MX Series router or EX Series switch to participate in Multiple VLAN Registration Protocol (MVRP), verify that the configuration is properly set and that MVRP messages are being sent and received on your switch.

Action

1. Confirm that the router is declaring VLANs.

   Show that MVRP is enabled:

   ```
   user@host> show mvrp
   ```

   ```
   MVRP configuration for routing instance 'default-switch'
   MVRP dynamic VLAN creation : Enabled
   MVRP BPDU MAC address      : Customer bridge group (01-80-C2-00-00-21)
   MVRP timers (ms)
     Interface       Join   Leave  LeaveAll
     ge-11/3/0        200     800     10000
   ```

   Show the MVRP applicant state:

   ```
   user@host> show mvrp applicant-state
   ```

   ```
   MVRP applicant state for routing instance 'default-switch'
    (VO) Very anxious observer, (VP) Very anxious passive, (VA) Very anxious new,
    (AN) Anxious new, (AA) Anxious active, (QA) Quiet active, (LA) Leaving active,

    (AO) Anxious observer, (QO) Quiet observer, (LO) Leaving observer,
    (AP) Anxious passive, (QP) Quiet passive
   ```

```
VLAN Id     Interface          State
    100     ge-11/3/0          Declaring (QA)
    200     ge-11/3/0          Declaring (QA)
    300     ge-11/3/0          Declaring (QA)
```

2. Confirm that VLANs are registered on interfaces.

   List VLANs in the registered state:

   `user@host>` **show mvrp registration-state**

```
MVRP registration state for routing instance 'default-switch'

VLAN Id   Interface   Registrar   Forced      Managed    STP
                      State       State       State      State
    100   ge-11/3/0   Registered  Registered  Normal     Forwarding
    200   ge-11/3/0   Registered  Registered  Normal     Forwarding
    300   ge-11/3/0   Empty       Empty       Normal     Forwarding
```

3. Display a list of VLANs created dynamically.

   List dynamic VLAN membership:

   `user@host>` **show mvrp dynamic-vlan-memberships**

```
MVRP dynamic vlans for routing instance 'default-switch'
 (s) static vlan, (f) fixed registration

VLAN Id            Interfaces
    100            ge-3/3/0
                   ge-3/0/5
    200            ge-3/3/0
                   ge-3/0/5
```

Meaning

The output of **show mvrp applicant-state** shows that trunk interface **ge-11/3/0** is declaring (sending out) interest in the VLAN IDs **100**, **200**, and **300**, and MVRP is operating properly.

The output of **show mvrp registrant-state** shows the registrar state for VLANs **100** and **200** as **Registered**, indicating that these VLANs are receiving traffic from a customer site. VLAN **300** is in an **Empty** state and is not receiving traffic from a customer site.

The output of the **show mvrp dynamic-vlan-membership** shows that VLANs **100** and **200** are created dynamically (here, on an MX Series router operating as an aggregation switch between MX Series routers operating as edge switches). VLANs created statically are marked with an **(s)** (which is not indicated in this output).

28

CHAPTER

Configuring Ethernet Ring Protection Switching

Example: Configuring Ethernet Ring Protection Switching on EX Series Switches

IN THIS SECTION

You can configure Ethernet ring protection switching (ERPS) on connected EX Series or QFX Series switches to prevent fatal loops from disrupting a network. (Platform support depends on the Junos OS release in your installation.) ERPS is similar to spanning-tree protocols, but ERPS is more efficient because it is customized for ring topologies. You must configure at least three switches to form a ring.

This example shows how to configure Ethernet ring protection switching on four switches that are connected to one another on a dedicated link in a ring topology.

> **NOTE:** This task uses Junos OS for EX Series switches without support for the Enhanced Layer 2 Software (ELS) configuration style. However, an ERPS ring can include different types of switches, with or without ELS support. If you are configuring an ERPS ring that also includes QFX Series or EX Series switches running software that supports ELS, see "Example: Configuring Ethernet Ring Protection Switching on QFX Series and EX Series Switches Supporting ELS" on page 874 for equivalent example configuration steps on those switches. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50.

Requirements

This example uses the following hardware and software components:

- Four connected EX Series switches that will function as nodes in the ring topology.

NOTE: Because Junos uses an ERPV2 state machine for ERPV1 support on both EX2300 and EX3400 switches, operation of ERPS on those two switches deviates from the ERPV1 ITU standard in the following ways:

- Wait to Restore (WTR) configuration values on EX2300 and EX3400 switches must be 5-12 minutes.
- The Wait To Block Timer (WTB) is always disabled on EX2300 and EX3400 switches because it is not supported in ERPSv1. Any configuration you make to the WTB setting has no effect. The output from the CLI command 'show protection-group ethernet-ring node-state detail' lists a WTB setting but that setting has no effect.
- During initial state machine initialization on EX2300 and EX3400 switches, both ERPV1 ring ports move to a discarding state on the non-RPL node.
- During ERPV1 initial state machine initialization on EX2300 and EX3400 switches, the Automatic Protection Switching (APS) state moves to an idle state on the non-RPL switch

- Junos OS Release 12.1 or later without support for the Enhanced Layer 2 Software (ELS) configuration style.

Before you begin, be sure you have:

- Configured two trunk interfaces on each of the four switches. See Table 120 on page 858 for a list of the interface names used in this example.
- Configured the same VLAN (**erp-control-vlan-1**) with ID 100 on all four switches and associated two network interfaces from each of the four switches with the VLAN. See "Configuring VLANs for EX Series Switches" on page 183. See Table 120 on page 858 for a list of the interface names used in this example.
- Configured two VLANs (**erp-data-1** and **erp-data-2**) with IDs 101 and 102, respectively, on all four switches and associated both the east and west interfaces on each switch with **erp-data-1** and **erp-data-2**. See Table 120 on page 858 for a list of the interface names used in this example.

NOTE: When EX2300 and EX3400 ERPS switches have a VLAN-ID configured with a name under an interface hierarchy, a commit error occurs. Avoid this by configuring VLAN-IDs using numbers when they are under an interface hierarchy with ERPS configured in the switch.

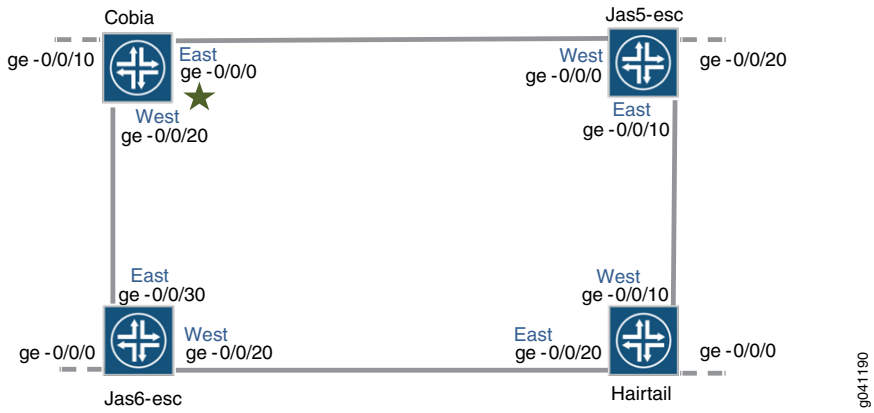
Overview and Topology

ERPS uses a dedicated physical link, including a control VLAN for trunk ports, between all of the switches to protect the active links. ERPS VLANs are all located on this link and are also blocked by default. When traffic between the switches is flowing with no problems, the active links take care of all traffic. Only if an error occurs on one of the data links would the ERPS control channel take over and start forwarding traffic.

> **NOTE:** Trunk ports on switches use a VLAN to create individual control channels for ERPS. When multiple ERPS instances are configured for a ring, there are multiple sets of ring protection links (RPLs) and RPL owners on the ERPS link, and a different channel is blocked for each instance. Nontrunk ports use the physical link as the control channel and protocol data units (PDUs) are untagged, with no VLAN information in the packet.

This example creates one protection ring (called a node ring) named erp1 on four switches connected in a ring by trunk ports as shown in Figure 44 on page 857. Because the links are trunk ports, the VLAN named erp-control-vlan-1 is used for erp1 traffic. The east interface of each switch is connected with the west interface of an adjacent switch. Cobia is the RPL owner, with interface ge-0/0/0 configured as an RPL end interface. The interface ge-0/0/0 of Jas5-esc is configured as the RPL neighbor interface. In the idle state, the RPL end blocks the control VLAN and data channel VLAN for this particular ERP instance—the blocked port on Cobia is marked with a star in Figure 44 on page 857.

Figure 44: Ethernet Ring Protection Switching Example



In this example, we configure the four switches with the interfaces indicated in both Figure 44 on page 857 and Table 120 on page 858.

Table 120: Components to Configure for This Example

Interfaces	Cobia	Jas5-esc	Jas6-esc	Hairtail
East	ge-0/0/0	ge-0/0/10	ge-0/0/30	ge-0/0/20
West	ge-0/0/20	ge-0/0/0	ge-0/0/20	ge-0/0/10
Third	ge-0/0/10	ge-0/0/20	ge-0/0/0	ge-0/0/0

Configuration

IN THIS SECTION

Configuring ERPS on Cobia, the RPL Owner Node

CLI Quick Configuration

To quickly configure Cobia, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

> **NOTE:** Spanning-tree protocols and ERPS cannot both be configured on a ring port. Because RSTP is the spanning-tree protocol enabled in the default switch configuration, this example shows disabling RSTP on each ring port before configuring ERPS. If another spanning-tree protocol is enabled, you must disable that first instead.

```
set protocols rstp interface ge-0/0/0 disable
set protocols rstp interface ge-0/0/20 disable
set protocols protection-group ethernet-ring erp1
```

```
set protocols protection-group ethernet-ring erp1 ring-protection-link-owner
set protocols protection-group ethernet-ring erp1 data-channel erp-data-1
set protocols protection-group ethernet-ring erp1 data-channel erp-data-2
set protocols protection-group ethernet-ring erp1 control-vlan erp-control-vlan-1
set protocols protection-group ethernet-ring erp1 east-interface control-channel ge-0/0/0.0
set protocols protection-group ethernet-ring erp1 east-interface ring-protection-link-end
set protocols protection-group ethernet-ring erp1 west-interface control-channel ge-0/0/20.0
```

Step-by-Step Procedure

To configure ERPS on Cobia:

1. Disable any spanning- tree protocols configured on the ERPS interfaces. STP, RSTP, VSTP, and MSTP are all available spanning tree protocols. RSTP is enabled in the default configuration, so this example shows disabling RSTP:

   ```
   [edit protocols]
   user@switch# set rstp interface ge-0/0/0 disable
   user@switch# set rstp interface ge-0/0/20 disable
   ```

2. Create a node ring named erp1:

   ```
   [edit protocols]
   user@switch# set protection-group ethernet-ring erp1
   ```

3. Designate Cobia as the RPL owner node:

   ```
   [edit protocols protection-group ethernet-ring erp1]
   user@switch# set ring-protection-link-owner
   ```

4. Configure the VLANs erp-data-1 and erp-data-2 as data channels:

   ```
   [edit protocols protection-group ethernet-ring erp1]
   user@switch# set data-channel erp-data-1
   user@switch# set data-channel erp-data-2
   ```

5. Configure the control VLAN erp-control-vlan-1 for this ERP instance on the trunk interface:

   ```
   [edit protocols protection-group ethernet-ring erp1]
   user@switch# set control-vlan erp-control-vlan-1
   ```

6. Configure the east interface of the node ring erp1 with the control channel ge-0/0/0.0 and indicate that this particular ring protection link ends here:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set east-interface control-channel ge-0/0/0.0
user@switch# set east-interface ring-protection-link-end
```

7. Configure the west interface of the node ring erp1 with the control channel ge-0/0/20.0:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set west-interface control-channel ge-0/0/20.0
```

Results

In configuration mode, check your ERPS configuration by entering the **show protocols** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@switch# show protocols
rstp {
    interface ge-0/0/20.0 {
        disable;
    }
    interface ge-0/0/0.0 {
        disable;
    }
}
protection-group {
    ethernet-ring erp1 {
        ring-protection-link-owner;
        east-interface {
            control-channel {
                ge-0/0/0.0;
            }
            ring-protection-link-end;
        }
        west-interface {
            control-channel {
                ge-0/0/20.0;
            }
        }
        control-vlan erp-control-vlan-1;
```

```
        data-channel {
            vlan [ 101-102 ];
        }
    }
}
```

In configuration mode, check your VLAN configuration by entering the **show vlans** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
  [edit]
  user@switch# show vlans
erp-control-vlan-1 {
    vlan-id 100;
    interface {
        ge-0/0/0.0;
        ge-0/0/20.0;
    }
}
erp-data-1 {
    vlan-id 101;
    interface {
        ge-0/0/10.0;
        ge-0/0/0.0;
        ge-0/0/20.0;
    }
}
erp-data-2 {
    vlan-id 102;
    interface {
        ge-0/0/10.0;
        ge-0/0/0.0;
        ge-0/0/20.0;
    }
}
```

In configuration mode, check your interface configurations by entering the **show interfaces** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
  [edit]
  user@switch# show interfaces
```

```
ge-0/0/0 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
        }
    }
}
ge-0/0/10 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
        }
    }
}
ge-0/0/20 {
    unit 0 {
        family ethernet-switching {
             port-mode trunk;
        }
    }
}
```

If you are finished configuring the device, enter **commit** in configuration mode.

Configuring ERPS on Jas5-esc

CLI Quick Configuration

To quickly configure Jas5-esc, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set protocols rstp interface ge-0/0/10 disable
set protocols rstp interface ge-0/0/0 disable
set protocols protection-group ethernet-ring erp1
set protocols protection-group ethernet-ring erp1 data-channel erp-data-1
set protocols protection-group ethernet-ring erp1 data-channel erp-data-2
set protocols protection-group ethernet-ring erp1 control-vlan erp-control-vlan-1
set protocols protection-group ethernet-ring erp1 east-interface control-channel ge-0/0/10.0
set protocols protection-group ethernet-ring erp1 west-interface control-channel ge-0/0/0.0
```

Step-by-Step Procedure

To configure ERPS on Jas5-esc:

1. Disable any spanning- tree protocols configured on the ERPS interfaces. RSTP is enabled in the default configuration, so this example shows disabling RSTP:

```
[edit protocols]
user@switch# set rstp interface ge-0/0/10 disable
user@switch# set rstp interface ge-0/0/0 disable
```

2. Create a node ring named erp1:

```
[edit protocols]
user@switch# set protection-group ethernet-ring erp1
```

3. Configure a control VLAN named erp-control-vlan-1 for the node ring erp1:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set control-vlan erp-control-vlan-1
```

4. Configure two data channels named erp-data-1 and erp-data-2 to define a set of VLAN IDs that belong to a ring instance.

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set data-channel erp-data-1
user@switch# set data-channel erp-data-2
```

5. Configure the east interface of the node ring erp1 with the control channel ge-0/0/10.0:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set east-interface control-channel ge-0/0/10.0
```

6. Configure the west interface of the node ring erp1 with the control channel ge-0/0/0.0:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set west-interface control-channel ge-0/0/0.0
```

Results

In configuration mode, check your ERPS configuration by entering the **show protocols** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@switch# show protocols
rstp {
    interface ge-0/0/10.0 {
        disable;
    }
    interface ge-0/0/0.0 {
        disable;
    }
}
protection-group {
    ethernet-ring erp1 {
        east-interface {
            control-channel {
                ge-0/0/10.0;
            }
        }
        west-interface {
            control-channel {
                ge-0/0/0.0;
            }
        }
        control-vlan erp-control-vlan-1;
        data-channel {
            vlan [ 101-102 ];
        }
    }
}
```

In configuration mode, check your VLAN configuration by entering the **show vlans** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@switch# show vlans
erp-control-vlan-1 {
    vlan-id 100;
    interface {
        ge-0/0/10.0;
        ge-0/0/0.0;
```

```
    }
}
erp-data-1 {
    vlan-id 101;
    interface {
        ge-0/0/20.0;
        ge-0/0/10.0;
        ge-0/0/0.0;
    }
}
erp-data-2 {
    vlan-id 102;
    interface {
        ge-0/0/20.0;
        ge-0/0/10.0;
        ge-0/0/0.0;
    }
}
```

In configuration mode, check your interface configurations by entering the **show interfaces** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
  [edit]
  user@switch# show interfaces
ge-0/0/0 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
        }
    }
}
ge-0/0/10 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
        }
    }
}
ge-0/0/20 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
        }
```

```
    }
}
```

If you are finished configuring the device, enter **commit** in configuration mode.

Configuring ERPS on Hairtail

CLI Quick Configuration

To quickly configure Hairtail, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set protocols rstp interface ge-0/0/10 disable
set protocols rstp interface ge-0/0/20 disable
set protocols protection-group ethernet-ring erp1
set protocols protection-group ethernet-ring erp1 data-channel erp-data-1
set protocols protection-group ethernet-ring erp1 data-channel erp-data-2
set protocols protection-group ethernet-ring erp1 control-vlan erp-control-vlan-1
set protocols protection-group ethernet-ring erp1 east-interface control-channel ge-0/0/20.0
set protocols protection-group ethernet-ring erp1 west-interface control-channel ge-0/0/10.0
```

Step-by-Step Procedure

To configure ERPS on Hairtail:

1. Disable any spanning- tree protocols configured on the ERPS interfaces. RSTP is enabled in the default configuration, so this example shows disabling RSTP:

   ```
   [edit protocols]
   user@switch# set rstp interface ge-0/0/10 disable
   user@switch# set rstp interface ge-0/0/20 disable
   ```

2. Create a node ring named erp1:

   ```
   [edit protocols]
   user@switch# set protection-group ethernet-ring erp1
   ```

3. Configure the control VLAN erp-control-vlan-1 for the node ring erp1:

   ```
   [edit protocols protection-group ethernet-ring erp1]
   user@switch# set control-vlan erp-control-vlan-1
   ```

4. Configure two data channels named erp-data-1 and erp-data-2 to define a set of VLAN IDs that belong to a ring instance:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set data-channel erp-data-1
user@switch# set data-channel erp-data-2
```

5. Configure the east interface of the node ring erp1 with the control channel ge-0/0/20.0 and indicate that it connects to a ring protection link:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set east-interface control-channel ge-0/0/20.0
```

6. Configure the west interface of the node ring erp1 with the control channel ge-0/0/10.0 and indicate that it connects to a ring protection link:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set west-interface control-channel ge-0/0/10.0
```

Results

In configuration mode, check your ERPS configuration by entering the **show protocols** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@switch# show protocols
rstp {
    interface ge-0/0/10.0 {
        disable;
    }
    interface ge-0/0/20.0 {
        disable;
    }
}
protection-group {
    ethernet-ring erp1 {
        east-interface {
            control-channel {
                ge-0/0/20.0;
            }
        }
```

```
        west-interface {
            control-channel {
            ge-0/0/10.0;
            }
        }
        control-vlan erp-control-vlan-1;
        data-channel {
            vlan [ 101-102 ];
        }
    }
}
```

In configuration mode, check your VLAN configuration by entering the **show vlans** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
  [edit]
  user@switch# show vlans
erp-control-vlan-1 {
    vlan-id 100;
    interface {
        ge-0/0/20.0;
        ge-0/0/10.0;
    }
}
erp-data-1 {
    vlan-id 101;
    interface {
        ge-0/0/0.0;
        ge-0/0/20.0;
        ge-0/0/10.0;
    }
}
erp-data-2 {
    vlan-id 102;
    interface {
        ge-0/0/0.0;
        ge-0/0/20.0;
        ge-0/0/10.0;
    }
}
```

In configuration mode, check your interface configurations by entering the **show interfaces** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
  [edit]
  user@switch# show interfaces
ge-0/0/0 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
        }
    }
}
ge-0/0/10 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
        }
ge-0/0/20 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
        }
    }
```

If you are finished configuring the device, enter **commit** in configuration mode.

Configuring ERPS on Jas6-esc

CLI Quick Configuration

To quickly configure Jas6-esc, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set protocols rstp interface ge-0/0/30 disable
set protocols rstp interface ge-0/0/20 disable
set protocols protection-group ethernet-ring erp1
set protocols protection-group ethernet-ring erp1 data-channel erp-data-1
set protocols protection-group ethernet-ring erp1 data-channel erp-data-2
set protocols protection-group ethernet-ring erp1 control-vlan erp-control-vlan-1
set protocols protection-group ethernet-ring erp1 east-interface control-channel ge-0/0/30.0
set protocols protection-group ethernet-ring erp1 west-interface control-channel ge-0/0/20.0
```

Step-by-Step Procedure

To configure ERPS on Jas6-esc:

1. Disable any spanning- tree protocols configured on the ERPS interfaces. RSTP is enabled in the default configuration, so this example shows disabling RSTP:

```
[edit protocols]
user@switch# set rstp interface ge-0/0/30 disable
user@switch# set rstp interface ge-0/0/20 disable
```

2. Create a node ring named erp1:

```
[edit protocols]
user@switch# set protection-group ethernet-ring erp1
```

3. Configure the control VLAN erp-control-vlan-1 for the node ring erp1:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set control-vlan erp-control-vlan-1
```

4. Configure two data channels named erp-data-1 and erp-data-2 to define a set of VLAN IDs that belong to a ring instance.

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set data-channel erp-data-1
user@switch# set data-channel erp-data-2
```

5. Configure the east interface of the node ring erp1 with the control channel ge-0/0/30.0 :

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set east-interface control-channel ge-0/0/30.0
```

6. Configure the west interface of the node ring erp1 with the control channel ge-0/0/20.0:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set west-interface control-channel ge-0/0/20.0
```

Results

In configuration mode, check your ERPS configuration by entering the **show protocols** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@switch# show protocols
rstp {
    interface ge-0/0/20.0 {
        disable;
    }
    interface ge-0/0/30.0 {
        disable;
    }
}
protection-group {
    ethernet-ring erp1 {
        east-interface {
            control-channel {
                ge-0/0/30.0;
            }
        }
        west-interface {
            control-channel {
                ge-0/0/20.0;
            }
        }
        control-vlan erp-control-vlan-1;
        data-channel {
            vlan [ 101-102 ];
        }
    }
}
```

In configuration mode, check your VLAN configuration by entering the **show vlans** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@switch# show vlans
erp-control-vlan-1 {
    vlan-id 100;
    interface {
        ge-0/0/30.0;
        ge-0/0/20.0;
```

```
    }
}
erp-data-1 {
    vlan-id 101;
    interface {
        ge-0/0/0.0;
        ge-0/0/30.0;
        ge-0/0/20.0;
    }
}
erp-data-2 {
    vlan-id 102;
    interface {
        ge-0/0/0.0;
        ge-0/0/30.0;
        ge-0/0/20.0;
    }
}
```

In configuration mode, check your interfaces configuration by entering the **show interfaces** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
  [edit]
  user@switch# show interfaces
ge-0/0/0 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
        }
    }
}
ge-0/0/20 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
        }
    }
}
ge-0/0/30 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
       }
```

```
    }
}
```

Verification

Verify that ERPS is working correctly.

Verifying That ERPS Is Working Correctly

Purpose

Verify that ERPS is working on the four EX switches that function as nodes in the ring topology.

Action

Check the state of the ring links in the output of the **show protection-group ethernet-ring interface** command. When the ring is configured but not being used (no error exists on the data links), one ERP interface is forwarding traffic and one is discarding traffic. Discarding blocks the ring.

```
user@switch> show protection-group ethernet-ring interface
```

```
Ethernet ring port parameters for protection group erp1
Interface    Forward State  RPL End  Signal Failure    Admin State
ge-0/0/2.0   discarding     yes      clear             ready
ge-0/0/0.0   forwarding     no       clear             ready
```

To find out what has occurred since the last restart, check the RPS statistics for ring-blocked events. **NR** is a No Request ring block, which means that the switch is not blocking either of the two ERP interfaces. **NR-RB** is a No Request Ring Blocked event, which means that the switch is blocking one of its ERP interfaces and sending a packet out to notify the other switches.

```
user@switch> show protection-group ethernet-ring statistics
```

```
Ring Name Local SF Remote SF NR Event NR-RB Event
erp1       2          1            2            3
```

Meaning

The **show protection-group ethernet-ring interface** command output from the RPL owner node indicates that one interface is forwarding traffic and one is discarding traffic, meaning that the ERP is ready but not active. If at least one interface in the ring is not forwarding, the ring is blocked and therefore ERP is working.

The **show protection-group ethernet-ring statistics** command output indicates that, since the last reboot, both local and remote signal failures have occurred (**Local SF** and **Remote SF**).

The **NR Event** count is 2, indicating that the NR state was entered into twice. **NR** stands for No Request. This means that the switch either originated NR PDUs or received an NR PDU from another switch and stopped blocking the interface to allow ERP to function.

The three **NR-RB** events indicate that on three occasions, this switch either sent out NR-RB PDUs or received NR-RB PDUs from another switch. This occurs when a network problem is resolved and the switch once again blocks the ERP link at one end.

RELATED DOCUMENTATION

- *Configuring Ethernet Ring Protection Switching on Switches (CLI Procedure)*
- *Ethernet Ring Protection Switching Overview*
- *Understanding Ethernet Ring Protection Switching Functionality*

Example: Configuring Ethernet Ring Protection Switching on QFX Series and EX Series Switches Supporting ELS

IN THIS SECTION

You can configure Ethernet ring protection switching (ERPS) on connected EX Series or QFX Series switches to prevent fatal loops from disrupting a network. ERPS is similar to the Spanning Tree Protocol, but ERPS is more efficient because it is customized for ring topologies. You must connect and configure at least three switches to form a ring.

This example shows how to configure Ethernet ring protection switching on four switches with ELS support, connected to one another on a dedicated link in a ring topology. You can include different types of switches

in an ERPS ring, including those with and without ELS support. If any of your EX Series switches runs software that does not support ELS, use these configuration directions: "Example: Configuring Ethernet Ring Protection Switching on EX Series Switches" on page 855. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50.

Requirements

This example uses the following hardware and software components:

- Four connected EX Series switches or QFX Series switches that support the Enhanced Layer 2 Software (ELS) to function as nodes in the ring topology. You could use any of these QFX Series switches: QFX5100, QFX5200, and QFX10000. This configuration also applies to EX Series switches that support the Enhanced Layer 2 Software (ELS) configuration style that runs on EX4300, EX4600, EX2300, and EX3400 switches.
- Junos OS Release 13.2X50-D10 or later for EX Series switches.
- Junos OS Release 14.1X53-D10 or later for QFX5100 switches. Junos OS Release 15.1X53-D30 or later for QFX5200, and QFX10000 switches.

Before you begin, be sure you have:

- Configured two trunk interfaces on each of the four switches. See Table 120 on page 858 for a list of the interface names used in this example.
- Configured a VLAN (with name **erp-control-vlan-1** and ID **100**) on all four switches and associated two network interfaces from each of the four switches with the VLAN. See *Configuring VLANs for the QFX Series* OR *Configuring VLANs for EX Series Switches with ELS Support (CLI Procedure)*. See Table 120 on page 858 for a list of the interface names used in this example.
- Configured two more VLANs (one with name **erp-data-1** and vlan ID **101** and a second vlan with the name **erp-data-2** and vlan ID **102**) on all four switches and associated both the east and west interfaces on each switch.

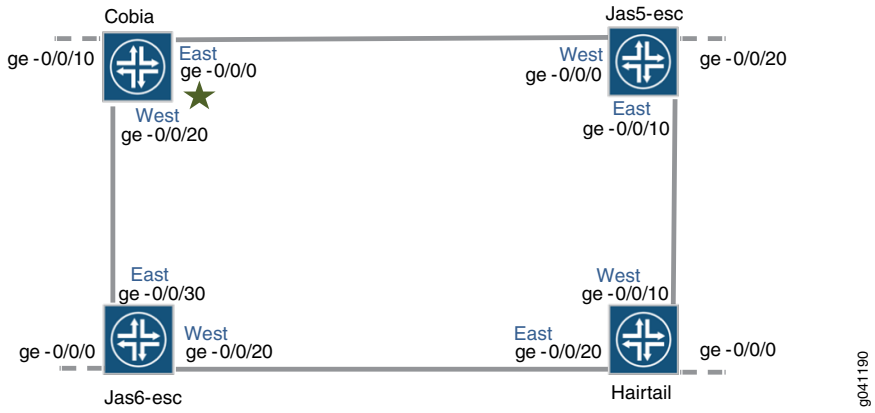
Overview and Topology

ERPS uses a dedicated physical link, including a control VLAN for trunk ports, between all of the switches to protect the active links. ERPS VLANs are all located on this link and are also blocked by default. When traffic between the switches is flowing with no problems, the active links take care of all traffic. Only if an error occurs on one of the data links would the ERPS control channel take over and start forwarding traffic.

NOTE: Trunk ports on switches use a VLAN to create individual control channels for ERPS. When multiple ERPS instances are configured for a ring, there are multiple sets of ring protection links (RPLs) and RPL owners on the ERPS link, and a different channel is blocked for each instance. Nontrunk ports use the physical link as the control channel and protocol data units (PDUs) are untagged, with no VLAN information in the packet.

This example creates one protection ring (called a node ring) named erp1 on four switches connected in a ring by trunk ports as shown in Figure 44 on page 857. Because the links are trunk ports, VLAN 100 is used for erp1 traffic. The east interface of each switch is connected with the west interface of an adjacent switch. Cobia is the RPL owner, with interface ge-0/0/0 configured as an RPL end interface. The interface ge-0/0/0 of Jas5-esc is configured as the RPL neighbor interface. In the idle state, the RPL end blocks the control VLAN and data channel VLAN for this particular ERP instance—the blocked port on Cobia is marked with a star in Figure 44 on page 857.

Figure 45: Ethernet Ring Protection Switching Example



In this example, we configure the four switches with the interfaces indicated in both Figure 44 on page 857 and Table 120 on page 858.

Table 121: Components to Configure for This Example

Interfaces	Cobia	Jas5-esc	Jas6-esc	Hairtail
East	ge-0/0/0	ge-0/0/10	ge-0/0/30	ge-0/0/20
West	ge-0/0/20	ge-0/0/0	ge-0/0/20	ge-0/0/10
Third	ge-0/0/10	ge-0/0/20	ge-0/0/0	ge-0/0/0

Configuration

IN THIS SECTION

Configuring ERPS on Cobia, the RPL Owner Node

CLI Quick Configuration

To quickly configure Cobia, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

ERPS cannot be configured on an interface if any spanning-tree protocol is configured. (RSTP is configured by default.) Therefore, in this example, RSTP is disabled on each ring port before configuring ERPS. Spanning-tree protocols are disabled two different ways, depending on which version of the Junos OS you are running. Therefore, the first two statements in this example vary: Junos release 15.1 or later uses one command to turn off RSTP and Junos releases prior to 15.1 uses another command.

Junos OS release 15.1 or later: **set protocols rstp interface ge-0/0/0 disable**
Junos OS release 15.1 or later: **set protocols rstp interface ge-0/0/20 disable**
Junos OS release prior to 15.1: **delete rstp interface ge-0/0/0**
Junos OS release prior to 15.1: **delete rstp interface ge-0/0/20**
set protocols protection-group ethernet-ring erp1
set protocols protection-group ethernet-ring erp1 ring-protection-link-owner
set protocols protection-group ethernet-ring erp1 data-channel 101
set protocols protection-group ethernet-ring erp1 data-channel 102
set protocols protection-group ethernet-ring erp1 control-vlan 100
set protocols protection-group ethernet-ring erp1 east-interface control-channel ge-0/0/0.0
set protocols protection-group ethernet-ring erp1 east-interface ring-protection-link-end
set protocols protection-group ethernet-ring erp1 east-interface control-channel ge-0/0/20.0
set protocols protection-group ethernet-ring erp1 west-interface control-channel vlan 100
set protocols protection-group ethernet-ring erp1 east-interface control-channel vlan 100

Step-by-Step Procedure

To configure ERPS on Cobia:

1. Disable any spanning-tree protocol currently configured on the ERPS interfaces. RSTP, VSTP, and MSTP are all available spanning-tree protocols. RSTP is enabled in the default configuration, so this example shows disabling RSTP. Spanning-tree protocols are disabled two different ways, depending on which version of the Junos OS you are running.

   If you are running Junos release 15.1 or later, disable any spanning-tree protocol with these commands. To disable RSTP:

   ```
   [edit protocols]
   user@switch# set rstp interface ge-0/0/0 disable
   user@switch# set rstp interface ge-0/0/20 disable
   ```

   If you are running a Junos release prior to 15.1, disable any spanning-tree protocol with these commands. To disable RSTP:

   ```
   [edit protocols]
   user@switch# delete rstp interface ge-0/0/0
   user@switch# delete rstp interface ge-0/0/20
   ```

2. Create a node ring named erp1:

   ```
   [edit protocols]
   user@switch# set protection-group ethernet-ring erp1
   ```

3. Designate Cobia as the RPL owner node:

   ```
   [edit protocols protection-group ethernet-ring erp1]
   user@switch# set ring-protection-link-owner
   ```

4. Configure the VLANs 101 and 102 as data channels:

   ```
   [edit protocols protection-group ethernet-ring erp1]
   user@switch# set data-channel 101
   user@switch# set data-channel 102
   ```

5. Configure the control vlan 100 for this ERPS instance on the trunk interface:

   ```
   [edit protocols protection-group ethernet-ring erp1]
   ```

```
user@switch# set control-vlan 100
```

6. Configure the east interface of the node ring erp1 with control channel ge-0/0/0.0 and indicate that this particular ring protection link ends here:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set east-interface control-channel ge-0/0/0.0
user@switch# set east-interface ring-protection-link-end
```

7. Configure the west interface of the node ring erp1 with control channel ge-0/0/20.0 :

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set west-interface control-channel ge-0/0/20.0
```

8. Every ring instance on a trunk port has one control VLAN in which ERP packets traverse. The control VLAN also controls data VLANs, if any are configured. Assign 100 as the control VLAN on both interfaces:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set west-interface control-channel vlan 100
user@switch# set east-interface control-channel
```

Configuring ERPS on Jas5-esc

CLI Quick Configuration

To quickly configure Jas5-esc, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

ERPS cannot be configured on an interface if any spanning tree protocol is configured. (RSTP is configured by default.) Therefore, in this example, RSTP is disabled on each ring port before configuring ERPS. Spanning tree is disabled two different ways, depending on which version of the Junos OS you are running. Therefore, the first two statements will vary: Junos release 15.1 or later uses one command to turn off RSTP and Junos releases prior to 15.1 uses another command.

```
Junos OS release 15.1 or later: set protocols rstp interface ge-0/0/10 disable
Junos OS release 15.1 or later: set protocols rstp interface ge-0/0/0 disable
Junos OS release prior to 15.1: delete rstp interface ge-0/0/10
Junos OS release prior to 15.1: delete rstp interface ge-0/0/0
set protocols protection-group ethernet-ring erp1
```

```
set protocols protection-group ethernet-ring erp1 data-channel 101
set protocols protection-group ethernet-ring erp1 data-channel 102
set protocols protection-group ethernet-ring erp1 control-vlan 100
set protocols protection-group ethernet-ring erp1 east-interface control-channel ge-0/0/10.0
set protocols protection-group ethernet-ring erp1 east-interface control-channel vlan 100 ge-0/0/0.0
set protocols protection-group ethernet-ring erp1 west-interface control-channel ge-0/0/20.0
set protocols protection-group ethernet-ring erp1 west-interface control-channel vlan 100
set protocols protection-group ethernet-ring erp1 west-interface control-channel vlan 100
```

Step-by-Step Procedure

To configure ERPS on Jas5-esc:

1. Disable any spanning-tree protocol currently configured on the ERPS interfaces. RSTP, VSTP, and MSTP are all available spanning-tree protocols. RSTP is enabled in the default configuration, so this example shows disabling RSTP. Spanning-tree protocols are disabled two different ways, depending on which version of the Junos OS you are running.

   If you are running Junos release 15.1 or later, disable any spanning-tree protocol with these commands. To disable RSTP:

   ```
   [edit protocols]
   user@switch# set rstp interface ge-0/0/10 disable
   user@switch# set rstp interface ge-0/0/0 disable
   ```

   If you are running a Junos release prior to 15.1, disable any version of spanning-tree protocol with these commands. To disable RSTP:

   ```
   [edit protocols]
   user@switch# delete rstp interface ge-0/0/10
   user@switch# delete rstp interface ge-0/0/0
   ```

2. Create a node ring named erp1:

   ```
   [edit protocols]
   user@switch# set protection-group ethernet-ring erp1
   ```

3. Configure two data channels named erp-data-1 and erp-data-2 to define a set of VLAN IDs that belong to a ring instance.

   ```
   [edit protocols protection-group ethernet-ring erp1]
   user@switch# set data-channel vlan 101
   ```

```
user@switch# set data-channel vlan 102
```

4. Configure a control VLAN with ID 100 for the node ring erp1:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set control-vlan 100
```

5. Configure the east interface of the node ring erp1 with the control channel ge-0/0/10.0:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set east-interface control-channel ge-0/0/10.0
```

6. Configure the west interface of the node ring erp1 with the control channel ge-0/0/0.0 vlan 100:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set west-interface control-channel ge-0/0/0.0
```

7. Every ring instance on a trunk port has one control VLAN in which ERP packets traverse. The control VLAN also controls data VLANs, if any are configured. Assign vlan # 100 as the control VLAN:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set west-interface control-channel vlan 100
user@switch# set east-interface control-channel vlan 100
```

Configuring ERPS on Hairtail

CLI Quick Configuration

To quickly configure Hairtail, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

ERPS cannot be configured on an interface if any spanning tree protocol is configured. (RSTP is configured by default.) Therefore, in this example, RSTP is disabled on each ring port before configuring ERPS. Spanning tree is disabled two different ways, depending on which version of the Junos OS you are running. Therefore, the first two statements will vary: Junos release 15.1 or later uses one command to turn off RSTP and Junos releases prior to 15.1 uses another command.

Junos OS release 15.1 or later: **set protocols rstp interface ge-0/0/10 disable**

```
Junos OS release 15.1 or later: set protocols rstp interface ge-0/0/20 disable
Junos OS release prior to 15.1: delete rstp interface ge-0/0/10
Junos OS release prior to 15.1: delete rstp interface ge-0/0/20
set protocols protection-group ethernet-ring erp1
set protocols protection-group ethernet-ring erp1 data-channel 101
set protocols protection-group ethernet-ring erp1 data-channel 102
set protocols protection-group ethernet-ring erp1 control-vlan 100
Set protocols protection-group ethernet-ring erp1 east-interface control-channel ge-0/0/0.0
set protocols protection-group ethernet-ring erp1 east-interface control-channel ge-0/0/20.0
set protocols protection-group ethernet-ring erp1 west-interface control-channel vlan 100
set protocols protection-group ethernet-ring erp1 east-interface control-channel vlan 100
```

Step-by-Step Procedure

To configure ERPS on Hairtail:

1. Disable any spanning-tree protocol currently configured on the ERPS interfaces. RSTP, VSTP, and MSTP are all available spanning-tree protocols. RSTP is enabled in the default configuration, so this example shows disabling RSTP. Spanning-tree protocols are disabled two different ways, depending on which version of the Junos OS you are running.

   If you are running Junos release 15.1 or later, disable any spanning-tree protocol with these commands. To disable RSTP:

   ```
   [edit protocols]
   user@switch# set rstp interface ge-0/0/10 disable
   user@switch# set rstp interface ge-0/0/20 disable
   ```

   If you are running a Junos release prior to 15.1, disable any spanning-tree protocol with these commands. To disable RSTP:

   ```
   [edit protocols]
   user@switch# delete rstp interface ge-0/0/10
   user@switch# delete rstp interface ge-0/0/20
   ```

2. Create a node ring named erp1:

   ```
   [edit protocols]
   user@switch# set protection-group ethernet-ring erp1
   ```

3. Configure the control vlan 100 for the node ring erp1:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set control-vlan 100
```

4. Configure two data channels numbered 101 and 102 to define a set of VLAN IDs that belong to a ring instance:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set data-channel vlan 101
user@switch# set data-channel vlan 102
```

5. Configure the east interface of the node ring erp1 with the control channel ge-0/0/20.0:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set east-interface control-channel ge-0/0/20.0
```

6. Configure the west interface of the node ring erp1 with the control channel ge-0/0/10.0:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set west-interface control-channel ge-0/0/10.0
```

7. Every ring instance on a trunk port has one control VLAN in which ERP packets traverse. The control VLAN also controls data VLANs, if any are configured. Assign 100 as the control VLAN:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set west-interface control-channel vlan 100
user@switch# set east-interface control-channel vlan 100
```

Configuring ERPS on Jas6-esc

CLI Quick Configuration

To quickly configure Jas6-esc, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

ERPS cannot be configured on an interface if any spanning tree protocol is configured. (RSTP is configured by default.) Therefore, in this example, RSTP is disabled on each ring port before configuring ERPS. Spanning tree is disabled two different ways, depending on which version of the Junos OS you are running. Therefore, the first two statements will vary: Junos release 15.1 or later uses one command to turn off RSTP and Junos releases prior to 15.1 uses another command.

```
Junos OS release 15.1 or later: set protocols rstp interface ge-0/0/30 disable
Junos OS release 15.1 or later: set protocols rstp interface ge-0/0/20 disable
Junos OS release prior to 15.1: delete rstp interface ge-0/0/30
Junos OS release prior to 15.1: delete rstp interface ge-0/0/20
set protocols protection-group ethernet-ring erp1
set protocols protection-group ethernet-ring erp1 data-channel 101
set protocols protection-group ethernet-ring erp1 data-channel 102
set protocols protection-group ethernet-ring erp1 control-vlan 100
set protocols protection-group ethernet-ring erp1 east-interface control-channel ge-0/0/30.0
set protocols protection-group ethernet-ring erp1 west-interface control-channel ge-0/0/20.0
set protocols protection-group ethernet-ring erp1 west-interface control-channel vlan 100
set protocols protection-group ethernet-ring erp1 east-interface control-channel vlan 100
```

Step-by-Step Procedure

To configure ERPS on Jas6-esc:

1. Disable any spanning-tree protocol currently configured on the ERPS interfaces. RSTP, VSTP, and MSTP are all available spanning-tree protocols. RSTP is enabled in the default configuration, so this example shows disabling RSTP. Spanning-tree protocols are disabled two different ways, depending on which version of the Junos OS you are running.

   If you are running Junos release 15.1 or later, disable any spanning-tree protocol with these commands. To disable RSTP:

   ```
   [edit protocols]
   user@switch# set rstp interface ge-0/0/30 disable
   user@switch# set rstp interface ge-0/0/20 disable
   ```

   If you are running a Junos release prior to 15.1, disable any spanning-tree protocol with these commands. To disable RSTP:

   ```
   [edit protocols]
   user@switch# delete rstp interface ge-0/0/30
   user@switch# delete rstp interface ge-0/0/20
   ```

2. Create a node ring named erp1:

   ```
   [edit protocols]
   user@switch# set protection-group ethernet-ring erp1
   ```

3. Configure the control vlan 100 for the node ring erp1:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set control-vlan 100
```

4. Configure two data channels numbered 101 and 102 to define VLAN IDs that belong to a ring instance.

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set data-channel 101
user@switch# set data-channel 102
```

5. Configure the east interface of the node ring erp1 with the control channel ge-0/0/30.0 :

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set east-interface control-channel ge-0/0/30.0
```

6. Configure the west interface of the node ring erp1 with the control channel ge-0/0/20.0:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set west-interface control-channel ge-0/0/20.0
```

7. Every ring instance on a trunk port has one control VLAN in which ERP packets traverse. The control VLAN also controls data VLANs, if any are configured. Assign vlan number 100 as the control VLAN:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set west-interface control-channel vlan 100
user@switch# set east-interface control-channel vlan 100
```

RELATED DOCUMENTATION

- *Configuring Ethernet Ring Protection Switching on Switches (CLI Procedure)*
- *Ethernet Ring Protection Switching Overview*
- *Understanding Ethernet Ring Protection Switching Functionality*

CHAPTER 29

Configuring Q-in-Q Tunneling and VLAN Translation

Configuring Q-in-Q Tunneling and VLAN Q-in-Q Tunneling and VLAN Translation

IN THIS SECTION

Understanding Q-in-Q Tunneling and VLAN Translation

IN THIS SECTION

Q-in-Q tunneling and VLAN translation allow service providers to create a Layer 2 Ethernet connection between two customer sites. Providers can segregate different customers' VLAN traffic on a link (for example, if the customers use overlapping VLAN IDs) or bundle different customer VLANs into a single service VLAN. Data centers can use Q-in-Q tunneling and VLAN translation to isolate customer traffic within a single site or to enable customer traffic flows between cloud data centers in different geographic locations.

Using Q-in-Q tunneling, providers can segregate or bundle customer traffic into fewer VLANs or different VLANs by adding another layer of 802.1Q tags. Q-in-Q tunneling is useful when customers have overlapping VLAN IDs, because the customer's 802.1Q (dot1Q) VLAN tags are prepended by the service VLAN (S-VLAN) tag. The Juniper Networks Junos operating system (Junos OS) implementation of Q-in-Q tunneling supports the IEEE 802.1ad standard.

This topic describes:

How Q-in-Q Tunneling Works

In Q-in-Q tunneling, as a packet travels from a customer VLAN (C-VLAN) to a service provider's VLAN, a customer-specific 802.1Q tag is added to the packet. This additional tag is used to segregate traffic into service-provider-defined service VLANs (S-VLANs). The original customer 802.1Q tag of the packet remains and is transmitted transparently, passing through the service provider's network. As the packet leaves the S-VLAN in the downstream direction, the extra 802.1Q tag is removed.

> **NOTE:** All of the VLANs in an implementation can be service VLANs. That is, if the total number of supported VLANs is 4090, all of them can be service VLANs.

When Q-in-Q tunneling is enabled on Juniper Networks EX Series Ethernet Switches, trunk interfaces are assumed to be part of the service provider network and access interfaces are assumed to be customer facing. An access interface can receive both tagged and untagged frames in this case.

> **NOTE:** Starting with Junos OS 14.1X53-D30, you can configure the same interface to be an S-VLAN/NNI interface and a C-VLAN/UNI interface. This means that the same physical interface can transmit single-tagged and double-tagged frames simultaneously. This allows you maximum flexibility in your network topology and lets you maximize the use of your interfaces.

An interface can be a member of multiple S-VLANs. You can map one C-VLAN to one S-VLAN (1:1) or multiple C-VLANs to one S-VLAN (N:1). Packets are double-tagged for an additional layer of segregating or bundling of C-VLANs. C-VLAN and S-VLAN tags are unique; so you can have both a C-VLAN 101 and an S-VLAN 101, for example. You can limit the set of accepted customer tags to a range of tags or to discrete values. Class-of-service (CoS) values of C-VLANs are unchanged in the downstream direction.

You may, optionally, copy ingress priority and CoS settings to the S-VLAN. On non-ELS switches, you can use private VLANs to isolate users to prevent the forwarding of traffic between user interfaces even if the interfaces are on the same VLAN.

When Q-in-Q tunneling is enabled, trunk interfaces are assumed to be part of the service provider or data center network. Access interfaces are assumed to be customer-facing and accept both tagged and untagged frames. When using many-to-one bundling or mapping a specific interface, you must use the **native** option to specify an S-VLAN for untagged and priority tagged packets if you want to accept these packets. (Priority tagged packets have their VLAN ID set to 0, and their priority code point bits might be configured with a CoS value.)

> **NOTE:** Priority tagged packets are not supported with Q-in-Q tunneling on QFX5100 and EX4600 switches.

If you do not specify an S-VLAN for them, untagged packets are discarded. The **native** option is not available for all-in-one bundling because there is no need to specify untagged and priority tagged packets when all packets are mapped to an S-VLAN.

You can use the **native** option to specify an S-VLAN for untagged and priority tagged packets when using many-to-one bundling and mapping a specific interface approaches to map C-VLANs to S-VLANs. (This does not apply to switches supporting ELS.) Otherwise the packets are discarded. The **native** option is not available for all-in-one bundling because there is no need to specify untagged and priority tagged packets when all packets are mapped to the S-VLAN. See the Mapping C-VLANs to S-VLANs section of this document for information on the methods of mapping C-VLANs to S-VLANs.

On QFabric systems only, you can use the **native** option to apply a specified inner tag to packets that ingress as untagged on access interfaces. This functionality is useful if your QFabric system connects to servers that host customer virtual machines that send untagged traffic and each customer's traffic requires its own VLAN while being transported through the QFabric. Instead of using individual VLANs for each customer (which can quickly lead to VLAN exhaustion), you can apply a unique inner (C-VLAN) tag to each customer's traffic and then apply a single outer tag (S-VLAN) tag for transport through the QFabric. This allows you to segregate your customers's traffic while consuming only one QFabric VLAN. Use the **inner-tag** option of the **mapping** statement to accomplish this.

On non-ELS switches, firewall filters allow you to map an interface to a VLAN based on a policy. Using firewall filters to map an interface to a VLAN is useful when you want a subset of traffic from a port to be mapped to a selected VLAN instead of the designated VLAN. To configure a firewall filter to map an interface to a VLAN, the **vlan** option has to be configured as part of the firewall filter and the **mapping policy** option must be specified in the interface configuration for each logical interface using the filter.

> **NOTE:** On an EX4300 switch, you can configure multiple logical interfaces on the same Ethernet port, but each logical interface supports only single-tagged packets and that tag must include a different VLAN ID than those supported by the other logical interfaces. Given this situation, you cannot enable Q-in-Q tunneling on Ethernet ports with multiple logical subinterfaces.

Q-in-Q tunneling does not affect any class-of-service (CoS) values that are configured on a C-VLAN. These settings are retained in the C-VLAN tag and can be used after a packet leaves an S-VLAN. CoS values are not copied from C-VLAN tags to S-VLAN tags.

Depending on your interface configuration, you might need to adjust the MTU value on your trunk or access ports to accommodate the 4 bytes used for the tag added by Q-in-Q tunneling. For example, if you use the default MTU value of 1514 bytes on your access and trunk ports, you need to make one of the following adjustments:

- Reduce the MTU on the access links by at least 4 bytes so that the frames do not exceed the MTU of the trunk link when S-VLAN tags are added.
- Increase the MTU on the trunk link so that the link can handle the larger frame size.

> **NOTE:** You can configure Q-in-Q tunneling only on access ports (not trunk ports).

How VLAN Translation Works

VLAN translation replaces an incoming C-VLAN tag with an S-VLAN tag instead of adding an additional tag. The C-VLAN tag is therefore lost, so a single-tagged packet is normally untagged when it leaves the S-VLAN (at the other end of the link). If an incoming packet has had Q-in-Q tunneling applied in advance, VLAN translation replaces the outer tag and the inner tag is retained when the packet leaves the S-VLAN at the other end of the link. Incoming packets whose tags do not match the C-VLAN tag are dropped, unless additional VLAN translation configuration for those tags exist.

To configure VLAN translation, use the **mapping** **swap** statement at the **[edit vlans interface]** hierarchy level. As long as the C-VLAN and S-VLAN tags are unique, you can configure more than one C-VLAN-to-S-VLAN translation on an access port. If you are translating only one VLAN on an interface, you do not need to include the **dot1q-tunneling** statement in the S-VLAN configuration. If you are translating more than one VLAN, you must use the **dot1q-tunneling** statement.

NOTE: You can configure VLAN translation on access ports only. You cannot configure it on trunk ports, and you cannot configure Q-in-Q tunneling on the same access port. You can configure only one VLAN translation for a given VLAN and interface. For example, you can create no more than one translation for VLAN 100 on interface xe-0/0/0.

NOTE: VLAN translation is not supported on QFabric systems.

Using Dual VLAN Tag Translation

Starting with Junos OS Release 14.1X53-D40, you can use the dual VLAN tag translation (also known as dual VLAN tag rewrite) feature to deploy switches in service-provider domains, allowing dual-tagged, single-tagged, and untagged VLAN packets to come into or exit from the switch. Table 122 on page 891 shows the operations that are added for dual VLAN tag translation.

Table 122: Operations Added with Dual VLAN Tag Rewrite

Operation	Function
swap-push	Swap a VLAN tag and push a new VLAN tag
pop-swap	Pop an outer VLAN tag and swap an inner VLAN tag
swap-swap	Swap both outer and inner VLAN tags

Dual VLAN tag translation supports:

- Configuration of S-VLANs (NNI) and C-VLANs (UNI) on the same physical interface
- Control protocols such as VSTP, OSPF, and LACP
- IGMP snooping
- Configuration of a private VLAN (PVLAN) and VLAN on a single-tagged interface
- Use of TPID 0x8100 on both inner and outer VLAN tags

See "Setting Up a Dual VLAN Tag Translation Configuration on QFX Switches" on page 929.

Sending and Receiving Untagged Packets

To enable an interface to send and receive untagged packets, you must specify a native VLAN for a physical interface. When the interface receives an untagged packet, it adds the VLAN ID of the native VLAN to

the packet in the C-VLAN field and adds the S-VLAN tag as well (so the packet is double-tagged), and sends the newly tagged packet to the mapped interface.

The preceding paragraph does *not* apply to:

- Non-ELS switches.
- EX4300 switches running under a Junos release prior to Junos OS Release 19.3R1.

When the switches in the short list above receive an untagged packet, they add the S-VLAN tag to the packet (so the packet is single-tagged) and send the newly tagged packet to the mapped interface.

> **NOTE:** Ensure that all switches configured in your Q-in-Q setup operate with either the single-tag approach or the double-tag approach. The setup will not work if the switches do not have the same approach.

Starting in Junos OS Release 19.3R1, you can configure EX4300 switches to use the double-tag approach. Set the configuration statement **input-native-vlan-push** to **enable** and ensure that the **input-vlan-map** configuration statement is set to **push**, as shown in the following example:

```
[edit interfaces ge-1/0/45]
flexible-vlan-tagging;
native-vlan-id 20;
input-native-vlan-push enable;
encapsulation extended-vlan-bridge;
unit 10 {
     vlan-id-list 10-100;
     input-vlan-map push;
     output-vlan-map pop;
}
```

> **NOTE:** On switches that support this feature, except for the EX4300 switch, the **input-native-vlan-push** statement is set to **enable** by default. (The **input-native-vlan-push** statement is set to **disable** by default on the EX4300 switch.) However, we recommend that you check the configuration to ensure that **input-vlan-map** is set to **push**—the feature does not work if that setting isn't in place.

To specify a native VLAN, use the **native-vlan-id** statement at the **[edit interfaces *interface-name*]** hierarchy level. The native VLAN ID must match the C-VLAN or S-VLAN ID or be included in the VLAN ID list specified on the logical interface.

For example, on a logical interface for a C-VLAN interface, you might specify a C-VLAN ID list of 100-200. Then, on the C-VLAN physical interface, you could specify a native VLAN ID of 150. This configuration would work because the native VLAN of 150 is included in the C-VLAN ID list of 100-200.

We recommend configuring a native VLAN when using any of the approaches to map C-VLANs to S-VLANs. See the Mapping C-VLANs to S-VLANs section in this topic for information about the methods of mapping C-VLANs to S-VLANs.

Disabling MAC Address Learning

In a Q-in-Q deployment, customer packets from downstream interfaces are transported without any changes to source and destination MAC addresses. You can disable MAC address learning at global, interface, and VLAN levels:

- To disable learning globally, disable MAC address learning for the switch.
- To disable learning for an interface, disable MAC address learning for all VLANs of which the specified interface is a member.
- To disable learning for a VLAN, disable MAC address learning for a specified VLAN.

Disabling MAC address learning on an interface disables learning for all the VLANs of which that interface is a member. When you disable MAC address learning on a VLAN, MAC addresses that have already been learned are flushed.

If you disable MAC address learning on an interface or a VLAN, you cannot include 802.1X authentication in that same VLAN configuration.

When a routed VLAN interface (RVI) is associated with either an interface or a VLAN on which MAC address learning is disabled, the Layer 3 routes resolved on that VLAN or that interface are not resolved with the Layer 2 component. This results in routed packets flooding all the interfaces associated with the VLAN.

Mapping C-VLANs to S-VLANs

IN THIS SECTION

There are multiple ways to map C-VLANs to an S-VLAN:

> **NOTE:** If you configure multiple mapping methods, the switch gives priority to mapping a specific interface, then to many-to-many bundling, and last to all-in-one bundling. However, for a particular mapping method, setting up overlapping rules for the same C-VLAN is not supported.

- All-in-one bundling—Use the **edit vlans *s-vlan-name* dot1q-tunneling** statement without specifying customer VLANs. All packets received on all access interfaces (including untagged packets) are mapped to the S-VLAN.
- Many-to-one bundling—Use the **edit vlans *s-vlan-name* dot1q-tunneling customer-vlans** statement to specify which C-VLANs are mapped to the S-VLAN. Use this method when you want a subset of the C-VLANs to be part of the S-VLAN. If you want untagged or priority tagged packets to be mapped to the S-VLAN, use the **native** option with the **customer-vlans** statement. (Priority tagged packets have their VLAN ID set to 0, and their priority code point bits might be configured with a CoS value.)
- Many-to-many bundling—Use many-to-many bundling when you want a subset of the C-VLANs on the access switch to be part of multiple S-VLANs.
- Mapping a specific interface—Use the **edit vlans *s-vlan-name* interface *interface-name* mapping** statement to specify a C-VLAN for a given S-VLAN. This configuration applies to only one interface—not all access interfaces as with all-in-one and many-to-one bundling. If you want untagged or priority tagged packets to be mapped to the S-VLAN, use the **native** option with the **customer-vlans** statement.

  This method has two options: swap and push. With the push option, a packet retains its tag and an additional VLAN tag is added. With the swap option, the incoming tag is replaced with an S-VLAN tag. (This is VLAN translation.)

  - You can configure multiple push rules for a given S-VLAN and interface. That is, you can configure an interface so that the same S-VLAN tag is added to packets arriving from multiple C-VLANs.
  - You can configure only one swap rule for a given S-VLAN and interface.

  This functionality is typically used to keep traffic from different customers separate or to provide individualized treatment for traffic on a certain interface.

If you configure multiple methods, the switch gives priority to mapping a specific interface, then to many-to-one bundling, and last to all-in-one bundling. However, you cannot have overlapping rules for the same C-VLAN under a given approach. For example, you cannot use many-to one bundling to map C-VLAN 100 to two different S-VLANs.

All-in-One Bundling

All-in-one bundling maps all packets from all C-VLAN interfaces to an S-VLAN.

The C-VLAN interface accepts untagged and single-tagged packets. An S-VLAN 802.1Q tag is then added to these packets, and the packets are sent to the S-VLAN interface, which accepts untagged, single-tagged, and double-tagged packets.

> **NOTE:** The C-VLAN and S-VLAN interfaces accept untagged packets provided that the **native-vlan-id** statement is configured on these interfaces.

Many-to-One Bundling

Many-to-one bundling is used to specify which C-VLANs are mapped to an S-VLAN. Many-to-one bundling is configured using the **customer-vlans** option.

Many-to-one bundling is used when you want a subset of the C-VLANs on the access switch to be part of the S-VLAN. When using many-to-one bundling, untagged and priority tagged packets can be mapped to the S-VLAN when the **native** option is specified along with the **customer-vlans** option.

Many-to-Many Bundling

Many-to-many bundling is used to specify which C-VLANs are mapped to which S-VLANs.

Use many-to-many bundling when you want a subset of the C-VLANs on the access switch to be part of multiple S-VLANs. With many-to-many bundling, the C-VLAN interfaces accept untagged and single-tagged packets. An S-VLAN 802.1Q tag is then added to these packets, and the packets are sent to the S-VLAN interfaces, which accept untagged, single-tagged, and double-tagged packets.

> **NOTE:** The C-VLAN and S-VLAN interfaces accept untagged packets provided that the **native-vlan-id** statement is configured on these interfaces.

Mapping a Specific Interface

Use specific interface mapping when you want to assign an S-VLAN to a specific C-VLAN on an interface. The configuration applies only to the specific interface, not to all access interfaces.

Specific interface mapping has two suboptions: **push** and **swap**. When traffic that is mapped to a specific interface is pushed, the packet retains its original tag as it moves from the C-VLAN to the S-VLAN and an additional S-VLAN tag is added to the packet. When traffic that is mapped to a specific interface is swapped, the incoming tag is replaced with a new VLAN tag. This is sometimes known VLAN rewriting or VLAN translation.

Typically, this method is used to keep data from different customers separate or to provide individualized treatment of the packets on a certain interface. You might also use this method map VLAN traffic from different customers to a single S-VLAN.

When using specific interface mapping, the C-VLAN interfaces accept untagged and single-tagged packets, while the S-VLAN interfaces accept untagged, single-tagged, and double-tagged packets.

> **NOTE:** The C-VLAN and S-VLAN interfaces accept untagged packets provided that the **native-vlan-id** statement is configured on these interfaces.

Combining Methods and Configuration Restrictions

If you configure multiple methods, the switch gives priority to mapping a specific interface, then to many-to-one bundling, and last to all-in-one bundling. An access interface configured under all-in-one bundle cannot be part of a many-to-one bundle. It can have additional mappings defined, however.

To ensure deterministic results, the following configuration restrictions apply:

- Mapping cannot be defined for untagged vlans.
- An access interface can have multiple customer VLAN ranges, but an interface cannot have overlapping tags across the VLANs.

  For example, the following configuration is not allowed:

```
vlans {
    customer-1 {
        vlan-id 100;                  /* S-VLAN */
        interfaces ge-0/0/0.0;        /* Downstream */
        interfaces ge-0/0/1.0;        /* Downstream */
        interfaces xe-0/1/0.0;        /* trunk */
        dot1q-tunnelling customer-vlans 100-200 300-400
        }
    }
    customer-2 {
        vlan-id 200;
        interfaces ge-0/0/0.0;        /* Downstream */
        interfaces xe-0/1/0.0;        /* trunk */
        dot1q-tunnelling customer-vlans 250-350
        }
    }
    customer-3 {
        vlan-id 300;
        interfaces ge-0/0/1.0;        /* Downstream */
        interfaces xe-0/1/0.0;        /* trunk */
        dot1q-tunnelling customer-vlans 500-600
    }
}
```

Because the **customer-2** configuration creates overlapping **customer-vlan** ranges for ge-0/0/0, it is invalid.

- An access interface can have a single rule that maps an untagged packet to a VLAN.
- Each interface can have at most one mapping swap rule per VLAN.
- You can push a VLAN tag only on the access ports of a Q-in-Q VLAN. This restriction applies to all three methods of pushing a VLAN tag: that is, all-in-one bundling, many-to-one-bunding, and mapping a specific interface using push.
- You can push different C-VLAN tags for a given S-VLAN on different interfaces. This could potentially result in traffic leaking across VLANs, depending on your configuration.

Routed VLAN Interfaces on Q-in-Q VLANs

Routed VLAN interfaces (RVIs) are supported on Q-in-Q VLANs.

Packets arriving on an RVI that is using Q-in-Q VLANs will get routed regardless of whether the packet is single or double tagged. The outgoing routed packets contain an S-VLAN tag only when exiting a trunk interface; the packets exit the interface untagged when exiting an access interface.

Constraints for Q-in-Q Tunneling and VLAN Translation

Be aware of the following constraints when configuring Q-in-Q tunneling and VLAN translation:

- Q-in-Q tunneling supports only two VLAN tags.
- Q-in-Q tunneling does not support most access port security features. There is no per-VLAN (customer) policing or per-VLAN (outgoing) shaping and limiting with Q-in-Q tunneling unless you configure these security features by using firewall filters.
- With releases of Junos OS Release 13.2X51 previous to Release 13.2X51-D20, you cannot create a regular VLAN on an interface if you have created an S-VLAN or C-VLAN on that interface for Q-in-Q tunneling. This means that you cannot create an integrated routing and bridging (IRB) interface on that interface because regular VLANs are a required part of IRB configuration. With Junos OS Release 13.2X51-D25, you can create a regular VLAN on a trunk interface that has an S-VLAN, which means that you can also create an IRB interface on the trunk. In this case, the regular VLAN and S-VLAN on the same trunk interface cannot share the same VLAN ID. Junos OS Release 13.2X51-D25 does not allow you to create a regular VLAN on an access interface that has a C-VLAN.
- Starting with Junos OS Release 14.1X53-D40, integrated routing and bridging (IRB) interfaces are supported on Q-in-Q VLANs—you can configure the IRB interface on the same interface as one used by an S-VLAN, and you can use the same VLAN ID for both the VLAN used by the IRB interface and for the VLAN used as an S-VLAN.

Packets arriving on an IRB interface that is using Q-in-Q VLANs will get routed regardless of whether the packet is single tagged or double tagged. The outgoing routed packets contain an S-VLAN tag only when exiting a trunk interface; the packets exit the interface untagged when exiting an access interface.

> **NOTE:** You can configure the IRB interface only on S-VLAN (NNI) interfaces, not on C-VLAN (UNI) interfaces.

- Most access port security features are not supported with Q-in-Q tunneling and VLAN translation.
- Configuring Q-in-Q tunneling and VLAN rewriting/VLAN translation on the same port is not supported.
- You can configure at most one VLAN rewrite/VLAN translation for a given VLAN and interface. For example, you can create no more than one translation for VLAN 100 on interface xe-0/0/0.
- The combined total of VLANs and rules for Q-in-Q tunneling and VLAN translation cannot exceed 6000. For example, you can configure and commit 4000 VLANs and 2000 rules for Q-in-Q tunneling and VLAN translation. However, you cannot configure 4000 VLANs and 2500 rules for Q-in-Q tunneling and VLAN translation. If you try to commit a configuration that exceeds the limit, you see CLI and syslog errors that inform you about the problem.
- You cannot use the native VLAN ID.
- MAC addresses are learned from S-VLANs, not C-VLANs.
- Broadcast, unknown unicast, and multicast traffic is forwarded to all members in the S-VLAN.
- The following features are not supported with Q-in-Q tunneling:
  - DHCP relay
  - Fibre Channel over Ethernet
  - IP Source Guard
- The following features are not supported with VLAN rewriting/VLAN translation:
  - Fibre Channel over Ethernet
  - Firewall filter applied to a port or VLAN in the output direction
  - Private VLANs
  - VLAN Spanning Tree Protocol
  - Reflective relay

Configuring Q-in-Q Tunneling on QFX Series Switches

Q-in-Q tunneling and VLAN translation allow service providers to create a Layer 2 Ethernet connection between two customer sites. Providers can segregate different customers' VLAN traffic on a link (for example, if the customers use overlapping VLAN IDs) or bundle different customer VLANs into a single service VLAN. Data centers can use Q-in-Q tunneling to isolate customer traffic within a single site or when customer traffic flows between cloud data centers in different geographic locations.

Starting in Junos OS Release 19.4R1, the QFX10000 line of switches support the third and fourth Q-in-Q tags as payload (also known as a pass-through tag) along with the existing two tags (for VLAN matching and operations). The QFX10000 switches support multiple Q-in-Q tags for both Layer 2 bridging and EVPN-VXLAN cases. The Layer 2 access interfaces accept packets with three or four tags (all tags with the TPID value 0x8100). All the tags beyond the fourth tag (that is, from the fifth tag onward) are considered part of the Layer 3 payload and are forwarded transparently.

> **NOTE:** In a one or two tagged packet, the tags, tag 1 and tag 2 can carry any TPID values such as 0x8100, 0x88a8, 0x9100, and 0x9200.

Before you begin setting up Q-in-Q tunneling, make sure you have created and configured the necessary customer VLANs on the neighboring switches. See "Configuring VLANs on Switches" on page 182.

To configure Q-in-Q tunneling:

1. Create the service VLAN (S-VLAN) and configure an ID for it:

   [edit vlans]

   ```
   user@switch# set s-vlan-name vlan-ids-vlan-ID
   ```

2. Enable Q-in-Q tunneling on the S-VLAN:

   [edit vlans]

   ```
   user@switch# set s-vlan-name dot1q-tunneling
   ```

3. Set the allowed customer VLANs (C-VLANs) on the S-VLAN (optional). Here, the C-VLANs are identified by a range:

   [edit vlans]

   ```
   user@switch# set s-vlan-name dot1q-tunneling customer-vlans range
   ```

4. Configure a global value for the tag protocol identifier (EtherType) of the service VLAN tags (optional):

   [edit]

```
user@switch# set ethernet-switching-options dot1q-tunneling ether-type ether-type-value
```

Depending on your interface configuration, you might need to adjust the MTU value on your trunk or access ports to accommodate the 4 bytes used for the tag added by Q-in-Q tunneling. For example, if you use the default MTU value of 1514 bytes on your access and trunk ports, you need to make one of the following adjustments:

- Reduce the MTU on the access links by at least 4 bytes so that the frames do not exceed the MTU of the trunk link when S-VLAN tags are added.
- Increase the MTU on the trunk link so that the link can handle the larger frame size.

Configuring Q-in-Q Tunneling on EX Series Switches with ELS Support

> **NOTE:** This task uses Junos OS for EX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does not support ELS, see "Configuring Q-in-Q Tunneling on EX Series Switches" on page 910. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50.

Q-in-Q tunneling enables service providers on Ethernet access networks to segregate or bundle customer traffic into different VLANs by adding another layer of 802.1Q tags. You can configure Q-in-Q tunneling on EX Series switches.

> **NOTE:** You cannot configure 802.1X user authentication on interfaces that have been enabled for Q-in-Q tunneling.

When Q-in-Q tunneling is configured on EX Series switches, trunk interfaces are assumed to be part of the service-provider network and access interfaces are assumed to be part of the customer network. Therefore, this topic also refers to trunk interfaces as service-provider VLAN (S-VLAN) interfaces (network-to-network interfaces [NNI]), and to access interfaces as customer VLAN (C-VLAN) interfaces (user-network interfaces [UNI]).

Before you begin configuring Q-in-Q tunneling, make sure you set up your VLANs. See *Configuring VLANs for EX Series Switches with ELS Support (CLI Procedure)* or *Configuring VLANs for EX Series Switches (J-Web Procedure)*.

Configure Q-in-Q tunneling by using one of the following methods to map C-VLANs to S-VLANs:

Configuring All-in-One Bundling

You can configure Q-in-Q tunneling by using the all-in-one bundling method, which maps packets from all C-VLAN interfaces on a switch to an S-VLAN.

To configure the all-in-one bundling method on a C-VLAN interface:

1. Enable the transmission of packets with no or a single 802.1Q VLAN tag:

   [edit interfaces *interface-name*]
   `user@switch#` **set flexible-vlan-tagging**

2. Enable extended VLAN bridge encapsulation:

   [edit interfaces *interface-name*]
   `user@switch#` **set encapsulation extended-vlan-bridge**

3. Map packets from all C-VLANs to a logical interface:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@switch#` **set vlan-id-list *vlan-id-numbers***

   NOTE: You can apply no more than eight VLAN identifier lists to a physical interface.

4. Enable a C-VLAN interface to send and receive untagged packets:

   [edit interfaces *interface-name*]
   `user@switch#` **set native-vlan-id *vlan-id***

   When specifying a native VLAN ID on a C-VLAN physical interface, the value must be included in the VLAN ID list specified on the C-VLAN logical interface in step 3.

5. Specify that packets traveling from a C-VLAN interface to an S-VLAN interface are tagged with the VLAN ID of the S-VLAN:

   [edit interfaces *interface-name* unit *logical-unit-number*]

   `user@switch#` **set input-vlan-map push**

6. Specify that the 802.1Q S-VLAN tag is removed as packets exit an S-VLAN interface.

   [edit interfaces *interface-name* unit *logical-unit-number*]

   `user@switch#` **set output-vlan-map pop**

7. Configure a name for the S-VLAN, and associate the logical interface configured in step 3 with the S-VLAN:

   [edit vlans *vlan-name*]

   `user@switch#` **set interface *interface-name.logical-unit-number***

The following configuration on the C-VLAN interface ge-0/0/1 enables Q-in-Q tunneling and maps packets from C-VLANs 100 through 200 to logical interface 10, which is in turn associated with S-VLAN v10. In this sample configuration, a packet originated in C-VLAN 100 includes a tag with the VLAN ID 100. When this packet travels from the interface ge-0/0/1 to the S-VLAN interface, a tag with VLAN ID 10 is added to it. As the packet exits the S-VLAN interface, the tag with the VLAN ID 10 is removed.

set interfaces ge-0/0/1 flexible-vlan-tagging

set interfaces ge-0/0/1 encapsulation extended-vlan-bridge

set interfaces ge-0/0/1 unit 10 vlan-id-list 100-200

set interfaces ge-0/0/1 native-vlan-id 150

set interfaces ge-0/0/1 unit 10 input-vlan-map push

set interfaces ge-0/0/1 unit 10 output-vlan-map pop

set vlans v10 interface ge-0/0/1.10

To configure the all-in-one bundling method on an S-VLAN interface:

1. Enable the transmission of packets with no, one, or two 802.1Q VLAN tags:

   [edit interfaces *interface-name*]

   `user@switch#` **set flexible-vlan-tagging**

2. Enable extended VLAN bridge encapsulation:

[edit interfaces *interface-name*]

`user@switch#` **set encapsulation extended-vlan-bridge**

3. Map packets from the logical interface specified in the C-VLAN interface configuration to the S-VLAN:

   [edit interfaces *interface-name* unit *logical-unit-number*]

   `user@switch#` **set vlan-id *number***

4. Enable the S-VLAN interface to send and receive untagged packets:

   [edit interfaces *interface-name*]

   `user@switch#` **set native-vlan-id *vlan-id***

   When specifying a native VLAN ID on an S-VLAN physical interface, the value must match the VLAN ID specified on the S-VLAN logical interface in step 3.

5. Associate the S-VLAN interface with the S-VLAN that was configured in the C-VLAN interface procedure:

   [edit vlans *vlan-name*]

   `user@switch#` **set interface *interface-name.logical-unit-number***

For example, the following configuration on the S-VLAN interface ge-1/1/1 enables Q-in-Q tunneling and maps packets with a VLAN ID tag of 10 to logical interface 10, which is in turn associated with S-VLAN v10. .

set interfaces ge-1/1/1 flexible-vlan-tagging

set interfaces ge-1/1/1 encapsulation extended-vlan-bridge

set interfaces ge-1/1/1 unit 10 vlan-id 10

set interfaces ge-1/1/1 native-vlan-id 10

set vlans v10 interface ge-1/1/1.10

Configuring Many-to-Many Bundling

You can configure Q-in-Q tunneling by using the many-to-many bundling method, which maps packets from multiple C-VLANs to multiple S-VLANs.

To configure the many-to-many bundling method on a C-VLAN interface:

1. Enable the transmission of packets with no or a single 802.1Q VLAN tag:

   [edit interfaces *interface-name*]

   `user@switch#` **set flexible-vlan-tagging**

2. Enable extended VLAN bridge encapsulation:

   [edit interfaces *interface-name*]

   `user@switch#` **set encapsulation extended-vlan-bridge**

3. Map packets from specified C-VLANs to a logical interface:

   [edit interfaces *interface-name* unit *logical-unit-number*]

   `user@switch#` **set vlan-id-list *vlan-id-numbers***

4. Enable a C-VLAN interface to send and receive untagged packets:

   [edit interfaces *interface-name*]

   `user@switch#` **set native-vlan-id *vlan-id***

   When specifying a native VLAN ID on a C-VLAN physical interface, the value must be included in the VLAN ID list specified on the C-VLAN logical interface in step 3.

5. Specify that packets traveling from a C-VLAN interface to an S-VLAN interface are tagged with the VLAN ID of the S-VLAN:

   [edit interfaces *interface-name* unit *logical-unit-number*]

   `user@switch#` **set input-vlan-map push**

6. Specify that the 802.1Q S-VLAN tag is removed as packets exit an S-VLAN interface:

   [edit interfaces *interface-name* unit *logical-unit-number*]

   `user@switch#` **set output-vlan-map pop**

7. Configure a name for an S-VLAN, and associate the logical interface configured in step 3 with the S-VLAN:

   [edit vlans *vlan-name*]

   `user@switch#` **set interface *interface-name.logical-unit-number***

The following configuration on the C-VLAN interface ge-0/0/1 for customer 1 enables Q-in-Q tunneling and maps packets from C-VLANs 100 through 120 to logical interface 10, which is in turn associated with S-VLAN v10.

The configuration on the C-VLAN interface ge-0/0/2 for customer 2 enables Q-in-Q tunneling and maps packets from C- VLANs 30 through 40, 50 through 60, and 70 through 80 to logical interface 30, which is in turn associated with S- VLAN v30.

In this sample configuration, a packet originated in C-VLAN 100 includes a tag with the VLAN ID 100. When this packet travels from the interface ge-0/0/1 to the S-VLAN interface, a tag with a VLAN ID of 10 is added to it. As the packet exits the S-VLAN interface, the tag with the VLAN ID of 10 is removed.

Customer 1

set interfaces ge-0/0/1 flexible-vlan-tagging

set interfaces ge-0/0/1 encapsulation extended-vlan-bridge

set interfaces ge-0/0/1 unit 10 vlan-id-list 100-120

set interfaces ge-0/0/1 native-vlan-id 100

set interfaces ge-0/0/1 unit 10 input-vlan-map push

set interfaces ge-0/0/1 unit 10 output-vlan-map pop

set vlans v10 interface ge-0/0/1.10

Customer 2

set interfaces ge-0/0/2 flexible-vlan-tagging

set interfaces ge-0/0/2 encapsulation extended-vlan-bridge

set interfaces ge-0/0/2 unit 30 vlan-id-list 30-40

set interfaces ge-0/0/2 unit 30 vlan-id-list 50-60

set interfaces ge-0/0/2 unit 30 vlan-id-list 70-80

set interfaces ge-0/0/2 native-vlan-id 30

set interfaces ge-0/0/2 unit 30 input-vlan-map push

set interfaces ge-0/0/2 unit 30 output-vlan-map pop

set vlans v30 interface ge-0/0/2.30

To configure the many-to-many bundling method on an S-VLAN interface:

1. Enable the transmission of packets with no, one, or two 802.1Q VLAN tags:

   [edit interfaces *interface-name*]
   `user@switch#` **set flexible-vlan-tagging**

2. Enable extended VLAN bridge encapsulation:

   [edit interfaces *interface-name*]
   `user@switch#` **set encapsulation extended-vlan-bridge**

3. Map packets from each logical interface specified in the C-VLAN interface configuration to an S-VLAN:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@switch#` **set native-vlan-id *number***

4. Enable an S-VLAN interface to send and receive untagged packets:

   [edit interfaces *interface-name*]
   `user@switch#` **set native-vlan-id *vlan-id***

   When specifying a native VLAN ID on an S-VLAN physical interface, the value must match an S-VLAN ID specified on the S-VLAN logical interface in step 3.

5. Associate the S-VLAN interface with the S-VLANs that were configured in the C-VLAN interface procedure:

   [edit vlans *vlan-name*]
   `user@switch#` **set interface *interface-name.logical-unit-number***

For example, the following configuration on the S-VLAN interface ge-1/1/1 enables Q-in-Q tunneling and maps incoming C-VLAN packets to logical interfaces 10 and 30, which are in turn associated with S-VLANs v10 and v30, respectively.

set interfaces ge-1/1/1 flexible-vlan-tagging

set interfaces ge-1/1/1 encapsulation extended-vlan-bridge

set interfaces ge-1/1/1 unit 10 vlan-id 10

set interfaces ge-1/1/1 unit 30 vlan-id 30

set interfaces ge-1/1/1 native-vlan-id 10

set vlans v10 interface ge-1/1/1.10

set vlans v30 interface ge-1/1/1.30

Configuring a Specific Interface Mapping with VLAN Rewrite Option

You can configure Q-in-Q tunneling by mapping packets from a specified C-VLAN to a specified S-VLAN. In addition, while the packets are transmitted to and from the S-VLAN, you can specify that the 802.1Q C-VLAN tag be removed and replaced with the S-VLAN tag or vice versa.

To configure a specific interface mapping with VLAN rewriting on the C-VLAN interface:

1. Enable the transmission of packets with no or one 802.1Q VLAN tag:

   [edit interfaces *interface-name*]
   `user@switch#` **set flexible-vlan-tagging**

2. Enable extended VLAN bridge encapsulation:

   [edit interfaces *interface-name*]
   `user@switch#` **set encapsulation extended-vlan-bridge**

3. Map packets from a specified C-VLAN to a logical interface:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@switch#` **set vlan-id *number***

4. Enable the C-VLAN interface to send and receive untagged packets:

   [edit interfaces *interface-name*]
   `user@switch#` **set native-vlan-id *vlan-id***

   When specifying a native VLAN ID on a C-VLAN physical interface, the value must match the VLAN ID specified on the C-VLAN logical interface in step 3.

5. Specify that the existing 802.1Q C-VLAN tag is removed from packets traveling from a C-VLAN interface to an S-VLAN interface and replaced with the 802.1Q S-VLAN tag:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@switch#` **set input-vlan-map swap**

6. Specify that the existing 802.1Q S-VLAN tag is removed from packets traveling from an S-VLAN interface to a C-VLAN interface and replaced with the 802.1Q C-VLAN tag:

   [edit interfaces *interface-name* unit *logical-unit-number*]

```
user@switch# set output-vlan-map swap
```

7. Configure a name for the S-VLAN, and associate the logical interface configured in step 3 with the S-VLAN:

   [edit vlans *vlan-name*]

   ```
   user@switch# set interface interface-name.logical-unit-number
   ```

For example, the following configuration on the C-VLAN interface ge-0/0/1 enables Q-in-Q tunneling and maps incoming packets from C-VLAN 150 to logical interface 200, which is in turn associated with VLAN v200. Also, as packets travel from the C-VLAN interface ge-0/0/1 to an S-VLAN interface, the C-VLAN tag 150 is removed and replaced with the S-VLAN tag 200. As packets travel from an S-VLAN interface to C-VLAN interface ge-0/0/1, the S-VLAN tag 200 is removed and replaced with the C-VLAN tag of 150.

```
set interfaces ge-0/0/1 flexible-vlan-tagging

set interfaces ge-0/0/1 encapsulation extended-vlan-bridge

set interfaces ge-0/0/1 unit 200 vlan-id 150

set interfaces ge-0/0/1 native-vlan-id 150

set interfaces ge-0/0/1 unit 200 input-vlan-map swap

set interfaces ge-0/0/1 unit 200 output-vlan-map swap

set vlans v200 interface ge-0/0/1.200
```

To configure a specific interface mapping with VLAN rewriting on the S-VLAN interface:

1. Enable the transmission of packets with no, one, or two 802.1Q VLAN tags:

   [edit interfaces *interface-name*]

   ```
   user@switch# set flexible-vlan-tagging
   ```

2. Enable extended VLAN bridge encapsulation:

   [edit interfaces *interface-name*]

   ```
   user@switch# set encapsulation extended-vlan-bridge
   ```

3. Map packets from the logical interface specified in the C-VLAN interface configuration to the S-VLAN:

   [edit interfaces *interface-name* unit *logical-unit-number*]

   ```
   user@switch# set vlan-id number
   ```

4. Enable the S-VLAN interface to send and receive untagged packets:

[edit interfaces *interface-name*]

`user@switch#` **set native-vlan-id *vlan-id***

When specifying a native VLAN ID on an S-VLAN physical interface, the value must match the VLAN ID specified on the S-VLAN logical interface in step 3.

5. Associate the S-VLAN interface with the S-VLAN that was configured in the C-VLAN interface procedure:
   :

   [edit vlans *vlan-name*]

   `user@switch#` **set interface *interface-name.logical-unit-number***

For example, the following configuration on the S-VLAN interface ge-1/1/1 enables Q-in-Q tunneling and maps packets with VLAN ID 200 to logical interface 200, which is in turn associated with S-VLAN v200.

set interfaces ge-1/1/1 flexible-vlan-tagging

set interfaces ge-1/1/1 encapsulation extended-vlan-bridge

set interfaces ge-1/1/1 unit 200 vlan-id 200

set interfaces ge-1/1/1 native-vlan-id 200

set vlans v200 interface ge-1/1/1.200

Configuring Q-in-Q Tunneling on EX Series Switches

> **NOTE:** This task uses Junos OS for EX Series switches that does not support the Enhanced Layer 2 Software (ELS) configuration style.

Q-in-Q tunneling allows service providers on Ethernet access networks to segregate or bundle customer traffic into different VLANs by adding another layer of 802.1Q tags. You can configure Q-in-Q tunneling on EX Series switches.

> **NOTE:** You cannot configure 802.1X user authentication on interfaces that have been enabled for Q-in-Q tunneling.

Before you begin configuring Q-in-Q tunneling, make sure you set up your VLANs. See "Configuring VLANs for EX Series Switches" on page 183 or *Configuring VLANs for EX Series Switches (J-Web Procedure)*.

To configure Q-in-Q tunneling:

1. Enable Q-in-Q tunneling on the S-VLAN:

   [edit vlans]

   ```
   user@switch# set s-vlan-name dot1q-tunneling
   ```

2. Set the allowed C-VLANs on the S-VLAN (optional). Here, the C-VLANs are identified by VLAN range:

   [edit vlans]

   ```
   user@switch# set s-vlan-name dot1q-tunneling customer-vlans range
   ```

3. Change the global Ethertype value (optional):

   [edit]

   ```
   user@switch# set ethernet-switching-options dot1q-tunneling ether-type ether-type-value
   ```

4. Disable MAC address learning on the S-VLAN (optional):

   [edit vlans]

   ```
   user@switch# set s-vlan-name no-mac-learning
   ```

Configuring Q-in-Q Tunneling Using All-in-One Bundling

You can configure Q-in-Q tunneling using the all-in-one bundling method, which forwards all packets that ingress on a C-VLAN interface to an S-VLAN. (Packets are forwarded to the S-VLAN regardless of whether they are tagged or untagged prior to ingress.) Using this approach saves you the effort of specifying a specific mapping for each C-VLAN.

First configure the S-VLAN and its interface:

1. Assign a logical interface (unit) to be a member of the S-VLAN.

   [edit vlans *vlan-name*]
   `user@switch#` **interface *interface-name.unit-number***

   > **NOTE:** Do not use logical interface unit 0. You must later bind a VLAN tag ID to the unit you specify in this step, and you cannot bind a VLAN tag ID to unit 0. Also note that you do not create a VLAN ID for the S-VLAN. The ID is created automatically for the appropriate logical interface.

2. Enable the interface to transmit packets with two 802.1Q VLAN tags:

   [edit interfaces *interface-name*]
   `user@switch#` **set flexible-vlan-tagging**

3. Enable extended VLAN bridge encapsulation on the interface:

   [edit interfaces *interface-name*]
   `user@switch#` **set encapsulation extended-vlan-bridge**

   > **NOTE:** If you configure an enterprise-style configuration such as PVLAN on the same physical interface on which you are configuring Q-in-Q tunneling, use **set encapsulation flexible-ethernet-services** . See *Understanding Flexible Ethernet Services Encapsulation on Switches*.

4. Enable the S-VLAN interface to send and receive untagged packets:

   [edit interfaces *interface-name*]

```
user@switch# set native-vlan-id vlan-id
```

5. Bind the logical interface (unit) of the interface that you specified in step 1 to the automatically created VLAN ID for the S-VLAN:

   [edit interfaces *interface-name* unit *logical-unit-number*]

   ```
   user@switch# set vlan-id number
   ```

> **NOTE:** If you configured **flexible-ethernet-services**, configure **vlan-bridge** encapsulation on the logical interface:
>
> **[edit interfaces *interface-name* unit *logical-unit-number*]**
>
> ```
> user@switch# set encapsulation vlan-bridge
> ```

For example, the following configuration makes xe-0/0/0.10 a member of VLAN 10, enables Q-in-Q tunneling on interface xe-0/0/0, enables xe-0/0/0 to accept untagged packets, and binds the VLAN ID of S-VLAN v10 to a logical interface of xe-0/0/0.

set vlans v10 interface xe-0/0/0.10

set interfaces xe-0/0/0 flexible-vlan-tagging

set interfaces xe-0/0/0 native-vlan-id 10

set interfaces xe-0/0/0 encapsulation extended-vlan-bridge

set interfaces xe-0/0/0 unit 10 vlan-id 10

Now configure all-in-one bundling on a C-VLAN interface:

1. Assign a logical interface (unit) of the C-VLAN interface to be a member of the S-VLAN.

   [edit vlans *vlan-name*]

   ```
   user@switch# set interface interface-name.unit-number
   ```

2. Enable the interface to transmit packets with 802.1Q VLAN tags :

   [edit interfaces *interface-name*]

   ```
   user@switch# set flexible-vlan-tagging
   ```

3. Enable extended VLAN bridge encapsulation on the interface:

   [edit interfaces *interface-name*]

```
user@switch# set encapsulation extended-vlan-bridge
```

4. Enable the C-VLAN interface to send and receive untagged packets:

   [edit interfaces *interface-name*]

   ```
   user@switch# set native-vlan-id vlan-id
   ```

5. Configure a logical interface to receive and forward any tagged packet whose VLAN ID tag matches the list of VLAN IDs you specify:

   [edit interfaces *interface-name* unit *logical-unit-number*]

   ```
   user@switch# set vlan-id-list vlan-id-numbers
   ```

   > **CAUTION:** You can apply no more than eight VLAN identifier lists to a physical interface. This limitation does not apply to QFX10000 switches.

6. Configure the system to add an S-VLAN tag (outer tag) as packets travel from a C-VLAN interface to the S-VLAN:

   [edit interfaces *interface-name* unit *logical-unit-number*]

   ```
   user@switch# set input-vlan-map push
   ```

   > **NOTE:** You can configure **vlan-id** on **input-vlan-map**, but doing so is optional.

7. Configure the system to remove the S-VLAN tag when packets are forwarded (internally) from the S-VLAN interface to the C-VLAN interface:

   [edit interfaces *interface-name* unit *logical-unit-number*]

   ```
   user@switch# set output-vlan-map pop
   ```

For example, the following configuration makes xe-0/0/1.10 a member of S-VLAN v10, enables Q-in-Q tunneling, maps packets from C-VLANs 100 through 200 to S-VLAN 10, and enables xe-0/0/1 to accept untagged packets. If a packet originates in C-VLAN 100 and needs to be sent across the S-VLAN, a tag with VLAN ID 10 is added to the packet. When a packet is forwarded (internally) from the S-VLAN interface to interface xe-0/0/1, the tag with VLAN ID 10 is removed.

set vlans v10 interface xe-0/0/1.10

set interfaces xe-0/0/1 flexible-vlan-tagging

set interfaces xe-0/0/1 encapsulation extended-vlan-bridge

set interfaces xe-0/0/1 unit 10 vlan-id-list 100-200

set interfaces xe-0/0/1 native-vlan-id 150

set interfaces xe-0/0/1 unit 10 input-vlan-map push

set interfaces xe-0/0/1 unit 10 output-vlan-map pop

Configuring Q-in-Q Tunneling Using Many-to-Many Bundling

You can configure Q-in-Q tunneling using the many-to-many bundling method, which maps packets from multiple C-VLANs to multiple S-VLANs. This method is convenient for mapping a range of C-VLANs without having to specify each one individually. (You can also use this method to configure only one C-VLAN to be mapped to an S-VLAN.)

First configure the S-VLANs and assign them to an interface:

1. Assign a logical interface (unit) to be a member of one of the S-VLANs. Do not use logical interface unit 0.

   [edit vlans *vlan-name*]
   `user@switch#` **set interface *interface-name.unit-number***

   > **NOTE:** Note that you do not create a VLAN ID for the S-VLAN. The ID is created automatically for the appropriate logical interface.

2. Repeat step 1 for the other S-VLANs.

3. Enable the physical interface to transmit packets with two 802.1Q VLAN tags:

   [edit interfaces *interface-name*]
   `user@switch#` **set flexible-vlan-tagging**

4. Enable extended VLAN bridge encapsulation on the interface:

   [edit interfaces *interface-name*]
   `user@switch#` **set encapsulation extended-vlan-bridge**

5. Enable the S-VLAN interface to send and receive untagged packets:

   [edit interfaces *interface-name*]
   `user@switch#` **set native-vlan-id *vlan-id***

6. Bind one of the logical units of the interface to the VLAN ID for one of the S-VLANs.

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@switch#` **set vlan-id *number***

7. Repeat step 6 to bind the automatically-created VLAN IDs for the other S-VLANs to the other logical units of the interface:

For example, the following configuration creates S-VLANs v10 and v30 and associates them with interface xe-0/0/0.10, enables Q-in-Q tunneling, enables xe-0/0/0 to accept untagged packets, and maps incoming C-VLAN packets to S-VLANs v10 and v30.

set vlans v10 interface xe-0/0/0.10

set vlans v30 interface xe-0/0/0.10

set interfaces xe-0/0/0 flexible-vlan-tagging

set interfaces xe-0/0/0 native-vlan-id 10

set interfaces xe-0/0/0 encapsulation extended-vlan-bridge

set interfaces xe-0/0/0 unit 10 vlan-id 10

set interfaces xe-0/0/0 unit 30 vlan-id 30

To configure the many-to-many bundling method on a C-VLAN interface, perform the following steps for each customer:

1. Assign a logical interface (unit) of one C-VLAN interface to be a member of one S-VLAN.

   [edit vlans *vlan-name*]
   `user@switch#` **set interface *interface-name.unit-number***

2. Repeat step 1 to assign another C-VLAN interface (physical interface) to be a member of another S-VLAN.

3. Enable the interface to transmit packets with 802.1Q VLAN tags:

   [edit interfaces *interface-name*]

```
user@switch# set flexible-vlan-tagging
```

4. Enable extended VLAN bridge encapsulation on the interface:

   [edit interfaces *interface-name*]

   ```
   user@switch# encapsulation extended-vlan-bridge
   ```

5. Enable the C-VLAN interface to send and receive untagged packets:

   [edit interfaces *interface-name*]

   ```
   user@switch# set native-vlan-id vlan-id
   ```

6. For each physical interface, configure a logical interface (unit) to receive and forward any tagged packet whose VLAN ID tag matches the list of VLAN IDs you specify:

   [edit interfaces *interface-name* unit *logical-unit-number*]

   ```
   user@switch# set vlan-id-list vlan-id-numbers
   ```

   To configure only one C-VLAN to be mapped to an S-VLAN, specify only one VLAN ID after *vlan-id-list*.

   > **CAUTION:** You can apply no more than eight VLAN identifier lists to a physical interface. This limitation does not apply to QFX10000 switches.

7. For each physical interface, configure the system to add an S-VLAN tag (outer tag) as packets travel from the C-VLAN interface to the S-VLAN:

   [edit interfaces *interface-name* unit *logical-unit-number*]

   ```
   user@switch# set input-vlan-map push
   ```

8. For each physical interface, configure the system to remove the S-VLAN tag when packets are forwarded from the S-VLAN interface to the C-VLAN interface:

   [edit interfaces *interface-name* unit *logical-unit-number*]

   ```
   user@switch# set output-vlan-map pop
   ```

For example, the following configuration makes xe-0/0/1.10 a member of S-VLAN v10, enables Q-in-Q tunneling, and maps packets from C-VLANs 10 through 20 to S-VLAN 10. The configuration for customer 2 makes xe-0/0/2.30 a member of S-VLAN v30, enables Q-in-Q tunneling, and maps packets from C-VLANs 30 through 40, 50 through 60, and 70 through 80 to S-VLAN 30. Both interfaces are configured to accept untagged packets.

If a packet originates in C-VLAN 10 and needs to be sent over the S-VLAN, a tag with a VLAN ID 10 is added to the packet. If a packet is forwarded internally from the S-VLAN interface to xe-0/0/1.10, the tag with VLAN ID 10 is removed. The same principles apply to the C-VLANs configured on interface xe-0/0/2.

> **NOTE:** Notice that you can use the same tag value for an S-VLAN and C-VLAN. For example, the configuration for customer 1 maps C-VLAN ID 10 to S-VLAN ID 10. C-VLAN and S-VLAN tags use separate name spaces, so this configuration is allowed.

Configuration for customer 1:

set vlans v10 interface xe-0/0/1.10

set interfaces xe-0/0/1 flexible-vlan-tagging

set interfaces xe-0/0/1 encapsulation extended-vlan-bridge

set interfaces xe-0/0/1 unit 10 vlan-id-list 10-20

set interfaces xe-0/0/1 native-vlan-id 15

set interfaces xe-0/0/1 unit 10 input-vlan-map push

set interfaces xe-0/0/1 unit 10 output-vlan-map pop

Configuration for customer 2:

set vlans v30 interface xe-0/0/2.30

set interfaces xe-0/0/2 flexible-vlan-tagging

set interfaces xe-0/0/2 encapsulation extended-vlan-bridge

set interfaces xe-0/0/2 unit 30 vlan-id-list 30-40

set interfaces xe-0/0/2 unit 30 vlan-id-list 50-60

set interfaces xe-0/0/2 unit 30 vlan-id-list 70-80

set interfaces xe-0/0/2 native-vlan-id 75

set interfaces xe-0/0/2 unit 30 input-vlan-map push

set interfaces xe-0/0/2 unit 30 output-vlan-map pop

Configuring a Specific Interface Mapping with VLAN ID Translation Option

You can configure Q-in-Q tunneling by mapping packets from a specified C-VLAN to a specified S-VLAN. In addition, you can configure the system to replace a C-VLAN tag with an S-VLAN tag or replace an S-VLAN tag with a C-VLAN tag (instead of double tagging). This is call VLAN translation or VLAN rewriting. VLAN translation is particularly useful if a service provider's Layer 2 network that connects a customer's sites does not support double tagged packets.

When you use VLAN translation, both ends of the link normally must be able to swap the tags appropriately. That is, both ends of the link must be configured to swap the C-VLAN tag for the S-VLAN tag and swap the S-VLAN tag for the C-VLAN tag so that traffic in both directions is tagged appropriately while in transit and after arrival.

First configure the S-VLAN and its interface:

1. Assign a logical interface to be a member of the S-VLAN. Do not use unit 0.

   [edit vlans *vlan-name*]
   `user@switch#` **set interface *interface-name.unit-number***

   > **NOTE:** Note that you do not create a VLAN ID for the S-VLAN. The ID is created automatically for the appropriate logical interface.

2. Enable the interface to transmit packets with 802.1Q VLAN tags:

   [edit interfaces *interface-name*]
   `user@switch#` **set flexible-vlan-tagging**

3. Enable the S-VLAN interface to send and receive untagged packets:

   [edit interfaces *interface-name*]
   `user@switch#` **set native-vlan-id *vlan-id***

4. Enable extended VLAN bridge encapsulation on the interface:

   [edit interfaces *interface-name*]
   `user@switch#` **set encapsulation extended-vlan-bridge**

5. Bind the logical interface (unit) of the interface that you specified earlier to the VLAN ID for the S-VLAN:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@switch#` **set vlan-id *number***

For example, the following configuration creates S-VLAN v200, makes xe-0/0/0.200 a member of that VLAN, enables Q-in-Q tunneling on interface xe-0/0/0, enables xe-0/0/0 to accept untagged packets, and binds a logical interface of xe-0/0/0 to the VLAN ID of VLAN v200.

set vlans v200 interface xe-0/0/0.200

set interfaces xe-0/0/0 flexible-vlan-tagging

set interfaces xe-0/0/0 native-vlan-id 150

set interfaces xe-0/0/0 encapsulation extended-vlan-bridge

set interfaces xe-0/0/0 unit 200 vlan-id 200

Now configure a specific interface mapping with optional VLAN ID translation on the C-VLAN interface:

1. Assign a logical interface of the C-VLAN interface to be a member of the S-VLAN.

   [edit vlans *vlan-name*]
   `user@switch#` **set interface *interface-name.unit-number***

2. Enable the interface to transmit packets with 802.1Q VLAN tags:

   [edit interfaces *interface-name*]
   `user@switch#` **set flexible-vlan-tagging**

3. Enable the C-VLAN interface to send and receive untagged packets:

   [edit interfaces *interface-name*]
   `user@switch#` **set native-vlan-id *vlan-id***

4. Enable extended VLAN bridge encapsulation on the interface:

   [edit interfaces *interface-name*]
   `user@switch#` **set encapsulation extended-vlan-bridge**

5. Configure a logical interface (unit) to receive and forward any tagged packet whose VLAN ID tag matches the VLAN IDs you specify:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@switch#` **set vlan-id *number***

6. Configure the system to remove the existing C-VLAN tag and replace it with the S-VLAN tag when packets ingress on the C-VLAN interface and are forwarded to the S-VLAN:

   [edit interfaces *interface-name* unit *logical-unit-number*]

```
user@switch# set input-vlan-map swap
```

7. Configure the system to remove the existing S-VLAN tag and replace it with the C-VLAN tag when packets are forwarded from the S-VLAN interface to the C-VLAN interface:

   [edit interfaces *interface-name* unit *logical-unit-number*]

   ```
   user@switch# set output-vlan-map swap
   ```

8. To configure an S-VLAN and associate it with the appropriate C-VLAN interface:

   [edit vlans *vlan-name*]

   ```
   user@switch# set interface interface-name
   ```

For example, the following configuration on C-VLAN interface xe-0/0/1.200 enables Q-in-Q tunneling, enables xe-0/0/1 to accept untagged packets, and maps incoming packets from C-VLAN 150 to logical interface 200, which is a member of S-VLAN 200. Also, when packets egress from C-VLAN interface xe-0/0/1 and travel to the S-VLAN interface, the C-VLAN tag of 150 is removed and replaced with the S-VLAN tag of 200. When packets travel from the S-VLAN interface to the C-VLAN interface, the S-VLAN tag of 200 is removed and replaced with the C-VLAN tag of 150.

set vlans v200 interface xe-0/0/1.200

set interfaces xe-0/0/1 flexible-vlan-tagging

set interfaces xe-0/0/1 native-vlan-id 150

set interfaces xe-0/0/1 encapsulation extended-vlan-bridge

set interfaces xe-0/0/1 unit 200 vlan-id 200

set interfaces xe-0/0/1 unit 200 output-vlan-map swap

set interfaces xe-0/0/1 unit 200 input-vlan-map swap

Example: Setting Up Q-in-Q Tunneling on QFX Series Switches

IN THIS SECTION

Service providers can use Q-in-Q tunneling to transparently pass Layer 2 VLAN traffic between customer sites without removing or changing the customer VLAN tags or class-of-service (CoS) settings. Data centers can use Q-in-Q tunneling to isolate customer traffic within a single site or when customer traffic flows between cloud data centers in different geographic locations.

NOTE: This example uses a Junos OS release that does *not* support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see *Configuring Q-in-Q Tunneling on QFX Series, NFX Series, and EX4600 Switches with ELS Support*.

This example describes how to set up Q-in-Q tunneling:

Requirements

This example requires one QFX Series device with Junos OS Release 12.1 or later.

Before you begin setting up Q-in-Q tunneling, make sure you have created and configured the necessary customer VLANs on the neighboring switches. See "Configuring VLANs on Switches" on page 182.

Overview and Topology

In this service provider network, there are multiple customer VLANs mapped to one service VLAN.

Table 123 on page 921 lists the settings for the sample topology.

Table 123: Components of the Topology for Setting Up Q-in-Q Tunneling

Interface	Description
xe-0/0/11.0	Tagged S-VLAN trunk port
xe-0/0/12.0	Untagged customer-facing access port
xe-0/0/13.0	Untagged customer-facing access port
xe-0/0/14.0	Tagged S-VLAN trunk port

Configuration

CLI Quick Configuration

To quickly create and configure Q-in-Q tunneling, copy the following commands and paste them into the switch terminal window:

```
[edit]
set vlans service-vlan vlan-id 1000
set vlans service-vlan dot1q-tunneling customer-vlans 1-100
set vlans service-vlan dot1q-tunneling customer-vlans 201-300
set interfaces xe-0/0/11 unit 0 family ethernet-switching port-mode trunk
set interfaces xe-0/0/11 unit 0 family ethernet-switching vlan members 1000
set interfaces xe-0/0/12 unit 0 family ethernet-switching port-mode access
set interfaces xe-0/0/12 unit 0 family ethernet-switching vlan members 1000
set interfaces xe-0/0/13 unit 0 family ethernet-switching port-mode access
set interfaces xe-0/0/13 unit 0 family ethernet-switching vlan members 1000
set interfaces xe-0/0/14 unit 0 family ethernet-switching port-mode trunk
set interfaces xe-0/0/14 unit 0 family ethernet-switching vlan members 1000
set ethernet-switching-options dot1q-tunneling ether-type 0x9100
```

Step-by-Step Procedure

To configure Q-in-Q tunneling:

1. Set the VLAN ID for the S-VLAN:

   [edit vlans]
   `user@switch#` **set service-vlan vlan-id 1000**

2. Enable Q-in-Q tunneling and specify the customer VLAN ranges:

   [edit vlans]
   `user@switch#` **set service-vlan dot1q-tunneling customer-vlans 1-100**
   `user@switch#` **set service-vlan dot1q-tunneling customer-vlans 201-300**

3. Set the port mode and VLAN information for the interfaces:

   [edit interfaces]

   ```
   user@switch# set xe-0/0/11 unit 0 family ethernet-switching port-mode trunk
   user@switch# set xe-0/0/11 unit 0 family ethernet-switching vlan members 1000
   user@switch# set xe-0/0/12 unit 0 family ethernet-switching port-mode access
   user@switch# set xe-0/0/12 unit 0 family ethernet-switching vlan members 1000
   user@switch# set xe-0/0/13 unit 0 family ethernet-switching port-mode access
   user@switch# set xe-0/0/13 unit 0 family ethernet-switching vlan members 1000
   user@switch# set xe-0/0/14 unit 0 family ethernet-switching port-mode trunk
   user@switch# set xe-0/0/14 unit 0 family ethernet-switching vlan members 1000
   ```

4. Set the Q-in-Q Ethertype value (optional):

   [edit]

   ```
   user@switch# set ethernet-switching-options dot1q-tunneling ether-type 0x9100
   ```

Results

Check the results of the configuration:

```
user@switch> show configuration vlans service-vlan
vlan-id 1000 {
    dot1q-tunneling {
    customer-vlans [ 1-100 201-300 ];
}
user@switch> show configuration interfaces
xe-0/0/11 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan members 1000;
        }
    }
}
xe-0/0/12 {
    unit 0 {
        family ethernet-switching {
            port-mode access;
            vlan members 1000;
        }
    }
}
```

```
xe-0/0/13 {
    unit 0 {
        family ethernet-switching {
            port-mode access;
            vlan members 1000;
        }
    }
}
xe-0/0/14 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan members 1000;
        }
    }
}
user@switch> show ethernet-switching-options
dot1q-tunneling {
    ether-type 0x9100;
}
```

Verification

Confirm that the configuration is working properly.

Verifying That Q-in-Q Tunneling Was Enabled

Purpose

Verify that Q-in-Q tunneling was properly enabled.

Action

Use the **show vlans** command:

```
user@switch> show vlans service-vlan extensive
```

```
VLAN: service-vlan, Created at: Wed Mar 14 07:17:53 2012
802.1Q Tag: 1000, Internal index: 18, Admin State: Enabled, Origin: Static
Dot1q Tunneling Status: Enabled
Customer VLAN ranges:
                    1-100
                    201-300
Protocol: Port Mode
Number of interfaces: Tagged 2 (Active = 0), Untagged  2 (Active = 0)
```

```
    xe-0/0/11.0, tagged, trunk
    xe-0/0/14.0, tagged, trunk
    xe-0/0/12.0, untagged, access
    xe-0/0/13.0, untagged, access
```

Meaning

The output indicates that Q-in-Q tunneling is enabled and that the VLAN is tagged and shows the associated customer VLANs.

Example: Setting Up Q-in-Q Tunneling on EX Series Switches

IN THIS SECTION

Service providers can use Q-in-Q tunneling to transparently pass Layer 2 VLAN traffic from a customer site, through the service provider network, to another customer site without removing or changing the customer VLAN tags or class-of-service (CoS) settings. You can configure Q-in-Q tunneling on EX Series switches.

This example describes how to set up Q-in-Q:

Requirements

This example requires one EX Series switch with Junos OS Release 9.3 or later for EX Series switches.

Before you begin setting up Q-in-Q tunneling, make sure you have created and configured the necessary customer VLANs. See "Configuring VLANs for EX Series Switches" on page 183 or *Configuring VLANs for EX Series Switches (J-Web Procedure)*.

Overview and Topology

In this service provider network, there are multiple customer VLANs mapped to one service VLAN.

Table 124 on page 926 lists the settings for the example topology.

Table 124: Components of the Topology for Setting Up Q-in-Q Tunneling

Interface	Description
ge-0/0/11.0	Tagged S-VLAN trunk port
ge-0/0/12.0	Untagged customer-facing access port
ge-0/0/13.0	Untagged customer-facing access port
ge-0/0/14.0	Tagged S-VLAN trunk port

Configuration

CLI Quick Configuration

To quickly create and configure Q-in-Q tunneling, copy the following commands and paste them into the switch terminal window:

```
[edit]
set vlans qinqvlan vlan-id 4001
set vlans qinqvlan dot1q-tunneling customer-vlans 1-100
set vlans qinqvlan dot1q-tunneling customer-vlans 201-300
set interfaces ge-0/0/11 unit 0 family ethernet-switching port-mode trunk
set interfaces ge-0/0/11 unit 0 family ethernet-switching vlan members 4001
set interfaces ge-0/0/12 unit 0 family ethernet-switching port-mode access
set interfaces ge-0/0/12 unit 0 family ethernet-switching vlan members 4001
set interfaces ge-0/0/13 unit 0 family ethernet-switching port-mode access
set interfaces ge-0/0/13 unit 0 family ethernet-switching vlan members 4001
set interfaces ge-0/0/14 unit 0 family ethernet-switching port-mode trunk
set interfaces ge-0/0/14 unit 0 family ethernet-switching vlan members 4001
set ethernet-switching-options dot1q-tunneling ether-type 0x9100
```

Step-by-Step Procedure

To configure Q-in-Q tunneling:

1. Set the VLAN ID for the S-VLAN:

   [edit vlans]

   ```
   user@switch# set qinqvlan vlan-id 4001
   ```

2. Enable Q-in-Q tuennling and specify the customer VLAN ranges:

   [edit vlans]

   ```
   user@switch# set qinqvlan dot1q-tunneling customer-vlans 1-100
   user@switch# set qinqvlan dot1q-tunneling customer-vlans 201-300
   ```

3. Set the port mode and VLAN information for the interfaces:

   [edit interfaces]

   ```
   user@switch# set ge-0/0/11 unit 0 family ethernet-switching port-mode trunk
   user@switch# set ge-0/0/11 unit 0 family ethernet-switching vlan members 4001
   user@switch# set ge-0/0/12 unit 0 family ethernet-switching port-mode access
   user@switch# set ge-0/0/12 unit 0 family ethernet-switching vlan members 4001
   user@switch# set ge-0/0/13 unit 0 family ethernet-switching port-mode access
   user@switch# set ge-0/0/13 unit 0 family ethernet-switching vlan members 4001
   user@switch# set ge-0/0/14 unit 0 family ethernet-switching port-mode trunk
   user@switch# set ge-0/0/14 unit 0 family ethernet-switching vlan members 4001
   ```

4. Set the Q-in-Q Ethertype value:

   [edit]

   ```
   user@switch# set ethernet-switching-options dot1q-tunneling ether-type 0x9100
   ```

Results

Check the results of the configuration:

```
user@switch> show configuration vlans qinqvlan
vlan-id 4001 {
    dot1q-tunneling {
    customer-vlans [ 1-100 201-300 ];
}
user@switch> show configuration interfaces
ge-0/0/11 {
```

```
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan members 4001;
        }
    }
}
ge-0/0/12 {
    unit 0 {
        family ethernet-switching {
            port-mode access;
            vlan members 4001;
        }
    }
}
ge-0/0/13 {
    unit 0 {
        family ethernet-switching {
            port-mode access;
            vlan members 4001;
        }
    }
}
ge-0/0/14 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan members 4001;
        }
    }
}
user@switch> show ethernet-switching-options
dot1q-tunneling {
    ether-type 0x9100;
}
```

Verification

IN THIS SECTION

- Verifying That Q-in-Q Tunneling Was Enabled | **929**

To confirm that the configuration is working properly, perform these tasks:

Verifying That Q-in-Q Tunneling Was Enabled

Purpose

Verify that Q-in-Q tunneling was properly enabled on the switch.

Action

Use the **show vlans** command:

```
user@switch> show vlans qinqvlan extensive
```

```
VLAN: qinqvlan, Created at: Thu Sep 18 07:17:53 2008
802.1Q Tag: 4001, Internal index: 18, Admin State: Enabled, Origin: Static
Dot1q Tunneling Status: Enabled
Customer VLAN ranges:
                    1-100
                    201-300
Protocol: Port Mode
Number of interfaces: Tagged 2 (Active = 0), Untagged  4 (Active = 0)
      ge-0/0/11.0, tagged, trunk
      ge-0/0/14.0, tagged, trunk
      ge-0/0/12.0, untagged, access
      ge-0/0/13.0, untagged, access
```

Meaning

The output indicates that Q-in-Q tunneling is enabled and that the VLAN is tagged and shows the associated customer VLANs.

Setting Up a Dual VLAN Tag Translation Configuration on QFX Switches

Starting with Junos OS Release 14.1X53-D40, you can use the dual VLAN tag translation (also known as dual VLAN tag rewrite) feature to deploy switches in service-provider domains, allowing dual-tagged, single-tagged, and untagged VLAN packets to come into or exit from the switch.

The following example configuration shows use of the swap-swap, pop-swap, and swap-push dual tag operations.

```
[edit]
set interfaces ge-0/0/1 unit 503 description UNI-3
set interfaces ge-0/0/1 unit 503 encapsulation vlan-bridge
```

```
set interfaces ge-0/0/1 unit 503 vlan-tags outer 503
set interfaces ge-0/0/1 unit 503 vlan-tags inner 504
set interfaces ge-0/0/1 unit 503 input-vlan-map swap-swap
set interfaces ge-0/0/1 unit 503 input-vlan-map vlan-id 500
set interfaces ge-0/0/1 unit 503 input-vlan-map inner-vlan-id 514
set interfaces ge-0/0/1 unit 503 output-vlan-map swap-swap
set interfaces ge-0/0/0 description NNI
set interfaces ge-0/0/0 flexible-vlan-tagging
set interfaces ge-0/0/0 encapsulation flexible-ethernet-services
set interfaces ge-0/0/0 unit 500 description "SVLAN500 port"
set interfaces ge-0/0/0 unit 500 encapsulation vlan-bridge
set interfaces ge-0/0/0 unit 500 vlan-id 500
set interfaces ge-0/0/0 unit 600 description "SVLAN600 port"
set interfaces ge-0/0/0 unit 600 encapsulation vlan-bridge
set interfaces ge-0/0/0 unit 600 vlan-id 600
set interfaces ge-0/0/0 unit 700 description "SVLAN700 port"
set interfaces ge-0/0/0 unit 700 encapsulation vlan-bridge
set interfaces ge-0/0/0 unit 700 vlan-id 700
set interfaces ge-0/0/0 unit 0 family ethernet-switching interface-mode trunk
set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members v1000
set interfaces ge-0/0/0 unit 1100 description UNI-SVLAN1100
set interfaces ge-0/0/0 unit 1100 encapsulation vlan-bridge
set interfaces ge-0/0/0 unit 1100 vlan-tags outer 1101
set interfaces ge-0/0/0 unit 1100 vlan-tags inner 1102
set interfaces ge-0/0/0 unit 1100 input-vlan-map swap-swap
set interfaces ge-0/0/0 unit 1100 input-vlan-map vlan-id 1100
set interfaces ge-0/0/0 unit 1100 input-vlan-map inner-vlan-id 2101
set interfaces ge-0/0/0 unit 1100 output-vlan-map swap-swap
```

```
set interfaces ge-0/0/0 unit 1200 description UNI-SVLAN1200
set interfaces ge-0/0/0 unit 1200 encapsulation vlan-bridge
set interfaces ge-0/0/0 unit 1200 vlan-id 1201
set interfaces ge-0/0/0 unit 1200 input-vlan-map swap-push
set interfaces ge-0/0/0 unit 1200 input-vlan-map inner-vlan-id 2200
set interfaces ge-0/0/0 unit 1200 output-vlan-map pop-swap
set interfaces ge-0/0/2 description UNI
set interfaces ge-0/0/2 flexible-vlan-tagging
set interfaces ge-0/0/2 encapsulation flexible-ethernet-services
set interfaces ge-0/0/2 unit 0 family ethernet-switching interface-mode trunk
set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan members v1000
set interfaces ge-0/0/2 unit 603 description UNI-3
set interfaces ge-0/0/2 unit 603 encapsulation vlan-bridge
set interfaces ge-0/0/2 unit 603 vlan-tags outer 603
set interfaces ge-0/0/2 unit 603 vlan-tags inner 604
set interfaces ge-0/0/2 unit 603 input-vlan-map swap-swap
set interfaces ge-0/0/2 unit 603 input-vlan-map vlan-id 600
set interfaces ge-0/0/2 unit 603 input-vlan-map inner-vlan-id 614
set interfaces ge-0/0/2 unit 603 output-vlan-map swap-swap
set interfaces ge-0/0/3 description UNI
set interfaces ge-0/0/3 flexible-vlan-tagging
set interfaces ge-0/0/3 encapsulation flexible-ethernet-services
set interfaces ge-0/0/3 unit 0 family ethernet-switching interface-mode trunk
set interfaces ge-0/0/3 unit 0 family ethernet-switching vlan members v1000
set interfaces ge-0/0/3 unit 703 description UNI-3
set interfaces ge-0/0/3 unit 703 encapsulation vlan-bridge
set interfaces ge-0/0/3 unit 703 vlan-tags outer 703
set interfaces ge-0/0/3 unit 703 vlan-tags inner 704
```

```
set interfaces ge-0/0/3 unit 703 input-vlan-map swap-swap
set interfaces ge-0/0/3 unit 703 input-vlan-map vlan-id 700
set interfaces ge-0/0/3 unit 703 input-vlan-map inner-vlan-id 714
set interfaces ge-0/0/3 unit 703 output-vlan-map swap-swap
set interfaces ge-0/0/3 unit 701 encapsulation vlan-bridge
set interfaces ge-0/0/3 unit 701 vlan-id 701
set interfaces ge-0/0/3 unit 701 input-vlan-map swap-push
set interfaces ge-0/0/3 unit 701 input-vlan-map inner-vlan-id 780
set interfaces ge-0/0/3 unit 701 output-vlan-map pop-swap
set interfaces ge-0/0/3 unit 1100 description SVLAN1100-NNI
set interfaces ge-0/0/3 unit 1100 encapsulation vlan-bridge
set interfaces ge-0/0/3 unit 1100 vlan-id 1100
set interfaces ge-0/0/3 unit 1200 description SVLAN1200-NNI
set interfaces ge-0/0/3 unit 1200 encapsulation vlan-bridge
set interfaces ge-0/0/3 unit 1200 vlan-id 1200
set vlans SVLAN500 interface ge-0/0/0.500
set vlans SVLAN500 interface ge-0/0/1.503
set vlans SVLAN600 interface ge-0/0/0.600
set vlans SVLAN600 interface ge-0/0/2.603
set vlans SVLAN600 interface ge-0/0/3.701
set vlans SVLAN700 interface ge-0/0/0.700
set vlans SVLAN700 interface ge-0/0/3.703
set vlans v1000 vlan-id 1000
set vlans SVLAN1100 interface ge-0/0/0.1100
set vlans SVLAN1100 interface ge-0/0/3.1100
set vlans SVLAN1200 interface ge-0/0/3.1200
set vlans SVLAN1200 interface ge-0/0/0.1200
```

Verifying That Q-in-Q Tunneling Is Working on Switches

Purpose

After creating a Q-in-Q VLAN, verify that it is set up properly.

Action

1. Use the **show configuration vlans** command to determine if you successfully created the primary and secondary VLAN configurations:

   ```
   user@switch> show configuration vlans
   ```

   ```
   svlan {
      vlan-id 300;
      dot1q-tunneling {
         customer-vlans [ 101–200 ];
      }
   }
   ```

2. Use the **show vlans** command to view VLAN information and link status:

   ```
   user@switch> show vlans s-vlan-name extensive
   ```

   ```
   VLAN: svlan, Created at: Thu Oct 23 16:53:20 2008
   802.1Q Tag: 300, Internal index: 2, Admin State: Enabled, Origin: Static
   Dot1q Tunneling Status: Enabled
   Customer VLAN ranges:
                       101–200
   Protocol: Port Mode
   Number of interfaces: Tagged 1 (Active = 0), Untagged  1 (Active = 0)
         xe-0/0/1, tagged, trunk
         xe-0/0/2, untagged, access
   ```

Meaning

The output confirms that Q-in-Q tunnling is enabled and that the VLAN is tagged, and lists the customer VLANs that are associated with the tagged VLAN.

Release History Table

Release	Description
19.4R1	Starting in Junos OS Release 19.4R1, the QFX10000 line of switches support the third and fourth Q-in-Q tags as payload (also known as a pass-through tag) along with the existing two tags (for VLAN matching and operations).
14.1X53-D40	Starting with Junos OS Release 14.1X53-D40, you can use the dual VLAN tag translation (also known as dual VLAN tag rewrite) feature to deploy switches in service-provider domains, allowing dual-tagged, single-tagged, and untagged VLAN packets to come into or exit from the switch.
14.1X53-D30	Starting with Junos OS 14.1X53-D30, you can configure the same interface to be an S-VLAN/NNI interface and a C-VLAN/UNI interface.

30
CHAPTER

Configuring Redundant Trunk Groups

Redundant Trunk Groups

IN THIS SECTION

Understanding Redundant Trunk Links (Legacy RTG Configuration)

In a typical enterprise network composed of distribution and access layers, a redundant trunk link provides a simple solution for network recovery when a trunk port on a switch goes down. In that case, traffic is routed to another trunk port, keeping network convergence time to a minimum.

> **NOTE:** For information on redundant trunk link configurations that include Q-in-Q support and use LAGs with link protection, see *Q-in-Q Support on Redundant Trunk Links Using LAGs with Link Protection*.

To configure a redundant trunk link, create a redundant trunk group. The redundant trunk group is configured on the access switch and contains two links: a primary or active link, and a secondary link. If the active link fails, the secondary link automatically starts forwarding data traffic without waiting for normal spanning-tree protocol convergence.

Data traffic is forwarded only on the active link. Data traffic on the secondary link is dropped and shown as dropped packets when you issue the operational mode command **show interfaces *interface-name* extensive**.

While data traffic is blocked on the secondary link, Layer 2 control traffic is still permitted. For example, an LLDP session can be run between two switches on the secondary link.

Rapid Spanning Tree Protocol (RSTP) is enabled by default on the switches to create a loop-free topology, but an interface is not allowed to be in both a redundant trunk group and in a spanning-tree protocol topology at the same time. You must disable RSTP on an interface if a redundant trunk group is configured on that interface. For example, in Figure 46 on page 938, in addition to disabling RSTP on the Switch 3 interfaces, you must also disable RSTP on the Switch 1 and Switch 2 interfaces connected to Switch 3. Spanning-tree protocols can, however, continue operating on other interfaces on those switches—for example on the link between Switch 1 and Switch 2.

Figure 46 on page 938 shows three switches in a basic topology for redundant trunk links. Switch 1 and Switch 2 make up the distribution layer, and Switch 3 makes up the access layer. Switch 3 is connected to the distribution layer through trunk ports ge-0/0/9.0 (Link 1) and ge-0/0/10.0 (Link 2). Link 1 and Link 2 are in a redundant trunk group called group1. Link 1 is designated as the primary link. Traffic flows between Switch 3 in the access layer and Switch 1 in the distribution layer through Link 1. While Link 1 is active, Link 2 blocks traffic.

Figure 46: Redundant Trunk Group, Link 1 Active

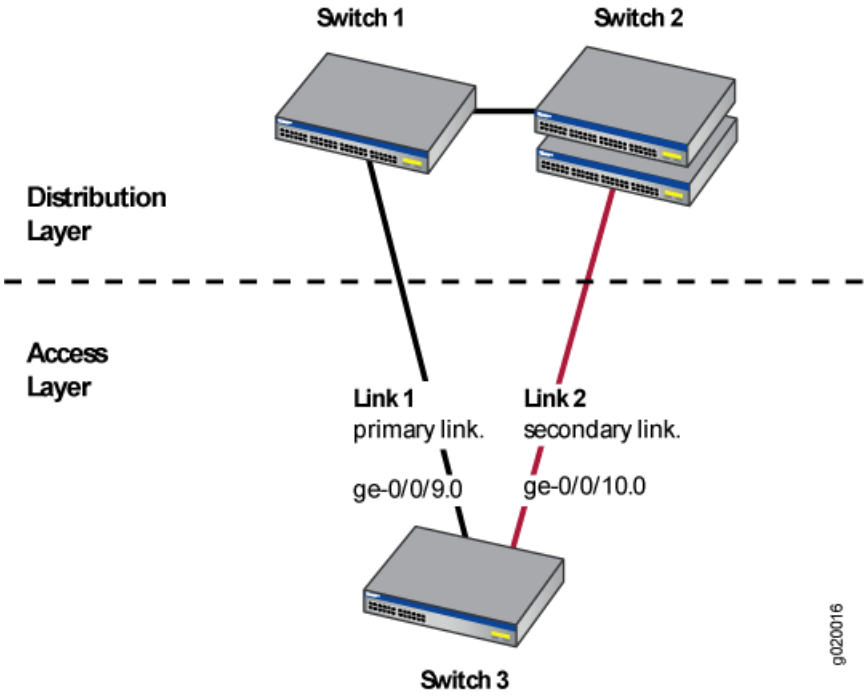
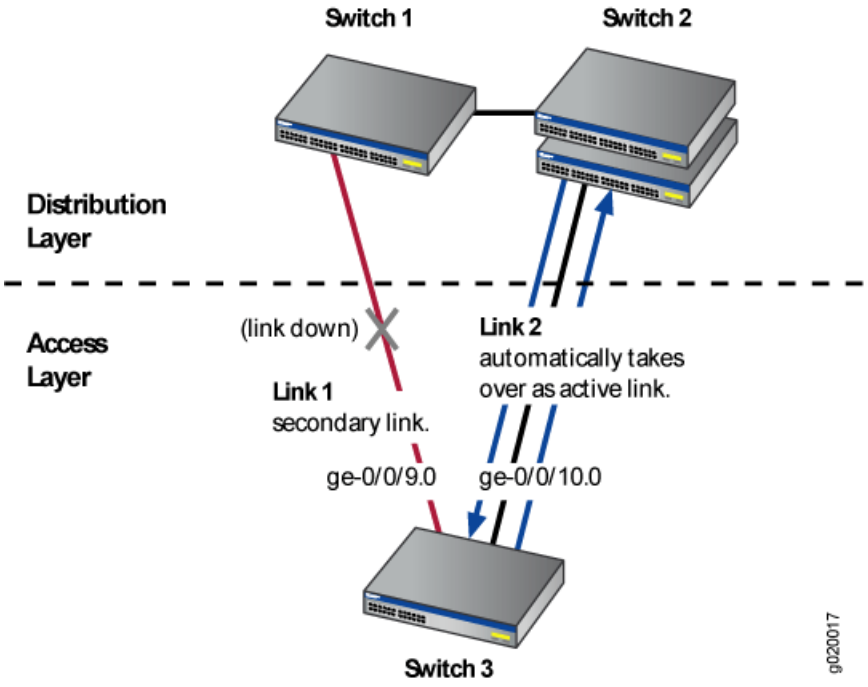


Figure 47 on page 938 illustrates how the redundant trunk link topology works when the primary link goes down.

Figure 47: Redundant Trunk Group, Link 2 Active



When Link 1 between Switch 1 and Switch 3 goes down, Link 2 takes over as the active link. Traffic between the access layer and the distribution layer is then automatically switched to Link 2 between Switch 3 and Switch 2.

Configuring Redundant Trunk Links for Faster Recovery on EX Series Switches

You can manage network convergence by configuring both a primary link and a secondary link on an EX Series switch; this is called a redundant trunk group (RTG). If the primary link in a redundant trunk group fails, it passes its known MAC address locations to the secondary link, which automatically takes over. You can configure a maximum of 16 redundant trunk groups on most standalone switches or on Virtual Chassis. The EX8200 switch and EX8200 Virtual Chassis, however, support up to 254 redundant trunk groups.

Generally, you configure a redundant trunk group by configuring one primary link (and its interface) and one unspecified link (and its interface) to serve as the secondary link. A second type of redundant trunk group, not shown in the procedure in this topic, consists of two unspecified links (and their interfaces); in this case, neither of the links is primary. In this second case, the software selects an active link by comparing the port numbers of the two links and activating the link with the higher port number. The procedure given here describes configuring a primary/unspecified configuration for a redundant trunk group because that configuration gives you more control and is more commonly used.

Rapid Spanning Tree Protocol (RSTP) is enabled by default on EX Series switches to create a loop-free topology, but an interface is not allowed to be in both a redundant trunk group and in a spanning-tree protocol topology at the same time.

A primary link takes over whenever it is able. You can, however, alter the number of seconds that the primary link waits before reestablishing control by configuring the primary link's preempt cutover timer.

Before you configure the redundant trunk group on the switch, be sure you have:

- Disabled RSTP on all switches that will be linked to your redundant trunk group.
- Configured at least two interfaces with their port mode set to **trunk**; be sure that these two interfaces are not part of any existing RTG. See *Configuring Gigabit Ethernet Interfaces (CLI Procedure)* .

To configure a redundant trunk group on a switch:

1. Turn off RSTP:

   ```
   [edit]
   user@switch# set protocols rstp disable
   ```

2. Name the redundant trunk group while configuring one primary and one unspecified trunk interface:

```
[edit ethernet-switching-options]
user@switch# set redundant-trunk-group group name interface interface-name primary
user@switch# set redundant-trunk-group group name interface interface-name
```

3. (Optional) Change the length of time (from the default of 1 second) that a re-enabled primary link waits to take over from an active secondary link:

```
[edit ethernet-switching-options]
set redundant-trunk-group group name preempt-cutover-timer seconds
```

Example: Configuring Redundant Trunk Links for Faster Recovery on Devices with ELS Support

IN THIS SECTION

NOTE: This example uses Junos OS for EX Series switches or QFX Series with support for the Enhanced Layer 2 Software (ELS) configuration style.. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50.

You can manage network convergence by configuring both a primary link and a secondary link on a switch; this is called a redundant trunk group (RTG). If the primary link in a redundant trunk group fails, it passes its known MAC address locations to the secondary link, which automatically takes over after one minute.

This example describes how to create a redundant trunk group with a primary and a secondary link:

Requirements

This example uses the following hardware and software components:

- Two EX Series or QFX Series distribution switches
- One EX Series or QFX Series access switch
- The appropriate software release for your platform:
  - For EX Series switches: Junos OS Release 13.2X50-D10 or later
  - For the QFX Series: Junos OS Release 13.2X50-D15 or later

Before you configure the redundant trunk links network on the access and distribution switches, be sure you have:

- Configured interfaces ge-0/0/9 and ge-0/0/10 on the access switch, Switch 3, as trunk interfaces.
- Configured one trunk interface on each distribution switch, Switch 1 and Switch 2.
- Connected the three switches as shown in the topology for this example (see Figure 48 on page 943).

Overview and Topology

In a typical enterprise network composed of distribution and access layers, a redundant trunk link provides a simple solution for trunk interface network recovery. When a trunk interface fails, data traffic is routed to another trunk interface after one minute, thereby keeping network convergence time to a minimum.

This example shows the configuration of a redundant trunk group that includes one primary link (and its interface) and one unspecified link (and its interface) that serves as the secondary link.

A second type of redundant trunk group, not illustrated in the example, consists of two unspecified links (and their interfaces); in this case, neither of the links is primary. The software selects an active link by comparing the port numbers of the two links and activating the link with the higher port number. For example, if the two link interfaces use interfaces ge-0/1/0 and ge-0/1/1, the software activates ge-0/1/1. (In the interface names, the final number is the port number.)

The two links in a redundant trunk group generally operate the same way, whether they are configured as primary/unspecified or unspecified/unspecified. Data traffic initially passes through the active link but is blocked on the inactive link. While data traffic is blocked on the secondary link, note that Layer 2 control traffic is still permitted if the link is active. For example, an LLDP session can be run between two switches on the secondary link. If the active link either goes down or is disabled administratively, it broadcasts a list of its known MAC addresses for data traffic; the other link immediately picks up and adds the MAC addresses to its address table, becomes active, and begins forwarding traffic.

The one difference in operation between the two types of redundant trunk groups occurs when a primary link is active, goes down, is replaced by the secondary link, and then reactivates. When a primary link is

re-enabled while the secondary link is active, the primary link waits 1 second (you can change the time interval by using the preempt cutover timer to accommodate your network) and then takes over as the active link. In other words, the primary link has priority and is always activated if it is available. This differs from the behavior of two unspecified links, both of which act as equals. Because the unspecified links are equal, the active link remains active until it either goes down or is disabled administratively; this is the only time that the other unspecified link learns the MAC addresses and immediately becomes active.

The example given here illustrates a primary/unspecified configuration for a redundant trunk group because that configuration gives you more control and is more commonly used.

> **NOTE:** Rapid Spanning Tree Protocol (RSTP) is enabled by default on the switches to create a loop-free topology, but an interface is not allowed to be in both a redundant trunk group and in a spanning-tree protocol topology at the same time. You will need to disable RSTP on the two distribution switches in the example, Switch 1 and Switch 2. Spanning-tree protocols can, however, continue operating in other parts of the network—for example, between the distribution switches and also in links between distribution switches and the enterprise core.

Figure 48 on page 943 displays an example topology containing three switches. Switch 1 and Switch 2 make up the distribution layer, and Switch 3 makes up the access layer. Switch 3 is connected to the distribution layer through trunk interfaces ge-0/0/9.0 (Link 1) and ge-0/0/10.0 (Link 2).

Table 125 on page 943 lists the components used in this redundant trunk group.

Because RSTP and RTGs cannot operate simultaneously on a switch, you disable RSTP on Switch 1 and Switch 2 in the first configuration task, and you disable RSTP on Switch 3 in the second task.

The second configuration task creates a redundant trunk group called example 1 on Switch 3. The trunk interfaces ge-0/0/9.0 and ge-0/0/10.0 are the two links configured in the second configuration task. You configure the trunk interface ge-0/0/9.0 as the primary link. You configure the trunk interface ge-0/0/10.0 as an unspecified link, which becomes the secondary link by default.

Figure 48: Topology for Configuring the Redundant Trunk Links

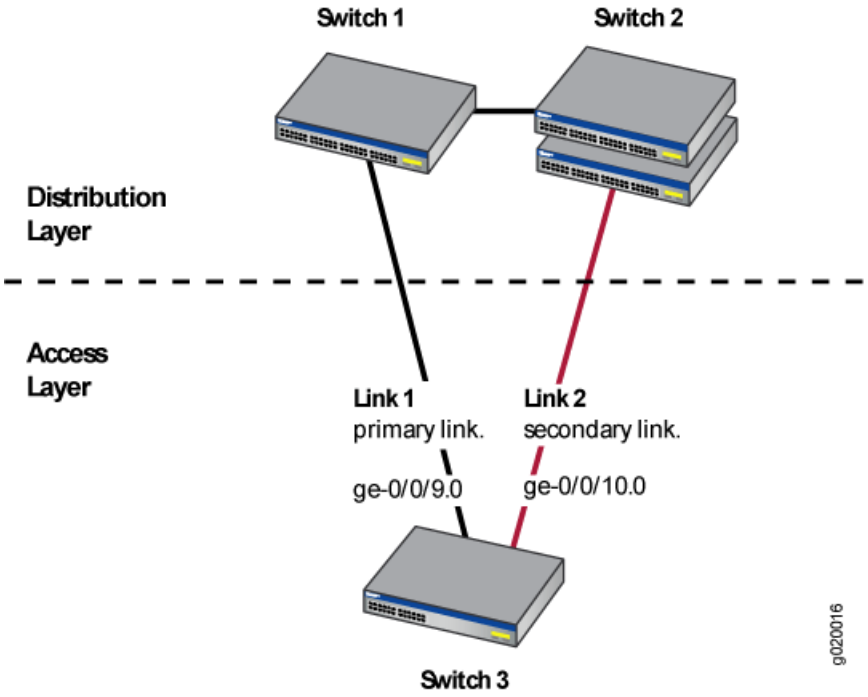


Table 125: Components of the Redundant Trunk Link Topology

Property	Settings
Switch hardware	• Switch 1–1 EX Series or QFX Series distribution switch • Switch 2–1 EX Series or QFX Series distribution switch • Switch 3–1 EX Series or QFX Series access switch
Trunk interfaces	On Switch 3 (access switch): ge-0/0/9.0 and ge-0/0/10.0
Redundant trunk group	rtg0

Disabling RSTP on Switches 1 and 2

To disable RSTP on Switch 1 and Switch 2, perform this task on each switch:

CLI Quick Configuration

To quickly disable RSTP on Switch 1 and Switch 2, copy the following command and paste it into each switch terminal window:

```
[edit]
```

set protocols rstp disable

Step-by-Step Procedure

To disable RSTP on Switch 1 and Switch 2:

1. Disable RSTP on Switch 1 and Switch 2:

```
[edit]
user@switch# set protocols rstp disable
```

Results

Check the results of the configuration:

```
[edit]
user@switch# show
protocols {
    rstp {
        disable;
    }
}
```

Configuring Redundant Trunk Links on Switch 3

To configure redundant trunk links on Switch 3, perform this task:

CLI Quick Configuration

To quickly configure the redundant trunk group rtg0 on Switch 3, copy the following commands and paste them into the switch terminal window:

```
[edit]
set protocols rstp disable
set switch-options redundant-trunk-group group rtg0 interface ge-0/0/9.0 primary
set switch-options redundant-trunk-group group rtg0 interface ge-0/0/10.0
set redundant-trunk-group group rtg0 preempt-cutover-timer 60
```

Step-by-Step Procedure

Configure the redundant trunk group rtg0 on Switch 3.

1. Turn off RSTP:

```
[edit]
user@switch# set protocols rstp disable
```

2. Name the redundant trunk group rtg0 while configuring trunk interface ge-0/0/9.0 as the primary link and ge-0/0/10 as an unspecified link to serve as the secondary link:

```
[edit switch-options]
user@switch#  set redundant-trunk-group group rtg0 interface ge-0/0/9.0 primary
user@switch#  set redundant-trunk-group group rtg0 interface ge-0/0/10.0
```

3. (Optional) Change the time interval (from the default of 1 second) that a re-enabled primary link waits to take over for an active secondary link:

```
[edit switch-options]
user@switch#  set switch-options redundant-trunk-group group rtg0 preempt-cutover-timer 60
```

Results

Check the results of the configuration:

```
[edit]
user@switch# show
switch-options
    redundant-trunk-group {
        group rtg0 {
            preempt-cutover-timer 60;
            interface ge-0/0/9.0 {
                primary;
            }
            interface ge-0/0/10.0;
        }
    }
protocols {
    rstp {
        disable;
    }
}
```

Verification

IN THIS SECTION

- Verifying That a Redundant Trunk Group Was Created | **946**

To confirm that the configuration is set up correctly, perform this task:

Verifying That a Redundant Trunk Group Was Created

Purpose

Verify that the redundant trunk group rtg0 has been created on Switch 1 and that trunk interfaces are members of the redundant trunk group.

Action

List all redundant trunk groups configured on the switch:

```
user@switch> show redundant-trunk-group
```

```
Group       Interface   State       Time of last flap          Flap
name                                                           count

rtg0        ge-0/0/9.0  Up/Pri      Never                        0
            ge-0/0/10.0 Up          Never                        0
```

Meaning

The **show redundant-trunk-group** command lists all redundant trunk groups configured on the switch as well as the interface names and their current states (up or down for an unspecified link, and up or down and primary for a primary link). For this configuration example, the output shows that the redundant trunk group rtg0 is configured on the switch. The **Up** beside the interfaces indicates that both link cables are physically connected. The **Pri** beside trunk interface ge-0/0/9.0 indicates that it is configured as the primary link.

Example: Configuring Redundant Trunk Links for Faster Recovery on EX Series Switches

IN THIS SECTION

NOTE: This example uses Junos OS for EX Series switches that does not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see "Example: Configuring Redundant Trunk Links for Faster Recovery on Devices with ELS Support" on page 940. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50.

You can manage network convergence by configuring both a primary link and a secondary link on a switch; this is called a redundant trunk group (RTG). If the primary link in a redundant trunk group fails, it passes its known MAC address locations to the secondary link, which automatically takes over after one minute.

This example describes how to create a redundant trunk group with a primary and a secondary link:

Requirements

This example uses the following hardware and software components:

- Two EX Series distribution switches
- One EX Series access switch
- Junos OS Release 10.4 or later for EX Series switches

Before you configure the redundant trunk links network on the access and distribution switches, be sure you have:

- Configured interfaces **ge-0/0/9** and **ge-0/0/10** on the access switch, Switch 3, as trunk interfaces. See *Configuring Gigabit Ethernet Interfaces (CLI Procedure)*.
- Configured one trunk interface on each distribution switch, Switch 1 and Switch 2.
- Connected the three switches as shown in the topology for this example (see Figure 49 on page 949).

Overview and Topology

In a typical enterprise network composed of distribution and access layers, a redundant trunk link provides a simple solution for trunk interface network recovery. When a trunk interface fails, data traffic is routed to another trunk interface after one minute, thereby keeping network convergence time to a minimum.

This example shows the configuration of a redundant trunk group that includes one primary link (and its interface) and one unspecified link (and its interface) that serves as the secondary link.

A second type of redundant trunk group, not illustrated in the example, consists of two unspecified links (and their interfaces); in this case, neither of the links is primary. In this second case, the software selects an active link by comparing the port numbers of the two links and activating the link with the higher port number. For example, if the two link interfaces use interfaces **ge-0/1/0** and **ge-0/1/1**, the software activates **ge-0/1/1.** (In the interface names, the final number is the port number.)

The two links in a redundant trunk group generally operate the same way, whether they are configured as primary/unspecified or unspecified/unspecified. Data traffic initially passes through the active link but is blocked on the inactive link. While data traffic is blocked on the secondary link, note that Layer 2 control traffic is still permitted if the link is active. For example, an LLDP session can be run between two switches on the secondary link. If the active link either goes down or is disabled administratively, it broadcasts a list of its known MAC addresses for data traffic; the other link immediately picks up and adds the MAC addresses to its address table, becomes active, and begins forwarding traffic.

The one difference in operation between the two types of redundant trunk groups occurs when a primary link is active, goes down, is replaced by the secondary link, and then reactivates. When a primary link is re-enabled while the secondary link is active, the primary link waits 1 second (you can change the length of time using the preempt cutover timer to accommodate your network) and then takes over as the active link. In other words, the primary link has priority and is always activated if it is available. This differs from the behavior of two unspecified links, which act as equals. Because the unspecified links are equal, the active link remains active until it either goes down or is disabled administratively; this is the only time that the other unspecified link learns the MAC addresses and immediately becomes active.

The example given here illustrates a primary/unspecified configuration for a redundant trunk group because that configuration gives you more control and is more commonly used.

> **NOTE:** Rapid Spanning Tree Protocol (RSTP) is enabled by default on EX Series switches to create a loop-free topology, but an interface is not allowed to be in both a redundant trunk group and in a spanning-tree protocol topology at the same time. You will need to disable RSTP on the two distribution switches in the example, Switch 1 and Switch 2. Spanning-tree protocols can, however, continue operating in other parts of the network—for example, between the distribution switches and also in links between distribution switches and the enterprise core.

Figure 49 on page 949 displays an example topology containing three switches. Switch 1 and Switch 2 make up the distribution layer, and Switch 3 makes up the access layer. Switch 3 is connected to the distribution layer through trunk interfaces **ge-0/0/9.0** (Link 1) and **ge-0/0/10.0** (Link 2).

Table 126 on page 949 lists the components used in this redundant trunk group.

Because RSTP and RTGs cannot operate simultaneously on a switch, you disable RSTP on Switch 1 and Switch 2 in the first configuration task, and you disable RSTP on Switch 3 in the second task.

The second configuration task creates a redundant trunk group called **example 1** on Switch 3. The trunk interfaces **ge-0/0/9.0** and **ge-0/0/10.0** are the two links configured in the second configuration task. You configure the trunk interface **ge-0/0/9.0** as the primary link. You configure the trunk interface **ge-0/0/10.0** as an unspecified link, which becomes the secondary link by default.

Figure 49: Topology for Configuring the Redundant Trunk Links

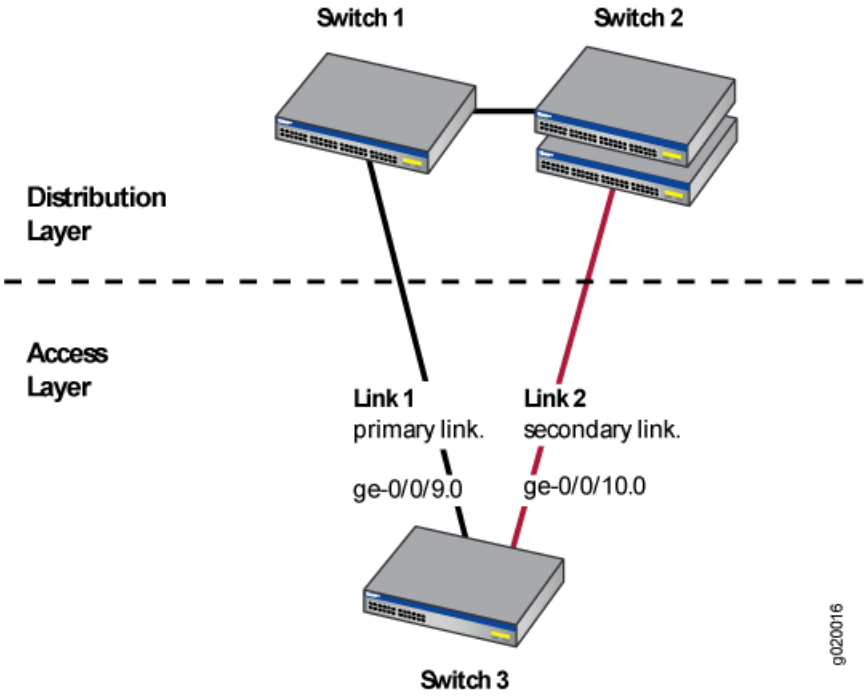


Table 126: Components of the Redundant Trunk Link Topology

Property	Settings
Switch hardware	• Switch 1–1 EX Series distribution switch • Switch 2–1 EX Series distribution switch • Switch 3–1 EX Series access switch
Trunk interfaces	On Switch 3 (access switch): **ge-0/0/9.0** and **ge-0/0/10.0**
Redundant trunk group	**example1**

Disabling RSTP on Switches 1 and 2

To disable RSTP on Switch 1 and Switch 2, perform this task on each switch:

CLI Quick Configuration

To quickly disable RSTP on Switch 1 and Switch 2, copy the following command and paste it into each switch terminal window:

```
[edit]
```

set protocols rstp disable

Step-by-Step Procedure

To disable RSTP on Switch 1 and Switch 2:

1. Disable RSTP on Switch 1 and Switch 2:

```
[edit]
user@switch# set protocols rstp disable
```

Results

Check the results of the configuration:

```
[edit]
user@switch# show
protocols {
    rstp {
        disable;
    }
}
```

Configuring Redundant Trunk Links on Switch 3

To configure redundant trunk links on Switch 3, perform this task:

CLI Quick Configuration

To quickly configure the redundant trunk group **example1** on Switch 3, copy the following commands and paste them into the switch terminal window:

```
[edit]
set protocols rstp disable
set ethernet-switching-options redundant-trunk-group group example1 interface ge-0/0/9.0 primary
set ethernet-switching-options redundant-trunk-group group example1 interface ge-0/0/10.0
set ethernet-switching-options redundant-trunk-group group example1 preempt-cutover-timer 60
```

Step-by-Step Procedure

Configure the redundant trunk group **example1** on Switch 3.

1. Turn off RSTP:

```
[edit]
user@switch# set protocols rstp disable
```

2. Name the redundant trunk group example1 while configuring trunk interface **ge-0/0/9.0** as the primary link and **ge-0/0/10** as an unspecified link to serve as the secondary link:

```
[edit ethernet-switching-options]
user@switch#  set redundant-trunk-group group example1 interface ge-0/0/9.0 primary
user@switch#  set redundant-trunk-group group example1 interface ge-0/0/10.0
```

3. (Optional) Change the length of time (from the default of 1 second) that a re-enabled primary link waits to take over for an active secondary link:

```
[edit ethernet-switching-options]
user@switch#  set redundant-trunk-group group example1 preempt-cutover-timer 60
```

Results

Check the results of the configuration:

```
[edit]
user@switch# show
ethernet-switching-options
    redundant-trunk-group {
        group example1 {
            preempt-cutover-timer 60;
            interface ge-0/0/9.0 {
                primary;
            }
            interface ge-0/0/10.0;
        }
    }
protocols {
    rstp {
        disable;
    }
}
```

Verification

IN THIS SECTION

- Verifying That a Redundant Trunk Group Was Created | **952**

To confirm that the configuration is set up correctly, perform this task:

Verifying That a Redundant Trunk Group Was Created

Purpose

Verify that the redundant trunk group **example1** has been created on Switch 1 and that trunk interfaces are members of the redundant trunk group.

Action

List all redundant trunk groups configured on the switch:

```
user@switch> show redundant-trunk-group
```

```
Group      Interface   State       Time of last flap          Flap
name                                                          count

example1   ge-0/0/9.0  Up/Pri      Never                        0
           ge-0/0/10.0 Up          Never                        0
```

Meaning

The **show redundant-trunk-group** command lists all redundant trunk groups configured on the switch, both links' interface addresses, and the links' current states (up or down for an unspecified link, and up or down and primary for a primary link). For this configuration example, the output shows that the redundant trunk group **example1** is configured on the switch. The **(Up)** beside the interfaces indicates that both link cables are physically connected. The **(Pri)** beside trunk interface **ge-0/0/9.0** indicates that it is configured as the primary link.

31
CHAPTER

Configuring Proxy ARP

Proxy ARP

IN THIS SECTION

Understanding Proxy ARP

IN THIS SECTION

You can configure proxy Address Resolution Protocol (ARP) to enable the switch to respond to ARP queries for network addresses by offering its own Ethernet media access control (MAC) address. With proxy ARP enabled, the switch captures and routes traffic to the intended destination.

Proxy ARP is useful in situations where hosts are on different physical networks and you do not want to use subnet masking. Because ARP broadcasts are not propagated between hosts on different physical networks, hosts will not receive a response to their ARP request if the destination is on a different subnet. Enabling the switch to act as an ARP proxy allows the hosts to transparently communicate with each other through the switch. Proxy ARP can help hosts on a subnet reach remote subnets without your having to configure routing or a default gateway.

Benefits of Using Proxy ARP

- Enables the switch to respond to ARP queries for network addresses by offering its own Ethernet media access control (MAC) address.
- Enables the switch to act as an ARP proxy allows the hosts to transparently communicate with each other through the switch.
- Helps hosts on a subnet reach remote subnets without your having to configure routing or a default gateway.

What Is ARP?

Ethernet LANs use ARP to map Ethernet MAC addresses to IP addresses. Each device maintains a cache containing a mapping of MAC addresses to IP addresses. The switch maintains this mapping in a cache that it consults when forwarding packets to network devices. If the ARP cache does not contain an entry for the destination device, the host (the DHCP client) broadcasts an ARP request for that device's address and stores the response in the cache.

Proxy ARP Overview

When proxy ARP is enabled, if the switch receives an ARP request for which it has a route to the target (destination) IP address, the switch responds by sending a proxy ARP reply packet containing its own MAC address. The host that sent the ARP request then sends its packets to the switch, which forwards them to the intended host.

> **NOTE:** For security reasons, the source address in an ARP request must be on the same subnet as the interface on which the ARP request is received.

You can configure proxy ARP for each interface. You can also configure proxy ARP for an integrated routing and bridging (IRB) interface named irb or a routed VLAN interface (RVI) named vlan. (On EX Series switches

that use Juniper Networks Junos operating system (Junos OS) with support for the Enhanced Layer 2 Software (ELS) configuration style, the feature is known as an IRB interface. On EX Series switches that use Junos OS that does not support ELS, the feature is known as an RVI.)

Two modes of proxy ARP are supported: restricted and unrestricted. Both modes require that the switch have an active route to the destination address of the ARP request.

- Restricted—The switch responds to ARP requests in which the physical networks of the source and target are different and does not respond if the source and target IP addresses are on the same subnet. In this mode, hosts on the same subnet communicate without proxy ARP. We recommend that you use this mode on the switch.
- Unrestricted—The switch responds to all ARP requests for which it has a route to the destination. This is the default mode (because it is the default mode in Juniper Networks Junos operating system (Junos OS) configurations other than those on the switch). We recommend using restricted mode on the switch.

Best Practices for Proxy ARP

We recommend these best practices for configuring proxy ARP on the switches:

- Set proxy ARP to restricted mode.
- Use restricted mode when configuring proxy ARP on RVIs or IRB interfaces.
- If you set proxy ARP to unrestricted, disable gratuitous ARP requests on each interface enabled for proxy ARP.

Configuring Proxy ARP on Devices with ELS Support

> **NOTE:** This task uses Junos OS for EX Series switches and QFX3500 and QFX3600 switches with support for the Enhanced Layer 2 Software (ELS) configuration style.If your switch runs software that does not support ELS, see "Configuring Proxy ARP on Switches" on page 958 or "Configuring Proxy ARP" on page 959. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50.

You can configure proxy Address Resolution Protocol (ARP) on your switch to enable the switch to respond to ARP queries for network addresses by offering its own media access control (MAC) address. With proxy ARP enabled, the switch captures and routes traffic to the intended destination.

To configure proxy ARP on a single interface:

```
[edit interfaces]
user@switch# set interface-name unit logical-unit-number proxy-arp (restricted
| unrestricted)
```

> **BEST PRACTICE:** We recommend that you configure proxy ARP in restricted mode. In restricted mode, the switch does not act as a proxy if the source and target IP addresses are on the same subnet. If you decide to use unrestricted mode, disable gratuitous ARP requests on the interface to avoid a situation wherein the switch's response to a gratuitous ARP request appears to the host to be an indication of an IP conflict.

To configure proxy ARP on an integrated routing and bridging (IRB) interface:

```
[edit interfaces]
user@switch# set irb.logical-unit-number proxy-arp restricted
```

Configuring Proxy ARP on Switches

> **NOTE:** This task uses Junos OS for EX Series switches that does not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see "Configuring Proxy ARP on Devices with ELS Support" on page 957. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50.

You can configure proxy Address Resolution Protocol (ARP) on your EX Series switch to enable the switch to respond to ARP queries for network addresses by offering its own media access control (MAC) address. With proxy ARP enabled, the switch captures and routes traffic to the intended destination.

To configure proxy ARP on a single interface:

```
[edit interfaces]
user@switch# set ge-0/0/3 unit 0 proxy-arp restricted
```

> **BEST PRACTICE:** We recommend that you configure proxy ARP in restricted mode. In restricted mode, the switch is not a proxy if the source and target IP addresses are on the same subnet. If you use unrestricted mode, disable gratuitous ARP requests on the interface to avoid the situation of the switch's response to a gratuitous ARP request appearing to the host to be an indication of an IP conflict:

To configure proxy ARP on a routed VLAN interface (RVI):

```
[edit interfaces]
user@switch# set vlan unit 100 proxy-arp restricted
```

SEE ALSO

- *Configuring Routed VLAN Interfaces on Switches (CLI Procedure)*

Configuring Proxy ARP

You can configure proxy Address Resolution Protocol (ARP) to enable the switch to respond to ARP queries for network addresses by offering its own media access control (MAC) address. With proxy ARP enabled, the switch captures and routes traffic to the intended destination.

To configure proxy ARP on a single interface:

```
[edit interfaces]
user@switch# set xe-0/0/3 unit 0 proxy-arp restricted
```

> **BEST PRACTICE:** We recommend that you configure proxy ARP in restricted mode. In restricted mode, the switch is not a proxy if the source and target IP addresses are on the same subnet. If you use unrestricted mode, disable gratuitous ARP requests on the interface to avoid the situation of the switch's response to a gratuitous ARP request appearing to the host to be an indication of an IP conflict:

To configure proxy ARP on a routed VLAN interface (RVI):

```
[edit interfaces]
user@switch# set vlan unit 100 proxy-arp restricted
```

SEE ALSO

Understanding Integrated Routing and Bridging | 728

Verifying That Proxy ARP Is Working Correctly

Purpose

Verify that the switch is sending proxy ARP messages.

Action

List the system statistics for ARP:

```
user@switch> show system statistics arp
```

```
arp:
         90060 datagrams received
         34 ARP requests received
         610 ARP replies received
         2 resolution request  received
         0 unrestricted proxy requests
         0 restricted proxy requests
         0 received proxy requests
         0 unrestricted proxy requests not proxied
         0 restricted proxy requests not proxied
         0 datagrams with bogus interface
         0 datagrams with incorrect length
         0 datagrams for non-IP protocol
         0 datagrams with unsupported op code
         0 datagrams with bad protocol address length
         0 datagrams with bad hardware address length
         0 datagrams with multicast source address
         0 datagrams with multicast target address
         0 datagrams with my own hardware address
         0 datagrams for an address not on the interface
         0 datagrams with a broadcast source address
         294 datagrams with source address duplicate to mine
         89113 datagrams which were not for me
         0 packets discarded waiting for resolution
         0 packets sent after waiting for resolution
         309 ARP requests sent
         35 ARP replies sent
         0 requests for memory denied
         0 requests dropped on entry
         0 requests dropped during retry
         0 requests dropped due to interface deletion
         0 requests on unnumbered interfaces
         0 new requests on unnumbered interfaces
         0 replies for from unnumbered interfaces
         0 requests on unnumbered interface with non-subnetted donor
         0 replies from unnumbered interface with non-subnetted donor
```

Meaning

The statistics show that two proxy ARP requests were received. The **unrestricted proxy requests not proxied** and **restricted proxy requests not proxied** fields indicate that all the unproxied ARP requests received have been proxied by the switch.

CHAPTER 32

Configuring Layer 2 Interfaces on Security Devices

Layer 2 Interfaces on Security Devices

IN THIS SECTION

Understanding Layer 2 Interfaces on Security Devices

Layer 2 logical interfaces are created by defining one or more logical units on a physical interface with the family address type **ethernet-switching**. If a physical interface has a **ethernet-switching** family logical interface, it cannot have any other family type in its logical interfaces. A logical interface can be configured in one of the following modes:

- Access mode—Interface accepts untagged packets, assigns the specified VLAN identifier to the packet, and forwards the packet within the VLAN that is configured with the matching VLAN identifier.
- Trunk mode—Interface accepts any packet tagged with a VLAN identifier that matches a specified list of VLAN identifiers. Trunk mode interfaces are generally used to interconnect switches. To configure a VLAN identifier for untagged packets received on the physical interface, use the **native-vlan-id** option. If the **native-vlan-id** option is not configured, untagged packets are dropped.

> **NOTE:** Multiple trunk mode logical interfaces can be defined, as long as the VLAN identifiers of a trunk interface do not overlap with those of another trunk interface. The **native-vlan-id** must belong to a VLAN identifier list configured for a trunk interface.

SEE ALSO

Ethernet Switching and Layer 2 Transparent Mode Overview | 41

Example: Configuring Layer 2 Logical Interfaces on Security Devices | 963

Example: Configuring Layer 2 Logical Interfaces on Security Devices

IN THIS SECTION

This example shows how to configure a Layer 2 logical interface as a trunk port so that the incoming packets can be selectively redirected to a firewall or other security device.

Requirements

Before you begin, configure the VLANs. See "Example: Configuring VLANs on Security Devices" on page 187.

Overview

In this example, you configure logical interface ge-3/0/0.0 as a trunk port that carries traffic for packets tagged with VLAN identifiers 1 through 10; this interface is implicitly assigned to the previously configured VLANs vlan-a and vlan-b. Then you assign a VLAN ID of 10 to any untagged packets received on physical interface ge-3/0/0.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set interfaces ge-3/0/0 unit 0 family ethernet-switching interface-mode trunk vlan members 1–10
set interfaces ge-3/0/0 vlan-tagging native-vlan-id 10
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure a Layer 2 logical interface as a trunk port:

1. Configure the logical interface.

```
[edit interfaces ge-3/0/0]
user@host# set unit 0 family ethernet-switching interface-mode trunk vlan members 1–10
```

2. Specify a VLAN ID for untagged packets.

```
[edit interfaces ge-3/0/0]
user@host# set vlan-tagging native-vlan-id 10
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show interfaces ge-3/0/0** and **show interfaces ge-3/0/0.0** commands.

SEE ALSO

Understanding Layer 2 Interfaces on Security Devices | **962**

Example: Configuring Layer 2 Security Zones | **980**

Understanding Mixed Mode (Transparent and Route Mode) on Security Devices

Mixed mode supports both transparent mode (Layer 2) and route mode (Layer 3); it is the default mode. You can configure both Layer 2 and Layer 3 interfaces simultaneously using separate security zones.

NOTE: For the mixed mode configuration, you must reboot the device after you commit the changes. However, for SRX5000 line devices, reboot is not required.

SRX4100 and SRX4200 devices support logical system in both transparent and route mode

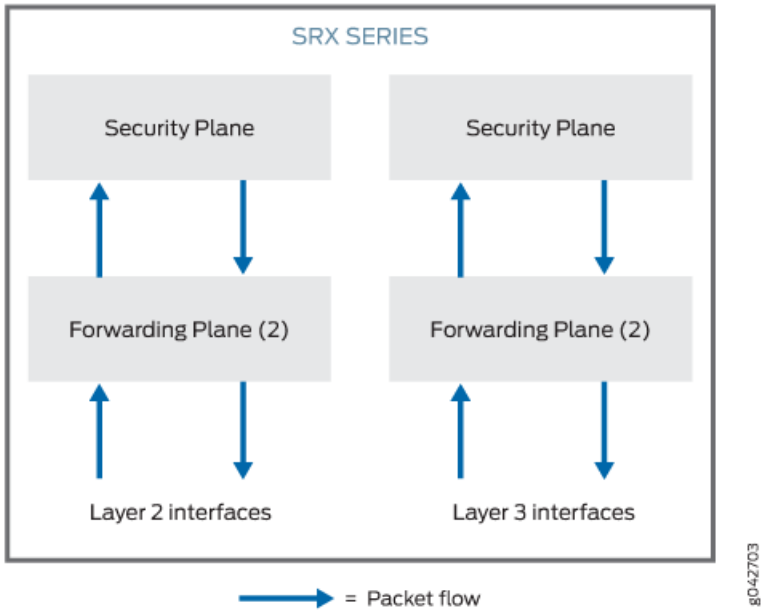
SRX4600 device supports logical system in route mode only

In mixed mode (Transparent and Route Mode):

- There is no routing among IRB interfaces and between IRB interfaces and Layer 3 interfaces.

The device in Figure 50 on page 965 looks like two separate devices. One device runs in Layer 2 transparent mode and the other device runs in Layer 3 routing mode. But both devices run independently. Packets cannot be transferred between the Layer 2 and Layer 3 interfaces, because there is no routing among IRB interfaces and between IRB interfaces and Layer 3 interfaces.

Figure 50: Architecture of Mixed Transparent and Route Mode



In mixed mode, the Ethernet physical interface can be either a Layer 2 interface or a Layer 3 interface, but the Ethernet physical interface cannot be both simultaneously. However, Layer 2 and Layer 3 families can exist on separate physical interfaces on the same device.

Table 127 on page 966 lists the Ethernet physical interface types and supported family types.

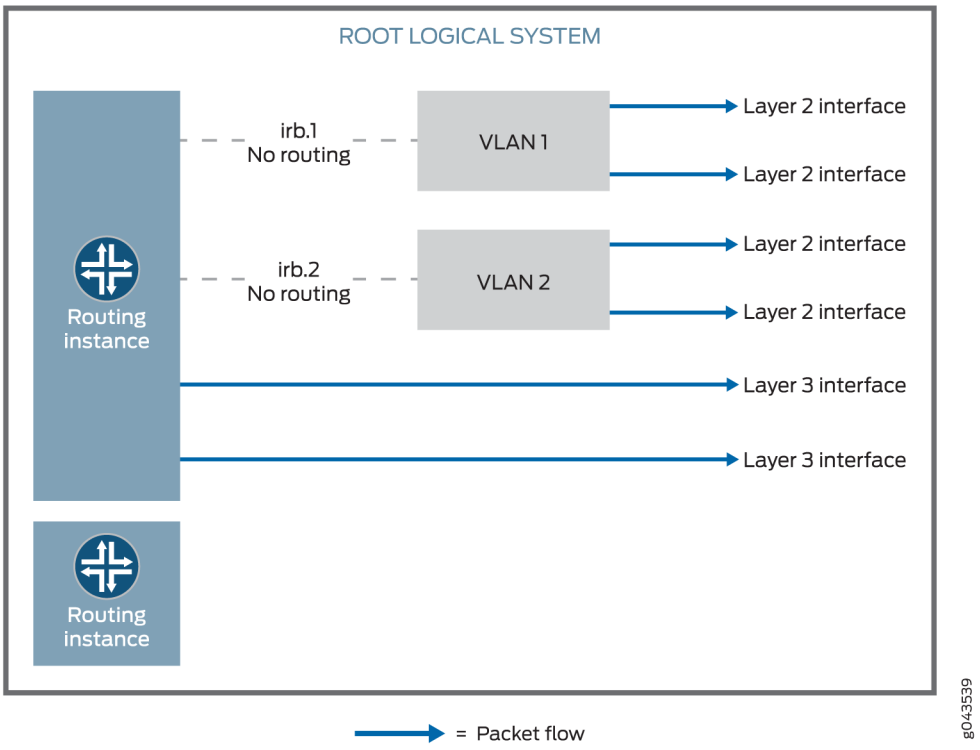
Table 127: Ethernet Physical Interface and Supported Family Types

Ethernet Physical Interface Type	Supported Family Type
Layer 2 Interface	**ethernet-switching**
Layer 3 Interface	**inet** and **inet6**

NOTE: Multiple routing instances are supported.

You can configure both the pseudointerface **irb.x** and the Layer 3 interface under the same default routing instance using either a default routing instance or a user-defined routing instance. See Figure 51 on page 966.

Figure 51: Mixed Transparent and Route Mode



Packets from the Layer 2 interface are switched within the same VLAN, or they connect to the host through the IRB interface. Packets cannot be routed to another IRB interface or a Layer 3 interface through their own IRB interface.

Packets from the Layer 3 interface are routed to another Layer 3 interface. Packets cannot be routed to a Layer 2 interface through an IRB interface.

Table 128 on page 967 lists the security features that are supported in mixed mode and the features that are not supported in transparent mode for Layer 2 switching.

Table 128: Security Features Supported in Mixed Mode (Transparent and Route Mode)

Mode Type	Supported	Not Supported
Mixed mode	• Application Layer Gateways (ALGs) • Firewall User Authentication (FWAUTH) • Intrusion Detection and Prevention (IDP) • Screen • AppSecure	• Unified Threat Management (UTM)
Route mode (Layer 3 interface)	• Network Address Translation (NAT) • VPN	—
Transparent mode (Layer 2 interface)	• Unified Threat Management (UTM)	• Network Address Translation (NAT) • VPN

Starting in Junos OS Release 12.3X48-D10 and Junos OS Release 17.3R1, some conditions apply to mixed-mode operations. Note the conditions here:

- On SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 devices, you cannot configure Ethernet switching and virtual private LAN service (VPLS) using mixed mode (Layer 2 and Layer 3).
- On SRX5400, SRX5600, and SRX5800 devices, you do not have to reboot the device when you configure VLAN.

SEE ALSO

Example: Improving Security Services by Configuring an SRX Series Device Using Mixed Mode (Transparent and Route Mode) | **968**

Understanding Secure Wire on Security Devices | **1006**

Example: Improving Security Services by Configuring an SRX Series Device Using Mixed Mode (Transparent and Route Mode)

IN THIS SECTION

You can configure an SRX Series device using both transparent mode (Layer 2) and route mode (Layer 3) simultaneously to simplify deployments and to improve security services.

This example shows how to pass the Layer 2 traffic from interface ge-0/0/1.0 to interface ge-0/0/0.0 and Layer 3 traffic from interface ge-0/0/2.0 to interface ge-0/0/3.0.

Requirements

This example uses the following hardware and software components:

- An SRX Series device
- Four PCs

Before you begin:

- Create a separate security zone for Layer 2 and Layer 3 interfaces. See "Understanding Layer 2 Security Zones" on page 979.

Overview

In enterprises where different business groups have either Layer 2 or Layer 3 based security solutions, using a single mixed mode configuration simplifies their deployments. In a mixed mode configuration, you can also provide security services with integrated switching and routing.

In addition, you can configure an SRX Series device in both standalone and chassis cluster mode using mixed mode.

In mixed mode (default mode), you can configure both Layer 2 and Layer 3 interfaces simultaneously using separate security zones.

NOTE: For the mixed mode configuration, you must reboot the device after you commit the changes. However, for SRX5000 line devices, reboot is not required.

In this example, first you configure a Layer 2 family type called Ethernet switching to identify Layer 2 interfaces. You set the IP address 10.10.10.1/24 to IRB interface. Then you create zone L2 and add Layer 2 interfaces ge-0/0/1.0 and ge-0/0/0.0 to it.

Next you configure a Layer 3 family type inet to identify Layer 3 interfaces. You set the IP address 192.0.2.1/24 to interface ge-0/0/2.0 and the IP address 192.0.2.3/24 to interface ge-0/0/3. Then you create zone L3 and add Layer 3 interfaces ge-0/0/2.0 and ge-0/0/3.0 to it.

Topology

Figure 52 on page 969 shows a mixed mode topology.

Figure 52: Mixed Mode Topology

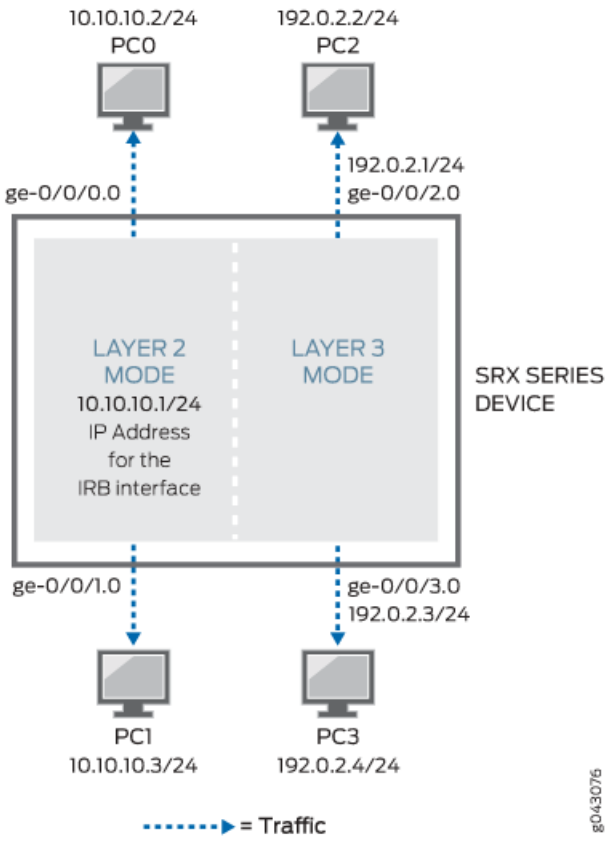


Table 129 on page 970 shows the parameters configured in this example.

Table 129: Layer 2 and Layer 3 Parameters

Parameter	Description
L2	Layer 2 zone.
ge-0/0/1.0 and ge-0/0/0.0	Layer 2 interfaces added to the Layer 2 zone.
L3	Layer 3 zone.
ge-0/0/2.0 and ge-0/0/3.0	Layer 3 interfaces added to the Layer 3 zone.
10.10.10.1/24	IP address for the IRB interface.
192.0.2.1/24 and 192.0.2.3/24	IP addresses for the Layer 3 interface.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set interfaces ge-0/0/0 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members 10
set interfaces ge-0/0/1 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members 10
set protocols l2-learning global-mode transparent-bridge
set interfaces irb unit 10 family inet address 10.10.10.1/24
set security zones security-zone L2 interfaces ge-0/0/1.0
set security zones security-zone L2 interfaces ge-0/0/0.0
set vlans vlan-10 vlan-id 10
set vlans vlan-10 l3-interface irb.10
set interfaces ge-0/0/2 unit 0 family inet address 192.0.2.1/24
set interfaces ge-0/0/3 unit 0 family inet address 192.0.2.3/24
set security policies default-policy permit-all
set security zones security-zone L2 host-inbound-traffic system-services any-service
set security zones security-zone L2 host-inbound-traffic protocols all
set security zones security-zone L3 host-inbound-traffic system-services any-service
set security zones security-zone L3 host-inbound-traffic protocols all
```

```
set security zones security-zone L3 interfaces ge-0/0/2.0
set security zones security-zone L3 interfaces ge-0/0/3.0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure Layer 2 and Layer 3 interfaces:

1. Create a Layer 2 family type to configure Layer 2 interfaces.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 family ethernet-switching interface-mode access
user@host# set ge-0/0/0 unit 0 family ethernet-switching vlan members 10
user@host# set ge-0/0/1 unit 0 family ethernet-switching interface-mode access
user@host# set ge-0/0/1 unit 0 family ethernet-switching vlan members 10
```

2. Configure Layer 2 interfaces to work under transparent-bridge mode.

```
[edit protocols]
user@host# set l2-learning global-mode transparent-bridge
```

3. Configure an IP address for the IRB interface.

```
[edit interfaces]
user@host# set irb unit 10 family inet address 10.10.10.1/24
```

4. Configure Layer 2 interfaces.

```
[edit security zones security-zone L2 interfaces]
user@host# set ge-0/0/1.0
user@host#  set ge-0/0/0.0
```

5. Configure VLAN.

```
[edit vlans vlan-10]
user@host# set vlan-id 10
user@host# set l3-interface irb.10
```

6. Configure IP addresses for Layer 3 interfaces.

```
[edit interfaces]
user@host# set ge-0/0/2 unit 0 family inet address 192.0.2.1/24
user@host# set ge-0/0/3 unit 0 family inet address 192.0.2.3/24
```

7. Configure the policy to permit the traffic.

```
[edit security policies]
user@host# set default-policy permit-all
```

8. Configure Layer 3 interfaces.

```
[edit security zones security-zone]
user@host# set L2 host-inbound-traffic system-services any-service
user@host# set L2 host-inbound-traffic protocols all
user@host# set L3 host-inbound-traffic system-services any-service
user@host# set L3 host-inbound-traffic protocols all
user@host# set L3 interfaces ge-0/0/2.0
user@host# set L3 interfaces ge-0/0/3.0
```

Results

From configuration mode, confirm your configuration by entering the **show interfaces**, **show security policies**, **show vlans**, and **show security zones** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
    ge-0/0/0 {
        unit 0 {
            family ethernet-switching {
                interface-mode access;
                vlan {
                    members 10;
                }
            }
        }
    }
    ge-0/0/1 {
        unit 0 {
```

```
            family ethernet-switching {
                interface-mode access;
                vlan {
                    members 10;
                }
            }
        }
    }
    ge-0/0/2 {
        unit 0 {
            family inet {
                address 192.0.2.1/24;
            }
        }
    }
    ge-0/0/3 {
        unit 0 {
            family inet {
                address 192.0.2.3/24;
            }
        }
    }
    irb {
        unit 10 {
            family inet {
                address 10.10.10.1/24;
            }
        }
    }
[edit]
user@host#  show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show vlans
    vlan-10 {
        vlan-id 10;
        l3-interface irb.10;
    }
[edit]
user@host#  show security zones
    security-zone L2 {
        host-inbound-traffic {
```

```
        system-services {
            any-service;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/1.0;
        ge-0/0/0.0;
    }
}
security-zone L3 {
    host-inbound-traffic {
        system-services {
            any-service;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/2.0;
        ge-0/0/3.0;
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying the Layer 2 and Layer 3 Interfaces and Zones

Purpose

Verify that the Layer 2 and Layer 3 interfaces and Layer 2 and Layer 3 zones are created.

Action

From operational mode, enter the **show security zones** command.

```
user@host> show security zones
```

```
 Security zone: HOST
  Send reset for non-SYN session TCP packets: Off
  Policy configurable: Yes
  Interfaces bound: 0
  Interfaces:

Security zone: L2
  Send reset for non-SYN session TCP packets: Off
  Policy configurable: Yes
  Interfaces bound: 2
  Interfaces:
    ge-0/0/0.0
    ge-0/0/1.0

Security zone: L3
  Send reset for non-SYN session TCP packets: Off
  Policy configurable: Yes
  Interfaces bound: 2
  Interfaces:
    ge-0/0/2.0
    ge-0/0/3.0

Security zone: junos-host
  Send reset for non-SYN session TCP packets: Off
  Policy configurable: Yes
  Interfaces bound: 0
  Interfaces:
```

Meaning

The output shows the Layer 2 (L2) and Layer 3 (L3) zone names and the number and names of Layer 2 and Layer 3 interfaces bound to the L2 and L3 zones.

Verifying the Layer 2 and Layer 3 Session

Purpose

Verify that the Layer 2 and Layer 3 sessions are established on the device.

Action

From operational mode, enter the **show security flow session** command.

```
user@host> show security flow session
```

```
Session ID: 1, Policy name: default-policy-logical-system-00/2, Timeout: 58, Valid

  In: 10.102.70.75/54395 --> 228.102.70.76/9876;udp, Conn Tag: 0x0, If: ge-0/0/0.0,
 Pkts: 1209, Bytes: 1695018,
  Out: 228.102.70.76/9876 --> 10.102.70.75/54395;udp, Conn Tag: 0x0, If: ge-0/0/1.0,
 Pkts: 0, Bytes: 0,

Session ID: 2, Policy name: default-policy-logical-system-00/2, Timeout: 58, Valid

  In: 10.102.70.19/23364 --> 228.102.70.20/23364;udp, Conn Tag: 0x0, If: ge-0/0/0.0,
 Pkts: 401, Bytes: 141152,
  Out: 228.102.70.20/23364 --> 10.102.70.19/23364;udp, Conn Tag: 0x0, If:
ge-0/0/1.0, Pkts: 0, Bytes: 0,
```

Meaning

The output shows active sessions on the device and each session's associated security policy.

- **Session ID 1**—Number that identifies the Layer 2 session. Use this ID to get more information about the Layer 2 session such as policy name or number of packets in and out.
- **default-policy-logical-system-00/2**—Default policy name that permitted the Layer 2 traffic.
- **In**—Incoming flow (source and destination Layer 2 IP addresses with their respective source and destination port numbers, session is ICMP, and the source interface for this session is ge-0/0/0.0).
- **Out**—Reverse flow (source and destination Layer 2 IP addresses with their respective source and destination port numbers, session is ICMP, and destination interface for this session is ge-0/0/1.0).
- **Session ID 2**—Number that identifies the Layer 2 session. Use this ID to get more information about the Layer 2 session such as policy name or number of packets in and out.
- **default-policy-logical-system-00/2**—Default policy name that permitted the Layer 2 traffic.
- **In**—Incoming flow (source and destination Layer 2 IP addresses with their respective source and destination port numbers, session is ICMP, and the source interface for this session is ge-0/0/0.0,).
- **Out**—Reverse flow (source and destination Layer 2 IP addresses with their respective source and destination port numbers, session is ICMP, and destination interface for this session is ge-0/0/1.0,).

SEE ALSO

Understanding Mixed Mode (Transparent and Route Mode) on Security Devices | 964

Understanding Secure Wire on Security Devices | 1006

33

CHAPTER

Configuring Security Zones and Security Policies on Security Devices

Security Zones and Security Policies on Security Devices

IN THIS SECTION

Understanding Layer 2 Security Zones

A Layer 2 security zone is a zone that hosts Layer 2 interfaces. A security zone can be either a Layer 2 or Layer 3 zone; it can host either all Layer 2 interfaces or all Layer 3 interfaces, but it cannot contain a mix of Layer 2 and Layer 3 interfaces.

The security zone type—Layer 2 or Layer 3—is implicitly set from the first interface configured for the security zone. Subsequent interfaces configured for the same security zone must be the same type as the first interface.

> **NOTE:** You cannot configure a device with both Layer 2 and Layer 3 security zones.

You can configure the following properties for Layer 2 security zones:

- Interfaces—List of interfaces in the zone.
- Policies—Active security policies that enforce rules for the transit traffic, in terms of what traffic can pass through the firewall, and the actions that need to take place on the traffic as it passes through the firewall.
- Screens—A Juniper Networks stateful firewall secures a network by inspecting, and then allowing or denying, all connection attempts that require passage from one security zone to another. For every security zone, and the MGT zone, you can enable a set of predefined screen options that detect and block various kinds of traffic that the device determines as potentially harmful.

> **NOTE:** You can configure the same screen options for a Layer 2 security zone as for a Layer 3 security zone.

- Address books—IP addresses and address sets that make up an address book to identify its members so that you can apply policies to them.
- TCP-RST—When this feature is enabled, the system sends a TCP segment with the reset flag set when traffic arrives that does not match an existing session and does not have the synchronize flag set.

In addition, you can configure a Layer 2 zone for host-inbound traffic. This allows you to specify the kinds of traffic that can reach the device from systems that are directly connected to the interfaces in the zone. You must specify all expected host-inbound traffic because inbound traffic from devices directly connected to the device's interfaces is dropped by default.

SEE ALSO

Ethernet Switching and Layer 2 Transparent Mode Overview | **41**

Understanding Layer 2 Interfaces on Security Devices | **962**

Example: Configuring Layer 2 Security Zones | **980**

Example: Configuring Layer 2 Logical Interfaces on Security Devices | **963**

Example: Configuring Layer 2 Security Zones

IN THIS SECTION

This example shows how to configure Layer 2 security zones.

Requirements

Before you begin, determine the properties you want to configure for the Layer 2 security zone. See "Understanding Layer 2 Security Zones" on page 979.

Overview

In this example, you configure security zone l2-zone1 to include a Layer 2 logical interface called ge-3/0/0.0 and security zone l2-zone2 to include a Layer 2 logical interface called ge-3/0/1.0. Then you configure l2-zone2 to allow all supported application services (such as SSH, Telnet, and SNMP) as host-inbound traffic.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set security-zone l2-zone1 interfaces ge-3/0/0.0
set security-zone l2-zone2 interfaces ge-3/0/1.0
set security-zone l2–zone2 host-inbound-traffic system-services all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure Layer 2 security zones:

1. Create a Layer 2 security zone and assign interfaces to it.

   ```
   [edit security zones]
   user@host# set security-zone l2-zone1 interfaces ge-3/0/0.0
   user@host# set security-zone l2-zone2 interfaces ge-3/0/1.0
   ```

2. Configure one of the Layer 2 security zones.

   ```
   [edit security zones]
   user@host# set security-zone l2–zone2 host-inbound-traffic system-services all
   ```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show security zones** command.

SEE ALSO

Example: Configuring Security Policies in Transparent Mode | 983

Example: Configuring Layer 2 Logical Interfaces on Security Devices | 963

Understanding Security Policies in Transparent Mode

In transparent mode, security policies can be configured only between Layer 2 zones. When packets are forwarded through the VLAN, the security policies are applied between security zones. A security policy for transparent mode is similar to a policy configured for Layer 3 zones, with the following exceptions:

- NAT is not supported.
- IPsec VPN is not supported.
- Application ANY is not supported.

Layer 2 forwarding does not permit any interzone traffic unless there is a policy explicitly configured on the device. By default, Layer 2 forwarding performs the following actions:

- Allows or denies traffic specified by the configured policy.
- Allows Address Resolution Protocol (ARP) and Layer 2 non-IP multicast and broadcast traffic.
- Continues to block all non-IP and non-ARP unicast traffic.

This default behavior can be changed for Ethernet switching packet flow by using either J-Web or the CLI configuration editor:

- Configure the **block-non-ip-all** option to block all Layer 2 non-IP and non-ARP traffic, including multicast and broadcast traffic.
- Configure the **bypass-non-ip-unicast** option to allow all Layer 2 non-IP traffic to pass through the device.

NOTE: You cannot configure both options at the same time.

Starting in Junos OS Release 12.3X48-D10 and Junos OS Release 17.3R1, you can create a separate security zone in mixed mode (the default mode) for Layer 2 and Layer 3 interfaces. However, there is no routing among IRB interfaces and between IRB interfaces and Layer 3 interfaces. Hence, you cannot configure security policies between Layer 2 and Layer 3 zones. You can only configure security policies between the Layer 2 zones or between Layer 3 zones.

SEE ALSO

Example: Configuring Security Policies in Transparent Mode | 983
Example: Configuring Layer 2 Security Zones | 980
Understanding Mixed Mode (Transparent and Route Mode) on Security Devices | 964

Example: Configuring Security Policies in Transparent Mode

IN THIS SECTION

This example shows how to configure security policies in transparent mode between Layer 2 zones.

Requirements

Before you begin, determine the policy behavior you want to include in the Layer 2 security zone. See “Understanding Security Policies in Transparent Mode” on page 982.

Overview

In this example, you configure a security policy to allow HTTP traffic from the 192.0.2.0/24 subnetwork in the l2–zone1 security zone to the server at 192.0.2.1/24 in the l2–zone2 security zone.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set security policies from-zone l2-zone1 to-zone l2-zone2 policy p1 match source-address 192.0.2.0/24
set security policies from-zone l2-zone1 to-zone l2-zone2 policy p1 match destination-address 192.0.2.1/24
set security policies from-zone l2-zone1 to-zone l2-zone2 policy p1 match application http
set security policies from-zone l2-zone1 to-zone l2-zone2 policy p1 then permit
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure security policies in transparent mode:

1. Create policies and assign addresses to the interfaces for the zones.

```
[edit security policies]
user@host# set from-zone l2-zone1 to-zone l2-zone2 policy p1 match source-address 192.0.2.0/24
user@host# set from-zone l2-zone1 to-zone l2-zone2 policy p1 match destination-address 192.0.2.1/24
```

2. Set policies for the application.

```
[edit security policies]
user@host# set from-zone l2-zone1 to-zone l2-zone2 policy p1 match application http
user@host# set from-zone l2-zone1 to-zone l2-zone2 policy p1 then permit
```

Results

From configuration mode, confirm your configuration by entering the **show security policies** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
```

```
    user@host> show security policies
    from-zone l2-zone1 to-zone l2-zone2
{
    policy p1 {
        match {
        source-address 192.0.2.0/24;
        destination-address 192.0.2.1/24;
        application junos-http;
    }
    then {
        permit;
    }
}
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Verifying Layer 2 Security Policies

Purpose

Verify that the Layer 2 security policies are configured properly.

Action

From configuration mode, enter the **show security policies** command.

SEE ALSO

Example: Configuring Layer 2 Security Zones | 980

Understanding Firewall User Authentication in Transparent Mode

A firewall user is a network user who must provide a username and password for authentication when initiating a connection across the firewall. Firewall user authentication enables administrators to restrict and permit users accessing protected resources behind a firewall based on their source IP address and other credentials. Junos OS supports the following types of firewall user authentication for transparent mode on the SRX Series device:

- Pass-through authentication—A host or a user from one zone tries to access resources on another zone. You must use an FTP, Telnet, or HTTP client to access the IP address of the protected resource and be

authenticated by the firewall. The device uses FTP, Telnet, or HTTP to collect username and password information, and subsequent traffic from the user or host is allowed or denied based on the result of this authentication.

- Web authentication—Users try to connect, by using HTTP, to an IP address on the IRB interface that is enabled for Web authentication. You are prompted for the username and password that are verified by the device. Subsequent traffic from the user or host to the protected resource is allowed or denied based on the result of this authentication.

SEE ALSO

Authentication and Integrated User Firewalls User Guide

Understanding Integrated Routing and Bridging | **728**

Example: Configuring an IRB Interface on a Security Device | **749**

34

CHAPTER

Configuring Ethernet Port Switching Modes on Security Devices

Ethernet Port Switching Modes on Security Devices

IN THIS SECTION

Understanding Switching Modes on Security Devices

There are two types of switching modes:

- Switching Mode–The uPIM appears in the list of interfaces as a single interface, which is the first interface on the uPIM. For example, ge-2/0/0. You can optionally configure each uPIM port only for autonegotiation, speed, and duplex mode. A uPIM in switching mode can perform the following functions:
  - Layer 3 forwarding—Routes traffic destined for WAN interfaces and other PIMs present on the chassis.
  - Layer 2 forwarding—Switches intra-LAN traffic from one host on the LAN to another LAN host (one port of uPIM to another port of same uPIM).
- Enhanced Switching Mode–Each port can be configured for switching or routing mode. This usage differs from the routing and switching modes, in which all ports must be in either switching or routing mode. The uPIM in enhanced switching mode provides the following benefits:

  Benefits of enhnanced switch mode:

  - Supports configuration of different types of VLANs and inter-VLAN routing.
  - Supports Layer 2 control plane protocol such as Link Aggregation Control Protocol (LACP).
  - Supports port-based Network Access Control (PNAC) by means of authentication servers.

  > **NOTE:** The SRX300 and SRX320 devices support enhanced switching mode only. When you set a multiport uPIM to enhanced switching mode, all the Layer 2 switching features are supported on the uPIM. (Platform support depends on the Junos OS release in your installation.)

You can set a multiport Gigabit Ethernet uPIM on a device to either switching or enhanced switching mode.

When you set a multiport uPIM to switching mode, the uPIM appears as a single entity for monitoring purposes. The only physical port settings that you can configure are autonegotiation, speed, and duplex mode on each uPIM port, and these settings are optional.

Ethernet Ports Switching Overview for Security Devices

IN THIS SECTION

Certain ports on Juniper Networks devices can function as Ethernet access switches that switch traffic at Layer 2 and route traffic at Layer 3.

You can deploy supported devices in branch offices as an access or desktop switch with integrated routing capability, thus eliminating intermediate access switch devices from your network topology. The Ethernet ports provide switching while the Routing Engine provides routing functionality, enabling you to use a single device to provide routing, access switching, and WAN interfaces.

This topic contains the following sections:

Supported Devices and Ports

Juniper Networks supports switching features on a variety of Ethernet ports and devices (see Table 130 on page 990). Platform support depends on the Junos OS release in your installation. The following ports and devices are included:

- Onboard Ethernet ports (Gigabit and Fast Ethernet built-in ports) on the SRX300, SRX320, SRX320 PoE, SRX340, SRX345, SRX550M and SRX1500 devices.
- Multiport Gigabit Ethernet XPIM on the SRX650 device.

Table 130: Supported Devices and Ports for Switching Features

Device	Ports
SRX100 devices	Onboard Fast Ethernet ports (fe-0/0/0 and fe-0/0/7)
SRX210 devices	Onboard Gigabit Ethernet ports (ge-0/0/0 and ge-0/0/1) and 1-Port Gigabit Ethernet SFP Mini-PIM port. Onboard Fast Ethernet ports (fe-0/0/2 and fe-0/0/7)
SRX220 devices	Onboard Gigabit Ethernet ports (ge-0/0/0 through ge-0/0/7) and 1-Port Gigabit Ethernet SFP Mini-PIM port.
SRX240 devices	Onboard Gigabit Ethernet ports (ge-0/0/0 through ge-0/0/15) and 1-Port Gigabit Ethernet SFP Mini-PIM port.
SRX300 devices	Onboard Gigabit Ethernet ports (ge-0/0/0 through ge-0/0/7)
SRX320 devices	Onboard Gigabit Ethernet ports (ge-0/0/0 through ge-0/0/7)
SRX340 devices	Onboard Gigabit Ethernet ports (ge-0/0/0 through ge-0/0/15)
SRX345 devices	Onboard Gigabit Ethernet ports (ge-0/0/0 through ge-0/0/15)
SRX550 devices	Onboard Gigabit Ethernet ports (ge-0/0/0 through ge-0/0/9, Multiport Gigabit Ethernet XPIM modules, and 1-Port Gigabit Ethernet SFP Mini-PIM port.
SRX550M devices	Onboard Gigabit Ethernet ports (ge-0/0/0 through ge-0/0/9 and Multiport Gigabit Ethernet XPIM modules.
SRX650 devices	Multiport Gigabit Ethernet XPIM modules **NOTE:** On SRX650 devices, Ethernet switching is not supported on Gigabit Ethernet interfaces (ge-0/0/0 through ge-0/0/3 ports).
SRX1500 devices	Onboard Gigabit Ethernet ports (ge-0/0/0 through ge-0/0/19)

On the SRX100. SRX220, SRX240, SRX300, SRX320, SRX340 and SRX345 devices, you can set the onboard Gigabit Ethernet ports to operate as either switched ports or routed ports. (Platform support depends on the Junos OS release in your installation.)

Integrated Bridging and Routing

Integrated bridging and routing (IRB) provides support for simultaneous Layer 2 swiching and Layer 3 routing within the same VLAN. Packets arriving on an interface of the VLAN are switched or routed based on the destination MAC address of the packet. Packets with the router's MAC address as the destination are routed to other Layer 3 interfaces.

Link Layer Discovery Protocol and LLDP-Media Endpoint Discovery

Devices use Link Layer Discovery Protocol (LLDP) and LLDP-Media Endpoint Discovery (MED) to learn and distribute device information about network links. The information allows the device to quickly identify a variety of systems, resulting in a LAN that interoperates smoothly and efficiently.

LLDP-capable devices transmit information in Type Length Value (TLV) messages to neighbor devices. Device information can include specifics, such as chassis and port identification and system name and system capabilities. The TLVs leverage this information from parameters that have already been configured in the Junos OS.

LLDP-MED goes one step further, exchanging IP-telephony messages between the device and the IP telephone. These TLV messages provide detailed information about Power over Ethernet (PoE) policy. The PoE Management TLVs let the device ports advertise the power level and power priority needed. For example, the device can compare the power needed by an IP telephone running on a PoE interface with available resources. If the device cannot meet the resources required by the IP telephone, the device could negotiate with the telephone until a compromise on power is reached.

The following basic TLVs are supported:

- Chassis Identifier—The MAC address associated with the local system.
- Port identifier—The port identification for the specified port in the local system.
- Port Description—The user-configured port description. The port description can be a maximum of 256 characters.
- System Name—The user-configured name of the local system. The system name can be a maximum of 256 characters.
- Switching Features Overview—This information is not configurable, but taken from the software.
- System Capabilities—The primary function performed by the system. The capabilities that system supports; for example, Ethernet switching or router. This information is not configurable, but based on the model of the product.
- Management Address—The IP management address of the local system.

The following LLDP-MED TLVs are supported:

- LLDP-MED Capabilities—A TLV that advertises the primary function of the port. The values range from 0 through 15:
  - 0—Capabilities
  - 1—Network policy
  - 2—Location identification
  - 3—Extended power through medium-dependent interface power-sourcing equipment (MDI-PSE)
  - 4—Inventory
  - 5–15—Reserved
- LLDP-MED Device Class Values:
  - 0—Class not defined
  - 1—Class 1 device
  - 2—Class 2 device
  - 3—Class 3 device
  - 4—Network connectivity device
  - 5–255— Reserved

    NOTE: Starting in Junos OS Release 15.1X49-D60 and Junos OS Release 17.3R1, Link Layer Discovery Protocol (LLDP) and LLDP-Media Endpoint Discovery (MFD) are enabled on SRX300, SRX320, SRX340, SRX345, SRX550M and SRX1500 devices.

- Network Policy—A TLV that advertises the port VLAN configuration and associated Layer 2 and Layer 3 attributes. Attributes include the policy identifier, application types, such as voice or streaming video, 802.1Q VLAN tagging, and 802.1p priority bits and Diffserv code points.
- Endpoint Location—A TLV that advertises the physical location of the endpoint.
- Extended Power via MDI—A TLV that advertises the power type, power source, power priority, and power value of the port. It is the responsibility of the PSE device (network connectivity device) to advertise the power priority on a port.

LLDP and LLDP-MED must be explicitly configured on uPIMs (in enhanced switching mode) on base ports on SRX100, SRX210, SRX240, SRX300, SRX320, SRX340, and SRX345 devices, and Gigabit Backplane Physical Interface Modules (GPIMs) on SRX650 devices. (Platform support depends on the Junos OS release in your installation.) To configure LLDP on all interfaces or on a specific interface, use the **lldp** statement at the [**set protocols**] hierarchy level. To configure LLDP-MED on all interfaces or on a specific interface, use the **lldp-med** statement at the [**set protocols**] hierarchy level.

Types of Switch Ports

The ports, or interfaces, on a switch operate in either access mode or trunk mode.

An interface in access mode connects to a network device, such as a desktop computer, an IP telephone, a printer, a file server, or a security camera. The interface itself belongs to a single VLAN. The frames transmitted over an access interface are normal Ethernet frames.

Trunk interfaces handle traffic for multiple VLANs, multiplexing the traffic for all those VLANs over the same physical connection. Trunk interfaces are generally used to interconnect switches to one another.

uPIM in a Daisy Chain

You cannot combine multiple uPIMs to act as a single integrated switch. However, you can connect uPIMs on the same chassis externally by physically connecting a port on one uPIM to a port on another uPIM in a daisy-chain fashion.

Two or more uPIMs daisy-chained together create a single switch with a higher port count than either individual uPIM. One port on each uPIM is used solely for the connection. For example, if you daisy-chain a 6-port uPIM and an 8-port uPIM, the result operates as a 12-port uPIM. Any port of a uPIM can be used for daisy chaining.

Configure the IP address for only one of the daisy-chained uPIMs, making it the primary uPIM. The secondary uPIM routes traffic to the primary uPIM, which forwards it to the Routing Engine. This results in some increase in latency and packet drops due to oversubscription of the external link.

Only one link between the two uPIMs is supported. Connecting more than one link between uPIMs creates a loop topology, which is not supported.

Q-in-Q VLAN Tagging

Q-in-Q tunneling, defined by the IEEE 802.1ad standard, allows service providers on Ethernet access networks to extend a Layer 2 Ethernet connection between two customer sites.

In Q-in-Q tunneling, as a packet travels from a customer VLAN (C-VLAN) to a service provider's VLAN, a service provider-specific 802.1Q tag is added to the packet. This additional tag is used to segregate traffic into service-provider-defined service VLANs (S-VLANs). The original customer 802.1Q tag of the packet remains and is transmitted transparently, passing through the service provider's network. As the packet leaves the S-VLAN in the downstream direction, the extra 802.1Q tag is removed.

> **NOTE:** When Q-in-Q tunneling is configured for a service provider’s VLAN, all Routing Engine packets, including packets from the routed VLAN interface, that are transmitted from the customer-facing access port of that VLAN will always be untagged.

There are three ways to map C-VLANs to an S-VLAN:

- All-in-one bundling—Use the **dot1q-tunneling** statement at the [**edit vlans**] hierarchy level to map without specifying customer VLANs. All packets from a specific access interface are mapped to the S-VLAN.
- Many-to-one bundling—Use the **customer-vlans** statement at the [**edit vlans**] hierarchy level to specify which C-VLANs are mapped to the S-VLAN.
- Mapping C-VLAN on a specific interface—Use the **mapping** statement at the [**edit vlans**] hierarchy level to map a specific C-VLAN on a specified access interface to the S-VLAN.

Table 131 on page 994 lists the C-VLAN to S-VLAN mapping supported on SRX Series devices. (Platform support depends on the Junos OS release in your installation.)

Table 131: Supported Mapping Methods

Mapping	SRX210	SRX240	SRX300	SRX320	SRX340	SRX345	SRX550M	SRX650
All-in-one bundling	Yes	Yes	No	No	Yes	Yes	Yes	Yes
Many-to-one bundling	No	No	No	No	Yes	Yes	Yes	Yes
Mapping C-VLAN on a specific interface	No	No	No	No	Yes	Yes	Yes	Yes

NOTE: VLAN translation is supported on SRX300 and SRX320 devices and these devices do not support Q-in-Q tunneling.

NOTE: On SRX650 devices, in the dot1q-tunneling configuration options, customer VLANs range and VLAN push do not work together for the same S-VLAN, even when you commit the configuration. If both are configured, then VLAN push takes priority over customer VLANs range.

IRB interfaces are supported on Q-in-Q VLANs for SRX210, SRX240, SRX340, SRX345, and SRX650 devices. Packets arriving on an IRB interface on a Q-in-Q VLAN are routed regardless of whether the packet is single or double tagged. The outgoing routed packets contain an S-VLAN tag only when exiting a trunk interface; the packets exit the interface untagged when exiting an access interface. (Platform support depends on the Junos OS release in your installation.)

In a Q-in-Q deployment, customer packets from downstream interfaces are transported without any changes to source and destination MAC addresses. You can disable MAC address learning at both the interface level and the VLAN level. Disabling MAC address learning on an interface disables learning for all the VLANs of which that interface is a member. When you disable MAC address learning on a VLAN, MAC addresses that have already been learned are flushed.

On SRX100, SRX210, SRX240, SRX300, SRX320, SRX340, SRX345, and SRX650 devices (with platform support depending on the Junos OS release in your installation), on the Layer 3 aggregated Ethernet, the following features are not supported:

- Encapsulations (such as CCC, VLAN CCC, VPLS, and PPPoE)
- J-Web
- Starting in Junos OS Release 19.4R2, you can configure the LLDP on redundant Ethernet (reth) interfaces. Use the **set protocol lldp interface *<reth-interface>*** command to configure LLDP on reth interface.
- On SRX550M devices the aggregate Ethernet (ae) interface with XE member interface cannot be configured with the Ethernet switching family.
- On SRX300, SRX320, SRX340, SRX345, and SRX550M devices, the Q-in-Q support on a Layer 3 interface has the following limitations:
  - Double tagging is not supported on reth and ae interfaces.
  - Multitopology routing is not supported in flow mode and in chassis clusters.
  - Dual tagged frames are not supported on encapsulations (such as CCC, TCC, VPLS, and PPPoE)
  - On Layer 3 logical interfaces, **input-vlan-map**, **output-vlan-map**, **inner-range**, and **inner-list** are not applicable
  - Only TPIDs with 0x8100 are supported, and the maximum number of tags is 2.
  - Dual tagged frames are accepted only for logical interfaces with IPV4 and IPv6 families.
- On SRX100, SRX210, SRX240, SRX300, SRX320, SRX340, SRX345, and SRX650 devices (with platform support depending on the Junos OS release in your installation), on the routed VLAN interface (RVI), the following features are not supported:
  - IS-IS (family ISO)
  - Encapsulations (Ether CCC, VLAN CCC, VPLS, PPPoE, and so on) on VLAN interfaces
  - CLNS
  - DVMRP
  - VLAN interface MAC change

- G-ARP
- Change VLAN-Id for VLAN interface

Example: Configuring Switching Modes on Security Devices

IN THIS SECTION

Requirements

Before you begin, see “Ethernet Ports Switching Overview for Security Devices” on page 989.

Overview

In this example, you configure **chassis** and set the l2-learning protocol to global mode switching. You then set a physical port parameter on the l2-learning protocols.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set protocols l2-learning global-mode switching
set interfaces ge-0/0/1 unit 0 family ethernet-switching interface-mode access
```

Step-by-Step Procedure

To configure switching mode:

1. Set l2-learning protocol to global mode switching.

```
[edit protocols l2-learning]
user@host# set protocols l2-learning global-mode switching
```

2. Set a physical port parameter on the l2-learning protocols.

```
[edit]
user@host# set interfaces ge-0/0/1 unit 0 family ethernet-switching interface-mode access
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Results

From configuration mode, confirm your configuration by entering the **show protocols** and **show interfaces** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show protocols
l2-learning {
    global-mode switching;
}
```

```
[edit]
user@host# show interfaces
ge-0/0/1 {
    unit 0 {
        family ethernet-switching {
            interface-mode access;
        }
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying the Switching Mode

Purpose

Make sure that the switching mode is configured as expected.

Action

From operational mode, enter the **show ethernet-switching global-information** command.

```
user@host> show ethernet-switching global-information
```

```
Global Configuration:

MAC aging interval    : 300
MAC learning          : Enabled
MAC statistics        : Disabled
MAC limit Count       : 16383
MAC limit hit         : Disabled
MAC packet action drop: Disabled
MAC+IP aging interval : IPv4 - 1200 seconds
                        IPv6 - 1200 seconds
MAC+IP limit Count    : 393215
MAC+IP limit reached  : No
LE  aging time        : 1200
LE  VLAN aging time   : 1200
Global Mode           : Switching
RE state              : Master
```

Meaning

The sample output shows that the global mode switching is configured as expected.

Verifying the Ethernet switching on Interface ge-0/0/1

Purpose

Make sure that the Ethernet switching is configured as expected on interface ge-0/0/1.

Action

From operational mode, enter the **show interfaces ge-0/0/1 brief** command.

```
user@host> show interfaces ge-0/0/1 brief
```

```
Physical interface: ge-0/0/1, Enabled, Physical link is Down
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Speed: 1000mbps, Loopback:
Disabled, Source filtering: Disabled, Flow control: Disabled, Auto-negotiation:
Enabled, Remote fault: Online
  Device flags   : Present Running Down
  Interface flags: Hardware-Down SNMP-Traps Internal: 0x0
  Link flags     : None

  Logical interface ge-0/0/1.0
    Flags: Device-Down SNMP-Traps 0x0 Encapsulation: Ethernet-Bridge
    Security: Zone: Null
    eth-switch
```

Meaning

The sample output shows that the Ethernet switching is configured on interface ge-0/0/1 as expected .

Release History Table

Release	Description
15.1X49-D60	Starting in Junos OS Release 15.1X49-D60 and Junos OS Release 17.3R1, Link Layer Discovery Protocol (LLDP) and LLDP-Media Endpoint Discovery (MFD) are enabled on SRX300, SRX320, SRX340, SRX345, SRX550M and SRX1500 devices.

CHAPTER 35

Configuring Ethernet Port VLANs in Switching Mode on Security Devices

Ethernet Port VLANs in Switching Mode on Security Devices

IN THIS SECTION

Understanding VLAN Retagging on Security Devices

VLAN retagging is not supported from Junos OS Release 15.1X49-D40 to Junos OS Release 15.1X49-D60.

Starting in Junos OS Release 15.1X49-D70, VLAN retagging in switching mode is supported on SRX300, SRX320, SRX340, SRX345, and SRX550M devices.

Starting in Junos OS Release 15.1X49-D80, VLAN retagging in switching mode is supported on SRX1500 devices.

To support VLAN retagging on SRX Series devices, configure **vlan-rewrite** in transparent mode and configure **swap** in switching mode.

The VLAN identifier in packets arriving on a Layer 2 trunk port can be rewritten or *retagged* with a different internal VLAN identifier. VLAN retagging is a symmetric operation; upon exiting the same trunk port, the retagged VLAN identifier is replaced with the original VLAN identifier. VLAN retagging provides a way to selectively screen incoming packets and redirect them to a firewall or other security device without affecting other VLAN traffic.

VLAN retagging can be applied only to interfaces configured as Layer 2 trunk interfaces. These interfaces can include redundant Ethernet interfaces in a Layer 2 transparent mode within a chassis cluster configuration.

> **NOTE:** If a trunk port is configured for VLAN retagging, untagged packets received on the port are not assigned a VLAN identifier with the VLAN retagging configuration. To configure a VLAN identifier for untagged packets received on the physical interface, use the **native-vlan-id** statement.

To configure VLAN retagging for a Layer 2 trunk interface, specify a one-to-one mapping of the following:

- Incoming VLAN identifier—VLAN identifier of the incoming packet that is to be retagged. This VLAN identifier must not be the same VLAN identifier configured with the **native-vlan-id** statement for the trunk port.
- Internal VLAN identifier—VLAN identifier for the retagged packet. This VLAN identifier must be in the VLAN identifier list for the trunk port and must not be the same VLAN identifier configured with the **native-vlan-id** statement for the trunk port.

SEE ALSO

Ethernet Switching and Layer 2 Transparent Mode Overview | 41

Example: Configuring Layer 2 Logical Interfaces on Security Devices | 963

Configuring VLAN Retagging on a Layer 2 Trunk Interface of a Security Device

VLAN retagging is a feature that works on IEEE standard 802.1Q virtual LAN tagging (VLAN tagging. VLAN retagging for SRX1500 devices is an enterprise style of VLAN retagging, in which a single command is sufficient on top of normal trunk configuration.

1. Create a Layer 2 trunk interface.

   [edit]

   `user@host#` **set interfaces ge-3/0/0 unit 0 family ethernet-switching interface-mode trunk vlan members 1–10**

2. Configure VLAN retagging.

   [edit]

   `user@host#` **set interfaces ge-3/0/0 unit 0 family ethernet-switching vlan-rewrite translate 11 2**

Example: Configuring a Guest VLAN on a Security Device

IN THIS SECTION

This example shows how to configure a guest VLAN for limited network access or for Internet-only access to avoid compromising a company's security.

Guest VLANs are not supported from Junos OS Release 15.1X49-D40 to Junos OS Release 15.1X49-D60.

Requirements

Before you begin, verify that the interfaces that will be used are in switch mode. See "Example: Configuring Switching Modes on Security Devices" on page 996 and "Understanding Switching Modes on Security Devices" on page 988.

Overview

In this example, you configure a VLAN called visitor-vlan with a VLAN ID of 300. Then you set protocols and configure visitor-vlan as the guest VLAN.

Configuration

Step-by-Step Procedure

To configure a guest VLAN:

1. Configure a VLAN.

```
[edit]
user@host# set vlans visitor-vlan vlan-id 300
```

2. Specify the guest VLAN.

```
[edit]
user@host# set protocols dot1x authenticator interface all guest-vlan visitor-vlan
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show vlans** and **show protocols dot1x** commands.

CHAPTER 36

Configuring Secure Wire on Security Devices

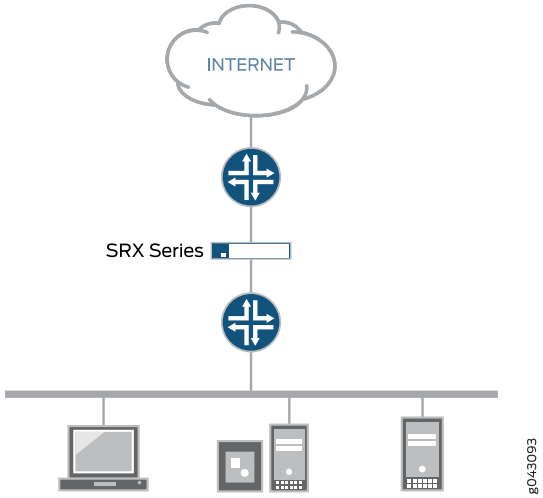
Secure Wire on Security Devices

IN THIS SECTION

Understanding Secure Wire on Security Devices

Traffic that arrives on a specific interface can be forwarded unchanged through another interface. This mapping of interfaces, called secure wire, allows an SRX Series to be deployed in the path of network traffic without requiring a change to routing tables or a reconfiguration of neighboring devices. Figure 53 on page 1006 shows a typical in-path deployment of an SRX Series with secure wire.

Figure 53: SRX Series In-Path Deployment with Secure Wire



Secure wire maps two peer interfaces. It differs from transparent and route modes in that there is no switching or routing lookup to forward traffic. As long as the traffic is permitted by a security policy, a packet arriving on one peer interface is immediately forwarded unchanged out of the other peer interface. There is no routing or switching decision made on the packet. Return traffic is also forwarded unchanged.

Secure wire mapping is configured with the **secure-wire** statement at the [**edit security forwarding-options**] hierarchy level; two Ethernet logical interfaces must be specified. The Ethernet logical interfaces must be configured with **family ethernet-switching** and each pair of interfaces must belong to the VLAN(s). The interfaces must be bound to security zones and a security policy configured to permit traffic between the zones.

This feature is available on Ethernet logical interfaces only; both IPv4 and IPv6 traffic are supported. You can configure interfaces for access or trunk mode. Secure wire supports chassis cluster redundant Ethernet interfaces. This feature does not support security features not supported in transparent mode, including NAT and IPsec VPN.

Secure wire supports Layer 7 features like AppSecure, SSL proxy, UTM and IPS/IDP.

Secure wire is a special case of Layer 2 transparent mode on SRX Series devices that provide point-to-point connections. This means that the two interfaces of a secure wire must ideally be directly connected to Layer 3 entities, such as routers or hosts. Secure wire interfaces can be connected to switches. However, note that a secure wire interface forwards all arriving traffic to the peer interface only if the traffic is permitted by a security policy.

Secure wire can coexist with Layer 3 mode. While you can configure Layer 2 and Layer 3 interfaces at the same time, traffic forwarding occurs independently on Layer 2 and Layer 3 interfaces.

Secure wire can coexist with Layer 2 transparent mode. If both features exist on the same SRX Series device, you need to configure them in different VLANs.

NOTE: Integrated routing and bridging (IRB) interfaces are not supported with secure wire.

SEE ALSO

Example: Simplifying SRX Series Device Deployment with Secure Wire over Access Mode Interfaces | **1008**

Example: Simplifying SRX Series Device Deployment with Secure Wire over Trunk Mode Interfaces | **1015**

Example: Simplifying SRX Series Device Deployment with Secure Wire over Aggregated Interface Member Links | **1019**

Example: Simplifying Chassis Cluster Deployment with Secure Wire over Redundant Ethernet Interfaces | **1025**

Example: Simplifying SRX Series Device Deployment with Secure Wire over Access Mode Interfaces

IN THIS SECTION

If you are connecting an SRX Series device to other network devices, you can use secure wire to simplify the device deployment in the network. No changes to routing or forwarding tables on the SRX Series device and no reconfiguration of neighboring devices is needed. Secure wire allows traffic to be forwarded unchanged between specified access mode interfaces on an SRX Series device as long as it is permitted by security policies or other security features. Follow this example if you are connecting an SRX Series device to other network devices through access mode interfaces.

This example shows how to configure a secure wire mapping for two access mode interfaces. This configuration applies to scenarios where user traffic is not VLAN tagged.

Requirements

No special configuration beyond device initialization is required before configuring this feature.

Overview

This example configures the secure wire access-sw that maps interface ge-0/0/3.0 to interface ge-0/0/4.0. The two peer interfaces are configured for access mode. The VLAN ID 10 is configured for the vlan-10 and the access mode interfaces.

NOTE: A specific VLAN ID must be configured for a VLAN.

Topology

Figure 54 on page 1009 shows the access mode interfaces that are mapped in secure wire access-sw.

Figure 54: Secure Wire Access Mode Interfaces

Configuration

CLI Quick Configuration

NOTE: Starting in Junos OS Release 15.1X49-D10 and Junos OS Release 17.3R1, some Layer 2 CLI configuration statements are enhanced, and some commands are changed. For detailed information about the modified hierarchies, see "Enhanced Layer 2 CLI Configuration Statement and Command Changes for Security Devices" on page 72.

The configuration statements shown below are for Junos OS Release 15.1X49-D10 or higher and Junos OS Release 17.3R1.

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set vlans vlan-10 vlan-id 10
set interfaces ge-0/0/3 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/3 unit 0 family ethernet-switching vlan members 10
set interfaces ge-0/0/4 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/4 unit 0 family ethernet-switching vlan members 10
set security forwarding-options secure-wire access-sw interface [ge-0/0/3.0 ge-0/0/4.0]
set security zones security-zone trust interfaces ge-0/0/3.0
set security zones security-zone untrust interfaces ge-0/0/4.0
set security address-book book1 address mail-untrust 203.0.113.1
set security address-book book1 attach zone untrust
set security address-book book2 address mail-trust 192.168.1.1
```

```
set security address-book book2 attach zone trust
set security policies from-zone trust to-zone untrust policy permit-mail match source-address mail-trust
set security policies from-zone trust to-zone untrust policy permit-mail match destination-address mail-untrust
set security policies from-zone trust to-zone untrust policy permit-mail match application junos-mail
set security policies from-zone trust to-zone untrust policy permit-mail then permit
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure a secure wire mapping for access mode interfaces:

1. Configure the VLAN.

```
[edit vlans vlan-10]
user@host# set vlan-id 10
```

2. Configure the access mode interfaces.

```
[edit interfaces ]
user@host# set ge-0/0/3 unit 0 family ethernet-switching interface-mode access
user@host# set ge-0/0/4 unit 0 family ethernet-switching interface-mode access
user@host# set ge-0/0/3 unit 0 family ethernet-switching vlan members 10
user@host# set ge-0/0/4 unit 0 family ethernet-switching vlan members 10
```

3. Configure the secure wire mapping.

```
[edit security forwarding-options]
user@host# set secure-wire access-sw interface [ge-0/0/3.0 ge-0/0/4.0]
```

4. Configure security zones.

```
[edit security zones]
user@host# set security-zone trust interfaces ge-0/0/3.0
user@host# set security-zone untrust interfaces ge-0/0/4.0
```

5. Create address book entries. Attach security zones to the address books.

```
[edit security address-book book1]
```

```
user@host# set address mail-untrust 203.0.113.1
user@host# set attach zone untrust
```

```
[edit security address-book book1]
user@host# set address mail-trust 192.168.1.1
user@host# set attach zone trust
```

6. Configure a security policy to permit mail traffic.

```
[edit security policies from-zone trust to-zone untrust]
user@host# set policy permit-mail match source-address mail-trust
user@host# set policy permit-mail match destination-address mail-untrust
user@host# set policy permit-mail match application junos-mail
user@host# set policy permit-mail then permit
```

Results

From configuration mode, confirm your configuration by entering the **show vlans**, **show interfaces**, **show security forwarding-options**, and **show security zones** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show vlans
   vlan-10 {
      vlan-id 10;
   }
   user@host# show interfaces
   ge-0/0/3 {
      unit 0 {
         family ethernet-switching {
            interface-mode access;
            vlan {
               members 10;
            }
         }
      }
   }
   ge-0/0/4 {
      unit 0 {
         family ethernet-switching {
            interface-mode access;
            vlan {
               members 10;
```

```
            }
          }
        }
    }
    user@host# show security forwarding-options
    secure-wire {
        access-sw {
            interface [ ge-0/0/3.0 ge-0/0/4.0 ];
        }
    }
user@host# show security zones
    security-zone trust {
        interfaces {
            ge-0/0/3.0;
        }
    }
    security-zone untrust {
        interfaces {
            ge-0/0/4.0;
        }
    }
user@host# show security policies
from-zone trust to-zone untrust {
    policy permit-mail {
        match {
            source-address mail-trust;
            destination-address mail-untrust;
            application junos-mail;
        }
        then {
            permit;
        }
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying Secure Wire Mapping

Purpose

Verify the secure wire mapping.

Action

From operational mode, enter the **show security forwarding-options secure-wire** command.

```
user@host> show security forward-options secure-wire
```

```
Secure wire                    Interface       Link   Interface       Link
access-sw                      ge-0/0/3.0      down   ge-0/0/4.0      down
Total secure wires: 1
```

Verifying the VLAN

Purpose

Verify the VLAN.

Action

From operational mode, enter the **show vlans vlan-10** command.

```
user@host> show vlans vlan-10
```

```
Routing instance        VLAN name            Tag     Interfaces
default-switch          vlan-10               10          ge-0/0/3.0
                                                          ge-0/0/4.0
```

Verifying Policy Configuration

Purpose

Verify information about security policies..

Action

From operational mode, enter the **show security policies detail** command.

```
user@host> show security policies detail
```

```
Default policy: deny-all
Pre ID default policy: permit-all
Policy: permit-mail, action-type: permit, State: enabled, Index: 4, Scope Policy:
 0
  Policy Type: Configured
  Sequence number: 1
  From zone: trust, To zone: untrust
  Source vrf group:
    any
  Destination vrf group:
    any
  Source addresses:
    mail-trust(book2): 192.168.1.1/32
  Destination addresses:
    mail-untrust(book1): 203.0.113.1/32
  Application: junos-mail
    IP protocol: tcp, ALG: 0, Inactivity timeout: 1800
      Source port range: [0-0]
      Destination ports: 25
  Per policy TCP Options: SYN check: No, SEQ check: No, Window scale: No
```

SEE ALSO

Understanding Secure Wire on Security Devices | 1006

Ethernet Switching and Layer 2 Transparent Mode Overview | 41

Example: Simplifying SRX Series Device Deployment with Secure Wire over Trunk Mode Interfaces

IN THIS SECTION

If you are connecting an SRX Series device to other network devices, you can use secure wire to simplify the device deployment in the network. No changes to routing or forwarding tables on the SRX Series device and no reconfiguration of neighboring devices is needed. Secure wire allows traffic to be forwarded unchanged between specified trunk mode interfaces on an SRX Series device as long as it is permitted by security policies or other security features. Follow this example if you are connecting an SRX Series device to other network devices through trunk mode interfaces.

Requirements

No special configuration beyond device initialization is required before configuring this feature.

Overview

This example configures the secure wire trunk-sw that maps interface ge-0/1/0.0 to interface ge-0/1/1.0. The two peer interfaces are configured for trunk mode and carry user traffic tagged with VLAN IDs from 100 to 102. The VLAN ID list 100-102 is configured for the VLAN vlan-100 and the trunk mode interfaces.

> **NOTE:** A specific VLAN ID must be configured for a VLAN.

Topology

Figure 55 on page 1016 shows the trunk mode interfaces that are mapped in secure wire trunk-sw.

Figure 55: Secure Wire Trunk Mode Interfaces

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set vlans vlan-100 vlan members 100-102
set interfaces ge-0/1/0 unit 0 family ethernet-switching interface-mode trunk vlan members 100-102
set interfaces ge-0/1/1 unit 0 family ethernet-switching interface-mode trunk vlan members 100-102
set security forwarding-options secure-wire trunk-sw interface [ge-0/1/0.0 ge-0/1/1.0]
set security zones security-zone trust interfaces ge-0/1/0.0
set security zones security-zone untrust interfaces ge-0/1/1.0
set security policies default-policy permit-all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure a secure wire mapping for trunk mode interfaces:

1. Configure the VLAN.

   ```
   [edit vlans vlan-100]
   user@host# set vlan members 100-102
   ```

2. Configure the trunk mode interfaces.

   ```
   [edit interfaces]
   user@host# set ge-0/1/0 unit 0 family ethernet-switching interface-mode trunk vlan members 100-102
   user@host# set ge-0/1/1 unit 0 family ethernet-switching interface-mode trunk vlan members 100-102
   ```

3. Configure the secure wire mapping.

```
[edit security forwarding-options]
user@host# set secure-wire trunk-sw interface [ge-0/1/0.0 ge-0/1/1.0]
```

4. Configure security zones.

```
[edit security zones]
user@host# set security-zone trust interfaces ge-0/1/0.0
user@host# set security-zone untrust interfaces ge-0/1/1.0
```

5. Configure a security policy to permit traffic.

```
[edit security policies]
user@host# set default-policy permit-all
```

Results

From configuration mode, confirm your configuration by entering the **show vlans**, **show interfaces**, **show security forwarding-options**, and **show security zones** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show vlans
   vlan-100 {
      vlan members 100-102;
   }
   user@host# show interfaces
   ge-0/1/0 {
      unit 0 {
         family ethernet-switching {
            interface-mode trunk;
            vlan members 100-102;
         }
      }
   }
   ge-0/1/1 {
      unit 0 {
         family ethernet-switching {
            interface-mode trunk;
            vlan members 100-102;
         }
      }
   }
```

```
    user@host# show security forwarding-options
    secure-wire trunk-sw {
        interfaces [ge-0/1/0.0 ge-0/1/1.0];
    }
user@host# show security zones
    security-zone trust {
        interfaces {
            ge-0/1/0.0;
        }
    }
    security-zone untrust {
        interfaces {
            ge-0/1/1.0;
        }
    }
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying Secure Wire Mapping

Purpose

Verify the secure wire mapping.

Action

From operational mode, enter the **show security forwarding-options secure-wire** command.

`user@host>` **show security forward-options secure-wire**

```
Secure wire                    Interface       Link   Interface       Link
```

```
trunk-sw                          ge-0/1/0.0      up     ge-0/1/1.0      up
Total secure wires: 1
```

Verifying the VLAN

Purpose

Verify the VLAN.

Action

From operational mode, enter the **show vlans** command.

```
user@host> show vlans
```

```
Routing instance        VLAN name            VLAN ID     Interfaces
default-switch          vlan-100-vlan-0100         100         ge-0/1/0.0
                                                                ge-0/1/1.0
default-switch          vlan-100-vlan-0101         101         ge-0/1/0.0
                                                                ge-0/1/1.0
default-switch          vlan-100-vlan-0102         102         ge-0/1/0.0
                                                                ge-0/1/1.0
```

NOTE: VLANs are automatically expanded, with one VLAN for each VLAN ID in the VLAN ID list.

SEE ALSO

Understanding Secure Wire on Security Devices | 1006

Example: Simplifying SRX Series Device Deployment with Secure Wire over Aggregated Interface Member Links

If you are connecting an SRX Series device to other network devices, you can use secure wire to simplify the device deployment in the network. No changes to routing or forwarding tables on the SRX Series device and no reconfiguration of neighboring devices is needed. Secure wire allows traffic to be forwarded unchanged between specified aggregated interface member links on an SRX Series device as long as it is

permitted by security policies or other security features. Follow this example if you are connecting an SRX Series device to other network devices through aggregated interface member links.

> **NOTE:** LACP is not supported. Secure wire mappings can be configured for member links of link bundles instead of directly mapping aggregated Ethernet interfaces. When the ports, or interfaces on SRX Series device are in trunk mode, the device do not transmit the LACP PDUs and fails the LACP. You must add a native vlan to secure wire interfaces, to bring LACP up.

> **NOTE:** On SRX210, SRX220, SRX240, SRX300, SRX320, SRX340, SRX345, SRX550, and SRX650 devices, when you create an aggregated interface with two or more ports and set the family to Ethernet switching, and if a link in the bundle goes down, the traffic forwarded through the same link will be rerouted two seconds later. This causes an outage for the traffic being sent to the link until reroute is complete.

Requirements

No special configuration beyond device initialization is required before configuring this feature.

Overview

This example configures secure wires for two aggregated Ethernet interface link bundles with two links each. Two separate secure wires ae-link1 and ae-link2 are configured using one link from each aggregated Ethernet link bundle. This static mapping requires that the two link bundles have the same number of links.

For link bundles, all logical interfaces of the secure wire mappings must belong to the same VLAN. VLAN ID 10 is configured for the VLAN vlan-10 and the logical interfaces. All logical interfaces of a link bundle must belong to the same security zone.

> **NOTE:** A specific VLAN ID or VLAN ID list must be configured for a VLAN.

Topology

Figure 56 on page 1021 shows the aggregated interfaces that are mapped in secure wire configurations.

Figure 56: Secure Wire Aggregated Interfaces

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set vlans vlan-10 vlan-id 10
set interfaces ge-0/0/0 unit 0 family ethernet-switching interface-mode access vlan-id 10
set interfaces ge-0/0/1 unit 0 family ethernet-switching interface-mode access vlan-id 10
set interfaces ge-0/1/0 unit 0 family ethernet-switching interface-mode access vlan-id 10
set interfaces ge-0/1/1 unit 0 family ethernet-switching interface-mode access vlan-id 10
set security forwarding-options secure-wire ae-link1-sw interface [ge-0/1/0.0 ge-0/1/1.0]
set security forwarding-options secure-wire ae-link2-sw interface [ge-0/0/0.0 ge-0/0/1.0]
set security zones security-zone trust interfaces ge-0/0/0.0
set security zones security-zone trust interfaces ge-0/1/0.0
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone untrust interfaces ge-0/1/1.0
set security policies default-policy permit-all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure a secure wire mapping for aggregated interface member links:

1. Configure the VLAN.

```
[edit vlans vlan-10]
user@host# set vlan-id10
```

2. Configure the interfaces.

```
[edit interfaces ]
user@host# set ge-0/0/0 unit 0 family ethernet-switching interface-mode access vlan-id 10
```

```
user@host# set ge-0/0/1 unit 0 family ethernet-switching interface-mode access vlan-id 10
user@host# set ge-0/1/0 unit 0 family ethernet-switching interface-mode access vlan-id 10
user@host# set ge-0/1/1 unit 0 family ethernet-switching interface-mode access vlan-id 10
```

3. Configure the secure wire mappings.

```
[edit security forwarding-options]
user@host# set secure-wire ae-link1-sw interface [ ge-0/1/0.0 ge-0/1/1.0 ]
user@host# set secure-wire ae-link2-sw interface [ ge-0/0/0.0 ge-0/0/1.0 ]
```

4. Configure security zones.

```
[edit security zones]
user@host# set security-zone trust interfaces ge-0/0/0.0
user@host# set security-zone trust interfaces ge-0/1/0.0
user@host# set security-zone untrust interfaces ge-0/0/1.0
user@host# set security-zone untrust interfaces ge-0/1/1.0
```

5. Configure a security policy to permit traffic.

```
[edit security policies]
user@host# set default-policy permit-all
```

Results

From configuration mode, confirm your configuration by entering the **show vlans**, **show interfaces**, **show security forwarding-options**, and **show security zones** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show vlans
   vlan-10 {
      vlan-id 10;
   }
   user@host# show interfaces
   ge-0/0/0 {
      unit 0 {
         family ethernet-switching {
            interface-mode access;
            vlan-id 10;
         }
```

```
        }
    }
    ge-0/0/1 {
        unit 0 {
            family ethernet-switching {
                interface-mode access;
                vlan-id 10;
            }
        }
    }
    ge-0/1/0 {
        unit 0 {
            family ethernet-switching {
                interface-mode access;
                vlan-id 10;
            }
        }
    }
    ge-0/1/1{
        unit 0 {
            family ethernet-switching {
                interface-mode access;
                vlan-id 10;
            }
        }
    }
    user@host# show security forwarding-options
    secure-wire ae-link1-sw {
        interfaces [ge-0/1/0.0 ge-0/1/1.0];
    }
    secure-wire ae-link2-sw {
        interfaces [ge-0/0/0.0 ge-0/0/1.0];
    }
user@host# show security zones
    security-zone trust {
        interfaces {
            ge-0/0/0.0;
            ge-0/1/0.0;
        }
    }
    security-zone untrust {
        interfaces {
            ge-0/0/1.0;
            ge-0/1/1.0;
```

```
    }
  }
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying Secure Wire Mapping

Purpose

Verify the secure wire mapping.

Action

From operational mode, enter the **show security forwarding-options secure-wire** command.

`user@host>` **show security forward-options secure-wire**

```
Secure wire                     Interface       Link   Interface       Link

ae-link1-sw                     ge-0/1/0.0      up     ge-0/1/1.0      up
ae-link2-sw                     ge-0/0/0.0      up     ge-0/0/1.0      up
Total secure wires: 2
```

Verifying the VLAN

Purpose

Verify the VLAN.

Action

From operational mode, enter the **show vlans vlan-10** command.

`user@host>` **show vlans vlan-10**

```
Routing instance        VLAN name            VLAN ID     Interfaces
default-switch          vlan-10                 10          ge-0/0/0.0
                                                            ge-0/0/1.0
                                                            ge-0/1/0.0
                                                            ge-0/1/1.0
```

SEE ALSO

Understanding Secure Wire on Security Devices | 1006

Example: Simplifying Chassis Cluster Deployment with Secure Wire over Redundant Ethernet Interfaces

IN THIS SECTION

If you are connecting an SRX Series chassis cluster to other network devices, you can use secure wire to simplify the cluster deployment in the network. No changes to routing or forwarding tables on the cluster and no reconfiguration of neighboring devices is needed. Secure wire allows traffic to be forwarded unchanged between specified redundant Ethernet interfaces on the SRX Series chassis cluster as long as it is permitted by security policies or other security features. Follow this example if you are connecting an SRX Series chassis cluster to other network devices through redundant Ethernet interfaces.

Requirements

Before you begin:

- Connect a pair of the same SRX Series devices in a chassis cluster.
- Configure the chassis cluster node ID and cluster ID.

- Set the number of redundant Ethernet interfaces in the chassis cluster.
- Configure the chassis cluster fabric.
- Configure chassis cluster redundancy group (in this example redundancy group 1 is used).

For more information, see the *Chassis Cluster User Guide for SRX Series Devices*.

Overview

Secure wire is supported over redundant Ethernet interfaces in a chassis cluster. The two redundant Ethernet interfaces must be configured in the same redundancy group. If failover occurs, both redundant Ethernet interfaces must fail over together.

> **NOTE:** Secure wire mapping of redundant Ethernet link aggregation groups (LAGs) are not supported. LACP is not supported.

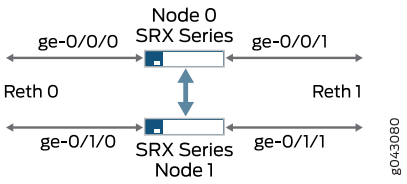
This example configures the secure wire reth-sw that maps ingress interface reth0.0 to egress interface reth1.0. Each redundant Ethernet interface consists of two child interfaces, one on each node of the chassis cluster. The two redundant Ethernet interfaces are configured for access mode. VLAN ID 10 is configured for the VLAN vlan-10 and the redundant Ethernet interfaces.

> **NOTE:** A specific VLAN ID or VLAN ID list must be configured for a VLAN.

Topology

Figure 57 on page 1026 shows the redundant Ethernet interfaces that are mapped in secure wire reth-sw.

Figure 57: Secure Wire Redundant Ethernet Interfaces



Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set vlans vlan-10 vlan-id 10
set interfaces ge-0/0/0 gigether-options redundant-parent reth0
set interfaces ge-0/0/1 gigether-options redundant-parent reth1
set interfaces ge-0/1/0 gigether-options redundant-parent reth0
set interfaces ge-0/1/1 gigether-options redundant-parent reth1
set interfaces reth0 unit 0 family ethernet-switching interface-mode access vlan-id 10
set interfaces reth1 unit 0 family ethernet-switching interface-mode access vlan-id 10
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth1 redundant-ether-options redundancy-group 1
set security forwarding-options secure-wire reth-sw interface [reth0.0 reth1.0]
set security zones security-zone trust interfaces reth0.0
set security zones security-zone untrust interfaces reth1.0
set security policies default-policy permit-all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure a secure wire mapping for chassis cluster redundant Ethernet interfaces:

1. Configure the VLAN.

```
[edit vlans vlan-10]
user@host# set vlan-id 10
```

2. Configure the redundant Ethernet interfaces.

```
[edit interfaces ]
user@host# set ge-0/0/0 gigether-options redundant-parent reth0
user@host# set ge-0/0/1 gigether-options redundant-parent reth1
user@host# set ge-0/1/0 gigether-options redundant-parent reth0
user@host# set ge-0/1/1 gigether-options redundant-parent reth1

user@host#set reth0 unit 0 family ethernet-switching interface-mode access vlan-id 10
user@host#set reth1 unit 0 family ethernet-switching interface-mode access vlan-id 10

user@host# set reth0 redundant-ether-options redundancy-group 1
user@host# set reth1 redundant-ether-options redundancy-group 1
```

3. Configure the secure wire mapping.

```
[edit security forwarding-options]
user@host# set secure-wire reth-sw interface [reth0.0 reth1.0]
```

4. Configure security zones.

```
[edit security zones]
user@host# set security-zone trust interfaces reth0.0
user@host# set security-zone untrust interfaces reth1.0
```

5. Configure a security policy to permit traffic.

```
[edit security policies]
user@host# set default-policy permit-all
```

Results

From configuration mode, confirm your configuration by entering the **show vlans**, **show interfaces**, **show security forwarding-options**, and **show security zones** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show vlans
   vlan-10 {
      vlan-id 10;
   }
   user@host# show interfaces
   ge-0/0/0 {
      gigether-options {
         redundant-parent reth0;
      }
   }
   ge-0/0/1 {
      gigether-options {
         redundant-parent reth1;
      }
   }
   ge-0/1/0 {
      gigether-options {
         redundant-parent reth0;
      }
   }
```

```
    ge-0/1/1 {
        gigether-options {
            redundant-parent reth1;
        }
    }
    reth0 {
        redundant-ether-options {
            redundancy-group 1;
        }
        unit 0 {
            family ethernet-switching {
                interface-mode access;
                vlan-id 10;
            }
        }
    }
    reth1 {
        redundant-ether-options {
            redundancy-group 1;
        }
        unit 0 {
            family ethernet-switching {
                interface-mode access;
                vlan-id 10;
            }
        }
    }
    user@host# show security forwarding-options
    secure-wire reth-sw {
        interfaces [reth0.0 reth1.0];
    }
user@host# show security zones
    security-zone trust {
        interfaces {
            reth0.0;
        }
    }
    security-zone untrust {
        interfaces {
            reth1.0;
        }
    }
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying Secure Wire Mapping

Purpose

Verify the secure wire mapping.

Action

From operational mode, enter the **show security forwarding-options secure-wire** command.

```
user@host> show security forward-options secure-wire
```

```
node0:
--------------------------------------------------------------------------
Secure wire                       Interface       Link   Interface       Link

reth-sw                           reth0.0         up     reth1.0         up

Total secure wires: 1

node1:
--------------------------------------------------------------------------
Secure wire                       Interface       Link   Interface       Link

reth-sw                           reth0.0         up     reth1.0         up

Total secure wires: 1
```

Verifying the VLAN

Purpose

Verify the VLAN.

Action

From operational mode, enter the **show vlan vlan-10** command.

```
user@host> show vlan vlan-10
```

```
Routing instance        VLAN Name             VLAN ID     Interfaces
default-switch          vlan-10                       10         reth0.0
                                                                 reth1.0
```

SEE ALSO

Understanding Secure Wire on Security Devices | 1006

Example: Simplifying Chassis Cluster Deployment with Secure Wire over Aggregated Redundant Ethernet Interfaces | 1031

Example: Simplifying Chassis Cluster Deployment with Secure Wire over Aggregated Redundant Ethernet Interfaces

IN THIS SECTION

If you are connecting an SRX Series chassis cluster to other network devices, you can use secure wire to simplify the cluster deployment in the network. No changes to routing or forwarding tables on the cluster and no reconfiguration of neighboring devices is needed. Secure wire allows traffic to be forwarded unchanged between specified redundant Ethernet interfaces on the SRX Series chassis cluster as long as it is permitted by security policies or other security features. Follow this example if you are connecting an SRX Series chassis cluster to other network devices through aggregated redundant Ethernet interfaces.

NOTE: Secure wires cannot be configured for redundant Ethernet interface link aggregation groups (LAGs). For the secure wire mapping shown in this example, there is no LAG configuration on the SRX Series chassis cluster. Each redundant Ethernet interface consists of two child interfaces, one on each node of the chassis cluster. Users on upstream or downstream devices connected to the SRX Series cluster can configure the redundant Ethernet interface child links in LAGs.

Requirements

Before you begin:

- Connect a pair of the same SRX Series devices in a chassis cluster.
- Configure the chassis cluster node ID and cluster ID.
- Set the number of redundant Ethernet interfaces in the chassis cluster.
- Configure the chassis cluster fabric.
- Configure the chassis cluster redundancy group (in this example, redundancy group 1 is used).

For more information, see the *Chassis Cluster User Guide for SRX Series Devices*.

Overview

This example configures secure wires for four redundant Ethernet interfaces: reth0, reth1, reth2, and reth3. Each redundant Ethernet interface consists of two child interfaces, one on each node of the chassis cluster. All four redundant Ethernet interfaces must be in the same VLAN—in this example, the VLAN is vlan-0. Two of the redundant Ethernet interfaces, reth0.0 and reth2.0, are assigned to the trust zone, while the other two interfaces, reth1.0 and reth3.0, are assigned to the untrust zone.

This example configures the following secure wires:

- reth-sw1 maps interface reth0.0 to interface reth1.0
- reth-sw2 maps interface reth2.0 to reth3.0

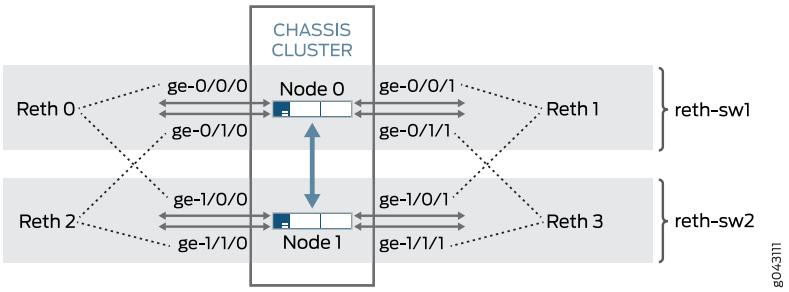
All redundant Ethernet interfaces are configured for access mode. VLAN ID 10 is configured for the VLAN vlan-0 and the redundant Ethernet interfaces.

NOTE: A specific VLAN ID or VLAN ID list must be configured for a VLAN.

Topology

Figure 58 on page 1033 shows the redundant Ethernet interface child links that are mapped in secure wire configurations reth-sw1 and reth-sw2. Each redundant Ethernet interface consists of two child interfaces, one on each node of the chassis cluster.

Figure 58: Secure Wire Redundant Ethernet Interface Child Links



Users on upstream or downstream devices connected to the SRX Series cluster can configure redundant Ethernet interface child links in a LAG as long as the LAG does not span chassis cluster nodes. For example, ge-0/0/0 and ge-0/1/0 and ge-0/0/1 and ge-0/1/1 on node 0 can be configured as LAGs on connected devices. In the same way, ge-1/0/0 and ge-1/1/0 and ge-1/0/1 and ge-1/1/1 on node 1 can be configured as LAGs on connected devices.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set vlans vlan-0 vlan-id 10
set interfaces ge-0/0/0 gigether-options redundant-parent reth0
set interfaces ge-0/0/1 gigether-options redundant-parent reth1
set interfaces ge-0/1/0 gigether-options redundant-parent reth2
set interfaces ge-0/1/1 gigether-options redundant-parent reth3
set interfaces ge-1/0/0 gigether-options redundant-parent reth0
set interfaces ge-1/0/1 gigether-options redundant-parent reth1
set interfaces ge-1/1/0 gigether-options redundant-parent reth2
set interfaces ge-1/1/1 gigether-options redundant-parent reth3
set interfaces reth0 unit 0 family ethernet-switching interface-mode access vlan-id 10
set interfaces reth1 unit 0 family ethernet-switching interface-mode access vlan-id 10
set interfaces reth2 unit 0 family ethernet-switching interface-mode access vlan-id 10
set interfaces reth3 unit 0 family ethernet-switching interface-mode access vlan-id 10
```

```
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth2 redundant-ether-options redundancy-group 1
set interfaces reth3 redundant-ether-options redundancy-group 1
set security forwarding-options secure-wire reth-sw1 interface [reth0.0 reth1.0]
set security forwarding-options secure-wire reth-sw2 interface [reth2.0 reth3.0]
set security zones security-zone trust interfaces reth0.0
set security zones security-zone trust interfaces reth2.0
set security zones security-zone untrust interfaces reth1.0
set security zones security-zone untrust interfaces reth3.0
set security policies default-policy permit-all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure a secure wire mapping for aggregated interface member links:

1. Configure the VLAN.

```
[edit vlans vlan-0]
user@host# set vlan-id 10
```

2. Configure the redundant Ethernet interfaces.

```
[edit interfaces ]
user@host# set ge-0/0/0 gigether-options redundant-parent reth0
user@host# set ge-0/0/1 gigether-options redundant-parent reth1
user@host# set ge-0/1/0 gigether-options redundant-parent reth2
user@host# set ge-0/1/1 gigether-options redundant-parent reth3
user@host# set ge-1/0/0 gigether-options redundant-parent reth0
user@host# set ge-1/0/1 gigether-options redundant-parent reth1
user@host# set ge-1/1/0 gigether-options redundant-parent reth2
user@host# set ge-1/1/1 gigether-options redundant-parent reth3

user@host# set reth0 unit 0 family ethernet-switching interface-mode access vlan-id 10
user@host# set reth1 unit 0 family ethernet-switching interface-mode access vlan-id 10
user@host# set reth2 unit 0 family ethernet-switching interface-mode access vlan-id 10
user@host# set reth3 unit 0 family ethernet-switching interface-mode access vlan-id 10

user@host# set reth0 redundant-ether-options redundancy-group 1
user@host# set reth1 redundant-ether-options redundancy-group 1
user@host# set reth2 redundant-ether-options redundancy-group 1
```

```
user@host# set reth3 redundant-ether-options redundancy-group 1
```

3. Configure the secure wire mappings.

```
[edit security forwarding-options]
user@host# set secure-wire reth-sw1 interface [reth0.0 reth1.0]
user@host# set secure-wire reth-sw2 interface [reth2.0 reth3.0]
```

4. Configure security zones.

```
[edit security zones]
user@host# set security-zone trust interfaces reth0.0
user@host# set security-zone trust interfaces reth2.0

user@host# set security-zone untrust interfaces reth1.0
user@host# set security-zone untrust interfaces reth3.0
```

5. Configure a security policy to permit traffic.

```
[edit security policies]
user@host# set default-policy permit-all
```

Results

From configuration mode, confirm your configuration by entering the **show vlans**, **show interfaces**, **show security forwarding-options**, and **show security zones** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show vlans
   vlan-0 {
      vlan-id 10;
   }
   user@host# show interfaces
   ge-0/0/0 {
      gigether-options {
         redundant-parent reth0;
      }
   }
   ge-0/0/1 {
      gigether-options {
```

```
        redundant-parent reth1;
    }
}
ge-0/1/0 {
    gigether-options {
        redundant-parent reth2;
    }
}
ge-0/1/1 {
    gigether-options {
        redundant-parent reth3;
    }
}
ge-1/0/0 {
    gigether-options {
        redundant-parent reth0;
    }
}
ge-1/0/1 {
    gigether-options {
        redundant-parent reth1;
    }
}
ge-1/1/0 {
    gigether-options {
        redundant-parent reth2;
    }
}
ge-1/1/1 {
    gigether-options {
        redundant-parent reth3;
    }
}
reth0 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family ethernet-switching {
            interface-mode access;
            vlan-id 10;
        }
    }
}
```

```
    reth1 {
        redundant-ether-options {
            redundancy-group 1;
        }
        unit 0 {
            family ethernet-switching {
                interface-mode access;
                vlan-id 10;
            }
        }
    }
    reth2 {
        redundant-ether-options {
            redundancy-group 1;
        }
        unit 0 {
            family ethernet-switching {
                interface-mode access;
                vlan-id 10;
            }
        }
    }
    reth3 {
        redundant-ether-options {
            redundancy-group 1;
        }
        unit 0 {
            family ethernet-switching {
                interface-mode access;
                vlan-id 10;
            }
        }
    }
    user@host# show security forwarding-options
    secure-wire reth-sw1 {
        interfaces [reth0.0 reth1.0];
    }
    secure-wire reth-sw2 {
        interfaces [reth2.0 reth3.0];
    }
user@host# show security zones
    security-zone trust {
        interfaces {
            reth0.0;
```

```
        reth2.0;
      }
    }
    security-zone untrust {
      interfaces {
        reth1.0;
        reth3.0;
      }
    }
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying Secure Wire Mapping

Purpose

Verify the secure wire mapping.

Action

From operational mode, enter the **show security forwarding-options secure-wire** command.

`user@host>` **show security forward-options secure-wire**

```
node0:
--------------------------------------------------------------------------
Secure wire                       Interface       Link   Interface       Link

reth-sw1                          reth0.0         up     reth1.0         up
reth-sw2                          reth2.0         up     reth3.0         up

Total secure wires: 2
```

```
node1:
--------------------------------------------------------------------------
Secure wire                        Interface       Link   Interface       Link

reth-sw1                           reth0.0         up     reth1.0         up
reth-sw2                           reth2.0         up     reth3.0         up

Total secure wires: 2
```

Verifying VLAN

Purpose

Verify the VLAN.

Action

From operational mode, enter the **show vlans vlan-0** command.

```
user@host> show vlans vlan-0
```

```
Routing instance        VLAN name            VLAN ID     Interfaces
default-switch          vlan-0                  10
                                                            reth0.0
                                                            reth1.0
                                                            reth2.0
                                                            reth3.0
```

SEE ALSO

Understanding Secure Wire on Security Devices | 1006

Example: Simplifying Chassis Cluster Deployment with Secure Wire over Redundant Ethernet Interfaces | 1025

CHAPTER 37

Configuring Reflective Relay on Switches

Reflective Relay on Switches

IN THIS SECTION

Understanding Reflective Relay for Use with VEPA Technology

IN THIS SECTION

Virtual Ethernet Port Aggregator (VEPA) technology aggregates packets generated by virtual machines located on the same server and relays them to a physical switch. The physical switch then provides connectivity between the virtual machines located on the server, so the virtual machines do not communicate with one another. Offloading switching activities from a virtual switch to a physical switch reduces the computing overhead on the virtual servers and takes advantage of the security, filtering, and management features of the physical switch. Reflective relay, also known as “hairpin turn,” enables the physical switch to receive aggregated packets from the virtual machines hosted on the server through the VEPA on the downstream port and send those packets out the same downstream port from which the physical switch received them.

Benefits of VEPA and Reflective Relay

- Reduces the computing overhead on the virtual servers and takes advantage of the security, filtering, and management features of the physical switch.

- Enables the physical switch to receive aggregated packets from the virtual machines hosted on the server through the VEPA on the downstream port and send those packets out the same downstream port from which the physical switch received them.

VEPA

Even though virtual machines are capable of sending packets directly to one another, it is more efficient to pass these aggregated packets from the VEPA to a physical switch. The switch can then send any packets destined for a virtual machine located on the same server to the VEPA.

Reflective Relay

Reflective relay, also known as a “hairpin turn” or “hairpin mode,” returns aggregated packets to the VEPA by using the same downstream port that initially delivered the aggregated packets from the VEPA to the switch. Reflective relay must be configured on the interface located on the physical switch that receives aggregated packets, such as VEPA packets, because some of these packets might need to be sent back to the server if they are destined for another virtual machine on the same server.

Reflective relay only occurs in two situations:

- When the destination address of the packet was learned on that downstream port
- When the destination has not yet been learned

Reflective relay does not otherwise change the operation of the switch. If the interface to which the virtual machine is connected and the MAC address of the virtual machine packet are not yet included in the Ethernet switching table for the virtual machine’s associated VLAN, an entry is added. If the source MAC address of an incoming packet under the respective VLAN is not yet present in the Ethernet switching table, the switch floods the packet on all the other ports that are members of the same VLAN, including the port on which the packet arrived.

SEE ALSO

Understanding Bridging and VLANs on Switches | 168

Configuring Reflective Relay on Switches

Configure reflective relay when a switch port must return packets on a downstream port. For example, configure reflective relay when a switch port receives aggregated virtual machine packets from a technology such as virtual Ethernet port aggregator (VEPA). When these packets are passed through the switch, reflective relay allows the switch to send those packets back on the same interface that was used for delivery.

> **NOTE:** This task uses Junos OS for QFX3500 and QFX3600 switches that do not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see "Configuring Reflective Relay on Switches with ELS Support" on page 1050.

Before you begin configuring reflective relay, ensure that you have:

- Configured packet aggregation on the server connected to the port. See your server documentation.
- Configured the port for all VLANs that could be included in aggregated packets..

To configure reflective relay:

1. Configure an Ethernet interface with a port mode of **tagged-acces**:

   ```
   [edit]
   user@switch# set interfaces interface-name unit number family family-type port-mode tagged-access
   ```

   For example:

   ```
   [edit]
   user@switch# set interfaces xe-0/0/2 unit 0 family ethernet-switching port-mode tagged-access
   ```

2. Configure the interface for reflective relay:

   ```
   [edit]
   user@switch# set interfaces interface-name unit number family family-type reflective-relay
   ```

   For example:

   ```
   [edit]
   user@switch# set interfaces xe-0/0/2 unit 0 family ethernet-switching reflective-relay
   ```

3. Configure the interface for the VLANs that exist on the VM server:

   ```
   [edit]
   user@switch# set interfaces interface-name unit number family family-type vlan members vlan-names
   ```

   For example:

```
[edit]
user@switch# set interfaces xe-0/0/2 unit 0 family ethernet-switching vlan members [VLAN_Purple
VLAN_Orange VLAN_Blue]
```

Example: Configuring Reflective Relay for Use with VEPA Technology on QFX Switches

IN THIS SECTION

Reflective relay must be configured on a switch that receives virtual machine aggregated packets, such as Virtual Ethernet Port Aggregator (VEPA) packets, because some of these packets might be sent back to the server destined for another virtual machine on the same server. Reflective relay returns those packets to the original device using the same downstream port that delivered the packets to the switch.

> **NOTE:** This example uses Junos OS for QFX3500 and QFX3600 switches that do not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does support ELS, see "Example: Configuring Reflective Relay for Use with VEPA Technology on QFX Switches with ELS Support" on page 1051.

This example shows how to configure a switch port interface to return packets sent by VEPA on the downstream interface back to the server using the same downstream interface:

Requirements

This example uses the following hardware and software components:

- One QFX3500 switch
- One server
- Junos OS Release 12.1 or later for the QFX Series

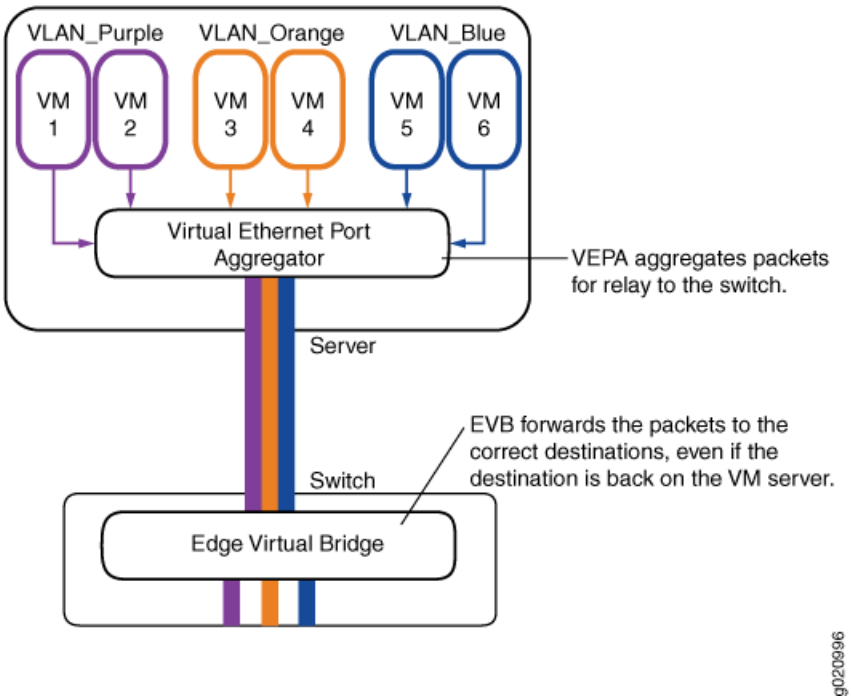
Before you configure reflective relay on a switch port, be sure you have:

- Configured a server with six virtual machines, VM 1 through VM 6.
- Configured the server with three VLANS named VLAN_Purple, VLAN_Orange, and VLAN_Blue and added two virtual machines to each VLAN.
- Configured the same three VLANs named VLAN_Purple, VLAN_Orange, and VLAN_Blue on one interface.
- Installed and configured VEPA to aggregate the virtual machine packets.

Overview and Topology

In this example, illustrated in Figure 59 on page 1046, a switch is connected to one server that is hosting six virtual machines and is configured with a VEPA for aggregating packets. The server's six virtual machines are VM 1 through VM 6, and each virtual machine belongs to one of the three server VLANs, VLAN_Purple, VLAN_Orange, or VLAN_Blue. Instead of the server directly passing packets between virtual machines, packets from any of the three VLANS that are destined for another one of the three VLANs are aggregated using VEPA technology and passed to the switch for processing. You must configure the switch port to accept these aggregated packets on the downstream interface and to return appropriate packets to the server on the same downstream interface after they are processed. Figure 59 on page 1046 shows the topology for this example.

Figure 59: Reflective Relay Topology



In this example, you configure the physical Ethernet switch port interface for tagged-access port mode and reflective relay. Configuring tagged-access port mode allows the interface to accept VLAN tagged packets. Configuring reflective relay allows the downstream port to return those packets on the same interface. Table 132 on page 1046 shows the components used in this example.

Table 132: Components of the Topology for Configuring Reflective Relay

Component	Description
QFX3500 switch	Switch that supports reflective relay.
xe-0/0/2	Switch interface to the server.
Server	Server with virtual machines and VEPA technology.
Virtual machines	Six virtual machines located on the server: V1, V2, V3, V4, V5, and V6.
VLANs	Three VLANs: VLAN_Purple, VLAN_Orange, and VLAN_Blue. Each VLAN has two virtual machine members.
VEPA	Virtual Ethernet port aggregator that aggregates virtual machine packets on the server before the resulting single stream is transmitted to the switch.

Configuration

IN THIS SECTION

To configure reflective relay, perform these tasks:

Configuring Reflective Relay on the Port

CLI Quick Configuration

To quickly configure reflective relay, copy the following commands and paste them into the switch window:

```
[edit]
set interfaces xe-0/0/2 unit 0 family ethernet-switching port-mode tagged-access
set interfaces xe-0/0/2 unit 0 family ethernet-switching reflective-relay
set interfaces xe-0/0/2 unit 0 family ethernet-switching vlan members [VLAN_Blue VLAN_Orange VLAN_Purple]
```

Step-by-Step Procedure

To configure reflective relay:

1. Configure the tagged-access port mode on the interface:

   > **NOTE:** Configure the port mode as tagged-access otherwise you will receive an error when you commit the configuration.

   ```
   [edit]
   user@switch# set interfaces xe-0/0/2 unit 0 family ethernet-switching port-mode tagged-access
   ```

2. Configure reflective relay on the interface to allow it to both accept and send packets:

   ```
   [edit]
   user@switch# set interfaces xe-0/0/2 unit 0 family ethernet-switching reflective-relay
   ```

3. Configure the interface for the three VLANs on the server:

```
[edit]
user@switch# set interfaces xe-0/0/2 unit 0 family ethernet-switching vlan members [VLAN_Purple
VLAN_Orange VLAN_Blue]
```

Results

Check the results of the configuration:

```
[edit interfaces xe-0/0/2]
user@switch# show
unit 0 {
        family ethernet-switching {
        port-mode tagged-access;
        reflective-relay;
        vlan {
         members [ VLAN_Purple VLAN_Orange VLAN_Blue ];
        }
    }
}
```

Verification

IN THIS SECTION

To confirm that reflective relay is enabled and working correctly, perform these tasks:

Verifying That Reflective Relay Is Enabled and Working Correctly

Purpose

Verify that reflective relay is enabled and working correctly.

Action

Use the **show ethernet-switching interfaces detail** command to display the reflective relay status:

```
user@switch> show ethernet-switching interfaces xe-0/0/2 detail
```

```
Interface: xe-0/0/2, Index: 66, State: down, Port mode: Tagged-access
Reflective Relay Status: Enabled
Ether type for the interface: 0x8100
VLAN membership:
    VLAN_Purple, 802.1Q Tag: 450, tagged, unblocked
    VLAN_Orange, 802.1Q Tag: 460, tagged, unblocked
    VLAN_Blue, 802.1Q Tag: 470, tagged, unblocked
Number of MACs learned on IFL: 0
```

Confirm that reflective relay is working by sending a Layer 2 broadcast message from one virtual machine to another virtual machine located on the same VLAN. Check the switch to verify that the switch sends the packets back on the same interface on which they were received. One way to check this is to set up port mirroring on the switch interface, connect a traffic generator to the mirrored interface, and use the traffic generator to examine packets.

Alternatively, if you do not have a traffic generator available, you can send traffic between two virtual machines with FTP, Telnet, or SSH, while running the **tcpdump** utility on the receiver virtual machine port to capture reflected packets.

Meaning

The reflective relay status is **Enabled**, meaning that interface **xe-0/0/2** is configured for the tagged-access port mode, which accepts VLAN-tagged packets, and for reflective relay, which accepts and returns packets on the same interface.

When the traffic generator shows packets arriving at the switch and returning to the server on the same interface, reflective relay is working.

SEE ALSO

Configuring VLANs on Switches | 182

Configuring Reflective Relay on Switches with ELS Support

Configure reflective relay when a switch port must return packets on a downstream port. For example, configure reflective relay when a switch port receives aggregated virtual machine packets from a technology such as virtual Ethernet port aggregator (VEPA). When these packets are passed through the switch, reflective relay allows the switch to send those packets back on the same interface that was used for delivery.

> **NOTE:** This task uses Junos OS for QFX3500 and QFX3600 switches that supports the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does not support ELS, see "Configuring Reflective Relay on Switches" on page 1043.

Before you begin configuring reflective relay, ensure that you have:

- Configured packet aggregation on the server connected to the port. See your server documentation.
- Configured the port for all VLANs that could be included in aggregated packets..

To configure reflective relay:

1. Configure an Ethernet interface with an interface mode of **trunk**:

   ```
   [edit]
   user@switch# set interfaces interface-name unit number family family-type interface-mode trunk
   ```

   For example:

   ```
   [edit]
   user@switch# set interfaces xe-0/0/2 unit 0 family ethernet-switching interface-mode trunk
   ```

2. Configure the interface for reflective relay:

   ```
   [edit]
   user@switch# set interfaces interface-name unit number family family-type reflective-relay
   ```

   For example:

   ```
   [edit]
   user@switch# set interfaces xe-0/0/2 unit 0 family ethernet-switching reflective-relay
   ```

3. Configure the interface for the VLANs that exist on the VM server:

   ```
   [edit]
   user@switch# set interfaces interface-name unit number family family-type vlan members vlan-names
   ```

   For example:

```
[edit]
user@switch# set interfaces xe-0/0/2 unit 0 family ethernet-switching vlan members [VLAN_Purple
VLAN_Orange VLAN_Blue]
```

Example: Configuring Reflective Relay for Use with VEPA Technology on QFX Switches with ELS Support

IN THIS SECTION

Reflective relay must be configured on a switch that receives virtual machine aggregated packets, such as Virtual Ethernet Port Aggregator (VEPA) packets, because some of these packets might be sent back to the server destined for another virtual machine on the same server. Reflective relay returns those packets to the original device using the same downstream port that delivered the packets to the switch.

> **NOTE:** This example uses Junos OS for QFX3500 and QFX3600 switches with support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does not support ELS, see "Example: Configuring Reflective Relay for Use with VEPA Technology on QFX Switches" on page 1044. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50.

This example shows how to configure a switch port interface to return packets sent by VEPA on the downstream interface back to the server using the same downstream interface:

Requirements

This example uses the following hardware and software components:

- One QFX3500 switch
- One server
- Junos OS Release 12.1 or later for the QFX Series

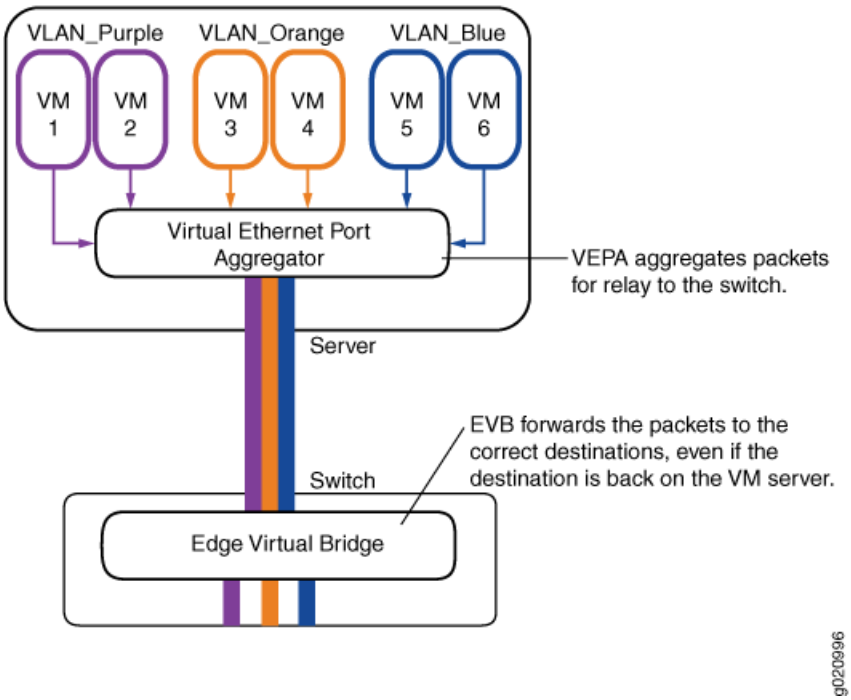
Before you configure reflective relay on a switch port, be sure you have:

- Configured a server with six virtual machines, VM 1 through VM 6.
- Configured the server with three VLANS named VLAN_Purple, VLAN_Orange, and VLAN_Blue and added two virtual machines to each VLAN.
- Configured the same three VLANs named VLAN_Purple, VLAN_Orange, and VLAN_Blue on one interface.
- Installed and configured VEPA to aggregate the virtual machine packets.

Overview and Topology

In this example, illustrated in Figure 60 on page 1053, a switch is connected to one server that is hosting six virtual machines and is configured with a VEPA for aggregating packets. The server's six virtual machines are VM 1 through VM 6, and each virtual machine belongs to one of the three server VLANs, VLAN_Purple, VLAN_Orange, or VLAN_Blue. Instead of the server directly passing packets between virtual machines, packets from any of the three VLANS that are destined for another one of the three VLANs are aggregated using VEPA technology and passed to the switch for processing. You must configure the switch port to accept these aggregated packets on the downstream interface and to return appropriate packets to the server on the same downstream interface after they are processed. Figure 60 on page 1053 shows the topology for this example.

Figure 60: Reflective Relay Topology



In this example, you configure the physical Ethernet switch port interface for trunk interface mode and reflective relay. Configuring trunk port mode allows the interface to accept VLAN tagged packets. Configuring reflective relay allows the downstream port to return those packets on the same interface. Table 133 on page 1053 shows the components used in this example.

Table 133: Components of the Topology for Configuring Reflective Relay

Component	Description
QFX3500 switch	Switch that supports reflective relay. .
xe-0/0/2	Switch interface to the server.
Server	Server with virtual machines and VEPA technology.
Virtual machines	Six virtual machines located on the server: V1, V2, V3, V4, V5, and V6.
VLANs	Three VLANs: VLAN_Purple, VLAN_Orange, and VLAN_Blue. Each VLAN has two virtual machine members.
VEPA	Virtual Ethernet port aggregator that aggregates virtual machine packets on the server before the resulting single stream is transmitted to the switch.

Configuration

IN THIS SECTION

To configure reflective relay, perform these tasks:

Configuring Reflective Relay on the Port

CLI Quick Configuration

To quickly configure reflective relay, copy the following commands and paste them into the switch window:

```
[edit]
```

set interfaces xe-0/0/2 unit 0 family ethernet-switching interface-mode trunk

set interfaces xe-0/0/2 unit 0 family ethernet-switching reflective-relay

set interfaces xe-0/0/2 unit 0 family ethernet-switching vlan members [VLAN_Blue VLAN_Orange VLAN_Purple]

Step-by-Step Procedure

To configure reflective relay:

1. Configure the trunk interface mode on the interface:

   ```
   [edit]
   user@switch# set interfaces xe-0/0/2 unit 0 family ethernet-switching interface-mode trunk
   ```

2. Configure reflective relay on the interface to allow it to both accept and send packets:

   ```
   [edit]
   user@switch# set interfaces xe-0/0/2 unit 0 family ethernet-switching reflective-relay
   ```

3. Configure the interface for the three VLANs on the server:

   ```
   [edit]
   user@switch# set interfaces xe-0/0/2 unit 0 family ethernet-switching vlan members [VLAN_Purple VLAN_Orange VLAN_Blue]
   ```

Results

Check the results of the configuration:

```
[edit interfaces xe-0/0/2]
user@switch# show
unit 0 {
        family ethernet-switching {
        interface-mode trunk;
        reflective-relay;
        vlan {
         members [ VLAN_Purple VLAN_Orange VLAN_Blue ];
        }
    }
}
```

Verification

IN THIS SECTION

To confirm that reflective relay is enabled and working correctly, perform these tasks:

Verifying That Reflective Relay Is Enabled and Working Correctly

Purpose

Verify that reflective relay is enabled and working correctly.

Action

Use the **show ethernet-switching interfaces detail** command to display the reflective relay status:

`user@switch>` **show ethernet-switching interfaces xe-0/0/2 detail**

```
Interface: xe-0/0/2, Index: 66, State: down, Interface mode: Trunk
Reflective Relay Status: Enabled
Ether type for the interface: 0x8100
VLAN membership:
    VLAN_Purple, 802.1Q Tag: 450, tagged, unblocked
    VLAN_Orange, 802.1Q Tag: 460, tagged, unblocked
    VLAN_Blue, 802.1Q Tag: 470, tagged, unblocked
```

```
Number of MACs learned on IFL: 0
```

Confirm that reflective relay is working by sending a Layer 2 broadcast message from one virtual machine to another virtual machine located on the same VLAN. Check the switch to verify that the switch sends the packets back on the same interface on which they were received. One way to check this is to set up port mirroring on the switch interface, connect a traffic generator to the mirrored interface, and use the traffic generator to examine packets.

Alternatively, if you do not have a traffic generator available, you can send traffic between two virtual machines with FTP, Telnet, or SSH, while running the **tcpdump** utility on the receiver virtual machine port to capture reflected packets.

Meaning

The reflective relay status is **Enabled**, meaning that interface **xe-0/0/2** is configured for the trunk interface mode, which accepts VLAN-tagged packets, and for reflective relay, which accepts and returns packets on the same interface.

When the traffic generator shows packets arriving at the switch and returning to the server on the same interface, reflective relay is working.

SEE ALSO

Configuring Port Mirroring

38
CHAPTER

Configuring Edge Virtual Bridging

Edge Virtual Bridging

IN THIS SECTION

Understanding Edge Virtual Bridging for Use with VEPA Technology on EX Series Switches

IN THIS SECTION

Servers using virtual Ethernet port aggregator (VEPA) do not send packets directly from one virtual machine (VM) to another. Instead, the packets are sent to virtual bridges on an adjacent switch for processing. EX Series switches use edge virtual bridging (EVB) as a virtual bridge to return the packets on the same interface that delivered the packets.

What Is EVB?

EVB is a software capability on a switch running Junos OS that allows multiple virtual machines to communicate with each other and with external hosts in the Ethernet network environment.

What Is VEPA?

VEPA is a software capability on a server that collaborates with an adjacent, external switch to provide bridging support between multiple virtual machines and external networks. The VEPA collaborates with the adjacent switch by forwarding all VM-originated frames to the adjacent switch for frame processing and frame relay (including hairpin forwarding) and by steering and replicating frames received from the VEPA uplink to the appropriate destinations.

Why Use VEPA Instead of VEB?

Even though virtual machines are capable of sending packets directly to one another with a technology called virtual Ethernet bridging (VEB), you typically want to use physical switches for switching because VEB uses expensive server hardware to accomplish the task. Instead of using VEB, you can install VEPA on a server to offload switching functionality to an adjacent, less expensive physical switch. Additional advantages of using VEPA include:

- VEPA reduces complexity and allows higher performance at the server.
- VEPA takes advantage of the physical switch's security and tracking features.
- VEPA provides visibility of inter-virtual-machine traffic to network management tools designed for an adjacent bridge.
- VEPA reduces the amount of network configuration required by server administrators, and as a consequence, reduces work for the network administrator.

How Does EVB Work?

EVB uses two protocols, Virtual Station Interface (VSI) Discovery and Configuration Protocol (VDP) and Edge Control Protocol (ECP), to program policies for each individual virtual switch instance—specifically, EVB maintains the following information for each VSI instance:

- VLAN ID
- VSI type
- VSI type version
- MAC address of the server

VDP is used by the VEPA server to propagate VSI information to the switch. This allows the switch to program policies on individual VSIs and supports virtual machine migration by implementing logic to preassociate a VSI with a particular interface.

ECP is a Link Layer Discovery Protocol (LLDP)-like transport layer that allows multiple upper layer protocols to send and receive protocol data units (PDUs). ECP improves upon LLDP by implementing sequencing, retransmission and an ack mechanism, while at the same time remaining lightweight enough to be

implemented on a single-hop network. ECP is implemented in an EVB configuration when you configure LLDP on interfaces that you have configured for EVB. That is, you configure LLDP, not ECP.

How Do I Implement EVB?

You can configure EVB on a switch when that switch is adjacent to a server that includes VEPA technology. In general, this is what you do to implement EVB:

- The network manager creates a set of VSI types. Each VSI type is represented by a VSI type ID and a VSI version--the network manager can deploy one or more VSI versions at any given time.
- The VM manager configures VSI (which is a virtual station interface for a VM that is represented by a MAC address and VLAN ID pair) . To accomplish this, the VM manager queries available VSI type IDs (VTIDs) and creates a VSI instance consisting of a VSI Instance ID and the chosen VTID. This instance is known as VTDB and contains a VSI manager ID, a VSI type ID, a VSI version, and a VSI instance ID.

SEE ALSO

Understanding Bridging and VLANs on Switches | 168

Example: Configuring Edge Virtual Bridging for Use with VEPA Technology on an EX Series Switch | 1062

Configuring Edge Virtual Bridging on an EX Series Switch

Configure edge virtual bridging (EVB) when a switch is connected to a virtual machine (VM) server using virtual Ethernet port aggregator (VEPA) technology. EVB does not convert packets; rather, it ensures that packets from one VM destined for another VM on the same VM server is switched. In other words, when the source and destination of a packet are the same port, EVB delivers the packet properly, which otherwise would not happen.

> **NOTE:** Configuring EVB also enables Virtual Station Interface (VSI) Discovery and Configuration Protocol (VDP).

Before you begin configuring EVB, ensure that you have:

- Configured packet aggregation on the server connected to the port that you will use on the switch for EVB. See your server documentation.
- Configured the EVB interface for all VLANs located on the virtual machines. See “Configuring VLANs for EX Series Switches” on page 183.

> **NOTE:** The port security features MAC move limiting and MAC limiting are supported on interfaces that are configured for EVB; however, the port security features IP source guard, dynamic ARP inspection (DAI), and DHCP snooping are not supported by EVB. For more information about these features, see *Port Security Features*.

To configure EVB on the switch:

1. Configure tagged-access mode for the interfaces on which you will enable EVB:

   ```
   [edit interfaces interface-name]
   user@switch# set unit 0 family ethernet-switching port-mode tagged-access
   ```

2. Enable the Link Layer Discovery Protocol (LLDP) on the interfaces on which you will enable EVB:.

   ```
   [edit protocols]
   user@switch# set lldp interface interface-name
   ```

3. Configure the interfaces for EVB as members of all VLANs located on the virtual machines.

   ```
   [edit protocols]
   user@switch# set vlans vlan-name vlan-id vlan-number
   ```

4. Enable VDP on the interfaces:

   ```
   [edit protocols]
   user@switch# set edge-virtual-bridging vsi-discovery interface interface-name
   ```

5. Define policies for VSI information, including a VSI manager ID, VSI type, VSI version, and VSI instance ID:

   ```
   [edit policy-options]
   user@switch# set vsi-policy policy-name from vsi-manager manager-number vsi-type type-number
   vsi-version version-number vsi-instance instance-number
   user@switch# set vsi-policy policy-name then filter filter-name
   ```

6. Define the firewall filters you mapped to in the previous step. When each incoming packet matches the filter, the count is incremented by 1. Other possible actions are accept and drop.

   ```
   [edit firewall family ethernet-switching]
   user@switch# set filter filter-name term term-name then action
   ```

7. Associate VSI policies with VDP:

```
[edit protocols]
user@switch# set edge-virtual-bridging vsi-discovery vsi-policy policy-name
```

8. Verify that the virtual machine successfully associated with the switch. After successful association of the VSI Profile with the switch interface, verify the learning of the VM's MAC address on MAC-Table or Forwarding database Table. The learn type of the VM's MAC addresses will be VDP, and upon successful shutdown of VM the corresponding MAC-VLAN entry will get flushed out from FDB table otherwise it will never shutdown.

   ```
   admin@host# run show ethernet-switching table
   ```

9. Verify that VSI profiles are being learned at the switch:

   ```
   user@switch# show edge-virtual-bridging vsi-profiles
   ```

10. Check the statistics of ECP packet exchanges between the switch and server:

    ```
    user@switch# show edge-virtual-bridging ecp statistics
    ```

SEE ALSO

Example: Configuring Edge Virtual Bridging for Use with VEPA Technology on an EX Series Switch | 1062

Understanding Edge Virtual Bridging for Use with VEPA Technology on EX Series Switches | 1058

Example: Configuring Edge Virtual Bridging for Use with VEPA Technology on an EX Series Switch

IN THIS SECTION

Virtual machines (VMs) can use a physical switch that is adjacent to the VMs' server to send packets both to other VMs and to the rest of the network when two conditions have been met:

- Virtual Ethernet packet aggregator (VEPA) is configured on the VM server.
- Edge virtual bridging (EVB) is configured on the switch.

This example shows how to configure EVB on the switch so that packets can flow to and from the virtual machines.

Requirements

This example uses the following hardware and software components:

- One EX4500 or EX8200 switch
- Junos OS Release 12.1 or later for EX Series switches

Before you configure EVB on a switch, be sure you have configured the server with virtual machines, the VLANs, and VEPA:

> **NOTE:** The following are the numbers of components used in this example, but you can use fewer or more to configure the feature.

- On the server, configure six virtual machines, VM 1 through VM 6 as shown in Figure 61 on page 1064. See your server documentation.
- On the server, configure three VLANs named VLAN_Purple, VLAN_Orange, and VLAN_Blue, and add two virtual machines to each VLAN. See your server documentation.
- On the server, install and configure VEPA to aggregate the virtual machine packets.
- On the switch, configure one interface with the same three VLANs as the server (VLAN_Purple, VLAN_Orange, and VLAN_Blue). See "Configuring VLANs for EX Series Switches" on page 183.

Overview and Topology

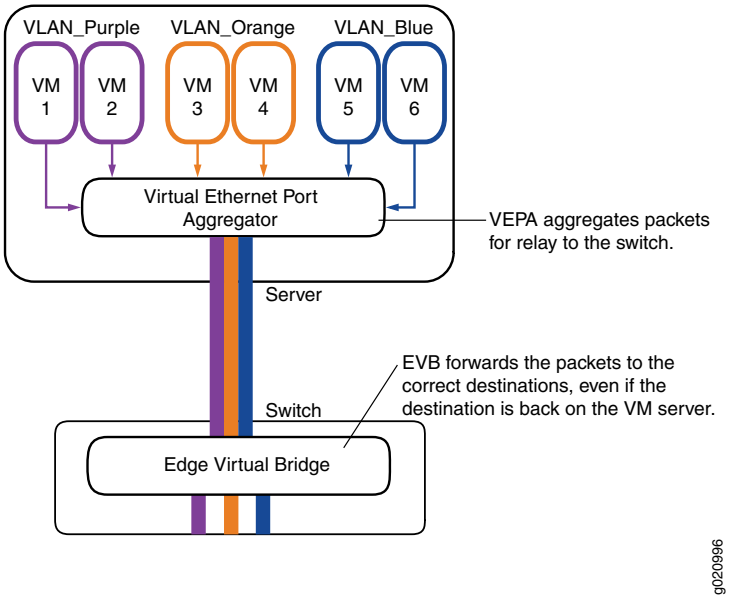
EVB is a software capability that provides multiple virtual end stations that communicate with each other and with external switches in the Ethernet network environment.

This example demonstrates the configuration that takes place on a switch when that switch is connected to a server with VEPA configured. In this example, a switch is already connected to a server hosting six virtual machines (VMs) and configured with VEPA for aggregating packets. The server's six virtual machines are VM 1 through VM 6, and each virtual machine belongs to one of the three server VLANs—VLAN_Purple, VLAN_Orange, or VLAN_Blue. Because VEPA is configured on the server, no two VMs can communicate

directly—all communication between VMs must happen via the adjacent switch. Figure 61 on page 1064 shows the topology for this example.

Edge Virtual Bridging Example Topology

Figure 61: Topology



The VEPA component of the server pushes all packets from any VM, regardless of whether the packets are destined to other VMs on the same server or to any external host, to the adjacent switch. The adjacent switch applies policies to incoming packets based on the interface configuration and then forwards the packets to appropriate interfaces based on the MAC learning table. If the switch has not yet learned a destination MAC, it floods the packet to all interfaces, including the source port on which the packet arrived.

Table 132 on page 1046 shows the components used in this example.

Table 134: Components of the Topology for Configuring EVB

Component	Description
EX Series switch	For a list of switches that support this feature, see *EX Series Switch Software Features Overview* or *EX Series Virtual Chassis Software Features Overview*.
ge-0/0/20	Switch interface to the server.
Server	Server with virtual machines and VEPA technology.
Virtual machines	Six virtual machines located on the server, named VM 1, VM 2, VM 3, VM 4, VM 5, and VM 6.

Table 134: Components of the Topology for Configuring EVB *(continued)*

Component	Description
VLANs	Three VLANs, named VLAN_Purple, VLAN_Orange, and VLAN_Blue. Each VLAN has two virtual machine members.
VEPA	A virtual Ethernet port aggregator (VEPA) is a software capability on a server that collaborates with an adjacent, external switch to provide bridging support between multiple virtual machines and with external networks. The VEPA collaborates with the switch by forwarding all VM-originated frames to the adjacent bridge for frame processing and frame relay (including hairpin forwarding) and by steering and replicating frames received from the VEPA uplink to the appropriate destinations.

NOTE: Configuring EVB also enables Virtual Station Interface (VSI) Discovery and Configuration Protocol (VDP).

Configuration

CLI Quick Configuration

To quickly configure EVB, copy the following commands and paste them into the switch's CLI at the **[edit]** hierarchy level.

```
set interfaces ge-0/0/20 unit 0 family ethernet-switching port-mode tagged-access
set protocols lldp interface ge-0/0/20.0
set vlans vlan_purple interface ge-0/0/20.0
set vlans vlan_orange interface ge-0/0/20.0
set vlans vlan_blue interface ge-0/0/20.0
set protocols edge-virtual-bridging vsi-discovery interface ge-0/0/20.0
set policy-options vsi-policy P1 from vsi-manager 98 vsi-type 998 vsi-version 4 vsi-instance
09b11c53-8b5c-4eeb-8f00-c84ebb0bb998
set policy-options vsi-policy P1 then filter f2
set policy-options vsi-policy P3 from vsi-manager 97 vsi-type 997 vsi-version 3 vsi-instance
09b11c53-8b5c-4eeb-8f00-c84ebb0bb997
set policy-options vsi-policy P3 then filter f3
set firewall family ethernet-switching filter f2 term t1 then accept
set firewall family ethernet-switching filter f2 term t1 then count f2_accept
set firewall family ethernet-switching filter f3 term t1 then accept
set firewall family ethernet-switching filter f3 term t1 then count f3_accept
```

set protocols edge-virtual-bridging vsi-discovery vsi-policy P1

set protocols edge-virtual-bridging vsi-discovery vsi-policy P3

Step-by-Step Procedure

To configure EVB on the switch:

1. Configure tagged-access mode for the interfaces on which you will enable EVB:

   ```
   [edit interfaces ge-0/0/20]
   user@switch# set unit 0 family ethernet-switching port-mode tagged-access
   ```

2. Enable the Link Layer Discovery Protocol (LLDP) on the ports interfaces on which you will enable EVB:

   ```
   [edit protocols]
   user@switch# set lldp interface ge-0/0/20.0
   ```

3. Configure the interface as a member of all VLANs located on the virtual machines.

   ```
   [edit]
   user@switch# set vlans vlan_purple interface ge-0/0/20.0
   user@switch# set vlans vlan_orange interface ge-0/0/20.0
   user@switch# set vlans vlan_blue interface ge-0/0/20.0
   ```

4. Enable the VSI Discovery and Control Protocol (VDP) on the interface:

   ```
   [edit protocols]
   user@switch# set edge-virtual-bridging vsi-discovery interface ge-0/0/20.0
   ```

5. Define policies for VSI information. VSI information is based on a VSI manager ID, VSI type, VSI version, and VSI instance ID:

   ```
   [edit policy-options]
   user@switch# set vsi-policy P1 from vsi-manager 98 vsi-type 998 vsi-version 4 vsi-instance
   09b11c53-8b5c-4eeb-8f00-c84ebb0bb998
   user@switch# set vsi-policy P1 then filter f2
   user@switch# set vsi-policy P3 from vsi-manager 97 vsi-type 997 vsi-version 3 vsi-instance
   09b11c53-8b5c-4eeb-8f00-c84ebb0bb997
   user@switch# set vsi-policy P3 then filter f3
   ```

6. Two VSI policies were defined in the previous step, each of them mapping to different firewall filters. Define the firewall filters:

   ```
   [edit firewall family ethernet-switching]
   ```

```
user@switch# set filter f2 term t1 then accept
user@switch# set filter f2 term t1 then count f2_accept
user@switch# set filter f3 term t1 then accept
user@switch# set filter f3 term t1 then count f3_accept
```

7. Associate VSI policies with VSI-discovery protocol

```
[edit]
user@switch# set protocols edge-virtual-bridging vsi-discovery vsi-policy P1
user@switch# set protocols edge-virtual-bridging vsi-discovery vsi-policy P3
```

Results

```
user@switch# show protocols
```

```
edge-virtual-bridging {
    vsi-discovery {
        interface {
            ge-0/0/20.0;
        }
        vsi-policy {
            P1;
            P3;
        }
    }
}
lldp {
    interface ge-0/0/20.0;
```

```
user@switch# show policy-options
```

```
vsi-policy P1 {
    from {
        vsi-manager 98 vsi-type 998 vsi-version 4 vsi-instance 09b11c53-8b5c-4ee
b-8f00-c84ebb0bb998;
    }
    then {
        filter f2;
    }
}
vsi-policy P3 {
```

```
    from {
        vsi-manager 97 vsi-type 997 vsi-version 3 vsi-instance 09b11c53-8b5c-4ee
b-8f00-c84ebb0bb997;
    }
    then {
        filter f3;
    }
}
```

user@switch# **show vlans**

```
vlan_blue {
    interface {
        ge-0/0/20.0;
    }
}
vlan_orange {
    interface {
        ge-0/0/20.0;
    }
}
vlan_purple {
    interface {
        ge-0/0/20.0;
        interface;
    }
}
```

user@switch# **show firewall**

```
family ethernet-switching {
        filter f2 {
        term t1 {
            then {
                accept;
                count f2_accept;
            }
        }
    }
    filter f3 {
        term t1 {
```

```
            then {
                accept;
                count f3_accept;
            }
        }
    }
}
```

Verification

IN THIS SECTION

To confirm that EVB is enabled and working correctly, perform these tasks:

Verifying That EVB is Correctly Configured

Purpose

Verify that EVB is correctly configured

Action

`user@switch#` **show edge-virtual-bridging**

```
Interface     Forwarding Mode      RTE   Number of VSIs  Protocols
ge-0/0/20.0   Reflective-relay     25    400             ECP, VDP, RTE
```

Meaning

When LLDP is first enabled, an EVB LLDP exchange takes place between switch and server using LLDP. As part of this exchange the following parameters are negotiated: Number of VSIs supported, Forwarding mode, ECP support, VDP support, and Retransmission Timer Exponent (RTE). If the output has values for the negotiated parameters, EVB is correctly configured.

Verifying That the Virtual Machine Successfully Associated With the Switch

Purpose

Verify that the virtual machine successfully associated with the switch. After successful association of VSI Profile with the switch interface, verify the learning of the VM's MAC address on MAC-Table or Forwarding database Table. The learn type of the VM's MAC addresses will be VDP, and upon successful shutdown of VM the corresponding MAC-VLAN entry will get flushed out from FDB table otherwise it will never shutdown.

Action

`user@switch#` **run show ethernet-switching table**

```
Ethernet-switching table: 10 entries, 4 learned
VLAN MAC address      Type  Age  Interfaces
v3  *        Flood        -      All-members
v3      00:02:a6:11:bb:1a    VDP          -      ge-1/0/10.0
v3      00:02:a6:11:cc:1a    VDP          -      ge-1/0/10.0
v3  00:23:9c:4f:70:01    Static       -      Router
v4      *                    Flood        -      All-members
v4      00:02:a6:11:bb:bb    VDP          -      ge-1/0/10.0
v4      00:23:9c:4f:70:01    Static       -      Router
v5      *                    Flood        -      All-members
v5      00:23:9c:4f:70:01    Static       -      Router
v5      52:54:00:d5:49:11    VDP          -      ge-1/0/20.0
```

Verifying That VSI Profiles Are Being Learned at the Switch

Purpose

Verify that VSI profiles are being learned at the switch.

Action

`user@switch#` **show edge-virtual-bridging vsi-profiles**

```
Interface: ge-0/0/20.0
Manager: 97, Type: 997, Version: 3, VSI State: Associate
 Instance: 09b11c53-8b5c-4eeb-8f00-c84ebb0bb997
      MAC                      VLAN
      00:10:94:00:00:04        3
```

Meaning

Whenever VMs configured for VEPA are started at the server, the VMs start sending VDP messages. As part of this protocol VSI profiles are learned at the switch.

If the output has values for Manager, Type, Version, VSI State, and Instance, VSI profiles are being learned at the switch.

SEE ALSO

Configuring Edge Virtual Bridging on an EX Series Switch | **1060**

Understanding Edge Virtual Bridging for Use with VEPA Technology on EX Series Switches | **1058**

39

CHAPTER

Troubleshooting Ethernet Switching

Troubleshooting Ethernet Switching

Problem

Description: Sometimes a MAC address entry in the switch's Ethernet switching table is not updated after the device with that MAC address has been moved from one interface to another on the switch. Typically, the switch does not wait for a MAC address expiration when a MAC move operation occurs. As soon as the switch detects the MAC address on the new interface, it immediately updates the table. Many network devices send a gratuitous ARP packet when switching an IP address from one device to another. The switch updates its ARP cache table after receipt of such gratuitous ARP messages, and then it also updates its Ethernet switching table.

Sometimes silent devices, such as syslog servers or SNMP trap receivers that receive UDP traffic but do not return acknowledgment (ACK) messages to the traffic source, fail to send gratuitous ARP packets when a device moves. If such a move occurs when the system administrator is not available to explicitly clear the affected interfaces by issuing the **clear ethernet-switching table** command, the entry for the moved device in the Ethernet switching table is not updated.

Solution

Set up the switch to handle unattended MAC address switchovers.

1. Reduce the system-wide ARP aging timer. (By default, the ARP aging timer is set at 20 minutes. The range of the ARP aging timer is from 1 through 240 minutes.)

   ```
   [edit system arp]
   user@switch# set aging-timer 3
   ```

2. Set the MAC aging timer to the same value as the ARP timer. (By default, the MAC aging timer is set to 300 seconds. The range is 60 to 1,000,000 seconds.)

   ```
   [edit protocols l2-learning]
   user@switch# set global-mac-table-aging-time 180
   ```

The ARP entry and the MAC address entry for the moved device expire within the times specified by the aging timer values. After the entries expire, the switch sends a new ARP message to the IP address of the device. The device responds to the ARP message, thereby refreshing the entries in the switch's ARP cache table and Ethernet switching table.

RELATED DOCUMENTATION

arp

global-mac-table-aging-time | **1182**

Troubleshooting Ethernet Switching on EX Series Switches

IN THIS SECTION

Troubleshooting issues for Ethernet switching on EX Series switches:

MAC Address in the Switch's Ethernet Switching Table Is Not Updated After a MAC Address Move

Problem

Description: Sometimes a MAC address entry in the switch's Ethernet switching table is not updated after the device with that MAC address has been moved from one interface to another on the switch. Typically, the switch does not wait for a MAC address expiration when a MAC move operation occurs. As soon as the switch detects the MAC address on the new interface, it immediately updates the table. Many network devices send a gratuitous ARP packet when switching an IP address from one device to another. The switch updates its ARP cache table after receipt of such gratuitous ARP messages, and then it also updates its Ethernet switching table. However, sometimes silent devices, such as SYSLOG servers or SNMP Trap receivers that receive UDP traffic but do not return acknowledgement (ACK) messages to the traffic source, do not send gratuitous ARP packets when a device moves. If such a move occurs when the system administrator is not available to explicitly clear the affected interfaces by issuing the **clear ethernet-switching table** command, the entry for the moved device in the Ethernet switching table is not updated.

Solution

Set up the switch to handle unattended MAC address switchovers.

1. Reduce the system-wide ARP aging timer. (By default, the ARP aging timer is set at 20 minutes. In Junos OS Release 9.4 and later, the range of the ARP aging timer is from 1 through 240 minutes.)

   ```
   [edit system arp]
   user@switch# set aging-timer 3
   ```

2. Set the MAC aging timer to the same value as the ARP timer. (By default, the MAC aging timer is set to 300 seconds. The range is 15 to 1,000,000 seconds.)

```
[edit vlans]
user@switch# set vlans sales mac-table-aging-time 180
```

The ARP entry and the MAC address entry for the moved device expire within the times specified by the aging timer values. After the entries expire, the switch sends a new ARP message to the IP address of the device. The device responds to the ARP, thereby refreshing the entries in the switch's ARP cache table and Ethernet switching table

RELATED DOCUMENTATION

arp

mac-table-aging-time | 1261

40
CHAPTER

Configuration Statements

address

List of Syntax

Syntax MX Series and EX Series (dynamic-profiles)

```
address (ip-address | ipv6-address);
```

Syntax QFX Series and QFabric (interfaces)

```
address address {
    arp ip-address (mac | multicast-mac) mac-address <publish>;
    broadcast address;
    destination address;
    destination-profile name;
    reui-64;
    master-only;
    multipoint-destination addressdlci dlci-identifier;
    multipoint-destination address {
        epd-threshold cells;
        inverse-arp;
        oam-liveness {
            up-count cells;
            down-count cells;
        }
        oam-period (disable | seconds);
        shaping {
            (cbr rate |rtvbr peak rate sustained rate burst length |vbr peak rate sustained rate burst length);
            queue-length number;
        }
        vci vpi-identifier.vci-identifier;
    }
    primary;
    preferred;
    (vrrp-group | vrrp-inet6-group) group-number {
        (accept-data | no-accept-data);
        advertise–interval seconds;
        authentication-type authentication;
        authentication-key key;
        fast-interval milliseconds;
        (preempt | no-preempt) {
```

```
            hold-time seconds;
        }
        priority-number number;
        track {
            priority-cost seconds;
            priority-hold-time interface-name {
                interface priority;
                bandwidth-threshold bits-per-second {
                    priority;
                }
            }
            route ip-address/mask routing-instance instance-name priority-cost cost;
        }
        virtual-address [ addresses ];
    }
}
```

MX Series and EX Series (dynamic-profiles)

```
[edit dynamic-profiles profile-name interfaces interface-name unit logical-unit-number family family],
[edit dynamic-profiles profile-name interfaces demux0 unit logical-unit-number family family],
[edit dynamic-profiles profile-name interfaces pp0 unit "$junos-interface-unit" family family],
[edit interfaces interface-name unit logical-unit-number family inet],
[edit logical-systems logical-system-name interfaces interface-name unit logical-unit-number family family]
```

QFX Series and QFabric (interfaces)

```
[edit interfaces interface-name unit logical-unit-number family family],
[edit logical-systems logical-system-name interfaces interface-name unit logical-unit-number family  family]
```

Release Information

Statement introduced in Junos OS Release 9.2.

Support at the **[edit dynamic-profiles *profile-name* interfaces pp0 unit "$junos-interface-unit" family *family*]** hierarchy level introduced in Junos OS Release 10.1.

Statement introduced before Junos OS Release 11.1 for QFX Series switches.

Support at the **[edit interfaces *interface-name* unit *logical-unit-number* family *inet*]** hierarchy level introduced in Junos OS Release 13.2X50-D10 for EX Series switches.

Description

Configure the interface address.

Options

ip-address—IPv4 address of the interface.

ipv6-address—IPv6 address of the interface. When configuring an IPv6 address on a dynamically created interface, use the ***$junos-ipv6-address*** dynamic variable.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring the Protocol Family

Format for Specifying IP Addresses, Network Masks, and Prefixes in Junos OS Configuration Statements

add-attribute-length-in-pdu

Syntax

```
add-attribute-length-in-pdu;
```

Hierarchy Level

```
[edit protocols mvrp]
```

Release Information
Statement introduced in Junos OS Release 11.3 for EX Series switches.

Description
Add an extra byte in protocol data units (PDUs) sent by the Multiple VLAN Registration Protocol (MVRP) in Junos OS Releases 11.3 and later for EX Series switches that do not support the Enhanced Layer 2 Software (ELS). By default, this MVRP version does not include the extra byte. You can add the extra byte in this MVRP version to address an incompatibility issue with the following MVRP versions:

- MVRP in Junos OS Releases 11.2 and earlier, which includes the extra byte.
- MVRP in Junos OS Releases 13.2 and later for EX Series switches with support for ELS, which includes the extra byte.

If this incompatibility issue arises, the MVRP versions that include the extra byte do not recognize PDUs that do not include the extra byte.

You can recognize an MVRP version problem by looking at a switch running an MVRP version that includes the extra byte. Because a switch running an MVRP version that includes the extra byte cannot interpret an unmodified PDU from an MVRP version that does not include the extra byte, the switch will not add VLANs from the MVRP version that does not include the extra byte. When you execute the command **show mvrp statistics** on the MVRP version that includes the extra byte, the values for *Join Empty received* and *Join In received* will incorrectly display zero, even though the value for *MRPDU received* has been increased. Another indication that MVRP is having a version problem is that unexpected VLAN activity, such as multiple VLAN creation, takes place on the switch running the MVRP version that includes the extra byte.

Required Privilege Level
routing—To view this statement in the configuration.
routing control—To add this statement to the configuration.

RELATED DOCUMENTATION

aggregated-ether-options

List of Syntax

Syntax (EX, MX Series)

```
aggregated-ether-options {
    ethernet-switch-profile {
        tag-protocol-id;
    }
    (flow-control | no-flow-control);
    lacp {
        (active | passive);
        admin-key key;
        periodic interval;
        system-id mac-address;
    }
    (link-protection | no-link-protection);
    link-speed speed;
    local-bias;
    logical-interface-fpc-redundancy;
    (loopback | no-loopback);
    mc-ae {
        chassis-id chassis-id;
        events {
            iccp-peer-down {
                force-icl-down;
                prefer-status-control-active;
            }
        }
        init-delay-time seconds;
        mc-ae-id mc-ae-id;
        mode (active-active | active-standby);
        redundancy-group group-id;
        revert-time revert-time;
        status-control (active | standby);
        switchover-mode (non-revertive |revertive);
    }
    minimum-links number;
    system-priority
}
```

Syntax (NFX, QFX Series, EX4600, OCX1100, QFabric)

The **fcoe-lag** and **mc-ae** statements are not supported on OCX Series switches.

```
aggregated-ether-options {
    configured-flow-control {
        rx-buffers (on | off);
        tx-buffers (on | off);
    }
    ethernet-switch-profile {
    tag-protocol-id;
    (fcoe-lag | no-fcoe-lag);
    (flow-control | no-flow-control);
    lacp mode {
        admin-key key;
        periodic interval;
        system-id mac-address;
        force-up;
        }
    }
    (link-protection | no-link-protection);
    link-speed speed;
    local-bias;
    local-minimum-links-threshold threshold-value;
    (loopback | no-loopback);
    mc-ae {
        chassis-id chassis-id;
        mc-ae-id mc-ae-id;
        mode (active-active);
        status-control (active | standby);
    }
    minimum-links number;
    rebalance-periodic;
    resilient-hash;
    source-address-filter filter;
    (source-filtering | no-source-filtering);
}
```

Hierarchy Level (EX Series, QFX Series)

```
[edit interfaces aex]
```

Release Information

Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.
Statement introduced in Junos OS Release 12.3R2.
Statements **fcoe-lag** and **no-fcoe-lag** introduced in Junos OS Release 13.2X52-D10 for the QFX Series.
Statements **force-up**, **lacp**, and **resilient-hash** introduced in Junos OS Release 14.1X53-D10 for the QFX Series.
Statement **local-minimum-links-threshold** introduced in Junos OS Release 14.1X53-D40 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Configure the aggregated Ethernet properties of a specific aggregated Ethernet interface.

NOTE:

- The **fcoe-lag** and **mc-ae** statements are not supported on OCX Series switches.
- The **force-up** statement is not supported on QFX10002 switches.
- The **resilient-hash** statement is not supported on QFX5200, QFX5210, or QFX10002 switches.

The remaining statements are explained separately. See CLI Explorer.

Default

Options are not enabled.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- *Understanding Aggregated Ethernet Interfaces and LACP for Switches*
- *Configuring Aggregated Ethernet LACP (CLI Procedure)*
- *Example: Configuring Link Aggregation with LACP Between a QFX Series Product and an Aggregation Switch*
- *Junos OS Network Interfaces Library for Routing Devices*
- *Example: Configuring Aggregated Ethernet High-Speed Uplinks Between an EX4200 Virtual Chassis Access Switch and an EX4200 Virtual Chassis Distribution Switch*
- *Example: Configuring Aggregated Ethernet High-Speed Uplinks with LACP Between an EX4200 Virtual Chassis Access Switch and an EX4200 Virtual Chassis Distribution Switch*
- *Configuring Aggregated Ethernet Links (CLI Procedure)*
- *Configuring Aggregated Ethernet LACP (CLI Procedure)*
- *Configuring LACP Link Protection of Aggregated Ethernet Interfaces for Switches*
- Configuring Q-in-Q Tunneling on EX Series Switches with ELS Support | **900**
- *Junos OS Ethernet Interfaces Configuration Guide*

autostate-exclude

Syntax

```
autostate-exclude;
```

Hierarchy Level

```
[edit interface interface-name ether-options]
```

Release Information

Statement introduced in Junos OS Release 14.1x53-D40 and Junos OS Release 17.3R1 on QFX5100 switches.

Description

Specify not to include an IRB interface in the state calculation for VLAN members. The default behavior is not to exclude an IRB interface in the state calculation unless all the ports on the interface go down. Because an IRB interface often has multiple ports in a single VLAN, the state calculation for a VLAN member might include a port that is down, possibly resulting in traffic loss. This feature enables you to exclude a trunk or access interface from the state calculation, which results in the IRB interface being marked as down as soon as the port specifically assigned to a VLAN goes down.

IRB interfaces are used to bind specific VLANs to Layer 3 interfaces, enabling a switch to forward packets between those VLANs— without having to configure another device, such as a router, to connect VLANs. In a typical scenario, a port on the interface is assigned to a specific VLAN, while a different port on that interface is assigned to an 802.1Q trunk interface to carry traffic between multiple VLANs, and a third port on that interface is assigned to an access interface used to connect the VLAN to network devices.

To ensure that an interface is marked as down and thereby excluded from the state calculation for VLAN members when the port assigned to the VLAN goes down, configure this statement on the trunk or access interface. The trunk or port interface is automatically excluded from the state calculation of the IRB interface. In this way, when a port assigned to a specified VLAN goes down, the IRB interface assigned to that VLAN is also marked as down.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration

RELATED DOCUMENTATION

Excluding an IRB Interface from State Calculations on a QFX Series Switch | 774

port-mode | **1306**

show ethernet-switching interface | **1481**

bpdu-destination-mac-address

Syntax

```
bpdu-destination-mac-address provider-bridge-group;
```

MX Series and EX Series

```
[edit logical-systems logical-system-name protocols mvrp],
[edit logical-systems logical-system-name routing-instances routing-instance-name protocols mvrp] (for virtual switch
    instance type),
[edit protocols mvrp],
[edit routing-instances routing-instance-name protocols mvrp] (for virtual switch instance type)
```

SRX Series

```
[edit protocols mvrp],
[edit routing-instances routing-instance-name protocols mvrp] (for virtual switch instance type)
```

Release Information

Statement introduced in Junos OS Release 10.1 for MX Series routers.
Statement introduced in Junos OS Release 15.1X49-D70 for SRX Series devices.

Description

For Multiple VLAN Registration Protocol (MVRP) configurations, specifies the multicast address for MVRP. If configured, the provider MVRP multicast MAC address is used; otherwise, the Junos OS uses the customer MVRP multicast MAC address.

Default

By default, the provider MVRP multicast MAC address is used (if configured). Otherwise, the customer MVRP MAC address is used.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- *Example: Configuring Automatic VLAN Administration Using MVRP on MX Series Routers*
- *Configuring Multiple VLAN Registration Protocol (MVRP) to Manage Dynamic VLAN Registration*

bridge-domains

Syntax

```
bridge-domains {
    bridge-domain-name {
        bridge-options {
            ...bridge-options-configuration...
        }
        domain-type bridge;
        interface interface-name;
        no-irb-layer-2-copy;
        no-local-switching;
        routing-interface routing-interface-name;
        vlan-id (all | none | number);
        vlan-id-list [ vlan-id-numbers ];
        vlan-tags outer number inner number;
        bridge-options {
            interface interface-name {
                mac-pinning
                static-mac mac-address;
            }
            interface-mac-limit limit;
            mac-statistics;
            mac-table-size limit;
            no-mac-learning;
        }
    }
}
```

Hierarchy Level

```
[edit],
[edit logical-systems logical-system-name routing-instances routing-instance-name],
[edit routing-instances routing-instance-name]
```

Release Information

Statement introduced in Junos OS Release 8.4.
Support for logical systems added in Junos OS Release 9.6.
Support for the **no-irb-layer-2-copy** statement added in Junos OS Release 10.2.

Description

(MX Series routers only) Configure a domain that includes a set of logical ports that share the same flooding or broadcast characteristics in order to perform Layer 2 bridging.

Options

bridge-domain-name—Name of the bridge domain.

> **NOTE:** You cannot use the slash (/) character as part of the bridge domain name. If you do, the configuration will not commit.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring a Bridge Domain

Configuring a Layer 2 Virtual Switch

bridge-priority

Syntax

bridge-priority *priority*;

Hierarchy Level

[edit logical-systems *logical-system-name* protocols (mstp | rstp)],
[edit logical-systems *logical-system-name* protocols mstp msti *msti-id*],
[edit logical-systems *logical-system-name* protocols vstp vlan *vlan-id*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols (mstp | rstp)],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mstp msti *msti-id*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vstp vlan *vlan-id*],

[edit routing-instances *routing-instance-name* protocols (mstp | rstp)],
[edit routing-instances *routing-instance-name* protocols mstp msti *msti-id*],
[edit routing-instances *routing-instance-name* protocols vstp vlan *vlan-id*]

[edit protocols mstp],
[edit protocols mstp msti *msti-id*],
[edit protocols rstp],
[edit protocols stp],
[edit protocols vstp vlan *vlan-id*]

Release Information

Statement introduced in Junos OS Release 8.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement updated in Junos OS Release 9.4 for EX Series switches to add VSTP support.
Support for logical systems added in Junos OS Release 9.6.
Statement introduced in Junos OS Release 13.2X50-D10 for EX Series switches.

Description

Configures the bridge priority, which determines which bridge is elected as the root bridge. If two bridges have the same path cost to the root bridge, the bridge priority determines which bridge becomes the designated bridge for a LAN segment.

Default

32,768

Options

priority—The bridge priority can be set only in increments of 4096.

Range: 0 through 61,440

Default: 32,768

Required Privilege Level

routing—To view this statement in the configuration.

routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- *Understanding MSTP*
- *Understanding VSTP*
- *Understanding Bridge Priority for Election of Root Bridge and Designated Bridge*
- *Example: Configuring Network Regions for VLANs with MSTP on Switches*
- *show spanning-tree bridge*
- *show spanning-tree interface*

community-vlan

Syntax

```
community-vlan vlan community-vlan-name;
```

Hierarchy Level

```
[edit vlans primary-vlan-name vlan-id primary-vlan-vlan-id]
```

Release Information

Statement introduced in Junos OS Release 14.1X53-D10 for EX Series switches.
Statement introduced in Junos OS Release 14.1X53-D15 for QFX Series switches.

Description

Configure the specified community VLAN to be a secondary VLAN of the specified primary VLAN. A *community* VLAN is used to transport frames among members of a community (a subset of users within the VLAN), and to forward frames upstream to the primary VLAN.

NOTE: Before you specify this configuration statement, you must have already configured the specified community VLAN and assigned a VLAN ID to it. See **private-vlan**.

Required Privilege Level

system—To view this statement in the configuration.
system–control—To add this statement to the configuration.

RELATED DOCUMENTATION

Creating a Private VLAN on a Single Switch with ELS Support (CLI Procedure) | **472**

Creating a Private VLAN Spanning Multiple EX Series Switches with ELS Support (CLI Procedure) | **481**

control-channel

Syntax

```
control-channel channel-name {
    vlan vlan-id;
    interface name interface-name
}
```

Hierarchy Level

[edit protocols **protection-group ethernet-ring** *name* (**east-interface** | **west-interface**)]

Release Information

Statement introduced in Junos OS Release 9.4.
Statement introduced in Junos OS Release 12.1 for EX Series switches.
Statement introduced in Junos OS Release 14.153-D10 for QFX Series switches.

Description

Configure the Ethernet RPS control channel logical interface to carry the RAPS PDU. The related physical interface is the physical ring port.

Options

vlan *vlan-id*—If the control channel logical interface is a trunk port, then a dedicated **vlan *vlan-id*** defines the dedicated VLAN channel to carry the RAPS traffic. Only configure the ***vlan-id*** when the control channel logical interface is the trunk port.

interface name *interface-name*—Interface name of the control channel.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Ethernet Ring Protection Switching Overview

Example: Configuring Ethernet Ring Protection Switching on EX Series Switches | **855**

Example: Configuring Ethernet Ring Protection Switching on QFX Series and EX Series Switches Supporting ELS | **874**

Configuring Ethernet Ring Protection Switching on Switches (CLI Procedure)

control-vlan

Syntax

control-vlan *(vlan-id | vlan-name)*

Hierarchy Level

[edit protocols **protection-group ethernet-ring**]

[edit protocols **protection-group ethernet-ring** name (east-interface |west-interface)]

Release Information

Statement introduced in Junos OS Release 9.4.
Statement introduced in Junos OS Release 12.1 for EX Series switches.
Statement introduced in Junos OS Release 14.153-D10 for QFX Series switches.

Description

Specify the VLAN that carries the protocol data units (PDUs) between the nodes in the protected Ethernet ring. This is a control VLAN, meaning that it carries data for one instance of an Ethernet ring protection switching (ERPS) in the control channel. Use a control VLAN on trunk port interfaces. One control channel can contain multiple control VLANs.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Example: Configuring Ethernet Ring Protection Switching on EX Series Switches | **855**

Example: Configuring Ethernet Ring Protection Switching on QFX Series and EX Series Switches Supporting ELS | **874**

Configuring Ethernet Ring Protection Switching on Switches (CLI Procedure)

customer-vlans

Syntax

```
customer-vlans (id | native | range);
```

Hierarchy Level

```
[edit vlans vlan-name dot1q-tunneling]
```

Release Information

Statement introduced in Junos OS Release 9.3 for EX Series switches.
Option **native** introduced in Junos OS Release 9.6 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description

Limit the set of accepted customer VLAN tags to a range or to discrete values when mapping customer VLANs to service VLANs.

Options

id—Numeric identifier for a VLAN.

native—Accepts untagged and priority-tagged packets from access interfaces and assigns the configured S-VLAN to the packet.

range—Range of numeric identifiers for VLANs. On the QFX series, you can include as many as eight separate customer VLAN ranges for a given service VLAN. Do not configure more than this number of ranges.

Required Privilege Level

system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- dot1q-tunneling | **1120**
- ether-type | **1146**
- Example: Setting Up Q-in-Q Tunneling on EX Series Switches | **925**
- Configuring Q-in-Q Tunneling on EX Series Switches | **910**
- Understanding Q-in-Q Tunneling and VLAN Translation | **887**

cut-through

Syntax

```
cut-through;
```

Hierarchy Level

```
[edit forwarding-options]
```

Description

Configures all the interfaces in the QFX series switch or QFabric to use cut-through forwarding mode instead of store-and-forward mode.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring Forwarding Mode on Switches | 104

data-channel

Syntax

```
data-channel {
    vlan number;
}
```

Hierarchy Level

[edit protocols **protection-group ethernet-ring** *ring-name*]

Release Information

Statement introduced in Junos OS Release 10.2.
Statement introduced in Junos OS Release 12.1 for EX Series switches.
Statement introduced in Junos OS Release 14.153-D10 for QFX Series switches.

Description

For Ethernet ring protection, configure a data channel to define a set of VLAN IDs that belong to a ring instance.

VLANs specified in the data channel use the same topology used by the ERPS PDU in the control channel. Therefore, if a ring interface is blocked in the control channel, all traffic in the data channel is also blocked on that interface.

Options

vlan *number*—Specify (by VLAN ID) one or more VLANs that belong to a ring instance.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- *Ethernet Ring Protection Using Ring Instances for Load Balancing*
- *Example: Configuring Load Balancing Within Ethernet Ring Protection for MX Series Routers*
- Example: Configuring Ethernet Ring Protection Switching on QFX Series and EX Series Switches Supporting ELS | **874**
- *Configuring Ethernet Ring Protection Switching on Switches (CLI Procedure)*

description (Interfaces)

Syntax

```
description text;
```

Hierarchy Level

```
[edit interfaces interface-name],
[edit interfaces interface-name unit logical-unit-number],
[edit logical-systems logical-system-name interfaces interface-name unit logical-unit-number]
```

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.
Statement introduced in Junos OS Release 12.2 for ACX Series Universal Metro Routers.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Provide a textual description of the interface or the logical unit. Any descriptive text you include is displayed in the output of the **show interfaces** commands, and is also exposed in the **ifAlias** Management Information Base (MIB) object. It has no effect on the operation of the interface on the router or switch.

The textual description can also be included in the extended DHCP relay option 82 Agent Circuit ID suboption.

Options

text—Text to describe the interface. If the text includes spaces, enclose the entire text in quotation marks.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- *Configuring Interface Description*
- *Adding a Logical Unit Description to the Configuration*
- *Configuring Gigabit Ethernet Interfaces (CLI Procedure)*
- *Configuring Gigabit and 10-Gigabit Ethernet Interfaces for OCX Series Switches*

description (VLAN)

Syntax

```
description text-description;
```

Hierarchy Level

```
[edit vlans vlan-name]
```

Release Information

Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.
Option **text-description** enhanced from supporting up to 128 characters to supporting up to 256 characters in Junos OS Release 10.2 for EX Series switches.

Description

Provide a textual description for the VLAN. The text has no effect on the operation of the VLAN or switch.

Options

text-description—Text to describe the interface. It can contain letters, numbers, and hyphens (-) and can be up to 255 characters long. If the text includes spaces, enclose the entire text in quotation marks.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.
system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- Understanding Bridging and VLANs on Switches | **168**
- Understanding Bridging and VLANs on Switches | **168**
- Example: Setting Up Basic Bridging and a VLAN on Switches | **203**
- Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch | **226**
- Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch with ELS Support | **190**
- show vlans | **1648**

destination-address (Security Policies)

Syntax

```
destination-address {
    [address];
    any;
    any-ipv4;
    any-ipv6;
}
```

Hierarchy Level

[edit security policies from-zone *zone-name* to-zone *zone-name* policy *policy-name* match]

[edit security policies global policy *policy-name* match]

Release Information

Statement introduced in Junos OS Release 8.5. Support for IPv6 addresses added in Junos OS Release 10.2. Support for IPv6 addresses in active/active chassis cluster configurations (in addition to the existing support of active/passive chassis cluster configurations) added in Junos OS Release 10.4. Support for wildcard addresses added in Junos OS Release 11.1.

Description

Define the matching criteria. You can specify one or more IP addresses, address sets, or wildcard addresses. You can specify wildcards **any**, **any-ipv4**, or **any-ipv6**.

Options

address—IP address (**any**, **any-ipv4**, **any-ipv6**), IP address set, or address book entry, or wildcard address (represented as A.B.C.D/wildcard-mask). You can configure multiple addresses or address prefixes separated by spaces and enclosed in square brackets.

Required Privilege Level

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Security Policies Overview

dhcp-relay

Syntax

```
dhcp-relay {
    active-server-group server-group-name;
    authentication {
        password password-string;
        username-include {
            circuit-type;
            delimiter delimiter-character;
            domain-name domain-name-string;
            interface-name;
            logical-system-name;
            mac-address;
            option-60;
            option-82 <circuit-id> <remote-id>;
            routing-instance-name;
            user-prefix user-prefix-string;
        }
    }
    dhcpv6 {
        active-server-group server-group-name;
        authentication {
        password password-string;
            username-include {
                circuit-type;
                client-id;
                delimiter delimiter-character;
                domain-name domain-name-string;
                interface-name;
                logical-system-name;
                relay-agent-interface-id;
                relay-agent-remote-id;
                relay-agent-subscriber-id;
                routing-instance-name;
                user-prefix user-prefix-string;
            }
        dynamic-profile profile-name {
            aggregate-clients (merge | replace);
            use-primary primary-profile-name;
        }
        }
    }
```

```
group group-name {
    active-server-group server-group-name;
    authentication {
        ...
    }
    dynamic-profile profile-name {
        ...
    }
    interface interface-name {
        exclude;
        liveness-detection {
            failure-action (clear-binding | clear-binding-if-interface-up | log-only);
            method {
                bfd {
                    version (0 | 1 | automatic);
                    minimum-interval milliseconds;
                    minimum-receive-interval milliseconds;
                    multiplier number;
                    no-adaptation;
                    transmit-interval {
                        minimum-interval milliseconds;
                        threshold milliseconds;
                    }
                    detection-time {
                        threshold milliseconds;
                    }
                    session-mode(automatic | multihop | singlehop);
                    holddown-interval milliseconds;
                }
            }
        }
        overrides {
            ...
        }
        service-profile dynamic-profile-name;
        trace;
        upto upto-interface-name;
    }
    service-profile dynamic-profile-name;
    }
    overrides {
        ...
    }
    relay-agent-interface-id {
```

```
        ...
      }
    service-profile dynamic-profile-name;
    liveness-detection {
      failure-action (clear-binding | clear-binding-if-interface-up | log-only);
      method {
        bfd {
          version (0 | 1 | automatic);
          minimum-interval milliseconds;
          minimum-receive-interval milliseconds;
          multiplier number;
          no-adaptation;
          transmit-interval {
            minimum-interval milliseconds;
            threshold milliseconds;
          }
          detection-time {
            threshold milliseconds;
          }
          session-mode(automatic | multihop | singlehop);
          holddown-interval milliseconds;
        }
      }
    }
    overrides {
      allow-snooped-clients;
      interface-client-limit number;
      no-allow-snooped-clients;
      no-bind-on-request;
      send-release-on-delete;
    }
    relay-agent-interface-id {
      prefix prefix;
      use-interface-description (logical | device):
    }
    server-group {
      server-group-name {
        server-ip-address;
      }
    }
  dynamic-profile profile-name {
    aggregate-clients (merge | replace);
    use-primary primary-profile-name;
  }
```

```
forward-snooped-clients (all-interfaces | configured-interfaces | non-configured-interfaces);
```

```
group group-name {
    active-server-group server-group-name;
    authentication {
        ...
    }
    dynamic-profile profile-name {
        ...
    }
    interface interface-name {
        exclude;
        liveness-detection {
            failure-action (clear-binding | clear-binding-if-interface-up | log-only);
            method {
                bfd {
                    version (0 | 1 | automatic);
                    minimum-interval milliseconds;
                    minimum-receive-interval milliseconds;
                    multiplier number;
                    no-adaptation;
                    transmit-interval {
                        minimum-interval milliseconds;
                        threshold milliseconds;
                    }
                    detection-time {
                        threshold milliseconds;
                    }
                    session-mode(automatic | multihop | singlehop);
                    holddown-interval milliseconds;
                }
            }
        }
        overrides {
            ...
        }
        service-profile dynamic-profile-name;
        trace;
        upto upto-interface-name;
    }
    overrides {
        ...
    }
    relay-option-82 {
        ...
    }
```

```
    service-profile dynamic-profile-name;
}
liveness-detection {
    failure-action (clear-binding | clear-binding-if-interface-up | log-only);
    method {
        bfd {
            version (0 | 1 | automatic);
            minimum-interval milliseconds;
            minimum-receive-interval milliseconds;
            multiplier number;
            no-adaptation;
            transmit-interval {
                minimum-interval milliseconds;
                threshold milliseconds;
            }
            detection-time {
                threshold milliseconds;
            }
            session-mode(automatic | multihop | singlehop);
            holddown-interval milliseconds;
        }
    }
}
overrides {
    allow-snooped-clients;
    always-write-giaddr;
    always-write-option-82;
    client-discover-match <option60-and-option82>;
    disable-relay;
    interface-client-limit number;
    layer2-unicast-replies;
    no-allow-snooped-clients;
    no-bind-on-request;
    proxy-mode;
    replace-ip-source-with;
    send-release-on-delete;
    trust-option-82;
}
relay-option-82 {
    circuit-id {
        prefix prefix;
        use-interface-description (logical | device);
    }
}
```

```
    server-group {
        server-group-name {
            server-ip-address;
        }
    }
    service-profile dynamic-profile-name;
}
```

Hierarchy Level

```
[edit forwarding-options],
[edit vlans forwarding-options]
```

Release Information

Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description

Configure extended Dynamic Host Configuration Protocol (DHCP) relay and DHCPv6 relay options on the switch and enable the switch to function as a DHCP relay agent. A DHCP relay agent forwards DHCP request and reply packets between a DHCP client and a DHCP server.

DHCP relay supports the attachment of dynamic profiles and also interacts with the local AAA Service Framework to use back-end authentication servers, such as RADIUS, to provide subscriber authentication. You can attach dynamic profiles and configure authentication support on a global basis or for a specific group of interfaces.

The extended DHCP and DHCPv6 relay agent options configured with the **dhcp-relay** and **dhcpv6** statements are incompatible with the DHCP/BOOTP relay agent options configured with the **bootp** statement. As a result, the extended DHCP or DHCPv6 relay agent and the DHCP/BOOTP relay agent cannot both be enabled on the router at the same time.

The remaining statements are explained separately.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring DHCP and BOOTP Relay

disable (MVRP)

Syntax

```
disable;
```

Hierarchy Level

```
[edit protocols mvrp],
[edit protocols mvrp interface(all | interface-name)]
```

Release Information

Statement introduced in Junos OS Release 10.0 for EX Series switches.
Statement introduced in Junos OS Release 13.1 for the QFX Series.

Description

Disable the MVRP configuration on the interface.

Default

MVRP is disabled by default.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring Multiple VLAN Registration Protocol (MVRP) on Switches | 797

domain-type (Bridge Domains)

Syntax

```
domain-type bridge;
```

ACX Series and MX Series

```
[edit bridge-domains bridge-domain-name],
[edit logical-systems logical-system-name routing-instances routing-instance-name bridge-domains bridge-domain-name],
[edit routing-instances routing-instance-name bridge-domains bridge-domain-name]
```

SRX Series

```
[edit bridge-domains bridge-domain-name]
```

Release Information

Statement introduced in Junos OS Release 8.4.
Statement modified in Junos OS Release 9.5.
Support for logical systems added in Junos OS Release 9.6.

Description

Define the domain type **bridge** for a Layer 2 bridge domain.

NOTE: There is only one domain type **bridge**, that can be configured on SRX Series devices. Domain type **bridge** is not enabled by default. An SRX Series device operates in the Layer 2 transparent mode when all physical bridge domains on the device are partitioned into logical bridge domains.

NOTE: Starting with Junos OS Release 15.1X49-D10 and Junos OS Release 17.3R1, the CLI **domain-type** is not available.

> **NOTE:** Starting in Junos OS Release 15.1X49-D10 and Junos OS Release 17.3R1, the hierarchy **[edit bridge-domains bridge-domain-name]** is renamed to **[edit vlans vlan-name]**. For detailed information about the modified hierarchies, see "Enhanced Layer 2 CLI Configuration Statement and Command Changes for Security Devices" on page 72.

Required Privilege Level

routing—To view this statement in the configuration.

routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Ethernet Switching and Layer 2 Transparent Mode Overview | 41

Configuring a Bridge Domain

Configuring a Layer 2 Virtual Switch

dot1q-tunneling

List of Syntax

Syntax (Ethernet Switching) on page 1120
Syntax (VLANs) on page 1120

Syntax (Ethernet Switching)

```
dot1q-tunneling {
   ether-type (0x8100 | 0x88a8 | 0x9100);
}
```

Syntax (VLANs)

```
dot1q-tunneling {
   customer-vlans (id | native | range);
   layer2-protocol-tunneling all | protocol-name {
      drop-threshold number;
      shutdown-threshold number;
   }
}
```

Ethernet Switching

```
[edit ethernet-switching-options]
```

VLANs

```
[edit vlans vlan-name]
```

Release Information

Statement introduced in Junos OS Release 9.3 for EX Series switches.
Option **native** introduced in Junos OS Release 9.6 for EX Series switches.
Options **layer2-protocol-tunneling**, **drop-threshold**, and **shutdown-threshold** introduced in Junos OS Release 10.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description

For Ethernet switching, sets a global value for the EtherType for Q-in-Q tunneling.

For VLANs, enables Q-in-Q tunneling on the specified VLAN.

NOTE:

- The VLAN on which you enable Q-in-Q tunneling must be a tagged VLAN.
- You cannot configure 802.1X user authentication on interfaces that have been enabled for Q-in-Q tunneling.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- Example: Setting Up Q-in-Q Tunneling on EX Series Switches | **925**
- Configuring Q-in-Q Tunneling on EX Series Switches | **910**
- Configuring Q-in-Q Tunneling on QFX Series Switches | **899**
- Example: Setting Up Q-in-Q Tunneling on QFX Series Switches | **920**

dot1x

Syntax

```
dot1x {
    authenticator {
        authentication-profile-name access-profile-name;
        interface (all | [ interface-names ]) {
            authentication-order (captive-portal | dot1x | mac-radius);
            disable;
            guest-bridge-domain guest-bridge-domain;
            guest-vlan guest-vlan;
            ignore-port-bounce;
            mac-radius {
                authentication-protocol {
                    eap-md5;
                    eap-peap {
                        resume;
                    }
                    pap;
                }
                flap-on-disconnect;
                restrict;
            }
            maximum-requests number;
            multi-domain {
                max-data-session max-data-session;
                packet-action (drop-and-log | shutdown);
                recovery-timeout seconds;
            }
            (no-reauthentication | reauthentication interval );
            no-tagged-mac-authentication;
            quiet-period seconds;
            redirect-url redirect-url;
            retries number;
            server-fail (bridge-domain bridge-domain | deny | permit | use-cache | vlan-name vlan-name);
            server-fail-voip (deny | permit | use-cache | vlan-name vlan-name);
            server-reject-bridge-domain bridge-domain {
                block-interval seconds;
                eapol-block;
            }
            server-reject-vlan (vlan-id | vlan-name) {
                block-interval block-interval;
                eapol-block;
```

```
        }
        server-timeout seconds;
        supplicant (single | single-secure | multiple);
        supplicant-timeout seconds;
        transmit-period seconds;
      }
      ip-mac-session-binding;
      no-mac-table-binding;
      radius-options {
        add-interface-text-description;
        use-vlan-id;
        use-vlan-name;
      }
      static mac-address {
        bridge-domain-assignment bridge-domain-assignment;
        interface interface;
        vlan-assignment vlan-identifier;
      }
    }
    }
    ssl-certificate-path path-name;
    traceoptions {
      file filename <files files> <size size> <(world-readable | no-world-readable)>;
      flag (all | config-internal | dot1x-debug | dot1x-event | dot1x-ipc | eapol | esw-if | general | iccp | normal | parse
        | state | task | timer | vlan) {
        disable;
      }
    }
}
```

Hierarchy Level

```
[edit logical-systems name protocols],
[edit protocols]
```

Release Information

Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 9.3 for MX Series routers.
Statement introduced in Junos OS Release 14.1X53-D30 for the QFX Series.
Statement introduced in Junos OS Release 15.1X49-D80 for SRX Series.
ssl-certificate-path introduced in Junos OS Release 19.4.

Description

Configure IEEE 802.1X authentication for Port-Based Network Access Control. 802.1X authentication is supported on interfaces that are members of private VLANs (PVLANs).

Default

802.1X is disabled.

Options

ssl-certificate-path *path-name*—Specify the file path for SSL certificates if you are not using the default path. The default path for SSL certificates is **/var/tmp**.

The remaining statements are explained separately. Search for a statement in CLI Explorer or click a linked statement in the Syntax section for details.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- *show dot1x*
- *Example: Setting Up 802.1X for Single-Supplicant or Multiple-Supplicant Configurations on an EX Series Switch*
- *Example: Setting Up 802.1X in Conference Rooms to Provide Internet Access to Corporate Visitors on an EX Series Switch*
- *Example: Setting Up VoIP with 802.1X and LLDP-MED on an EX Series Switch*
- *Example: Configuring Static MAC Bypass of 802.1X and MAC RADIUS Authentication on an EX Series Switch*
- *Example: Configuring MAC RADIUS Authentication on an EX Series Switch*
- *Configuring RADIUS Server Fail Fallback (CLI Procedure)*

drop-threshold

Syntax

```
drop-threshold number;
```

Hierarchy Level

```
[edit vlans vlan-name dot1q-tunneling layer2-protocol-tunneling (all | protocol-name)]
```

Release Information

Statement introduced in Junos OS Release 10.0 for EX Series switches.

Description

Specify the maximum number of Layer 2 PDUs of the specified protocol that can be received per second on the interfaces in a specified VLAN before the switch begins dropping the Layer 2 PDUs. The drop threshold value must be less than or equal to the shutdown threshold value.

L2PT processing is done by the CPU, and L2PT traffic to the CPU is rate-limited to a maximum of 1000 pps. If traffic is received at a rate faster than this limit, the rate limit causes the traffic to be dropped before it hits the threshold and the dropped packets are not reported in L2PT statistics. This can also occur if you configure a drop threshold that is less than 1000 pps but traffic is received at a faster rate. For example, if you configure a drop threshold of 900 pps and the VLAN receives traffic at rate of 1100 pps, L2PT statistics will show that 100 packets were dropped. The 100 packets dropped because of the rate limit will not be reported. Similarly, if you do not configure a drop threshold and the VLAN receives traffic at rate of 1100 pps, the 100 packets dropped because of the rate limit are not reported.

> **NOTE:** If the drop threshold value is greater than the shutdown threshold value and you try to commit the configuration, the commit operation fails.

You can specify a drop threshold value without specifying a shutdown threshold value.

Default

No drop threshold is specified.

Options

number—Maximum number of Layer 2 PDUs of the specified protocol that can be received per second on the interfaces in a specified VLAN before the switch begins dropping the Layer 2 PDUs.

Range: 1 through 1000

Required Privilege Level

system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Example: Configuring Layer 2 Protocol Tunneling on EX Series Switches Without ELS Support | 701

Configuring Layer 2 Protocol Tunneling on EX Series Switches Without ELS Support | 699

shutdown-threshold | 1348

east-interface

Syntax

```
east-interface {
    node-id mac-address;
    control-channel channel-name {
        vlan number;
        interface name interface-name
    }
    interface-none
    ring-protection-link-end;
}
```

Hierarchy Level

```
[edit protocols protection-group ethernet-ring ring-name]
```

Release Information

Statement introduced in Junos OS Release 9.4.
Statement introduced in Junos OS Release 12.1 for EX Series switches.
Statement introduced in Junos OS Release 14.153-D10 for QFX Series switches.

Description

Define one of the two interface ports for Ethernet ring protection, the other being defined by the **west-interface** statement at the same hierarchy level. The interface must use the control channel's logical interface name. The control channel is a dedicated VLAN channel for the ring port.

EX Series switches do not use the node-id statement--the node ID is automatically configured on the switches using the MAC address.

NOTE: Always configure this port first, before configuring the **west-interface** statement.

NOTE: The Node ID is not configurable on EX Series switches. The node ID is automatically configured using the MAC address.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- *Ethernet Ring Protection Switching Overview*
- *Ethernet Ring Protection Using Ring Instances for Load Balancing*
- west-interface | **1429**
- ethernet-ring | **1148**
- Example: Configuring Ethernet Ring Protection Switching on EX Series Switches | **855**
- Example: Configuring Ethernet Ring Protection Switching on QFX Series and EX Series Switches Supporting ELS | **874**
- *Configuring Ethernet Ring Protection Switching on Switches (CLI Procedure)*

edge-virtual-bridging

Syntax

```
edge-virtual-bridging {
    traceoptions {
        file filename <files number> <no-stamp> <replace> <size size> <world-readable | no-world-readable>;
        flag flag ;
    }
    vsi-discovery {
        interface interface-name
        vsi-policy vsi-policy-name
    }
}
```

Hierarchy Level

[edit protocols]

Release Information

Statement introduced in Junos OS Release 12.1 for EX Series switches.

Description

Configure edge virtual bridging (EVB). EVB enables a virtualized station (a physical end station, a server, connected to virtual machines (VMs)) to network with an adjacent switch so that applications residing on the virtual machines can interact with each other and external networks through a technology called virtual Ethernet packet aggregator (VEPA).

The remaining statements are explained separately. See CLI Explorer.

Default

EVB is disabled by default.

Required Privilege Level

system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Example: Configuring Edge Virtual Bridging for Use with VEPA Technology on an EX Series Switch | 1062

Configuring Edge Virtual Bridging on an EX Series Switch | 1060

enable-all-ifl

Syntax

```
enable-all-ifl;
```

Hierarchy Level

[edit protocols **layer2-control mac-rewrite interface** *interface-name*]

Release Information

Statement introduced in Junos OS Release 13.3.

Description

Enable tunneling for STP, VTP, CDP, and other supported protocols on all logical interfaces (VLANs) configured on the interface.

NOTE: Tunneling on all logical interfaces is enabled automatically for PVST/PVST+.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Understanding Layer 2 Protocol Tunneling | 684

protocol | 1326

encapsulation

List of Syntax

Syntax for Physical Interfaces: M Series, MX Series, QFX Series, T Series, PTX Series

```
encapsulation (atm-ccc-cell-relay | atm-pvc | cisco-hdlc | cisco-hdlc-ccc | cisco-hdlc-tcc | ethernet-bridge | ethernet-ccc
    | ethernet-over-atm | ethernet-tcc | ethernet-vpls | ethernet-vpls-fr | ether-vpls-over-atm-llc | ethernet-vpls-ppp
    | extended-frame-relay-ccc | extended-frame-relay-ether-type-tcc | extended-frame-relay-tcc |
    extended-vlan-bridge | extended-vlan-ccc | extended-vlan-tcc | extended-vlan-vpls | flexible-ethernet-services |
    flexible-frame-relay | frame-relay | frame-relay-ccc | frame-relay-ether-type | frame-relay-ether-type-tcc |
    frame-relay-port-ccc | frame-relay-tcc | generic-services | multilink-frame-relay-uni-nni | ppp | ppp-ccc | ppp-tcc
    | vlan-ccc | vlan-vci-ccc | vlan-vpls);
```

Syntax for Physical Interfaces: SRX Series

```
encapsulation (ether-vpls-ppp | ethernet-bridge | ethernet-ccc | ethernet-tcc | ethernet-vpls |
    extended-frame-relay-ccc | extended-frame-relay-tcc | extended-vlan-bridge | extended-vlan-ccc |
    extended-vlan-tcc | extended-vlan-vpls | flexible-ethernet-services | frame-relay-port-ccc | vlan-ccc | vlan-vpls);
```

Syntax for Logical Interfaces: SRX Series

```
encapsulation ( dix | ether-vpls-fr | frame-relay-ppp | ppp-over-ether | vlan-bridge | vlan-ccc | vlan-tcc | vlan-vpls );
```

Physical Interfaces: M Series, MX Series, QFX Series, T Series, PTX Series

[edit interfaces *interface-name*],
[edit interfaces rlsq *number*:*number*]

Logical Interfaces

[edit interfaces *interface-name* unit *logical-unit-number*]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.5.
Statement introduced in Junos OS Release 11.1 for EX Series switches.

Statement introduced in Junos OS Release 12.1X48 for PTX Series Packet Transport Routers (**flexible-ethernet-services**, **ethernet-ccc**, and **ethernet-tcc** options only).

Description

For M Series, MX Series, QFX Series, T Series, PTX Series, specify the physical link-layer encapsulation type.

For SRX Series, specify logical link layer encapsulation.

> **NOTE:** Not all encapsulation types are supported on the switches. See the switch CLI.

Default

ppp—Use serial PPP encapsulation.

Physical Interface Options and Logical Interface Options

[Warning: element unresolved in stylesheets: <title> (in <config-options>). This is probably a new element that is not yet supported in the stylesheets.]

Physical Interface Options and Logical Interface Options

For physical interfaces:

> **NOTE:** Frame Relay, ATM, PPP, SONET, and SATSOP options are not supported on EX Series switches.

- **atm-ccc-cell-relay**—Use ATM cell-relay encapsulation.
- **atm-pvc**—Defined in RFC 2684, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*. When you configure physical ATM interfaces with ATM PVC encapsulation, an RFC 2684-compliant ATM Adaptation Layer 5 (AAL5) tunnel is set up to route the ATM cells over a Multiprotocol Label Switching (MPLS) path that is typically established between two MPLS-capable routers using the Label Distribution Protocol (LDP).
- **cisco-hdlc**—Use Cisco-compatible High-Level Data Link Control (HDLC) framing. E1, E3, SONET/SDH, T1, and T3 interfaces can use Cisco HDLC encapsulation. Two related versions are supported:
  - CCC version (**cisco-hdlc-ccc**)—The logical interface does not require an encapsulation statement. When you use this encapsulation type, you can configure the **ccc** family only.
  - TCC version (**cisco-hdlc-tcc**)—Similar to CCC and has the same configuration restrictions, but used for circuits with different media on either side of the connection.
- **cisco-hdlc-ccc**—Use Cisco-compatible HDLC framing on CCC circuits.
- **cisco-hdlc-tcc**—Use Cisco-compatible HDLC framing on TCC circuits for connecting different media.
- **ethernet-bridge**—Use Ethernet bridge encapsulation on Ethernet interfaces that have bridging enabled and that must accept all packets.
- **ethernet-over-atm**—For interfaces that carry IPv4 traffic, use Ethernet over ATM encapsulation. When you use this encapsulation type, you cannot configure multipoint interfaces. As defined in RFC 2684, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*, this encapsulation type allows ATM interfaces to connect to devices that support only bridge protocol data units (BPDUs). Junos OS does not completely support bridging, but accepts BPDU packets as a default gateway. If you use the router as an edge device, then the router acts as a default gateway. It accepts Ethernet LLC/SNAP frames with IP or ARP in the payload, and drops the rest. For packets destined to the Ethernet LAN, a route lookup is done using the destination IP address. If the route lookup yields a full address match, the packet is encapsulated with an LLC/SNAP and MAC header, and the packet is forwarded to the ATM interface.
- **ethernet-tcc**—For interfaces that carry IPv4 traffic, use Ethernet TCC encapsulation on interfaces that must accept packets carrying standard TPID values. For 8-port, 12-port, and 48-port Fast Ethernet PICs, TCC is not supported.

- **ethernet-vpls**—Use Ethernet VPLS encapsulation on Ethernet interfaces that have VPLS enabled and that must accept packets carrying standard TPID values. On M Series routers, except the M320 router, the 4-port Fast Ethernet TX PIC and the 1-port, 2-port, and 4-port, 4-slot Gigabit Ethernet PICs can use the Ethernet VPLS encapsulation type.
- **ethernet-vpls-fr**—Use in a VPLS setup when a CE device is connected to a PE device over a time division multiplexing (TDM) link. This encapsulation type enables the PE device to terminate the outer Layer 2 Frame Relay connection, use the 802.1p bits inside the inner Ethernet header to classify the packets, look at the MAC address from the Ethernet header, and use the MAC address to forward the packet into a given VPLS instance.
- **ethernet-vpls-ppp**—Use in a VPLS setup when a CE device is connected to a PE device over a time division multiplexing (TDM) link. This encapsulation type enables the PE device to terminate the outer Layer 2 PPP connection, use the 802.1p bits inside the inner Ethernet header to classify the packets, look at the MAC address from the Ethernet header, and use it to forward the packet into a given VPLS instance.
- **ether-vpls-over-atm-llc**—For ATM intelligent queuing (IQ) interfaces only, use the Ethernet virtual private LAN service (VPLS) over ATM LLC encapsulation to bridge Ethernet interfaces and ATM interfaces over a VPLS routing instance (as described in RFC 2684, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*). Packets from the ATM interfaces are converted to standard ENET2/802.3 encapsulated Ethernet frames with the frame check sequence (FCS) field removed.
- **extended-frame-relay-ccc**—Use Frame Relay encapsulation on CCC circuits. This encapsulation type allows you to dedicate DLCIs 1 through 1022 to CCC. When you use this encapsulation type, you can configure the **ccc** family only.
- **extended-frame-relay-ether-type-tcc**—Use extended Frame Relay ether type TCC for Cisco-compatible Frame Relay for DLCIs 1 through 1022. This encapsulation type is used for circuits with different media on either side of the connection.
- **extended-frame-relay-tcc**—Use Frame Relay encapsulation on TCC circuits to connect different media. This encapsulation type allows you to dedicate DLCIs 1 through 1022 to TCC.
- **extended-vlan-bridge**—Use extended VLAN bridge encapsulation on Ethernet interfaces that have IEEE 802.1Q VLAN tagging and bridging enabled and that must accept packets carrying TPID 0x8100 or a user-defined TPID.
- **extended-vlan-ccc**—Use extended VLAN encapsulation on CCC circuits with Gigabit Ethernet and 4-port Fast Ethernet interfaces that must accept packets carrying 802.1Q values. Extended VLAN CCC encapsulation supports TPIDs 0x8100, 0x9100, and 0x9901. When you use this encapsulation type, you can configure the **ccc** family only. For 8-port, 12-port, and 48-port Fast Ethernet PICs, extended VLAN CCC is not supported. For 4-port Gigabit Ethernet PICs, extended VLAN CCC is not supported.
- **extended-vlan-tcc**—For interfaces that carry IPv4 traffic, use extended VLAN encapsulation on TCC circuits with Gigabit Ethernet interfaces on which you want to use 802.1Q tagging. For 4-port Gigabit Ethernet PICs, extended VLAN TCC is not supported.

- **extended-vlan-vpls**—Use extended VLAN VPLS encapsulation on Ethernet interfaces that have VLAN 802.1Q tagging and VPLS enabled and that must accept packets carrying TPIDs 0x8100, 0x9100, and 0x9901. On M Series routers, except the M320 router, the 4-port Fast Ethernet TX PIC and the 1-port, 2-port, and 4-port, 4-slot Gigabit Ethernet PICs can use the Ethernet VPLS encapsulation type.

  NOTE: The built-in Gigabit Ethernet PIC on an M7i router does not support extended VLAN VPLS encapsulation.

- **flexible-ethernet-services**—For Gigabit Ethernet IQ interfaces and Gigabit Ethernet PICs with small form-factor pluggable transceivers (SFPs) (except the 10-port Gigabit Ethernet PIC and the built-in Gigabit Ethernet port on the M7i router), and for Gigabit Ethernet interfaces, use flexible Ethernet services encapsulation when you want to configure multiple per-unit Ethernet encapsulations. Aggregated Ethernet bundles can use this encapsulation type. This encapsulation type allows you to configure any combination of route, TCC, CCC, Layer 2 virtual private networks (VPNs), and VPLS encapsulations on a single physical port. If you configure flexible Ethernet services encapsulation on the physical interface, VLAN IDs from 1 through 511 are no longer reserved for normal VLANs.
- **flexible-frame-relay**—For IQ interfaces only, use flexible Frame Relay encapsulation when you want to configure multiple per-unit Frame Relay encapsulations. This encapsulation type allows you to configure any combination of TCC, CCC, and standard Frame Relay encapsulations on a single physical port. Also, each logical interface can have any DLCI value from 1 through 1022.
- **frame-relay**—Use Frame Relay encapsulation is defined in RFC 1490, *Multiprotocol Interconnect over Frame Relay*. E1, E3, link services, SONET/SDH, T1, T3, and voice services interfaces can use Frame Relay encapsulation.
- **frame-relay-ccc**—Use Frame Relay encapsulation on CCC circuits. This encapsulation is same as standard Frame Relay for DLCIs 0 through 511. DLCIs 512 through 1022 are dedicated to CCC. The logical interface must also have **frame-relay-ccc** encapsulation. When you use this encapsulation type, you can configure the **ccc** family only.
- **frame-relay-ether-type**—Use Frame Relay ether type encapsulation for compatibility with the Cisco Frame Relay. IETF frame relay encapsulation identifies the payload format using NLPID and SNAP formats. Cisco-compatible Frame Relay encapsulation uses the Ethernet type to identify the type of payload.

  NOTE: When the encapsulation type is set to Cisco-compatible Frame Relay encapsulation, ensure that the LMI type is set to ANSI or Q933-A.

- **frame-relay-ether-type-tcc**—Use Frame Relay ether type TCC for Cisco-compatible Frame Relay on TCC circuits to connect different media. This encapsulation is Cisco-compatible Frame Relay for DLCIs 0 through 511. DLCIs 512 through 1022 are dedicated to TCC.

- **frame-relay-port-ccc**—Use Frame Relay port CCC encapsulation to transparently carry all the DLCIs between two customer edge (CE) routers without explicitly configuring each DLCI on the two provider edge (PE) routers with Frame Relay transport. The connection between the two CE routers can be either user-to-network interface (UNI) or network-to-network interface (NNI); this is completely transparent to the PE routers. When you use this encapsulation type, you can configure the **ccc** family only.
- **frame-relay-tcc**—This encapsulation is similar to Frame Relay CCC and has the same configuration restrictions, but used for circuits with different media on either side of the connection.
- **generic-services**—Use generic services encapsulation for services with a hierarchical scheduler.
- **multilink-frame-relay-uni-nni**—Use MLFR UNI NNI encapsulation. This encapsulation is used on link services, voice services interfaces functioning as FRF.16 bundles, and their constituent T1 or E1 interfaces, and is supported on LSQ and redundant LSQ interfaces.
-
- **ppp**—Use serial PPP encapsulation. This encapsulation is defined in RFC 1661, *The Point-to-Point Protocol (PPP) for the Transmission of Multiprotocol Datagrams over Point-to-Point Links*. PPP is the default encapsulation type for physical interfaces. E1, E3, SONET/SDH, T1, and T3 interfaces can use PPP encapsulation.
- **ppp-ccc**—Use serial PPP encapsulation on CCC circuits. When you use this encapsulation type, you can configure the **ccc** family only.
- **ppp-tcc**—Use serial PPP encapsulation on TCC circuits for connecting different media. When you use this encapsulation type, you can configure the **tcc** family only.
- **vlan-ccc**—Use Ethernet VLAN encapsulation on CCC circuits. VLAN CCC encapsulation supports TPID 0x8100 only. When you use this encapsulation type, you can configure the **ccc** family only.
- **vlan-vci-ccc**—Use ATM-to-Ethernet interworking encapsulation on CCC circuits. When you use this encapsulation type, you can configure the **ccc** family only. All logical interfaces configured on the Ethernet interface must also have the encapsulation type set to **vlan-vci-ccc**.
- **vlan-vpls**—Use VLAN VPLS encapsulation on Ethernet interfaces with VLAN tagging and VPLS enabled. Interfaces with VLAN VPLS encapsulation accept packets carrying standard TPID values only. On M Series routers, except the M320 router, the 4-port Fast Ethernet TX PIC and the 1-port, 2-port, and 4-port, 4-slot Gigabit Ethernet PICs can use the Ethernet VPLS encapsulation type.

  NOTE:
  - Label-switched interfaces (LSIs) do not support VLAN VPLS encapsulation. Therefore, you can only use VLAN VPLS encapsulation on a PE-router-to-CE-router interface and not a core-facing interface.
  - Starting with Junos OS release 13.3, a commit error occurs when you configure **vlan-vpls** encapsulation on a physical interface and configure **family inet** on one of the logical units. Previously, it was possible to commit this invalid configuration.

For logical interfaces:

- **frame-relay**—Configure a Frame Relay encapsulation when the physical interface has multiple logical units, and the units are either point to point or multipoint.
- **multilink-frame-relay-uni-nni**—Link services interfaces functioning as FRF.16 bundles can use Multilink Frame Relay UNI NNI encapsulation.
- **ppp**—For normal mode (when the device is using only one ISDN B-channel per call). Point-to-Point Protocol is for communication between two computers using a serial interface.
- **ppp-over-ether**—This encapsulation is used for underlying interfaces of pp0 interfaces.

Required Privilege Level
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Understanding Physical Encapsulation on an Interface

Configuring Interface Encapsulation on Physical Interfaces

Configuring CCC Encapsulation for Layer 2 VPNs

Configuring Layer 2 Switching Cross-Connects Using CCC

Configuring TCC Encapsulation for Layer 2 VPNs and Layer 2 Circuits

Configuring ATM Interface Encapsulation

Configuring ATM-to-Ethernet Interworking

Configuring VLAN and Extended VLAN Encapsulation | **315**

Configuring VLAN and Extended VLAN Encapsulation | **315**

Configuring Encapsulation for Layer 2 Wholesale VLAN Interfaces

Configuring Interfaces for Layer 2 Circuits

Configuring Interface Encapsulation on PTX Series Packet Transport Routers

Configuring MPLS LSP Tunnel Cross-Connects Using CCC

Configuring TCC

Configuring VPLS Interface Encapsulation

Configuring Interfaces for VPLS Routing

Defining the Encapsulation for Switching Cross-Connects

Configuring an MPLS-Based Layer 2 VPN (CLI Procedure)

ether-options

List of Syntax

Junos OS Syntax

```
ether-options {
    802.3ad {
        aex;
        (backup | primary);
        lacp {
            force-up;
            (primary |backup);
            port-priority
        }
    }
    asynchronous-notification;
    (auto-negotiation| no-auto-negotiation);
    autostate-exclude
    configured-flow-control {
        rx-buffers (on | off);
        tx-buffers (on | off);
    }
    ethernet-switch-profile {
        ethernet-policer-profile
        (mac-learn-enable | no-mac-learn-enable);
        recovery-timeouttime-in-seconds;
        storm-control storm-control-profile;
        tag-protocol-id;
    }
    (flow-control | no-flow-control);
    ieee-802-3az-eee;
    ignore-l3-incompletes;
    link-mode (automatic | full-duplex | half-duplex);
    (loopback | no-loopback);
    mdi-mode (auto | force | mdi | mdix);
    mpls {
        pop-all-labels <required-depth (1 | 2 | all)>;
    }
    no-auto-mdix;
    redundant-parent (Interfaces) parent;
    source-address-filter name;
```

```
    (source-filtering| no-source-filtering);
    speed {
        (auto-negotiation <auto-negotiate-10-100> | ethernet-100m | ethernet-10g | ethernet-10m | ethernet-1g);
    }
}
```

Junos OS Evolved Syntax

```
ether-options {
    802.3ad {
        aex;
        (backup | primary);
        lacp {
            force-up;
            (primary |backup);
            port-priority
        }
    }
    asynchronous-notification;
    (auto-negotiation| no-auto-negotiation);
    autostate-exclude
    ethernet-switch-profile {
        ethernet-policer-profile
        (mac-learn-enable | no-mac-learn-enable);
        recovery-timeout time-in-seconds;
        storm-control storm-control-profile;
        tag-protocol-id;
    }
    fec (gigether)
    (flow-control | no-flow-control);
    ignore-l3-incompletes;
    (loopback | no-loopback);
    loopback-remote;
    mpls {
        pop-all-labels <required-depth (1 | 2 | all)>;
    }
    source-address-filter name;
    (source-filtering | no-source-filtering);
}
```

Hierarchy Level

[edit interfaces *interface-name*]
[edit interfaces interface-range *range*]

Release Information

Statement introduced in Junos OS Release 9.0.
autostate-exclude option introduced in Junos OS Release 14.1x53-D40 for QFX5100 switches only.
fec and **loopback-remote** options introduced in Junos OS Evolved Release 20.1R1.

Description

Configure **ether-options** properties for a Gigabit Ethernet or 10-Gigabit Ethernet interface.

In Junos OS Evolved, when you configure **set interfaces *interface* ether-options 802.3ad *ae name*** at the same time as you apply a second configuration to the same interface at the **[edit interfaces *interface*]** hierarchy, the second configuration will not take effect until the interface joins the aggregated Ethernet interface ***ae name***.

> **NOTE:** The **ether-options** statement is not supported for subscriber management on aggregated Ethernet member link interfaces. You must configure **gigether-options** instead.

Table 135 on page 1142 shows the supported and unsupported platforms.

Table 135: Supported Platform Information

Supported Platforms for gigether-options	Supported Platforms for ether-options	Notes
ACX Series Routers (Junos OS) • ACX500 • ACX1000 • ACX1100 • ACX2100 • ACX2200 • ACX4000 • ACX5400 • ACX6000	ACX Series Routers (Junos OS) • ACX5048 • ACX5096	None
Not Supported	EX Series Switches (Junos OS) • EX2300 • EX2300 Multigigabit • EX2300-C • EX3400 • EX4300 • EX4300 Multigigabit • EX4600 • EX4650 • EX9200 • EX9250	None
MX Series Routers (Junos OS) • MX5 • MX10 • MX40 • MX80 • MX104 • MX150 • MX204 • MX240 • MX480 • MX960 • MX20008	Not Supported	None

Table 135: Supported Platform Information *(continued)*

Supported Platforms for gigether-options	Supported Platforms for ether-options	Notes
• MX2010 • MX2020 • MX10003 • MX10008 and MX10016		
PTX Series Routers (Junos OS) • PTX1000 • PTX3000 • PTX5000 • PTX10001 • PTX10002 • PTX10003 • PTX10008 and PTX10016	PTX Series Routers (Junos OS) • PTX1000 • PTX10001 • PTX10002 • PTX10003 • PTX10008 and PTX10016	PTX Series Routers (Junos OS) In Junos OS Release 17.3R3S7, PTX1000 Series routers support both **ether-options** and **gigether-options**. In Junos OS Releases 17.3R1, 17.4R1, and 17.4R2, PTX10000 Series routers support both **ether-options** and **gigether-options**.
PTX Series routers (Junos OS Evolved) • PTX10003 • PTX10008 and PTX100016	PTX Series routers (Junos OS Evolved) • PTX10003 • PTX10008 and PTX100016	PTX Series routers (Junos OS Evolved) Starting in Junos OS Evolved Release 20.1R1, PTX Series routers support **ether-options** only.
Not Supported		None

Table 135: Supported Platform Information *(continued)*

Supported Platforms for gigether-options	Supported Platforms for ether-options	Notes
	QFX Series Switches (Junos OS) • QFX5100 (48S) • QFX5100 (48T) • QFX5100 (24Q) • QFX5100 (96S) • QFX5110 (48S) • QFX5110 (32Q) • QFX5120 (48Y) • QFX5120 (32C) • QFX5200 (48Y) • QFX5200 (32C) • QFX5210 (64C) • QFX5210 (64C-S) • QFX5220 (32CD) • QFX5220 (128C) • QFX100002 • QFX10008 and QFX10016	
Not Supported	QFX Series Switches (Junos OS Evolved) • QFX5200-32C-L • QFX5220-32CD • QFX5220-128C	None
SRX Series (Junos OS) • SRX300 • SRX550 • SRX1500 • SRX4100 and SRX4200 • SRX4600 • SRX5400 • SRX5600	SRX Series (Junos OS) • SRX300 • SRX550 • SRX1500 • SRX4100 and SRX4200 • SRX4600 • SRX5400 • SRX5600	SRX Series (Junos OS) To configure gigabit-Ethernet interfaces (ge-), use **gigether-options**. To configure ethernet interfaces (et-) and fast ethernet interfaces (fe-), use **ether-options**.

Default

Enabled.

Options

> **NOTE:** The **auto-negotation** and **speed** statements are not supported on the OCX Series.

loopback-remote—Starting in Junos OS Evolved Release 20.1R1, enable remote loopback.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- *Gigabit Ethernet Interface*
- *Configuring LACP Link Protection of Aggregated Ethernet Interfaces for Switches*
- Configuring Q-in-Q Tunneling on EX Series Switches with ELS Support | **900**
- *gigether-options*

ether-type

Syntax

```
ether-type (0x8100 | 0x88a8 | 0x9100)
```

Hierarchy Level

[edit **ethernet-switching-options dot1q-tunneling**]

Release Information

Statement introduced in Junos OS Release 9.3 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description

Configure a global value for the tag protocol identifier (EtherType) of the service VLAN tags (outer tags) in Q-in-Q tunneling. Only one Ethertype value is supported at a time. The Ethertype value appears in the Ethernet type field of the packet. It specifies the protocol being transported in the Ethernet frame.

Required Privilege Level

system—To view this statement in the configuration.
system-control—To add this statement to the configuration.
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- dot1q-tunneling | **1120**
- Example: Setting Up Q-in-Q Tunneling on EX Series Switches | **925**
- Configuring Q-in-Q Tunneling on EX Series Switches | **910**
- Configuring Q-in-Q Tunneling on QFX Series Switches | **899**
- Example: Setting Up Q-in-Q Tunneling on QFX Series Switches | **920**

ethernet (Chassis Cluster)

Syntax

```
ethernet {
    device-count number;
    lacp {
        link-protection {
            non-revertive;
        }
        system-priority number;
    }
}
```

Hierarchy Level

```
[edit chassis aggregated-devices]
```

Release Information

Statement introduced in Junos OS Release 10.2.

Description

Configure properties for aggregated Ethernet devices.

Options

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

ethernet-ring

Syntax

```
ethernet-ring ring-name {
    control-vlan (vlan-id | vlan-name);
    data-channel {
        vlan number
    }
    east-interface {
        control-channel channel-name {
            vlan number;
            interface name interface-name
        }
    }
    guard-interval number;
    node-id mac-address;
    restore-interval number;
    ring-protection-link-owner;
    west-interface {
        control-channel channel-name {
            vlan number;
        }
    }
}
```

Hierarchy Level

```
[edit protocols protection-group]
```

Release Information

Statement introduced in Junos OS Release 9.4.
Statement introduced in Junos OS Release 12.1 for EX Series switches.
Statement introduced in Junos OS Release 14.153-D10 for QFX Series switches.

Description

For Ethernet PICs on MX Series routers or for EX Series switches, , specify the Ethernet ring in an Ethernet ring protection switching configuration.

Options

ring-name—Name of the Ethernet protection ring.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

interface—To view this statement in the configuration.

interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Ethernet Ring Protection Switching Overview

Example: Configuring Ethernet Ring Protection Switching on EX Series Switches | **855**

Example: Configuring Ethernet Ring Protection Switching on QFX Series and EX Series Switches Supporting ELS | **874**

Configuring Ethernet Ring Protection Switching on Switches (CLI Procedure)

ethernet-switch-profile

Syntax

```
ethernet-switch-profile {
    ethernet-policer-profile {
        input-priority-map {
            ieee802.1p premium [values];
        }
        output-priority-map {
            classifier {
                premium {
                    forwarding-class class-name {
                        loss-priority (high | low);
                    }
                }
            }
        }
        policer cos-policer-name {
            aggregate {
                bandwidth-limit bps;
                burst-size-limit bytes;
            }
            premium {
                bandwidth-limit bps;
                burst-size-limit bytes;
            }
        }
        storm-control storm-control-profile;
        tag-protocol-id tpid;
    }
    mac-learn-enable;
}
```

Hierarchy Level

[edit interfaces *interface-name* gigether-options],
[edit interfaces *interface-name* aggregated-ether-options],
[edit interfaces *interface-name* **aggregated-ether-options**],
[edit interfaces *interface-name* ether-options]

Release Information

Statement introduced before Junos OS Release 7.4.

Statement introduced in Junos OS Release 12.2 for ACX Series Universal Metro Routers.
Statement introduced in Junos OS Release 13.2 for the QFX Series.
Statement introduced in Junos OS Release 13.2X50-D15 for the EX Series switches.

Description

> **NOTE:** On QFX Series standalone switches, the **ethernet-policer-profile** CLI hierarchy and the **mac-learn-enable** statement are supported only on the Enhanced Layer 2 Switching CLI.

For Gigabit Ethernet IQ, 10-Gigabit Ethernet IQ2 and IQ2-E, and Gigabit Ethernet PICs with SFPs (except the 10-port Gigabit Ethernet PIC, aggregated Ethernet with Gigabit Ethernet IQ interfaces, the built-in Gigabit Ethernet port on the M7i router); 100-Gigabit Ethernet Type 5 PIC with CFP; and Gigabit Ethernet, 10-Gigabit Ethernet, 40-Gigabit Ethernet, and aggregated Ethernet interfaces on EX Series switches, configure VLAN tag and MAC address accounting and filtering properties.

The remaining statements are explained separately. See CLI Explorer.

> **NOTE:** When you gather interfaces into a bridge domain, the **no-mac-learn-enable** statement at the **[edit interfaces *interface-name* gigether-options ethernet-switch-profile]** hierarchy level is not supported. You must use the **no-mac-learning** statement at the **[edit bridge-domains *bridge-domain-name* bridge-options interface *interface-name*]** hierarchy level to disable MAC learning on an interface in a bridge domain. For information on disabling MAC learning for a bridge domain, see the *MX Series Layer 2 Configuration Guide*.

Default

If the **ethernet-switch-profile** statement is not configured, Gigabit Ethernet IQ and Gigabit Ethernet PICs with SFPs (except the 10-port Gigabit Ethernet PIC and the built-in Gigabit Ethernet port on the M7i router) behave like Gigabit Ethernet interfaces.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring Gigabit Ethernet Policers

Configuring Gigabit Ethernet Policers

Stacking and Rewriting Gigabit Ethernet VLAN Tags Overview | **381**

Configuring Q-in-Q Tunneling on EX Series Switches with ELS Support | **900**

ethernet-switching

List of Syntax

Syntax (EX Series and QFX Series)

```
ethernet-switching {
    filter input filter-name;
    filter output filter-name;
    native-vlan-id vlan-id;
    port-mode mode;
    reflective-relay;
    vlan {
        members [ (all | names | vlan-ids) ];
    }
}
```

Syntax (SRX Series)

```
ethernet-switching {
    block-non-ip-all;
    bpdu-vlan-flooding;
    bypass-non-ip-unicast;
    no-packet-flooding {
        no-trace-route;
    }
}
```

Hierarchy Level

For EX Series and QFX Series switches:

```
[edit interfaces ge-chassis/slot/port unit logical-unit-number] family
```

For SRX Series devices:

```
[edit security flow]
```

Release Information

Statement introduced in Junos OS Release 9.5 for SRX Series.
Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description

Configure Ethernet switching protocol family information for the logical interface. Changes default Layer 2 forwarding behavior.

The remaining statements are explained separately. See CLI Explorer.

Default

You must configure a logical interface to be able to use the physical device.

Options

- **block-non-ip-all**—Block all Layer 2 non-IP and non-ARP traffic, including multicast and broadcast traffic.
- **bpdu-vlan-flooding**—Set 802.1D bridge protocol data unit (BPDU) flooding based on VLAN on which BPDU originate. The default behavior is to receive BPDUs and then flood BPDUs out to all active ports on the SRX Series devices.
- **bypass-non-ip-unicast**—Allow all Layer 2 non-IP traffic to pass through the device.
- **no-packet-flooding**—Stop IP flooding and send ARP or ICMP requests to discover the destination MAC address for a unicast packet.

  > **NOTE:** On all SRX Series devices in transparent mode, packet flooding is enabled by default. If you have manually disabled packet flooding with the set security flow ethernet-switching no-packet-flooding command, then multicast packets such as OSPFv3 hello packets are dropped.

- **no-trace-route**—Do not send ICMP requests to discover the destination MAC address for a unicast packet. Only ARP requests are sent. This option only allows the device to discover the destination MAC address for a unicast packet if the destination IP address is in the same subnetwork as the ingress IP address.

> **NOTE:** The **block-non-ip-all** and **bypass-non-ip-unicast** options cannot be configured at the same time.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.
security—To view this in the configuration.
security-control—To add this to the configuration.

RELATED DOCUMENTATION

Configuring Gigabit and 10-Gigabit Ethernet Interfaces for EX4600 and QFX Series Switches

Understanding Traffic Processing on Security Devices

JUNOS Software Network Interfaces Configuration Guide

ethernet-switching-options

List of Syntax

EX Series on page 1156
QFX Series, QFabric, EX4600 on page 1161

EX Series

```
ethernet-switching-options {
    analyzer (Port Mirroring) {
        name {
            loss-priority priority;
            ratio number;
            input {
                ingress {
                    interface (all | interface-name);
                    vlan (vlan-id | vlan-name);
                }
                egress {
                    interface (all | interface-name);
                }
            }
            output {
                interface interface-name;
                vlan (vlan-id | vlan-name) {
                no-tag;
                }
            }
        }
    }
    bpdu-block {
        disable-timeout timeout;
        interface (all | [interface-name]) {
            (disable | drop | shutdown);
            }
    }
    dot1q-tunneling {
        ether-type (0x8100 | 0x88a8 | 0x9100);
    }
    interfaces interface-name {
        no-mac-learning;
    }
    mac-lookup-length number-of-entries;
}
```

```
mac-notification {
    notification-interval seconds;
}
mac-table-aging-time seconds;
nonstop-bridging;
port-error-disable {
    disable-timeout timeout;
}
redundant-trunk-group {
    group name {
        interface interface-name <primary>;
        interface interface-name;
    }
}
```

```
secure-access-port {
    dhcp-snooping-file {
        location local_pathname | remote_URL;
        timeout seconds;
        write-interval seconds;
    }
    dhcpv6-snooping-file {
        location local_pathname | remote_URL;
        timeout seconds;
        write-interval seconds;
    }
    interface (all | interface-name) {
        allowed-mac {
            mac-address-list;
        }
        (dhcp-trusted | no-dhcp-trusted);
        fcoe-trusted;
        mac-limit limit action (drop | log | none | shutdown);
        no-allowed-mac-log;
        persistent-learning;
        static-ip ip-address {
            vlan  vlan-name;
            mac mac-address;
        }
        static-ipv6 ip-address {
            vlan  vlan-name;
            mac mac-address;
        }
    }
    vlan (all | vlan-name) {
        (arp-inspection | no-arp-inspection) [
            forwarding-class class-name;
        }
        dhcp-option82 {
            circuit-id {
                prefix hostname;
                use-interface-description;
                use-vlan-id;
            }
            remote-id {
                prefix hostname | mac | none;
                use-interface-description;
                use-string string;
            }
```

```
        vendor-id [string];
    }
    (examine-dhcp | no-examine-dhcp) {
        forwarding-class class-name;
    }
    (examine-dhcpv6 | no-examine-dhcpv6) {
        forwarding-class class-name;
    }
    examine-fip {
        fc-map fc-map-value;
    }
    (ip-source-guard | no-ip-source-guard);
    (ipv6-source-guard | no-ipv6-source-guard);
    mac-move-limit limit action (drop | log | none | shutdown);
    }
    (neighbor-discovery-inspection | no-neighbor-discovery-inspection);
no-option-37;
static {
    vlan name {
        mac mac-address {
            next-hop interface-name;
        }
    }
}
storm-control {
    action-shutdown;
    interface (all | interface-name) {
        bandwidth bandwidth;
        level level;
        multicast;
        no-broadcast;
        no-multicast;
        no-registered-multicast;
        no-unknown-unicast;
        no-unregistered-multicast;
    }
}
traceoptions {
    file filename <files number> <no-stamp> <replace> <size size> <world-readable | no-world-readable>;
    flag flag <disable>;
}
```

```
    unknown-unicast-forwarding {
        vlan (all | vlan-name) {
            interface interface-name;
        }
    }
    voip {
        interface (all | [interface-name | access-ports]) {
            forwarding-class forwarding-class;
            vlan vlan-name;
        }
    }
}
```

QFX Series, QFabric, EX4600

```
ethernet-switching-options {
    analyzer {
        name {
            input {
                egress {
                    interface (all | interface-name);
                }
                ingress {
                    interface (all | interface-name);
                    vlan (vlan-id | vlan-name);
                }
            output {
                interface interface-name;
                ip-address ip-address;
                vlan (vlan-id | vlan-name);
            }
        }
    }
    bpdu-block {
        interface (all | [interface-name]);
        disable-timeout timeout;
    }
    dot1q-tunneling {
        ether-type (0x8100 | 0x88a8 | 0x9100)
    }
    interfaces interface-name {
        no-mac-learning;
    }
    mac-table-aging-time seconds {
    }
    port-error-disable {
        disable-timeout timeout;
    }
    secure-access-port {
        dhcp-snooping-file {
            location local_pathname | remote_URL;
            timeout seconds;
            write-interval seconds;
        }
        interface (all | interface-name) {
            allowed-mac {
                mac-address-list;
            }
```

```
        (dhcp-trusted | no-dhcp-trusted);
        fcoe-trusted;
        mac-limit limit action action;
        no-allowed-mac-log;
    }
    vlan (all | vlan-name) {
        (arp-inspection | no-arp-inspection) [
            forwarding-class (for DHCP Snooping or DAI Packets) class-name;
        }
        dhcp-option82 {
            circuit-id {
                prefix (Circuit ID for Option 82) hostname;
                use-interface-description;
                use-vlan-id;
            }
            remote-id {
                prefix (Remote ID for Option 82) hostname | mac | none;
                use-interface-description;
                use-string string;
            }
            vendor-id <string>;
        }
        (examine-dhcp | no-examine-dhcp) {
            forwarding-class (for DHCP Snooping or DAI Packets) class-name;
        }
        examine-fip {
            examine-vn2vn {
                beacon-period milliseconds;
            }
            fc-map fc-map-value;
        }
        mac-move-limit limit <fabric-limit limit action action;
    }
}
static {
    vlan vlan-id {
        mac mac-address next-hop interface-name;
    }
}
```

```
    storm-control {
        interface (all | interface-name) {
            bandwidth bandwidth;
            no-broadcast;
            no-multicast;
            no-unknown-unicast;
        }
    }
    traceoptions {
        file filename <files number> <no-stamp> <replace> <size size> <world-readable | no-world-readable>;
        flag flag <disable>;
    }
}
```

Hierarchy Level

```
[edit]
```

Release Information

Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description

Configure Ethernet switching options.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- *Understanding Port Mirroring and Analyzers*
- *Understanding How to Protect Access Ports from Common Attacks*
- *Port Security Features*
- *Understanding BPDU Protection for STP, RSTP, and MSTP*
- Understanding Redundant Trunk Links (Legacy RTG Configuration) | **937**
- *Understanding Storm Control*
- *Understanding 802.1X and VoIP on EX Series Switches*
- Understanding Q-in-Q Tunneling and VLAN Translation | **887**
- *Understanding and Preventing Unknown Unicast Forwarding*
- Understanding MAC Notification on EX Series Switches | **128**
- *Understanding FIP Snooping*
- *Understanding Nonstop Bridging on EX Series Switches*

exclusive-mac

Syntax

exclusive-mac *virtual-mac-mac-address/mask*;

Hierarchy Level

[edit protocols l2-learning global-mac-move]

Release Information
Statement introduced in Junos OS Release 14.1X53-D45.

Description
Exclude MAC addresses from the MAC move limit algorithm.

The global MAC move feature is used to track MAC addresses when they appear on a different physical interface or within a different unit of the same physical interface. When you configure the **exclusive-mac *virtual-mac-mac-address/mask*** parameter at the [**edit protocols l2-learning global-mac-move**] hierarchy level, specified MAC addresses are excluded and will not be tracked. In addtion, excluded MAC addresses cannot be pinned when **mac-pinning** is set on an interface.

This feature can be useful in OVSDB-managed topologies with VRRP servers deployed in a redundancy configuration (master/slave), and when MAC move limit is configured. Both servers could negotiate mastership, and the same MAC address could be learned under the global MAC move feature while negotiation is occurring. In such cases, excluding the MAC address of the VRRP servers by using the **exclusive-mac** statement prevents this “false” move from being tracked.

The following example excludes VRRP V2 virtual router MAC addresses, as defined in RFC 3768:

```
[edit]
```

set protocols l2-learning global-mac-move exclusive-mac 00:00:5e:00:01:00/40

The following example excludes VRRP V3 virtual router MAC addresses, as defined in RFC 5798:

```
[edit]
```

set protocols l2-learning global-mac-move exclusive-mac 00:00:5e:00:02:00/40

Options

virtual-mac-mac-address/mask— Specify a MAC address and a mask. If the mask is 48, only the exact MAC address is excluded. If the mask is 40, all the MAC addresses that have the same first 5 bytes are excluded.

Required Privilege Level

view-level—To view this statement in the configuration.

control-level—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring MAC Move Parameters | 108

Amy: add mac-pinning exclusion topic here:

extend-secondary-vlan-id

Syntax

```
extend-secondary-vlan-id number;
```

Hierarchy Level

```
[edit vlans vlan-name pvlan]
```

Release Information

Statement introduced in Junos OS Release 12.2 for the QFX Series.

Description

Configure traffic that egresses from a secondary VLAN trunk port to retain its secondary VLAN tag instead of getting the tag of the primary VLAN that the secondary port is a member of.

Required Privilege Level

routing—To view this statement in the configuration.
routing–control—To add this statement to the configuration.

RELATED DOCUMENTATION

id-understanding-secondary-vlan-trunk-ports-and-promiscuous-access-ports-on-pvlans

Example: Configuring PVLANs with Secondary VLAN Trunk Ports and Promiscuous Access Ports on a QFX Series Switch | 566

fabric-control

Syntax

```
fabric-control {
    graceful-restart {
        restart-timeseconds;
        stale-routes-time seconds;
    }
}
```

Hierarchy Level

```
[edit fabric protocols]
```

Release Information

Statement introduced in Junos OS Release 13.2X52-D10 for the QFX Series.

Description

Specify attributes for the fabric control protocol.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- *Understanding Routing Engines in the QFabric System*

filter (VLANs)

Syntax

```
filter (input | output) filter-name;
```

Hierarchy Level

```
[edit vlans vlan-name]
```

```
[edit vlans vlan-name forwarding-options]
```

Release Information

Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.
Statement introduced in Junos OS Release 13.2X50-D10 for EX Series switches.

Description

Apply a firewall filter to traffic entering or exiting a VLAN.

Default

All incoming traffic is accepted unmodified to the VLAN, and all outgoing traffic is sent unmodified from the VLAN.

Options

filter-name —Name of a firewall filter defined in a **filter** statement.

- **input**—Apply a firewall filter to VLAN ingress traffic.
- **output**—Apply a firewall filter to VLAN egress traffic.

Required Privilege Level

system—To view this statement in the configuration.
system-control—To add this statement to the configuration.
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Example: Configuring Firewall Filters for Port, VLAN, and Router Traffic on EX Series Switches

Configuring Firewall Filters

Configuring Firewall Filters (CLI Procedure)

Overview of Firewall Filters (QFX Series)

Firewall Filters for EX Series Switches Overview

Configuring VLANs for EX Series Switches with ELS Support (CLI Procedure)

flexible-vlan-tagging

Syntax

```
flexible-vlan-tagging;
```

Hierarchy Level

```
[edit interfaces aex],
[edit interfaces ge-fpc/pic/port],
[edit interfaces et-fpc/pic/port],
[edit interfaces ps0],
[edit interfaces xe-fpc/pic/port]
```

Release Information

Statement introduced in Junos OS Release 8.1.
Support for aggregated Ethernet added in Junos OS Release 9.0.
Statement introduced in Junos OS Release 12.1x48 for PTX Series Packet Transport Routers.
Statement introduced in Junos OS Release 13.2X50-D15 for EX Series switches.
Statement introduced in Junos OS Release 13.2X51-D20 for the QFX Series.

Description

Support simultaneous transmission of 802.1Q VLAN single-tag and dual-tag frames on logical interfaces on the same Ethernet port, and on pseudowire logical interfaces.

This statement is supported on M Series and T Series routers, for Fast Ethernet and Gigabit Ethernet interfaces only on Gigabit Ethernet IQ2 and IQ2-E, IQ, and IQE PICs, and for aggregated Ethernet interfaces with member links in IQ2, IQ2-E, and IQ PICs or in MX Series DPCs, or on Ethernet interfaces for PTX Series Packet Transport Routers or 100-Gigabit Ethernet Type 5 PIC with CFP.

This statement is supported on Gigabit Ethernet, 10-Gigabit Ethernet, 40-Gigabit Ethernet, and aggregated Ethernet interfaces on EX Series and QFX Series switches.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Enabling VLAN Tagging | 298

Configuring Flexible VLAN Tagging on PTX Series Packet Transport Routers | 303

forwarding-options

Syntax

```
forwarding-options {
    dhcp-security {
        arp-inspection;
        group group-name {
            interface interface-name {
                static-ip ip-address {
                    mac mac-address;
                }
            }
            overrides {
                no-option82;
                (trusted | untrusted);
            }
        }
        ip-source-guard;
        no-dhcp-snooping;
        option-82 {
            circuit-id {
                prefix {
                    host-name;
                    logical-system-name;
                    routing-instance-name;
                }
                use-interface-description (device | logical);
                use-vlan-id;
            }
            remote-id {
                host-name hostname;
                use-interface-description (device | logical);
                use-string string;
            }
            vendor-id {
                use-string string;
            }
        }
    }
    filter (VLANs) {
        input filter-name;
        output filter-name;
    }
```

```
flood {
    input filter-name;
}
```

Chassis: EX4600 and QFX Series

```
forwarding options profile-name {
    num-65-127-prefix number;
}
```

Chassis: EX4600 and QFX Series

```
forwarding-options lpm-profile {
    prefix-65-127-disable;
    unicast-in-lpm;
}
```

Chassis: EX4600 and QFX Series

```
forwarding-options custom-profile {
    l2-entries | l3-entries | lpm-entries {
        num-banks number;
    }
}
```

Hierarchy Level

```
[edit],
[edit bridge-domains bridge-domain-name],
[edit vlans vlan-name]
```

```
[edit chassis (QFX Series)]
```

Release Information

Statement introduced before Junos OS Release 7.4.

Statement introduced in Junos OS Release 11.3 for QFX Series switches.

Hierarchy level **[edit vlans *vlan-name*]** introduced in Junos OS Release 13.2X50-D10 for EX Series switches.

Hierarchy level **[edit bridge-domains *bridge-domain-name*]** introduced in Junos OS Release 14.1 for MX Series routers.

custom-profile option introduced in Junos OS Release 15.1x53-D30 for QFX5200 Series switches only.

Description

Configure a unified forwarding table profile to allocate the amount of memory available for the following:

- MAC addresses.
- Layer 3 host entries.
- Longest prefix match table entries.

This feature enables you to select a profile that optimizes the amount of memory available for various types of forwarding-table entries based on the needs of your network. For example, for a switch that handles a great deal of Layer 2 traffic, such as a virtualized network with many servers and virtualized machines, you would choose the **l2-profile-one**, which allocates the highest amount of memory to MAC addresses.

You configure the memory allocation for LPM table entries differently, depending on whether you using Junos OS Release 13.2X51-D10 or Junos OS Release 13.2X51-D15 and later. For more information about configuring memory allocation for LPM table entries, see "Configuring the Unified Forwarding Table on Switches" on page 94.

The **num-65-127-prefix *number*** statement is not supported on the **custom-profile** and the **lpm-profile**. The **prefix-65-127-disable** and **unicast-in-lpm** statements are supported only on the **lpm-profile**.

When you commit a configuration with a forwarding table profile change, in most cases the Packet Forwarding Engine restarts automatically to apply the new parameters, which brings the data interfaces down and then up again.

However, starting with Junos OS Releases 14.1X53-D40, 15.1R5, and 16.1R3, for a Virtual Chassis or Virtual Chassis Fabric (VCF) comprised of EX4600 or QFX5100 switches, the Packet Forwarding Engine in member switches does not automatically restart upon configuring and committing a unified forwarding table profile change. This behavior avoids having Virtual Chassis or VCF instability and a prolonged convergence period if a profile change is propagated to member switches and multiple Packet Forwarding Engines all restart at the same time. In this environment, instead of automatically restarting when you initially commit a profile configuration change, the message **Reboot required for configuration to take effect** is displayed at the master switch CLI prompt, notifying you that the profile change does not take effect until the next time you restart the Virtual Chassis or VCF. The profile configuration change is propagated to member switches that support this feature, and a reminder that a reboot is required to apply this pending configuration change appears in the system log of the master switch and applicable member switches. You then enable the profile change subsequently during a planned downtime period using the **request system reboot** command, which quickly establishes a stable Virtual Chassis or VCF with the new configuration.

NOTE: You should plan to make unified forwarding table profile changes only when you are ready to perform a Virtual Chassis or VCF system reboot *immediately* after committing the configuration update. Otherwise, in the intervening period between committing the configuration change and rebooting the Virtual Chassis or VCF, the system can become inconsistent if a member experiences a problem and restarts. In that case, the new configuration takes effect on the member that was restarted, while the change is not yet activated on all the other members.

The remaining statements are explained separately. See CLI Explorer.

Options

profile-name—name of the profile to use for memory allocation in the unified forwarding table.

Table 136 on page 1177 lists the profiles you can choose that have set values and the associated values for each type of entry.

On QFX5200 Series switches only, you can also select **custom-profile**. This profile enables you to allocate from one to four banks of shared hash memory to a specific type of forwarding-table entry. Each shared hash memory bank can store a maximum of the equivalent of 32,000 IPv4 unicast addresses.

Table 136: Unified Forwarding Table Profiles

Profile Name	MAC Table	Host Table (unicast and multicast addresses)					
	MAC Addresses	IPv4 unicast	IPv6 unicast	IPv4 (*, G)	IPv4 (S, G)	IPv6 (*, G)	IPv6 (S, G)
l2-profile-one	288K	16K	8K	8K	8K	4K	4K
l2-profile-two	224K	80K	40K	40K	40K	20K	20K
l2-profile-three (default)	160K	144K	72K	72K	72K	36K	36K
l3-profile	96K	208K	104K	104K	104K	52K	52K
lpm-profile*	32K	16K	8K	8K	8K	4K	4K

* This profile supports only IPv4 in Junos OS Release13.2X51-D10. Starting in Junos OS Relase13.2X51-D15, the **lpm-profile** supports IPv4 and IPv6 entries.

NOTE: If the host stores the maximum number of entries for any given type, the entire table is full and is unable to accommodate *any* entries of any other type. For information about valid combinations of table entries see *Understanding the Unified Forwarding Table*.

l2-entries | l3-entries | lpm-entries—(**custom-profile only**) Select a type of forwarding-table entry—Layer 2, Layer 3, or LPM—to allocate a specific number of shared memory banks. You configure the amount of memory to allocate for each type of entry separately.

num-banks *number*—(**custom-profile only**) Specify the number of shared memory banks to allocate for a specific type of forwarding-table entry. Each shared memory bank stores the equivalent of 32,000 IPv4 unicast addresses.

Range: 0 through 4.

NOTE: There are four shared memory banks, which can be allocated flexibly among the three types of forwarding-table entries. To allocate no shared memory for a particular entry type, specify the number **0**. When you commit the configuration, the system issues a commit check to ensure that you have not configured more than four memory banks. You do not have to configure all four shared memory banks. By default, each entry type is allocated the equivalent of 32,000 IPv4 unicast addresses in shared memory.

Required Privilege Level

interface—To view this statement in the configuration.

interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Understanding the Unified Forwarding Table

Example: Configuring a Unified Forwarding Table Custom Profile | **90**

Configuring Traffic Forwarding and Monitoring

global-mac-limit (Protocols)

Syntax

```
global-mac-limit limit {
    packet-action drop;
}
```

Hierarchy Level

```
[edit protocols l2-learning]
```

Release Information

Statement modified in Junos OS Release 9.5.

Description

Limit the number of media access control (MAC) addresses learned from the logical interfaces on the router.

Default

131,071 MAC addresses

> **NOTE:** SRX300, SRX320, SRX340, and SRX345 devices support 16,383 addresses, and SRX1500 devices support 24,575 addresses.

Options

limit—Number of MAC addresses that can be learned on the device.

Range: 20 through 13,1071 addresses

The remaining statement is explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Example: Configuring VLANs on Security Devices | 187

global-mac-move

Syntax

```
global-mac-move {
    cooloff-time seconds;
    disable-action;
    exclusive-mac virtual-mac-mac-address/mask;
    interface-recovery-time seconds;
    notification-time seconds;
    reopen-time seconds;
    statistical-approach-wait-time seconds;
    threshold-count count;
    threshold-time seconds;
    virtual-mac mac-address /mask;
}
```

Hierarchy Level

```
[edit protocols l2-learning]
```

Release Information

Statement introduced in Junos OS Release 9.4.
Support for logical systems added in Junos OS Release 9.6.
Support for disable-action and reopen-time added in Junos OS Release 13.2.
Support for exclusive-mac added in Junos OS Release 14.1X53-D45.
Statements **cooloff-time**, **interface-recovery-time**, **statistical-approach-wait-time**, and **virtual-mac** moved from vpls-mac-move to global-mac-move hierarchy level in Junos OS Release 17.4.

Description

Set parameters for media access control (MAC) address move reporting.

Default

By default, MAC moves notify every second, with a threshold time of 1 second and a threshold count of 50.

Required Privilege Level

view-level—To view this statement in the configuration.
control-level—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring MAC Move Parameters | 108

MAC Moves Loop Prevention in VPLS Network Overview

Example: Configuring Loop Prevention in VPLS Network Due to MAC Moves

virtual-mac

global-mac-statistics

Syntax

```
global-mac-statistics;
```

Hierarchy Level

[edit protocols l2-learning]

Release Information

Statement introduced in Junos OS Release 9.2.
Support for logical systems added in Junos OS Release 9.6.

Description

(MX Series routers and EX Series switches only) Enable MAC accounting for the entire router or switch.

Default

disabled

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Enabling MAC Accounting

global-mac-table-aging-time

Syntax

```
global-mac-table-aging-time seconds;
```

Hierarchy Level

```
[edit protocols l2-learning]
```

Release Information

Statement introduced in Junos OS Release 9.2.
Statement modified in Junos OS Release 9.5.
Support for logical systems added in Junos OS Release 9.6.

Description

Configure the timeout interval for entries in the MAC table.

> **NOTE:** The **global-mac-table-aging-time** statement appears in the Junos OS CLI for devices that support the Enhanced Layer 2 Software (ELS) configuration style. If your device runs software that does *not* support ELS, use the **mac-table-aging-time** statement, which appears in the **[edit ethernet-switching-options]** and the **[edit vlans]** hierarchies. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50.

Default

300 seconds

Options

seconds—Time elapsed before MAC table entries are timed out and entries are deleted from the table.

Range: For MX Series routers: 10 through 1 million; for EX Series and QFX Series switches: 60 through 1 million; for SRX devices: 10 through 64,000 seconds

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring the MAC Table Timeout Interval

global-mode (Protocols)

Syntax

```
global-mode (switching | transparent-bridge) ;
```

Hierarchy Level

```
[edit protocols l2-learning]
```

Release Information

Statement introduced in Junos OS Release 15.1X49-D40.

Description

Specify the global mode for the SRX Series device as Layer 2 transparent bridge mode or switching mode. After changing the mode, you must reboot the device for the configuration to take effect.

Default

On SRX1500, the default Layer 2 global mode is transparent-bridge mode.

Starting with Junos OS Release 15.1X49-D100, on SRX300, SRX320, SRX340, SRX345, SRX550, and SRX550M devices, the default Layer 2 global mode configuration is changed from transparent-bridge to switching mode.

> **NOTE:** You must explicitly configure Layer 2 transparent-bridge mode for the SRX300, SRX320, SRX340, SRX345, SRX550, and SRX550M devices that work in transparent mode. Use the command **set protocols l2-learning global-mode transparent-bridge** before rebooting the devices with Junos OS 15.1X49-D100 image.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

l2-learning | **1229**

Ethernet Switching and Layer 2 Transparent Mode Overview | **41**

global-no-mac-learning

Syntax

```
global-no-mac-learning;
```

Hierarchy Level

```
[edit protocols l2-learning],
[edit protocols l2-learning]
```

Release Information

Statement introduced in Junos OS Release 9.2.
Statement modified for SRX Series in Junos OS Release 9.5.
Support for logical systems added in Junos OS Release 9.6.

Description

Disable MAC learning on the entire device.

Default

MAC learning is enabled.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Disabling Layer 2 Learning and Forwarding | 104
Understanding Q-in-Q Tunneling and VLAN Translation | 887
Example: Configuring VLANs on Security Devices | 187

gratuitous-arp-reply

Syntax

```
(gratuitous-arp-reply | no-gratuitous-arp-reply);
```

Hierarchy Level

```
[edit interfaces interface-name]
[edit interfaces interface-range interface-range-name]
```

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 in EX Series switches.
Statement introduced in Junos OS Release 12.2 for ACX Series Universal Metro Routers.

Description

For Ethernet interfaces, enable updating of the Address Resolution Protocol (ARP) cache for gratuitous ARPs.

Default

Updating of the ARP cache is disabled on all Ethernet interfaces.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring Gratuitous ARP | 349
no-gratuitous-arp-request | 1282

group (Redundant Trunk Groups)

Syntax

```
group name {
   interface interface-name <primary>;
      interface interface-name;
   preempt-cutover-timer seconds;
}
```

Hierarchy Level

- For platforms with ELS:

  ```
  [edit switch-options redundant-trunk-group]
  ```

- For platforms without ELS:

  ```
  [edit ethernet-switching-options redundant-trunk-group]
  ```

Release Information

Statement introduced in Junos OS Release 9.0 for EX Series switches.
Hierarchy level **[edit switch-options]** introduced in Junos OS Release 13.2X50-D10 (ELS). (See "Using the Enhanced Layer 2 Software CLI" on page 50 for information about ELS.)
Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series.

Description

Create a redundant trunk group.

Options

name—The name of the redundant trunk group.

- For platforms with ELS:

  The group name must be a string "rtg*n*" where *n* is a number from 0 through 15, such as "rtg2" or "rtg10".

- For platforms without ELS:

  The group name must start with a letter and can consist of letters, numbers, dashes, and underscores.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

system—To view this statement in the configuration.
system–control—To add this statement to the configuration.

RELATED DOCUMENTATION

Example: Configuring Redundant Trunk Links for Faster Recovery on EX Series Switches | 946

Example: Configuring Redundant Trunk Links for Faster Recovery on Devices with ELS Support | 940

Understanding Redundant Trunk Links (Legacy RTG Configuration) | 937

guard-interval

Syntax

```
guard-interval number;
```

Hierarchy Level

```
[edit protocols protection-group ethernet-ring ring-name]
```

Release Information

Statement introduced in Junos OS Release 9.4.
Statement introduced in Junos OS Release 12.1 for EX Series switches.
Statement introduced in Junos OS Release 14.153-D10 for QFX Series switches.

Description

When a link goes down, the ring protection link (RPL) activates. When the downed link comes back up, the RPL link receives notification, restores the link, and waits for the restore interval before issuing another block on the same link. This configuration is a global configuration and applies to all Ethernet rings if the Ethernet ring does not have a more specific configuration for this value. If no parameter is configured at the protection group level, the global configuration of this parameter uses the default value.

Options

number—Guard timer interval, in milliseconds.

Range: 10 through 2000 ms

Default: 500 ms

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Ethernet Ring Protection Switching Overview

Example: Configuring Ethernet Ring Protection Switching on EX Series Switches | **855**

Example: Configuring Ethernet Ring Protection Switching on QFX Series and EX Series Switches Supporting ELS | **874**

Configuring Ethernet Ring Protection Switching on Switches (CLI Procedure)

hold-interval (Protection Group)

Syntax

```
hold-interval number;
```

Hierarchy Level

```
[edit protocols protection-group ethernet-ring name]
```

Release Information

Statement introduced in Junos OS Release 9.4.

Description

Specify the hold-off timer interval *for all rings* in 100 millisecond (ms) increments.

Options

number—Hold-timer interval, in milliseconds.

Range: 0 through 10,000 ms

Default: 100 ms

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Ethernet Ring Protection Switching Overview

Example: Configuring Ethernet Ring Protection Switching on QFX Series and EX Series Switches Supporting ELS | **874**

host-inbound-traffic

Syntax

```
host-inbound-traffic {
    protocols protocol-name {
        except;
    }
    system-services service-name {
        except;
    }
}
```

Hierarchy Level

[edit security zones functional-zone management],
[edit security zones functional-zone management interfaces *interface-name*],
[edit security zones security-zone *zone-name*],
[edit security zones security-zone *zone-name* interfaces *interface-name*]

Release Information

Statement introduced in Junos OS Release 8.5.

Description

Control the type of traffic that can reach the device from interfaces bound to the zone.

Options

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Understanding How to Control Inbound Traffic Based on Traffic Types

Understanding How to Control Inbound Traffic Based on Protocols

inner-tag-protocol-id

Syntax

```
inner-tag-protocol-id tpid;
```

Hierarchy Level

```
[edit interfaces interface-name unit logical-unit-number input-vlan-map],
[edit interfaces interface-name unit logical-unit-number output-vlan-map],
[edit logical-systems logical-system-name interfaces interface-name unit logical-unit-number input-vlan-map],
[edit logical-systems logical-system-name interfaces interface-name unit logical-unit-number output-vlan-map]
```

Release Information

Statement introduced in Junos OS Release 8.1.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.

Description

Configure the IEEE 802.1Q TPID value to rewrite for the inner tag.

All TPIDs you include in input and output VLAN maps must be among those you specify at the **[edit interfaces *interface-name* gigether-options ethernet-switch-profile tag-protocol-id [*tpids*]]** hierarchy level.

On MX Series routers, you can use this statement for Gigabit Ethernet IQ, IQ2 and IQ2-E interfaces, and for aggregated Ethernet interfaces using Gigabit Ethernet IQ2 and IQ2-E or 10-Gigabit Ethernet PICs.

Default

If the **inner-tag-protocol-id** statement is not configured, the TPID value is 0x8100.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring Inner and Outer TPIDs and VLAN IDs | 388

inner-vlan-id

Syntax

```
inner-vlan-id number;
```

Hierarchy Level

```
[edit interfaces interface-name unit logical-unit-number input-vlan-map],
[edit interfaces interface-name unit logical-unit-number output-vlan-map],
[edit logical-systems logical-system-name interfaces interface-name unit logical-unit-number input-vlan-map],
[edit logical-systems logical-system-name interfaces interface-name unit logical-unit-number output-vlan-map]
```

Release Information

Statement introduced in Junos OS Release 8.1.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.

Description

For Gigabit Ethernet IQ, IQ2 and IQ2-E interfaces, and for aggregated Ethernet interfaces using Gigabit Ethernet IQ2 and IQ2-E or 10-Gigabit Ethernet PICs on MX Series routers or 100-Gigabit Ethernet Type 5 PIC with CFP, or on Ethernet interfaces on EX Series switches, specify the VLAN ID to rewrite for the inner tag of the final packet.

You cannot include the **inner-vlan-id** statement with the **swap** statement, **swap-push** statement, **push-push** statement, or **push-swap** statement and the **inner-vlan-id** statement at the **[edit interfaces *interface-name* unit *logical-unit-number*output-vlan-map]** hierarchy level. If you include any of those statements in the output VLAN map, the VLAN ID in the outgoing frame is rewritten to the **inner-vlan-id** statement you include at the **[edit interfaces *interface-name* unit *logical-unit-number*]** hierarchy level.

Options

number—VLAN ID number.

Range: 0 through 4094

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring Inner and Outer TPIDs and VLAN IDs | 388

input-native-vlan-push

Syntax

```
input-native-vlan-push (disable | enable);
```

Hierarchy Level

```
[edit interfaces interface-name]
```

Release Information

Statement introduced in Junos OS Release 19.3R1 on EX Series and QFX Series switches.

Description

(On EX2300, EX3400, EX4300, EX4300-MP, EX4600, EX4650, and the QFX5000 line of switches) For a Q-in-Q tunneling configuration, enable or disable whether the switch inserts a native VLAN identifier in untagged frames received on the C-VLAN interface when the configuration statement **input-vlan-map** with a **push** operation is configured.

Options

disable—Disable **input-native-vlan-push**.

enable—Enable **input-native-vlan-push**.

Default:

- (EX4300 only) **disable**
- (EX2300, EX3400, EX4300-MP, EX4600, EX4650, and QFX5000 line) **enable**

Required Privilege Level

interface

RELATED DOCUMENTATION

Configuring Q-in-Q Tunneling and VLAN Q-in-Q Tunneling and VLAN Translation | 887

input-vlan-map

Syntax

```
input-vlan-map {
    (pop | pop-pop | pop-swap | push | push-push | swap | swap-push | swap-swap);
    inner-tag-protocol-id tpid;
    inner-vlan-id number;
    tag-protocol-id tpid;
    vlan-id number;
}
```

Hierarchy Level

```
[edit interfaces interface-name unit logical-unit-number],
[edit logical-systems logical-system-name interfaces interface-name unit logical-unit-number]
```

Release Information

Statement introduced before Junos OS Release 7.4.
pop-pop, **pop-swap**, **push-push**, **swap-push**, and **swap-swap** statements introduced in Junos OS Release 8.1.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.
Statement introduced in Junos OS Release 13.2X50-D15 for EX Series switches.
Statement introduced in Junos OS Release 13.2X51-D20 for the QFX Series.

Description

For Gigabit Ethernet IQ, 10-Gigabit Ethernet SFPP interfaces, 100-Gigabit Ethernet Type 5 PIC with CFP only as well as Gigabit Ethernet, 10-Gigabit Ethernet, 40-Gigabit Ethernet, and aggregated Ethernet interfaces, define the rewrite profile to be applied to incoming frames on this logical interface.

The remaining statements are explained separately. See CLI Explorer.

> **NOTE:** Connectivity fault management (CFM) sessions for all interfaces in which **input-vlan-map** is configured are supported only if the interface also has an explicit configuration for **output-vlan-map** as **output-vlan-map pop;**. See **output-vlan-map**. This configuration is required for all the interfaces in the topology even when the CFM session is on that interface or on a different interface in the data path of the same topology.

Required Privilege Level

interface—To view this statement in the configuration.

interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Stacking a VLAN Tag | **393**

output-vlan-map | **1294**

Configuring Q-in-Q Tunneling on EX Series Switches with ELS Support | **900**

instance-type

Syntax

```
instance-type type;
```

Hierarchy Level

```
[edit logical-systems logical-system-name routing-instances routing-instance-name],
[edit routing-instances routing-instance-name]
```

Release Information

Statement introduced before Junos OS Release 7.4.
virtual-switch and **layer2-control** options introduced in Junos OS Release 8.4.
Statement introduced in Junos OS Release 9.2 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 12.3 for ACX Series routers.
mpls-internet-multicast option introduced in Junos OS Release 11.1 for the EX Series, M Series, MX Series, and T Series.
evpn option introduced in Junos OS Release 13.2 for MX 3D Series routers.
evpn option introduced in Junos OS Release 17.3 for the QFX Series.
forwarding option introduced in Junos OS Release 14.2 for the PTX Series.
mpls-forwarding option introduced in Junos OS Release 16.1 for the MX Series.
evpn-vpws option introduced in Junos OS Release 17.1 for MX Series routers.
Support for logical systems on MX Series routers added in Junos OS Release 17.4R1.

Description

Define the type of routing instance.

Options

> **NOTE:** On ACX Series routers, you can configure only the forwarding, virtual router, and VRF routing instances.

type—Can be one of the following:

- **evpn**—(MX 3D Series routers, QFX switches and EX9200 switches)—Enable an Ethernet VPN (EVPN) on the routing instance.
  hierarchy level.
- **evpn-vpws**—Enable an Ethernet VPN (EVPN) Virtual Private Wire Service (VPWS) on the routing instance.

- **forwarding**—Provide support for filter-based forwarding, where interfaces are not associated with instances. All interfaces belong to the default instance. Other instances are used for populating RPD learned routes. For this instance type, there is no one-to-one mapping between an interface and a routing instance. All interfaces belong to the default instance inet.0.
- **l2backhaul-vpn**—Provide support for Layer 2 wholesale VLAN packets with no existing corresponding logical interface. When using this instance, the router learns both the outer tag and inner tag of the incoming packets, when the **instance-role** statement is defined as **access**, or the outer VLAN tag only, when the **instance-role** statement is defined as **nni**.
- **l2vpn**—Enable a Layer 2 VPN on the routing instance. You must configure the **interface**, **route-distinguisher**, **vrf-import**, and **vrf-export** statements for this type of routing instance.
- layer2-control—(MX Series routers only) Provide support for RSTP or MSTP in customer edge interfaces of a VPLS routing instance. This instance type cannot be used if the customer edge interface is multihomed to two provider edge interfaces. If the customer edge interface is multihomed to two provider edge interfaces, use the default BPDU tunneling.
- **mpls-forwarding**—(MX Series routers only) Allow filtering and translation of route distinguisher (RD) values in IPv4 and IPv6 VPN address families on both routes received and routes sent for selected BGP sessions. In particular, for Inter-AS VPN Option-B networks, this option can prevent the malicious injection of VPN labels from one peer AS boundary router to another.
- **mpls-internet-multicast**—(EX Series, M Series, MX Series, and T Series routers only) Provide support for ingress replication provider tunnels to carry IP multicast data between routers through an MPLS cloud, using MBGP or next-generation MVPN.
- **no-forwarding**—This is the default routing instance. Do not create a corresponding forwarding instance. Use this routing instance type when a separation of routing table information is required. There is no corresponding forwarding table. All routes are installed into the default forwarding table. IS-IS instances are strictly nonforwarding instance types.
- **virtual-router**—Enable a virtual router routing instance. This instance type is similar to a VPN routing and forwarding instance type, but used for non-VPN-related applications. You must configure the **interface** statement for this type of routing instance. You do not need to configure the **route-distinguisher**, **vrf-import**, and **vrf-export** statements.
- **virtual-switch**—(MX Series routers, EX9200 switches, and QFX switches only) Provide support for Layer 2 bridging. Use this routing instance type to isolate a LAN segment with its Spanning Tree Protocol (STP) instance and to separate its VLAN identifier space.

- **vpls**—Enable VPLS on the routing instance. Use this routing instance type for point-to-multipoint LAN implementations between a set of sites in a VPN. You must configure the **interface**, **route-distinguisher**, **vrf-import**, and **vrf-export** statements for this type of routing instance.
- **vrf**—VPN routing and forwarding (VRF) instance. Provides support for Layer 3 VPNs, where interface routes for each instance go into the corresponding forwarding table only. Required to create a Layer 3 VPN. Create a VRF table (***instance-name*.inet.0**) that contains the routes originating from and destined for a particular Layer 3 VPN. For this instance type, there is a one-to-one mapping between an interface and a routing instance. Each VRF instance corresponds with a forwarding table. Routes on an interface go into the corresponding forwarding table. You must configure the **interface**, **route-distinguisher**, **vrf-import**, and **vrf-export** statements for this type of routing instance.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- *Configuring the Instance Type*
- *Configuring EVPN Routing Instances*
- *Configuring EVPN Routing Instances on EX9200 Switches*
- *Configuring Virtual Router Routing Instances*
- *Example: Configuring Filter-Based Forwarding on the Source Address*
- *Example: Configuring Filter-Based Forwarding on Logical Systems*

inter-switch-link

Syntax

```
inter-switch-link vlan members primary-vlan-name;
```

Hierarchy Level

```
[edit interfaces interface-name unit logical-unit-number family ethernet-switching interface-mode trunk]
```

Release Information

Statement introduced in Junos OS Release 14.1X53-D10 for EX Series switches.

Description

Use this configuration statement when a private VLAN (PVLAN) spans multiple switches. The Inter-Switch Link protocol (ISL) must be configured on a trunk port of the primary VLAN in order to connect the switches composing the PVLAN to each other. You do not need to configure an ISL when a PVLAN is configured on a single switch.

Required Privilege Level

system—To view this statement in the configuration.
system–control—To add this statement to the configuration.

RELATED DOCUMENTATION

Creating a Private VLAN Spanning Multiple EX Series Switches with ELS Support (CLI Procedure) | 481

interface

Syntax

```
interface interface-name;
```

Hierarchy Level

```
[edit bridge-domains bridge-domain-name],
[edit logical-systems logical-system-name routing-instances routing-instance-name bridge-domains bridge-domain-name],
[edit routing-instances routing-instance-name bridge-domains bridge-domain-name],
[edit vlans vlan-name]
```

Release Information

Statement introduced in Junos OS Release 8.4.
Support for top-level configuration for the **virtual-switch** type of routing instance added in Junos OS Release 9.2.
In Junos OS Release 9.1 and earlier, the routing instances hierarchy supported this statement only for a VPLS instance or a bridge domain configured within a virtual switch.
Support for logical systems added in Junos OS Release 9.6.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.
Statement introduced in Junos OS Release 12.3X52 for ACX Series routers.
Statement introduced in Junos OS Release 15.1.

Description

(MX Series routers and EX Series switches only) Specify the logical interfaces to include in the bridge domain, VLAN, VPLS instance, or virtual switch.

Options

interface-name—Name of a logical interface. For more information about how to configure logical interfaces, see the *Junos OS Network Interfaces Library for Routing Devices*.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- *Configuring a Bridge Domain*
- *Configuring a Layer 2 Virtual Switch*

interface (MVRP)

Syntax

```
interface (all | interface-name) {
    disable;
    join-timer milliseconds;
    leave-timer milliseconds;
    leaveall-timer milliseconds;
    registration (forbidden | normal);
}
```

Syntax

```
interface (all | interface-name) {
    join-timer milliseconds;
    leave-timer milliseconds;
    leaveall-timer milliseconds;
    point-to-point;
    registration (forbidden | normal | restricted);
}
```

Hierarchy Level

[edit protocols mvrp]

[edit routing-instances *routing-instance-name* protocols mvrp] (for virtual switch instance type)

[edit logical-systems *logical-system-name* protocols mvrp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvrp] (for virtual switch instance type),

Release Information

Statement introduced in Junos OS Release 10.0 for EX Series switches.
Statement introduced in Junos OS Release 10.1 for MX Series routers.
Statement introduced in Junos OS Release 13.1 for the QFX Series.
Statement introduced in Junos OS Release 15.1X49-D70 for SRX Series devices.

Description

Specify interfaces on which to configure Multiple VLAN Registration Protocol (MVRP).

NOTE: On QFX Series switches, you must configure specific interfaces—you cannot specify **interface all**. You can enable MVRP on an interface range.

Default

By default, MVRP is disabled.

Options

all—All interfaces on the switch.

interface-name—Names of interface to be configured for MVRP.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches | **832**
- Configuring Multiple VLAN Registration Protocol (MVRP) on Switches | **797**
- Understanding Multiple VLAN Registration Protocol (MVRP) for Dynamic VLAN Registration | **793**
- Understanding Multiple VLAN Registration Protocol (MVRP) for Dynamic VLAN Registration | **793**
- *Example: Configuring Automatic VLAN Administration Using MVRP on MX Series Routers*
- *Configuring Multiple VLAN Registration Protocol (MVRP) to Manage Dynamic VLAN Registration*
- Verifying That MVRP Is Working Correctly | **851**

interface (Layer 2 Protocol Tunneling)

Syntax

```
interface interface-name {
    enable-all-ifl;
    protocol protocol-name;
}
```

Hierarchy Level

```
[edit logical-systems name protocols layer2-control mac-rewrite],
[edit protocols layer2-control mac-rewrite]
```

Release Information

Statement introduced in Junos OS Release 9.1.
Statement introduced in Junos OS Release 13.2X50-D10 for EX Series switches.
enable-all-if statement added in Junos OS Release 13.3.
Support for PVSTP protocol introduced in Junos OS Release 13.3.
Statement introduced in Junos OS Release 14.1X53-D10 and 17.3R1 for EX4300 switches.
Statement introduced in Junos OS Release 15.1X53-D55 and 18.2R1 for EX2300 and EX3400 switches.
Statement introduced in Junos OS Release 17.4R1 for EX4600 switches.
Statement introduced in Junos OS Release 19.1R1 for QFX Series switches.

Description

Configure an interface for Layer 2 protocol tunneling.

> **NOTE:** The **enable-all-ifl** option is available on EX9200 switches but not on other EX Series switches.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Understanding Layer 2 Protocol Tunneling | 684

interface (Redundant Trunk Groups)

Syntax

```
interface interface-name <primary>;
interface interface-name;
```

Hierarchy Level

For platforms with ELS:

```
[edit switch-options redundant-trunk-group group name]
```

For platforms without ELS:

```
[edit ethernet-switching-options redundant-trunk-group group name]
```

Release Information

Statement introduced in Junos OS Release 9.0 for EX Series switches.
Hierarchy level **[edit switch-options]** introduced in Junos OS Release 13.2X50-D10 (ELS). (See "Using the Enhanced Layer 2 Software CLI" on page 50 for information about ELS.)
Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series.

Description

Configure a primary link and secondary link on trunk ports. If the primary link fails, the secondary link automatically takes over as the primary link without waiting for normal STP convergence.

Options

interface *interface-name*—A logical interface or an aggregated interface containing multiple ports.

primary—(Optional) Specify one of the interfaces in the redundant group as the primary link. The interface without this option is the secondary link in the redundant group. If a link is not specified as **primary**, the software compares the two links and selects the link with the highest port number as the active link. For example, if the two interfaces are **ge-0/1/0** and **ge-0/1/1**, the software assigns **ge-0/1/1** as the active link.

Required Privilege Level

system—To view this statement in the configuration.
system–control—To add this statement to the configuration.

RELATED DOCUMENTATION

interface (Routing Instances)

Syntax

```
interface interface-name {
    description text;
}
```

Hierarchy Level

```
[edit logical-systems logical-system-name routing-instances routing-instance-name],
[edit routing-instances routing-instance-name]
```

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.2 for EX Series switches.
Statement introduced in Junos OS Release 12.3 for ACX Series routers.
Statement introduced in Junos OS Release 13.2 for MX 3D Series routers.

Description

Specify the interface over which the VPN traffic travels between the PE device and CE device. You configure the interface on the PE device. If the value **vrf** is specified for the **instance-type** statement included in the routing instance configuration, this statement is required.

Options

interface-name—Name of the interface.

The remaining statement is explained separately. Search for a statement in CLI Explorer or click a linked statement in the Syntax section for details.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- *Configuring Interfaces for VPN Routing*
- *Configuring EVPN Routing Instances*
- *Configuring EVPN Routing Instances on EX9200 Switches*
- *interface (VPLS Routing Instances)*

interface (Switching Options)

Syntax

```
interface interface-name {
    encapsulation-type;
    ignore-encapsulation-mismatch;
    pseudowire-status-tlv;
    static-mac mac-address {
        vlan-id vlan-id;
    }
}
```

Hierarchy Level

```
[edit vlans vlans-name switch-options]
```

Release Information

Statement modified in Junos OS Release 9.5.

Description

Specify the logical interfaces to include in the VLAN.

Options

- ***interface-name***—Name of a logical interface.
- ***encapsulation-type***—Encapsulation type for VPN.
- ***ignore-encapsulation-mismatch***—Allow different encapsulation types on local and remote devices.
- ***pseudowire-status-tlv***—Send pseudowire status.
- ***mac-address***—Static MAC address assigned to the logical interface.
- ***vlan-id***—VLAN identifier.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

interface (VLANs)

List of Syntax

Syntax (QFX Series, QFabric, NFX Series and EX4600)

```
interface interface-name {
    mapping (native (push | swap) | tag (push | swap));
}
```

Syntax (EX Series and SRX210)

```
interface interface-name; {
    egress;
    ingress;
    mapping (native (push | swap) | policy | tag (push | swap));
    pvlan-trunk;
}
```

QFX Series, QFabric, NFX Series and EX4600

```
[edit vlans vlan-name]
```

EX Series and SRX210

```
[edit vlans vlan-name],
[edit vlans vlan-name],
[edit vlans vlan-name vlan-id number],
[edit vlans vlan-name vlan-id number],
[edit vlans vlan-name vlan-id-list number]
```

Release Information

Statement introduced in Junos OS Release 9.3 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description

For a specific VLAN, configure an interface.

Options

interface-name—Name of the Ethernet interface

The remaining statement is explained separately. See CLI Explorer.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.
system—To view this statement in the configuration.system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- Example: Setting Up Basic Bridging and a VLAN on Switches | 203
- Configuring VLANs on Switches | 182
- Configuring VLANs for EX Series Switches | 183
- Configuring Q-in-Q Tunneling on EX Series Switches | 910
- Configuring Q-in-Q Tunneling on EX Series Switches with ELS Support | 900

interface-mac-limit

Syntax

```
interface-mac-limit {
    limit
    disable;
    packet-action ;
}
```

Hierarchy Level

[edit **bridge-domains** *bridge-domain-name* bridge-options],
[edit **bridge-domains** *bridge-domain-name* bridge-options interface *interface-name*],
[edit logical-systems *logical-system-name* **bridge-domains** *bridge-domain-name* bridge-options],
[edit logical-systems *logical-system-name* **bridge-domains** *bridge-domain-name* bridge-options interface *interface-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* **bridge-domains** *bridge-domain-name* bridge-options],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* **bridge-domains** *bridge-domain-name* bridge-options interface *interface-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* switch-options],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* switch-options interface *interface-name*],
[edit logical-systems *logical-system-name* switch-options],
[edit logical-systems *logical-system-name* switch-options interface *interface-name*],
[edit routing-instances *routing-instance-name* **bridge-domains** *bridge-domain-name* bridge-options],
[edit routing-instances *routing-instance-name* **bridge-domains** *bridge-domain-name* bridge-options interface *interface-name*],
[edit routing-instances *routing-instance-name* switch-options],
[edit routing-instances *routing-instance-name* switch-options interface *interface-name*],
[edit switch-options],
[edit switch-options],
[edit switch-options interface *interface-name*],
[edit switch-options interface *interface-name*],
[edit vlans *vlan-name* switch-options],
[edit vlans *vlan-name* switch-options interface *interface-name*]

Release Information

Statement introduced in Junos OS Release 8.4.
Support for the **switch-options** statement added in Junos OS Release 9.2.
Support for top-level configuration for the **virtual-switch** type of routing instance added in Junos OS Release 9.2. In Junos OS Release 9.1 and earlier, the routing instances hierarchy supported this statement only for a VPLS instance or a bridge domain configured within a virtual switch.

Support for logical systems added in Junos OS Release 9.6.
[edit switch-options], [edit switch-options interface *interface-name*], [edit vlans *vlan-name* switch-options], and [edit vlans *vlan-name* switch-options interface *interface-name*] hierarchy levels introduced in Junos OS Release 12.3R2 for EX Series switches.

Description

Configure a limit to the number of MAC addresses that can be learned from a bridge domain, VLAN, virtual switch, or set of bridge domains or VLANs.

> **NOTE:** For multichassis link aggregation (MC-LAG) peers in active-active mode, configuring the **interface-mac-limit** statement or changing the **interface-mac-limit** configuration when traffic is flowing can cause the MAC entries to be out of synchronization between the two MC-LAG peers, which might result in flooding. To avoid flooding, you must either halt traffic forwarding and then configure the **interface-mac-limit** statement or use the **commit at** configuration statement to commit the changes at the same time in both the peer nodes.
>
> Alternatively, if flooding does occur, you can clear the bridge MAC table on both the routers or switches by using the **clear bridge mac-table** command. Running this command ensures that the MAC entries are re-learned and in synchronization between both the peers.

Default

The default MAC limit varies with the platform.

Options

disable—Disables the global interface-mac-limit configuration on an interface and sets the maximum interface-mac-limit that is permitted on the device.

limit—Sets the maximum number of MAC addresses learned from an interface.

Range: 1 through <default MAC limit> MAC addresses per interface. Range is platform specific.

If you configure both **disable** and **limit**, disable takes precedence and packet-action is set to **none**. The remaining statement is explained separately.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Understanding Layer 2 Learning and Forwarding for Bridge Domains

Layer 2 Learning and Forwarding for VLANs Overview | **78**

Understanding Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports

Layer 2 Learning and Forwarding for VLANs Acting as a Switch for a Layer 2 Trunk Port | **80**

interface-mode

Syntax

```
interface-mode (access | trunk <inter-switch-link>);
```

Hierarchy Level

[edit interfaces *interface-name* unit *logical-unit-number* family bridge],
[edit interfaces *interface-name* unit *logical-unit-number* family ethernet-switching],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family bridge]

Release Information

Statement introduced in Junos OS Release 9.2.
Statement introduced in Junos OS Release 13.2X50-D10 for EX Series switches.
Statement introduced in Junos OS Release 13.2 for the QFX Series.
Statement introduced in Junos OS Release 15.1.
inter-switch-link option introduced in Junos OS Release 14.2 for MX240, MX480, and MX960 routers in enhanced LAN mode.

Description

NOTE: This statement supports the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does not support ELS, see **port-mode**. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50.

QFX3500 and QFX3600 standalone switches—Determine whether the logical interface accepts or discards packets based on VLAN tags. Specify the **trunk** option to accept packets with a VLAN ID that matches the list of VLAN IDs specified in the **vlan-id** or **vlan-id-list** statement, then forward the packet within the bridge domain or VLAN configured with the matching VLAN ID. Specify the **access** option to accept packets with no VLAN ID, then forward the packet within the bridge domain or VLAN configured with the VLAN ID that matches the VLAN ID specified in the **vlan-id** statement.

NOTE: On MX Series routers, if you want IGMP snooping to be functional for a bridge domain, then you should not configure **interface-mode** and **irb** for that bridge. Such a configuration commit succeeds, but IGMP snooping is not functional, and a message informing the same is displayed. For more information, see "Configuring a Trunk Interface on a Bridge Network" on page 325.

Options

access—Configure a logical interface to accept untagged packets. Specify the VLAN to which this interface belongs using the **vlan-id** statement.

trunk—Configure a single logical interface to accept packets tagged with any VLAN ID specified with the **vlan-id** or **vlan-id-list** statement.

trunk inter-switch-link—For a private VLAN, configure the InterSwitch Link protocol (ISL) on a trunk port of the primary VLAN in order to connect the switches composing the PVLAN to each other. You do not need to configure an ISL when a PVLAN is configured on a single switch. This configuration specifies whether the particular interface assumes the role of interswitch link for the PVLAN domains of which it is a member. This option is supported only on MX240, MX480, and MX960 routers in enhanced LAN mode.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- Configuring Access Mode on a Logical Interface | **322**
- Configuring a Logical Interface for Trunk Mode | **323**
- Example: Connecting Access Switches with ELS Support to a Distribution Switch with ELS Support | **251**
- *Tunnel Services Overview*
- *Tunnel Interface Configuration on MX Series Routers Overview*

interfaces (Q-in-Q Tunneling)

Syntax

```
interfaces interface-name {
    no-mac-learning;
}
```

Hierarchy Level

```
[edit ethernet-switching-options]
```

Release Information

Statement introduced in Junos OS Release 9.5 for EX Series switches.

Description

Configure settings for interfaces that have been assigned to family **ethernet-switching**.

Options

interface-name --Name of an interface that is configured for **family ethernet-switching**.

The remaining statement is explained separately. See CLI Explorer.

Required Privilege Level

system—To view this statement in the configuration.
system–control—To add this statement to the configuration.

RELATED DOCUMENTATION

Understanding Q-in-Q Tunneling and VLAN Translation | 887

interfaces (Security Zones)

Syntax

```
interfaces interface-name {
    host-inbound-traffic {
        protocols protocol-name {
            except;
        }
        system-services service-name {
            except;
        }
    }
}
```

Hierarchy Level

```
[edit security zones functional-zone management],
[edit security zones security-zone zone-name]
```

Release Information

Statement introduced in Junos OS Release 8.5.

Description

Specify the set of interfaces that are part of the zone.

Options

interface-name —Name of the interface.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Understanding Security Zones

interfaces

List of Syntax

Syntax (QFX Series) on page 1219
Syntax (EX Series, MX Series and T Series) on page 1219

Syntax (QFX Series)

```
interfaces interface-name {
    no-mac-learning;
}
```

Syntax (EX Series, MX Series and T Series)

```
interfaces { ... }
```

QFX Series

```
[edit ethernet-switching-options]
```

EX Series, MX Series and T Series

```
[edit]
```

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description

Configure settings for interfaces that have been assigned to family **ethernet-switching**.

Default

The management and internal Ethernet interfaces are automatically configured. You must configure all other interfaces.

Options

interface-name —Name of an interface that is configured for **family ethernet-switching**.

The remaining statement is explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.

routing-control—To add this statement to the configuration.
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Physical Interface Properties Overview

Configuring Aggregated Ethernet Link Protection

isid

Syntax

```
isid isid-number vlan-id-list [ vlan-ids ] {
    source-bmac <mac-address> <length>
}
```

Hierarchy Level

```
[edit routing-instances routing-instance-name service-groups service-group-name ]
```

Release Information

Statement introduced in JUNOS Release 10.0.

Description

For IEEE 802.1ah provider backbone bridge (PBB) configurations, configure the service identifier (I-SID) for the customer routing instance (I-component) service group.

Options

isid—Service identifier. Enter an I-SID in the range from **256** through **16777214**.

vlan-id-list [*vlan-ids*]—List of service VLANs (S-VLANs).

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

isid-list

Syntax

```
isid-list all-service-groups;
```

Hierarchy Level

```
[edit interfaces pseudo-logical-interface-name unit logical-unit-number family bridge]
```

Release Information
Statement introduced in JUNOS Release 10.0.

Description
For IEEE 802.1ah provider backbone bridge (PBB) configurations, map all service identifiers (I-SIDs) specified for the service groups.

Options
all-service-groups—Map all service identifiers (I-SIDs) for the specified service groups.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

isolated

Syntax

```
isolated;
```

Hierarchy Level

```
[edit vlans vlan-name interface interface-name]
```

Release Information

Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description

Configure an access or trunk port to be isolated. You configure a trunk port to be isolated so that it can be a secondary VLAN trunk port–that is, it can carry secondary VLAN traffic.

Required Privilege Level

system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- Creating a Private VLAN on a Single QFX Switch | **475**
- Creating a Private VLAN Spanning Multiple QFX Series Switches | **479**
- id-understanding-secondary-vlan-trunk-ports-and-promiscuous-access-ports-on-pvlans
- Example: Configuring PVLANs with Secondary VLAN Trunk Ports and Promiscuous Access Ports on a QFX Series Switch | **566**

isolated-vlan

Syntax

```
isolated-vlan vlan-name isolated-vlan-name vlan-id isolated-vlan-id;
```

Hierarchy Level

```
[edit vlans primary-vlan-name vlan-id primary-vlan-vlan-id]
```

Release Information

Statement introduced in Junos OS Release 14.1X53-D10 for EX Series switches.
Statement introduced in Junos OS Release 14.1X53-D15 for QFX Series switches.

Description

Configure the specified isolated VLAN to be a secondary VLAN of the specified primary VLAN. An isolated VLAN receives packets only from the primary VLAN and forwards frames upstream to the primary VLAN.

NOTE: Before you specify this configuration statement, you must have already configured an isolated VLAN and assigned a VLAN ID to it. See **private-vlan**.

Required Privilege Level

system—To view this statement in the configuration.
system–control—To add this statement to the configuration.

RELATED DOCUMENTATION

Creating a Private VLAN on a Single Switch with ELS Support (CLI Procedure) | **472**
Creating a Private VLAN Spanning Multiple EX Series Switches with ELS Support (CLI Procedure) | **481**

isolation-id

Syntax

```
isolation-id number;
```

Hierarchy Level

```
[edit vlans vlan-name vlan-id number]
```

Release Information

Statement introduced in Junos OS Release 10.4 for EX Series switches.

Description

Configure an inter-switch isolated VLAN within a private VLAN (PVLAN) that spans multiple switches.

Options

number—VLAN tag identifier.

Range: 0 through 4093

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Creating a Private VLAN Spanning Multiple EX Series Switches (CLI Procedure) | 484

isolation-vlan-id

Syntax

```
isolation-vlan-id number;
```

Hierarchy Level

```
[edit vlans vlan-name pvlan]
```

Release Information

Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description

Configure an interswitch isolated VLAN within a private VLAN that spans multiple switches.

Options

number—VLAN tag identifier.

Range: 0 through 4093

Required Privilege Level

routing—To view this statement in the configuration.
routing–control—To add this statement to the configuration.

RELATED DOCUMENTATION

Creating a Private VLAN on a Single QFX Switch | 475

Creating a Private VLAN Spanning Multiple QFX Series Switches | 479

join-timer (MVRP)

Syntax

```
join-timer milliseconds;
```

Hierarchy Level

```
[edit logical-systems logical-system-name protocols mvrp],
[edit logical-systems logical-system-name routing-instances routing-instance-name protocols mvrp interface (all |
    interface-name)] (for virtual switch instance type),
[edit logical-systems logical-system-name routing-instances routing-instance-name protocols mvrp] (for virtual switch
    instance type),
[edit logical-systems logical-system-name protocols mvrp interface (all | interface-name)],
```

```
[edit protocols mvrp interface (all | interface-name)]
```

```
[edit routing-instances routing-instance-name protocols mvrp] (for virtual switch instance type),
[edit routing-instances routing-instance-name protocols mvrp interface (all | interface-name)] (for virtual switch
    instance type)
```

Release Information

Statement introduced in Junos OS Release 10.0 for EX Series switches.
Statement introduced in Junos OS Release 10.1 for MX Series routers.
Statement introduced in Junos OS Release 13.1 for the QFX Series.
Statement introduced in Junos OS Release 15.1X49-D70 for SRX Series devices.

Description

Configure the maximum number of milliseconds interfaces must wait before sending Multiple VLAN Registration Protocol (MVRP) protocol data units (PDUs).

Maintain default timer settings unless there is a compelling reason to change the settings. Modifying timers to inappropriate values might cause an imbalance in the operation of MVRP.

Default

200 milliseconds

Options

milliseconds—Interval that the interface must wait before sending MVRP PDUs (range from 100 milliseconds through 500 milliseconds). Maintain default timer settings unless there is a compelling reason to change the settings. Modifying timers to inappropriate values might cause an imbalance in the operation of MVRP.

Required Privilege Level

routing—To view this statement in the configuration.

routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

leave-timer | **1238**

leaveall-timer | **1240**

point-to-point (MVRP) | **1301**

registration | **1340**

Example: Configuring Automatic VLAN Administration Using MVRP on MX Series Routers

Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches | **832**

Configuring Multiple VLAN Registration Protocol (MVRP) to Manage Dynamic VLAN Registration

Configuring Multiple VLAN Registration Protocol (MVRP) on Switches | **797**

Understanding Multiple VLAN Registration Protocol (MVRP) for Dynamic VLAN Registration | **793**

Understanding Multiple VLAN Registration Protocol (MVRP) for Dynamic VLAN Registration | **793**

Verifying That MVRP Is Working Correctly | **851**

l2-learning

List of Syntax

Syntax (MX Series, QFX Series, EX Series)

```
l2-learning {
    global-le-bridge-domain-aging-time;
    global-mac-ip-limit number;
    global-mac-ip-table-aging-time seconds;
    global-mac-limit limit;
    global-mac-statistics;
    global-mac-table-aging-time seconds;
    global-no-mac-learning;
    global-mac-move;
}
```

Syntax (SRX Series)

```
l2-learning {
    global-mac-limit limit {
    packet-action-drop
    }
    global-mac-table-aging-time seconds;
    global-mode (switching | transparent-bridge) ;
    global-no-mac-learning;
}
```

Hierarchy Level

```
[edit protocols]
```

Release Information

Statement introduced in Junos OS Release 8.4.
Statement modified in Junos OS Release 9.5. Support for global mode added in Junos OS Release 15.1X49-D40.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.
Statement introduced in Junos OS Release 13.2X51-D10 for QFX Series.
global-le-bridge-domain-aging-time option introduced in Junos OS Release 14.2R5 for the MX Series.

global-mac-ip-limit and **global-mac-ip-table-aging-time** options introduced in Junos OS Release 17.4R1 for MX Series routers and EX9200 switches.

Description

Configure Layer 2 address learning and forwarding properties globally.

The remaining statements are explained separately. See CLI Explorer.

Options

global-le-bridge-domain-aging-time—Specify the aging time of LE bridge-domain. The MAC address is learnt after next hop(NH) and bridge-domain(BD), also called NHBD. This aging time delays the deletion of NHBD. Configuring lesser time, in seconds, results in faster deletion of NHBD.

Range: 120 to 1000000 seconds

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Understanding Layer 2 Learning and Forwarding

global-mac-table-aging-time | **1182**

global-mac-limit (Protocols) | **1179**

global-no-mac-learning | **1185**

global-mode (Protocols) | **1184**

l3-interface (VLAN)

Syntax

```
l3-interface (vlan.logical-interface-number | irb.logical-interface-number);
```

```
l3-interface l3-interface-name.logical-interface-number {
    l3-interface-ingress-counting;
}
```

```
l3-interface interface-name-logical-unit-number;
```

Hierarchy Level

```
[edit vlans vlan-name]
```

```
[edit interfaces ge-chassis/slot/port unit logical-unit-number family ethernet-switching]
[edit vlans vlan-name]
```

Release Information

Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.
[edit vlans *vlan-name*] hierarchy level introduced in Junos OS Release 12.3R2 for EX Series switches and MX Series routers.
irb option introduced in Junos OS Release 13.2 for the QFX Series.

Description

Associate a Layer 3 interface with the VLAN. Configure Layer 3 interfaces on trunk ports to allow the interface to transfer traffic between VLANs. Within a VLAN, traffic is bridged, while across VLANs, traffic is routed. Because traffic between VLANs must be routed, a common Layer 3 interface is required.

Default

No Layer 3 (routing) interface is associated with the VLAN.

Options

interface-name–logical-unit-number—Name of a logical interface.

vlan.***logical-interface-number***—Number of the logical interface. Use the **unit** number that you used when you created the **vlan** interface with a **set interfaces (QFX Series) vlan unit** statement.

> **NOTE:** Use this statement with versions of Junos OS that do not support Enhanced Layer 2 Software (ELS).

irb.***logical-interface-number***—Logical interface defined with a **set interfaces (QFX Series) irb** statement.

> **NOTE:** Use this statement with versions of Junos OS that support Enhanced Layer 2 Software (ELS).

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring Routed VLAN Interfaces on Switches (CLI Procedure)

Configuring Integrated Routing and Bridging Interfaces on Switches (CLI Procedure) | **739**

show ethernet-switching interfaces | **1485**

show ethernet-switching interface | **1481**

show vlans | **1648**

l3-interface-ingress-counting

Syntax

```
l3-interface-ingress-counting layer-3-interface-name;
```

Hierarchy Level

```
[edit vlans vlan-name]
```

Release Information
Statement introduced in Junos OS Release 11.3 for EX Series switches.

Description
(EX8200 standalone switch and EX8200 Virtual Chassis) Enable routed VLAN interface (RVI) input counters on an EX8200 switch to collect RVI source statistics for tracking or billing purposes. The input counter is maintained by a firewall filter. The switch can maintain a limited number of firewall filter counters—these counters are allocated on a first-come, first-served basis.

Output (egress) counters for EX8200 switches are always present and cannot be removed.

Reset ingress-counting statistics with the *clear interfaces statistics* command.

Default
The input (ingress) counters (both packets and bytes) are disabled on an RVI by default.

Required Privilege Level
system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- show vlans | **1648**
- *clear interfaces statistics*
- *Configuring Firewall Filters (CLI Procedure)*
- *firewall*
- *Configuring Routed VLAN Interfaces on Switches (CLI Procedure)*

layer2-control

Syntax

```
layer2-control {
    bpdu-block {
        disable-timeout seconds;
        interface interface-name;
    }
    mac-rewrite {
        interface interface-name {
            enable-all-ifl;
            protocol protocol-name;
        }
    }
    nonstop-bridging;
    traceoptions {
        file filename <files number> <size maximum-file-size> <world-readable | no-world-readable>;
        flag flag <disable>;
    }
}
```

Hierarchy Level

```
[edit protocols]
```

Release Information

Statement introduced in Junos OS Release 8.4.
bpdu-block statement added in Junos OS Release 9.4.
enable-all-if statement added in Junos OS Release 13.3.
Statement introduced in Junos OS Release 14.1X53-D10 for EX4300 switches.
Statement introduced in Junos OS Release 15.1X53-D50 for EX2300 and EX3400 switches.
Statement introduced in Junos OS Release 17.4R1 for EX4600 switches.
Statement introduced in Junos OS Release 19.1R1 for QFX Series switches.

Description

Configure Layer 2 control protocols to enable features such as Layer 2 protocol tunneling or nonstop bridging.

The remaining statements are explained separately. See CLI Explorer.

> **NOTE:** For a detailed description of configuring the **nonstop-bridging** statement, see the *High Availability User Guide*. When you configure this statement on routing platforms with two Routing Engines, a master Routing Engine switches over gracefully to a backup Routing Engine and preserves Layer 2 Control Protocol (L2CP) information.

Required Privilege Level

interface—To view this statement in the configuration.

interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- Understanding Layer 2 Protocol Tunneling | **684**
- Configuring Layer 2 Protocol Tunneling | **694**
- instance-type | **1197**

layer2-protocol-tunneling

Syntax

```
layer2-protocol-tunneling all | protocol-name {
    drop-threshold number;
    shutdown-threshold number;
}
```

Hierarchy Level

```
[edit vlans vlan-name dot1q-tunneling]
```

Release Information

Statement introduced in Junos OS Release 10.0 for EX Series switches.

Description

Enable Layer 2 protocol tunneling (L2PT) on a VLAN on switches that do not use the the Enhanced Layer 2 Software (ELS) configuration style (which includes EX2200, EX3300, EX4200, EX4500, and EX4450 switches).

> **NOTE:** This command is not available on switches that use ELS configuration style (including EX2300, EX3400, EX4300, EX4600, EX9200, and any QFX Series switches that support L2PT). On those switches, you enable L2PT using the **protocol** statement at the **[edit protocols layer2-control mac-rewrite interface *interface-name*]** hierarchy level. See "Configuring Layer 2 Protocol Tunneling" on page 694.
>
> For details on ELS, see "Using the Enhanced Layer 2 Software CLI" on page 50.

The remaining statements are explained separately. See CLI Explorer.

Default

L2PT is not enabled.

Options

all—Enable all supported Layer 2 protocols.

protocol-name—Name of the Layer 2 protocol. Values are:

- **802.1x**—IEEE 802.1X authentication
- **802.3ah**—IEEE 802.3ah Operation, Administration, and Maintenance (OAM) link fault management (LFM)

> **NOTE:** If you enable L2PT for untagged OAM LFM packets, do not configure LFM on the corresponding access interface.

- **cdp**—Cisco Discovery Protocol
- **e-lmi**—Ethernet local management interface
- **gvrp**—GARP VLAN Registration Protocol
- **lacp**—Link Aggregation Control Protocol

> **NOTE:** If you enable L2PT for untagged LACP packets, do not configure LACP on the corresponding access interface.

- **lldp**—Link Layer Discovery Protocol
- **mmrp**—Multiple MAC Registration Protocol
- **mvrp**—Multiple VLAN Registration Protocol
- **stp**—Spanning Tree Protocol, Rapid Spanning Tree Protocol, and Multiple Spanning Tree Protocol
- **udld**—Unidirectional Link Detection (UDLD)
- **vstp**—VLAN Spanning Tree Protocol
- **vtp**—VLAN Trunking Protocol

Required Privilege Level
system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

show ethernet-switching layer2-protocol-tunneling interface | **1495**

show ethernet-switching layer2-protocol-tunneling statistics | **1497**

show ethernet-switching layer2-protocol-tunneling vlan | **1500**

Example: Configuring Layer 2 Protocol Tunneling on EX Series Switches Without ELS Support | **701**

Configuring Layer 2 Protocol Tunneling on EX Series Switches Without ELS Support | **699**

leave-timer (MVRP)

Syntax

```
leave-timer milliseconds;
```

EX Series, QFX Series, QFabric

```
[edit protocols mvrp interface (all | interface-name)]
```

M Series, SRX Series, MX Series, T Series

```
[edit logical-systems logical-system-name protocols mvrp],
[edit logical-systems logical-system-name routing-instances routing-instance-name protocols mvrp interface (all |
    interface-name)] (for virtual switch instance type),
[edit logical-systems logical-system-name routing-instances routing-instance-name protocols mvrp] (for virtual switch
    instance type),
[edit logical-systems logical-system-name protocols mvrp interface (all | interface-name)],
```

```
[edit protocols mvrp interface (all | interface-name)],
```

```
[edit routing-instances routing-instance-name protocols mvrp] (for virtual switch instance type),
[edit routing-instances routing-instance-name protocols mvrp interface (all | interface-name)] (for virtual switch
    instance type)
```

Release Information

Statement introduced in Junos OS Release 10.0 for EX Series switches.
Statement introduced in Junos OS Release 10.1 for MX Series routers.
Statement introduced in Junos OS Release 13.1 for the QFX Series.
Statement introduced in Junos OS Release 15.1X49-D70 for SRX Series devices.

Description

For Multiple VLAN Registration Protocol (MVRP), configure the number of milliseconds the switch retains a VLAN in the Leave state before the VLAN is unregistered. If the interface receives a join message before this timer expires, the VLAN remains registered.

Maintain default timer settings unless there is a compelling reason to change the settings. Modifying timers to inappropriate values might cause an imbalance in the operation of MVRP.

Default

1000 milliseconds

Options

milliseconds—Interval that the switch retains a VLAN in the Leave state before the VLAN is unregistered. At a minimum, set the **leave-timer** interval at twice the join-timer interval (range from 300 milliseconds through 1000 milliseconds). Maintain default timer settings unless there is a compelling reason to change the settings. Modifying timers to inappropriate values might cause an imbalance in the operation of MVRP.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Example: Configuring Automatic VLAN Administration Using MVRP on MX Series Routers

Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches | **832**

Configuring Multiple VLAN Registration Protocol (MVRP) on Switches | **797**

Configuring Multiple VLAN Registration Protocol (MVRP) to Manage Dynamic VLAN Registration

Verifying That MVRP Is Working Correctly | **851**

Understanding Multiple VLAN Registration Protocol (MVRP) for Dynamic VLAN Registration | **793**

Understanding Multiple VLAN Registration Protocol (MVRP) for Dynamic VLAN Registration | **793**

join-timer (MVRP) | **1227**

leaveall-timer | **1240**

point-to-point (MVRP) | **1301**

registration | **1340**

leaveall-timer (MVRP)

Syntax

```
leaveall-timer interval;
```

EX Series and QFX Series

- For platforms with ELS:

```
[edit protocols mvrp],
[edit protocols mvrp interface interface-name]
```

- For platforms without ELS:

```
[edit protocols mvrp interface (all | interface-name)]
```

SRX 1500, SRX 300, SRX 550M, SRX 345, SRX 340, SRX 320

```
[edit routing-instances routing-instance-name protocols mvrp] (for virtual switch instance type),
[edit routing-instances routing-instance-name protocols mvrp interface (all | interface-name)] (for virtual switch
    instance type)
```

EX Series, M Series, SRX Series, T Series, MX Series

```
[edit logical-systems logical-system-name protocols mvrp],
[edit logical-systems logical-system-name routing-instances routing-instance-name protocols mvrp interface (all |
    interface-name)] (for virtual switch instance type),
[edit logical-systems logical-system-name routing-instances routing-instance-name protocols mvrp] (for virtual switch
    instance type),
[edit logical-systems logical-system-name protocols mvrp interface (all | interface-name)],
[edit protocols mvrp interface (all | interface-name)],
[edit routing-instances routing-instance-name protocols mvrp] (for virtual switch instance type),
[edit routing-instances routing-instance-name protocols mvrp interface (all | interface-name)] (for virtual switch
    instance type)
```

Release Information

Statement introduced in Junos OS Release 10.0 for EX Series switches.
Statement introduced in Junos OS Release 10.1 for MX Series routers.
Hierarchy level **[edit protocols mvrp]** introduced in Junos OS Release 13.2X50-D10 (ELS). (See “Using the Enhanced Layer 2 Software CLI” on page 50 for information about ELS.)
Statement introduced in Junos OS Release 13.1 for the QFX Series.

Statement introduced in Junos OS Release 15.1X49-D70 for SRX Series devices.

Description

For Multiple VLAN Registration Protocol (MVRP), configure the interval at which the LeaveAll state operates on the interface.

Maintain default timer settings unless there is a compelling reason to change the settings. Modifying timers to inappropriate values might cause an imbalance in the operation of MVRP. However, if you choose to change the default values, keep in mind that on an EX Series switch that uses Junos OS with support for ELS, if the timer value set on an interface level is different from the value set on a switch level, then the value on the interface level takes precedence.

Options

EX Series and QFX Series:

interval—Number of seconds or milliseconds between the sending of Leave All messages.

Default: 10 seconds, or 10,000 milliseconds

SRX 1500, SRX 300, SRX 550M, SRX 345, SRX 340, SRX 320:

seconds—Interval between the sending of Leave All messages (range from 10 seconds through 60 seconds. Maintain default timer settings unless there is a compelling reason to change the settings. Modifying timers to inappropriate values might cause an imbalance in the operation of MVRP.

Default: 60 seconds

EX Series, M Series, SRX Series, T Series, MX Series:

milliseconds—Interval between the sending of Leave All messages. Maintain default timer settings unless there is a compelling reason to change the settings. Modifying timers to inappropriate values might cause an imbalance in the operation of MVRP.

Default: 10000 milliseconds

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

lldp

Syntax

```
lldp {
    advertisement-interval seconds;
    (disable | enable);
    hold-multiplier number;
    interface (all | [interface-name]) {
        (disable | enable);
        OBSOLETE: power-negotiation <(disable | enable)>;
        tlv-filter;
        tlv-select;
        trap-notification (disable | enable);
    }
    lldp-configuration-notification-interval seconds;
    management-address ip-management-address;;
    mau-type;
    netbios-snooping;
    no-tagging;
    neighbour-port-info-display (port-description | port-id);
    port-description-type (interface-alias | interface-description);
    port-id-subtype (interface-name | locally-assigned);
    ptopo-configuration-maximum-hold-time seconds;
    ptopo-configuration-trap-interval seconds;
    tlv-filter;
    tlv-select;
    traceoptions {
        file filename <files number> <size maximum-file-size> <world-readable | no-world-readable>;
        flag flag <disable>;
    }
    transmit-delay (LLDP) seconds;
    vlan-name-tlv-option (name | vlan-id);
}
```

Hierarchy Level

```
[edit protocols],
[edit routing-instances routing-instance-name protocols]
```

Release Information

Statement introduced in Junos OS Release 9.0 for EX Series switches.
management-address introduced in Junos OS Release 9.5.

Statement introduced in Junos OS Release 9.6 for MX Series.
Statement introduced in Junos OS Release 11.1 for QFX Series.
netbios-snooping introduced in Junos OS Release 11.1 for EX2200, EX3200, EX3300, EX4200, EX4500, EX4550, EX6210, EX8208, and EX8216 switches.
port-description-type introduced in Junos OS Release 13.3R5, 14.2R2, 14.1R4, and 12.3R9.
no-tagging introduced in Junos OS Release 14.1X53-D10 for EX2200, EX3200, EX3300, EX4200, EX4500, EX4550, EX6210, EX8208, and EX8216 switches.
Statement introduced in Junos OS Release 14.1X53-D20 for OCX Series switches.
neighbour-port-info-display introduced in Junos OS Release 14.1X53-D40 and Release 15.1R5 and Release 16.1R3.
mau-type introduced in Junos OS Release 15.1 for EX4300, EX9200, and EX9250 switches.

Description

Configure Link Layer Discovery Protocol (LLDP). The switch uses LLDP to advertise its identity and capabilities on a LAN, as well as to receive information about other network devices. LLDP is defined in the IEEE standard 802.1AB-2005.

NOTE: The transmit-delay and netbios-snooping options are not available on QFabric systems.

NOTE: On EX4300 switches, LLDP cannot be configured on the me0 or vme interface. Issuing the command **set protocols lldp interface me0** generates the following error message:

```
error: name: 'me0': Invalid interface
error: statement creation failed: interface
```

Issuing the command **set protocols lldp interface vme** generates the following error message:

```
error: name: 'vme': Invalid interface
error: statement creation failed: interface
```

Default

LLDP is disabled. If you configure LLDP for all interfaces, you can later disable a particular interface.

NOTE: The **interface-name** must be the physical interface and not a logical interface (unit).

Options

advertisement-interval *seconds*—Specify the frequency at which LLDP advertisements are sent. This value is also used in combination with the hold-multiplier value to determine the length of time LLDP information is held before it is discarded.

The **advertisement-interval** value must be greater than or equal to four times the **transmit-delay** value, or an error will be returned when you attempt to commit the configuration.

> **NOTE:** The default value of **transmit-delay** is 2 seconds. If you configure **advertisement-interval** as less than 8 seconds and you do not configure a value for **transmit-delay**, the value of **transmit-delay** is automatically changed to 1 second to satisfy the requirement that the **advertisement-interval** value be greater than or equal to four times the **transmit-delay** value.

Default: 30 seconds

Range: 5 through 32768 seconds

disable | enable—Disable or enable LLDP on the device.

Default: If you do not configure LLDP, it is disabled on the device.

hold-multiplier *number*—Specify the multiplier used in combination with the advertisement-interval value to determine the length of time LLDP information is held before it is discarded.

Range: 2 through 10

Default: 4 (or 120 seconds with the default of 30 seconds for advertisement-interval)

lldp-configuration-notification-interval *seconds*—Specify how often SNMP trap notifications are generated as a result of LLDP database changes.

Range: 5 through 3600 seconds

Default: Disabled

management-address *ip-management-address;*—Specify the management address to be used in LLDP Management Address type, length, and value (TLV) messages. The Management Address TLV typically contains the IPv4 or IPv6 management addresses of the local system. Only out-of-band management addresses can be used for the management-address. Other remote managers can use this address to obtain information related to the local device.

Default: The LLDP Management Address TLV uses the IP address of the switch's management Ethernet interface (me0), or the IP address of the virtual management Ethernet (VME) interface if the switch is a Virtual Chassis member.

mau-type—(EX4300, EX9200, and EX9250 switches only) Configure the switch to advertise information about the medium attachment unit (MAU) type. The MAU is a transceiver that interconnects the attachment unit interface (AUI) port on an attached host computer to an Ethernet cable. MAU types are defined in the IEEE 802.3 standard.

The MAU type is included in the MAC/PHY Configuration Status type, length, and value (TLV) message. TLVs are used by LLDP-capable devices to transmit information to neighbor devices. The MAC/PHY Configuration Status TLV is an organizationally defined TLV that advertises information about the physical interface. In addition to the MAU type, the MAC/PHY Configuration Status TLV also includes information such as autonegotiation status, support, and advertised capabilities.

The MAU type cannot be changed by configuration; however, you must configure the **mau-type** statement to include the MAU type value in the MAC/PHY Configuration Status TLV.

Default: If the **mau-type** statement is not configured, the MAU type field of the MAC/PHY Configuration Status TLV contains the value **Unknown**.

netbios-snooping—(EX2200, EX3200, EX3300, EX4200, EX4500, EX4550, EX6210, EX8208, and EX8216 switches only) Enable NetBIOS snooping to learn information about NetBIOS hosts that are connected to the switch.

no-tagging—(EX2200, EX3200, EX3300, EX4200, EX4500, EX4550, EX6210, EX8208, and EX8216 switches only) Configure the switch to send LLDPDUs without including VLAN tags on the interfaces on which VLAN tagging is enabled (tagged interfaces).

Default: Interfaces for which VLAN tagging is enabled include a VLAN tag (tag 0) in LLDPDUs if the **no-tagging** option is not configured.

neighbour-port-info-display (port-description | port-id)—Configure the type of LLDP neighbor port information that the device displays in the **Port info** field in the output of the **show lldp neighbors** CLI command.

Devices in a network use LLDP to learn about and identify neighbor devices. LLDP-capable devices transmit information in type, length, and value (TLV) messages to neighbor devices.

The **Port info** field of the **show lldp neighbors** command displays the port information received from LLDP neighbors. This information is sent from the LLDP neighbor to the device in a type, length, and value (TLV) message. You can use the **neighbor-port-info-display** statement to configure the device to display the information contained in either the Port Description TLV or the Port Identification TLV.

Values: Configure one of the following:

- port-description—Display the information from the Port Description TLV in the **Port info** field of the **show lldp neighbors** CLI command.

  The Port Description TLV contains the textual description of the logical unit or the port. The description for the logical unit is used, if available; otherwise, the description for the physical interface (port) is used. For example, LAG member interfaces do not contain a logical unit; therefore, only the description configured on the physical interface is used.

- port-id—Display the information from the Port Identification TLV in the **Port info** field of the **show lldp neighbors** CLI command.

  The Port Identification TLV contains the identifier for the neighbor port. The SNMP index of the interface is used as the port identifier.

Default: port-description—The information contained in the Port Description TLV is displayed in the **Port info** field.

port-description-type (interface-alias | interface-description)—Configure the value to be used to generate the Port Description TLV that the device advertises to neighbors.

Values: Configure one of the following:

- interface-alias—Use the *ifAlias* MIB object value to generate the port description TLV. The LLDP MIB variable *lldpLocPortDesc* then contains the same value as *ifAlias*, which is the same as the description of the interface.

- interface-description—Use the *ifDescr* MIB object value to generate the port description TLV. The LLDP MIB variable *lldpLocPortDesc* then contains the same value as *ifDescr*, which is the same as the interface name.

Default: interface-alias—The **interface-alias** value is same as the description of an interface configured with **set interface *name* description *description* command**.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- Configuring LLDP | **674**
- *show lldp*
- *Configuring LLDP (CLI Procedure)*
- *Understanding LLDP*
- *Understanding LLDP and LLDP-MED on EX Series Switches*
- *Configuring NetBIOS Snooping (CLI Procedure)*

mac

Syntax

```
mac mac-address;
```

Hierarchy Level

```
[edit interfaces interface-name]
```

Release Information
Statement introduced before Junos OS Release 7.4.

Description
Set the MAC address of the interface.

Use this statement at the **[edit interfaces ... ps0]** hierarchy level to configure the MAC address for a pseudowire logical device that is used for subscriber interfaces over point-to-point MPLS pseudowires.

Options
mac-address—MAC address. Specify the MAC address as six hexadecimal bytes in one of the following formats: ***nnnn.nnnn.nnnn*** or ***nn:nn:nn:nn:nn:nn***. For example, **0000.5e00.5355** or **00:00:5e:00:53:55**.

Required Privilege Level
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring the MAC Address on the Management Ethernet Interface

Configuring a Pseudowire Subscriber Logical Interface Device

mac (Static MAC-Based VLANs)

Syntax

```
mac mac-address {
    next-hop interface-name;
}
```

Hierarchy Level

```
[edit ethernet-switching-options static vlan vlan-name]
```

Description

Specify the MAC address to add to the Ethernet switching table.

The remaining statement is explained separately. See CLI Explorer.

Options

mac-address—MAC address

Required Privilege Level

system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Adding a Static MAC Address Entry to the Ethernet Switching Table | 113

mac-limit

List of Syntax

Syntax (QFX Series and EX4600) on page 1251
Syntax (SRX Series and EX Series) on page 1251

Syntax (QFX Series and EX4600)

```
mac-limit number;
```

Syntax (SRX Series and EX Series)

```
mac-limit limit action action;
```

Hierarchy Level

```
[edit vlans vlan-name]
```

Release Information

Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.
The short description of **interface-mac-limit** at the CLI command hierarchy is changed from **Maximum number of MAC addresses per interface (1..16383)** to **Maximum number of MAC addresses per interface (1..5120)** at the **[edit vlans vlan-name switch-options]** hierarchy level from Junos OS Release 18.2R1.

Description

Specify the maximum number of MAC addresses to be associated with a VLAN—the default is **unlimited**, which can leave the network vulnerable to flooding. Change **unlimited** to any number from 2 to the switch's maximum VLAN MAC limit. The maximum number of MAC addresses allowed in a switching table per VLAN varies depending on the EX Series switch. To see the maximum number of MAC addresses per VLAN allowed on your switch, issue the **set vlans *vlan-name* mac-limit ?** configuration-mode command.

> **NOTE:** Do not set the **mac-limit** value to 1. The first learned MAC address is often inserted into the forwarding database automatically—for instance, for a routed VLAN interface (RVI), the first MAC address inserted into the forwarding database is the MAC address of the RVI. For aggregated Ethernet bundles (LAGs) using LACP, the first MAC address inserted into the forwarding database in the Ethernet switching table is the source address of the protocol packet. In these cases, the switch does not learn MAC addresses other than the automatic address when **mac-limit** is set to 1, and this causes problems with MAC learning and forwarding.

When the MAC limit set by this statement is reached, no more MAC addresses are added to the Ethernet switching table. You can also, optionally, have a system log entry generated when the limit is exceeded by adding the option **action log**.

> **NOTE:** When you reconfigure the number of MAC addresses, the Ethernet switching table is not automatically cleared. Therefore, if you reduce the number of addresses from the default (unlimited) or a previously set limit, you could already have more entries in the table than the new limit allows. Previous entries remain in the table after you reduce the number of addresses, so you should clear the Ethernet switching table for a specified interface, MAC address, or VLAN when you reduce the MAC limit. Use the command **clear ethernet-switching table** to clear existing MAC addresses from the table before using the **mac-limit** configuration statement.

Default

The MAC limit is disabled, so entries are unlimited.

Options

QFX Series and EX4600:

number—Maximum number of MAC addresses.

Range: 1 through 32768

> **NOTE:** This statement is not supported on QFabric systems.

EX Series:

limit—Maximum number of MAC addresses.

Range: 1 through *switch maximum*

SRX Series:

number—Maximum number of MAC addresses.

Range: 1 through 5120

action—**Log** is the only action available. Configure **action log** to add a message to the system log when the mac-limit value is exceeded. A typical logged message looks like this:

May 5 06:18:31 bmp-199p1-dev edwd[5665]:
ESWD_VLAN_MAC_LIMIT_EXCEEDED: vlan default mac
00:1f:12:37:af:5b (tag 40). vlan limit exceeded

Required Privilege Level

routing—To view this statement in the configuration.

routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- show vlans | **1648**
- Example: Setting Up Basic Bridging and a VLAN on Switches | **203**
- Configuring MAC Table Aging on Switches | **137**
- Understanding Bridging and VLANs on Switches | **168**

mac-lookup-length

Syntax

```
mac-lookup-length number-of-entries;
```

Hierarchy Level

```
[edit ethernet-switching-options]
```

Release Information

Statement introduced in Junos OS Release 11.4 for EX Series switches.

Description

Increase the maximum number of searchable hash indexes to mitigate situations in which hash index collisions are causing problems with the learning of MAC addresses in the forwarding database (FDB).

The FDB on EX3200, EX3300, EX4200, EX4500, EX4550, and EX6210 switches is a hash table with 8192 hash indexes (rows) of MAC addresses and four entries per hash index. When the FDB is searched, a configured hash function calculates the hash index at which to start the search. By default, after the search starts at the determined hash index, the maximum number of hash indexes that can be searched is one hash index, or four entries

> **NOTE:** Increasing the number of hash indexes increases the chances of finding an open entry in which to add a newly learned MAC address. However, searching more hash indexes requires more bandwidth and may impact the FDB performance and line-rate traffic.

Default

4

Options

number-of-entries—Maximum number of searchable hash indexes in the FDB.

Range: 4, 8, 12

Required Privilege Level

system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Understanding Bridging and VLANs on Switches | 168

mac-notification

Syntax

```
mac-notification {
    notification-interval seconds;
}
```

Hierarchy Level

```
[edit ethernet-switching-options]
[edit switch-options]
```

Release Information

Statement introduced in Junos OS Release 9.6 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.
Hierarchy level **[edit switch-options]** added in Junos OS Release 14.1X53-D10 for EX Series and QFX Series.

Description

Enable MAC notification for a switch. If you configure this statement without setting a notification interval, MAC notification is enabled with the default MAC notification interval of 30 seconds.

The remaining statement is explained separately. See CLI Explorer.

Default

MAC notification is disabled by default.

Required Privilege Level

system—To view this statement in the configuration.
system-control—To add this statement to the configuration.
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring Non-ELS MAC Notification | 130

Configuring MAC Notification on Switches with ELS Support | 129

mac-rewrite

Syntax

```
mac-rewrite {
   interface interface-name {
      enable-all-ifl;
      protocol protocol-name;
      }
   }
}
```

Hierarchy Level

```
[edit protocols layer2-control]
```

Release Information

Statement introduced in Junos OS Release 9.1.
enable-all-if statement added in Junos OS Release 13.3.
Support for PVSTP protocol introduced in Junos OS Release 13.3 for MX Series routers and EX9200 switches.
Statement introduced in Junos OS Release 14.1X53-D10 and 17.3R1 for EX4300 switches.
Statement introduced in Junos OS Release 15.1X53-D55 and 18.2R1 for EX2300 and EX3400 switches.
Statement introduced in Junos OS Release 17.4R1 for EX4600 switches.
Statement introduced in Junos OS Release 19.1R1 for QFX Series switches.

Description

Enable rewriting of the MAC address for Layer 2 protocol tunneling (L2PT).

When a service provider edge port configured for L2PT receives a control packet for a supported protocol, the device rewrites the multicast destination MAC address with the predefined multicast tunneling MAC address 01:00:0c:cd:cd:d0. The packet travels across the provider network transparently to the other end of the tunnel, and the destination device restores the original multicast destination MAC address to deliver the packet at its destination.

Refer to **protocol** for the list of protocols that you can configure for L2PT on different devices.

To see the protocols for which you enabled L2PT on an interface, enter the **show mac-rewrite interface** command.

On MX Series and ACX Series routers and EX9200 switches with L2PT configured, customer-facing ports should not receive packets with the L2PT MAC address as the destination address unless there is a network

tolopogy or configuration error. Any such interface receiving an L2PT packet becomes “Disabled”, and you must subsequently re-enable it using the **clear error mac-rewrite** command.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- Understanding Layer 2 Protocol Tunneling | **684**
- Configuring Layer 2 Protocol Tunneling | **694**
- show mac-rewrite interface | **1572**
- clear error mac-rewrite | **1437**

mac-statistics

Syntax

```
mac-statistics;
```

Hierarchy Level

```
[edit bridge-domains bridge-domain-name bridge-options],
[edit logical-systems logical-system-name bridge-domains bridge-domain-name bridge-options],
[edit logical-systems logical-system-name routing-instances routing-instance-name bridge-domains bridge-domain-name
    bridge-options],
[edit logical-systems logical-system-name routing-instances routing-instance-name switch-options],
[edit logical-systems logical-system-name switch-options],
[edit routing-instances routing-instance-name bridge-domains bridge-domain-name bridge-options],
[edit routing-instances routing-instance-name switch-options],
[edit routing-instances routing-instance-name protocols evpn],
[edit switch-options],
[edit switch-options],
[edit vlans vlan-name switch-options]
```

Release Information

Statement introduced in Junos OS Release 8.4.
Support for the **switch-options** statement added in Junos OS Release 9.2.
Support for top-level configuration for the **virtual-switch** type of routing instance added in Junos OS Release 9.2. In Junos OS Release 9.1 and earlier, the routing instances hierarchy supported this statement only for a VPLS instance or a bridge domain configured within a virtual switch.
Support for logical systems added in Junos OS Release 9.6.
[edit switch-options] and [edit vlans *vlan-name* switch-options] hierarchy levels introduced in Junos OS Release 12.3R2 for EX Series switches.
Support for EVPNs added in Junos OS Release 13.2 for MX 3D Series routers.
[edit switch-options] and [edit vlans *vlan-name* switch-options] hierarchy levels introduced in Junos OS Release 13.2 for the QFX Series.

Description

(MX Series routers, EX Series switches, and QFX Series only) For bridge domains or VLANs, enable MAC accounting either for a specific bridge domain or VLAN, or for a set of bridge domains or VLANs associated with a Layer 2 trunk port.

Default

disabled

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- *Understanding Layer 2 Learning and Forwarding for Bridge Domains*
- Layer 2 Learning and Forwarding for VLANs Overview | **78**
- *Understanding Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports*
- Layer 2 Learning and Forwarding for VLANs Acting as a Switch for a Layer 2 Trunk Port | **80**
- *Configuring EVPN Routing Instances*
- *Configuring EVPN Routing Instances on EX9200 Switches*

mac-table-aging-time

Syntax

```
mac-table-aging-time (seconds | unlimited);
```

Hierarchy Level

```
[edit ethernet-switching-options],
[edit vlans vlan-name]
```

Release Information

Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement updated in Junos OS Release 9.4 for EX Series switches to include **[edit ethernet-switching-options]** hierarchy level.

Description

You configure how long MAC addresses remain in the Ethernet switching table using the **mac-table-aging-time** statement in either the **[edit ethernet-switching-options]** or the**[edit vlans]** hierarchy, depending on whether you want to configure it for the entire device or only for specific VLANs.

If you specify the time as **unlimited**, entries are never removed from the table. Generally, use this setting only if the device or the VLAN has a fairly static number of end devices; otherwise the table will eventually fill up. You can use this setting to minimize traffic loss and flooding that might occur when traffic arrives for MAC addresses that have been removed from the table.

> **NOTE:** The **mac-table-aging-time** statement appears in the Junos OS CLI for devices that do *not* support the Enhanced Layer 2 Software (ELS) configuration style. If your device runs software that supports ELS, use the **global-mac-table-aging-time** statement, which appears in the **[edit protocols l2-learning]** hierarchy. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50.

Default

Entries remain in the Ethernet switching table for 300 seconds.

Options

seconds—Time that entries remain in the Ethernet switching table before being removed.

Range: 60 through 1,000,000 seconds

Default: 300 seconds

unlimited—Entries remain in the Ethernet switching table.

Required Privilege Level
system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- show ethernet-switching statistics aging | **1512**
- Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch | **226**
- Configuring MAC Table Aging on Switches | **137**
- *Controlling Authentication Session Timeouts (CLI Procedure)*
- Configuring VLANs for EX Series Switches | **183**

mac-table-size

Syntax

```
mac-table-size limit {
    packet-action drop;
}
```

Hierarchy Level

```
[edit bridge-domains bridge-domain-name bridge-options],
[edit logical-systems logical-system-name bridge-domains bridge-domain-name bridge-options],
[edit logical-systems logical-system-name routing-instances routing-instance-name bridge-domains bridge-domain-name
    bridge-options],
[edit logical-systems logical-system-name routing-instances routing-instance-name switch-options],
[edit logical-systems logical-system-name switch-options],
[edit routing-instances routing-instance-name bridge-domains bridge-domain-name bridge-options],
[edit routing-instances routing-instance-name switch-options],
[edit switch-options],
[edit switch-options],
[edit vlans vlan-name switch-options]
```

Release Information

Statement introduced in Junos OS Release 8.4.
Support for the **switch-options** statement added in Junos OS Release 9.2.
Support for top-level configuration for the **virtual-switch** type of routing instance added in Junos OS Release 9.2. In Junos OS Release 9.1 and earlier, the routing instances hierarchy supported this statement only for a VPLS instance or a bridge domain configured within a virtual switch.
Support for logical systems added in Junos OS Release 9.6.
[edit switch-options] and **[edit vlans *vlan-name* switch-options]** hierarchy levels introduced in Junos OS Release 12.3R2 for EX Series switches.
Support at the **[edit vlans *vlan-name* switch-options** hierarchy level introduced in Junos OS Release 13.2 for the QFX Series.

Description

Modify the size of the MAC address table for the bridge domain or VLAN, a set of bridge domains or VLANs associated with a trunk port, or a virtual switch. The default is 5120 MAC addresses.

NOTE: For multichassis link aggregation (MC-LAG) peers in active-active mode, configuring the **mac-table-size** statement or changing the **mac-table-size** configuration when traffic is flowing can cause the MAC entries to be out of synchronization between the two MC-LAG peers, which might result in flooding. To avoid flooding, you must either halt traffic forwarding and then configure the **mac-table-size** statement or use the **commit at** configuration statement to commit the changes at the same time in both the peer nodes.

Alternatively, if flooding does occur, you can clear the bridge MAC table on both the routers by using the **clear bridge mac-table** command. Running this command ensures that the MAC entries are re-learned and in synchronization between both the peers.

Options

limit—Specify the maximum number of addresses in the MAC address table.

Range: 16 through 1,048,575 MAC addresses

Default: 5120 MAC addressesThere is no default MAC address limit for the **mac-table-size** statement at the **[edit switch-options]** hierarchy level. The number of MAC addresses that can be learned is only limited by the platform, 65,535 MAC addresses for EX Series switches and 1,048,575 MAC addresses for other devices.

The remaining statement is explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Understanding Layer 2 Learning and Forwarding for Bridge Domains

Layer 2 Learning and Forwarding for VLANs Overview | **78**

Understanding Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports

Layer 2 Learning and Forwarding for VLANs Acting as a Switch for a Layer 2 Trunk Port | **80**

mapping

List of Syntax

Syntax (EX Series) on page 1265
Syntax (QFX Series) on page 1265

Syntax (EX Series)

```
mapping (native (push | swap) | policy | tag (push | swap));
```

Syntax (QFX Series)

```
mapping (native (push | swap) | tag (push | swap));
mapping native inner-tag tag push;
mapping native push inner-tag tag;
```

Hierarchy Level

```
[edit vlans vlan-name interface interface-name egress],
[edit vlans vlan-name interface interface-name ingress],
[edit vlans vlan-name interface interface-name]
```

Release Information

Statement introduced in Junos OS Release 9.6 for EX Series switches.
Option **swap** introduced in Junos OS Release 10.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description

Map a specific C-VLAN to an S-VLAN. By default, the received incoming or outgoing tag is replaced with the new tag.

This statement is also required if you are configuring firewall filters to map traffic from an interface to a VLAN. If you are configuring firewall filters to map traffic from an interface to a VLAN, the **mapping policy** option must be configured using this command. The firewall filter also has to be configured using the **vlan** action for a match condition in the firewall filter stanza for firewall filters to map traffic from an interface for a VLAN.

Options

For EX Series:

native—Maps untagged and priority-tagged packets to an S-VLAN.

policy—Maps the interface to a firewall filter policy to an S-VLAN.

push—Retains the incoming tag and add an additional VLAN tag instead of replacing the original tag.

swap—Swaps the incoming VLAN tag with the VLAN ID tag of the S-VLAN. Use of this option is also referred to as VLAN ID translation.

tag—Retains the incoming 802.1Q tag on the interface.

For QFX Series:

inner-tag (QFabric systems only)—apply the specified tag as an inner tag to packets that are received as untagged on an access interface.

native—Map untagged and priority-tagged packets to an S-VLAN.

push—Retain the incoming tag (as an inner tag) and adds an additional VLAN tag. When you use this option, the TPID of the outer tag is set as follows:

- If Q-in-Q tunneling is not enabled in the VLAN, then the Ethertype for outer tag is set to 0x8100.
- If Q-in-Q tunneling is enabled in the VLAN and a packet is egressing from a trunk port, then the Ethertype is set to 0x88a8 (or as configured by an **ether-type** statement).

swap—Replaces the incoming VLAN tag with the VLAN ID tag of the S-VLAN. Using this option is also referred to as VLAN ID translation. When you use this option on a trunk port for which Q-in-Q tunneling is enabled, use the **ether-type** statement to set the Ethertype.

tag—Original VLAN tag that will be replaced (with **swap**) or that will become an inner tag (with **push**).

Required Privilege Level
system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring Q-in-Q Tunneling on QFX Series Switches | **899**

Example: Setting Up Q-in-Q Tunneling on QFX Series Switches | **920**

Configuring VLANs for EX Series Switches | **183**

Understanding Q-in-Q Tunneling and VLAN Translation | **887**

Understanding Bridging and VLANs on Switches | **168**

mapping-range

Syntax

```
mapping-range C-VLAN-range (push | swap) <vlan-id-start S-VLAN-ID>;
```

Hierarchy Level

```
[edit vlans vlan-name vlans interface-name]
```

Release Information

Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description

Configure an access interface to map a range of C-VLANs to a range of S-VLANs. Use this statement instead of using multiple **set vlans *VLAN-name* interface *interface-name* mapping (push | swap)** statements to configure Q-in-Q tunneling or VLAN translation on a per-VLAN basis. This statement is particularly useful if you have used the **vlan-range** statement to create multiple VLANs.

Options

push—Retain the incoming tag and adds an additional VLAN tag (Q-in-Q tunneling).

swap—Swap the incoming VLAN tag with the VLAN ID tag of the S-VLAN (VLAN translation).

vlan-ID-start *S-VLAN-ID*—(Optional) Set the start of the S-VLAN range that the C-VLANs will be mapped to. If you omit this option, mapping begins with the first ID in the range of S-VLAN IDs (which you configure using the **set vlans vlan-range** statement).

Required Privilege Level

system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- Configuring Q-in-Q Tunneling on QFX Series Switches | **899**
- Example: Setting Up Q-in-Q Tunneling on QFX Series Switches | **920**
- vlan-range | **1397**

members

Syntax

```
members [(all | names | vlan-ids)];
```

Hierarchy Level

```
[edit interfaces (QFX Series) interface-name unit 0 family ethernet-switching vlan]
```

Release Information

Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement updated with enhanced **?** (CLI completion feature) functionality in Junos OS Release 9.5 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description

For trunk interfaces, configure the VLANs for which the interface can carry traffic.

TIP: To display a list of all configured VLANs on the system, including VLANs that are configured but not committed, type **?** after **vlan** or **vlans** in your configuration mode command line. Note that only one VLAN is displayed for a VLAN range.

NOTE: The number of VLANs supported per switch varies for each model. Use the configuration-mode command **set vlans id vlan-id ?** to determine the maximum number of VLANs allowed on a switch. You cannot exceed this VLAN limit because each VLAN is assigned an ID number when it is created. You can, however, exceed the recommended VLAN member maximum.

On an EX Series switch that runs Junos OS that does not support the Enhanced Layer 2 Software (ELS) configuration style, the maximum number of VLAN members allowed on the switch is 8 times the maximum number of VLANs the switch supports (vmember limit = vlan max * 8). If the switch configuration exceeds the recommended VLAN member maximum, you see a warning message when you commit the configuration. If you ignore the warning and commit such a configuration, the configuration succeeds but you run the risk of crashing the Ethernet switching process (**eswd**) due to memory allocation failure.

On an EX Series switch that runs Junos OS that supports ELS, the maximum number of VLAN members allowed on the switch is 24 times the maximum number of VLANs the switch supports (vmember limit = vlan max * 24). If the configuration of one of these switches exceeds the recommended VLAN member maximum, a warning message appears in the system log (syslog).

Options

all—Specifies that this trunk interface is a member of all VLANs that are configured on this switch. When a new VLAN is configured on the switch, this trunk interface automatically becomes a member of the VLAN.

> **NOTE:** Since VLAN members are limited, specifying **all** could cause the number of VLAN members to exceed the limit at some point.

> **NOTE:** Each VLAN that is configured must have a specified VLAN ID when you attempt to commit the configuration; otherwise, the configuration commit fails. Also, **all** cannot be the name of a VLAN on the switch.

names—Name of one or more VLANs. VLAN IDs are applied automatically in this case.

vlan-ids—Numeric identifier of one or more VLANs. For a series of tagged VLANs, specify a range; for example, **10–20** or **10–20 23 27–30**.

> **NOTE:** Each configured VLAN must have a specified VLAN ID to successfully commit the configuration; otherwise, the configuration commit fails.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- *Configuring Gigabit Ethernet Interfaces (CLI Procedure)*
- *Configuring Gigabit Ethernet Interfaces for EX Series Switches with ELS support*
- *Configuring Gigabit Ethernet Interfaces (J-Web Procedure)*
- Configuring VLANs for EX Series Switches | **183**
- *Configuring VLANs for EX Series Switches with ELS Support (CLI Procedure)*
- Example: Setting Up Basic Bridging and a VLAN on Switches | **203**
- Understanding Bridging and VLANs on Switches | **168**
- show ethernet-switching interfaces | **1485**
- show vlans | **1648**

mvrp

List of Syntax

Syntax (EX Series with ELS Support)

```
mvrp {
    interface interface-name {
        join-timer milliseconds;
        leave-timer milliseconds;
        leaveall-timer seconds;
        registration (forbidden | normal);
    }
    join-timer milliseconds;
    leave-timer milliseconds;
    leaveall-timer seconds;
    no-attribute-length-in-pdu
    no-dynamic-vlan;
    traceoptions {
        file filename <files number > <size size> < world-readable | no-world-readable>;
        flag <flag> <disable>;
    }
}
```

Syntax (EX Series)

```
mvrp {
    add-attribute-length-in-pdu;
    disable (MVRP);
    interface (MVRP) (all | interface-name) {
        disable (MVRP);
        join-timer (MVRP) milliseconds;
        leave-timer (MVRP) milliseconds;
        leaveall-timer milliseconds;
        registration (forbidden | normal);
    }
    no-dynamic-vlan;
    traceoptions {
        file filename <files number > <size size> <no-stamp | world-readable | no-world-readable>;
        flag flag;
```

```
    }
}
```

Syntax (MX Series, EX Series, SRX 1500, SRX 300, SRX 550M, SRX 345, SRX 340, SRX 320)

```
mvrp {
    bpdu-destination-mac-address provider-bridge-group;
    join-timer (MVRP) milliseconds;
    leave-timer milliseconds;
    leaveall-timer milliseconds;
    interface (all | interface-name) {
        join-timer (MVRP) milliseconds;
        leave-timer milliseconds;
        leaveall-timer milliseconds;
        point-to-point;
        registration (forbidden | normal | restricted);
    }
    no-attribute-length-in-pdu
    no-dynamic-vlan;
    traceoptions {
        file filename <files number > <size size> <no-stamp | world-readable | no-world-readable>;
        flag flag;
    }
}
```

EX Series with ELS Support

```
[edit protocols]
```

EX Series and MX Series

```
[edit logical-systems logical-system-name protocols],
[edit logical-systems logical-system-name routing-instances routing-instance-name protocols] (for virtual switch
    instance type),
```

```
[edit protocols],
```

```
[edit routing-instances routing-instance-name protocols] (for virtual switch instance type),
```

SRX 1500, SRX 300, SRX 550M, SRX 345, SRX 340, SRX 320

[edit protocols],
[edit routing-instances *routing-instance-name* protocols] (for virtual switch instance type),

Release Information

Statement introduced in Junos OS Release 10.0 for EX Series switches.
Statement introduced in Junos OS Release 10.1 for MX Series routers.
Statement introduced in Junos OS Release 13.2X50-D10 for EX Series switches with ELS support.
Statement introduced in Junos OS Release 15.1X49-D70 for SRX Series devices.

Description

For Layer 2 networks, configure Multiple VLAN Registration Protocol (MVRP) to dynamically share VLAN information and dynamically configure needed VLANs. Maintaining VLAN configurations based on active VLANs reduces the amount of traffic traveling in the network, saving network resources. MVRP is configured on trunk interfaces.

Configure Multiple VLAN Registration Protocol (MVRP) on a trunk interface to ensure that the VLAN membership information on the trunk interface is updated as the switch's access interfaces become active or inactive in the configured VLANs.

NOTE: At Junos OS Release 11.3, MVRP was updated to conform to the IEEE standard 802.1ak. This update might result in compatibility issues in mixed release networks. For details, see "Configuring Multiple VLAN Registration Protocol (MVRP) on Switches" on page 797.

The remaining statements are explained separately. See CLI Explorer.

Default

MVRP is disabled by default.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Example: Configuring Automatic VLAN Administration Using MVRP on MX Series Routers

Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches | 832

Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches with ELS Support | 815

Configuring Multiple VLAN Registration Protocol (MVRP) to Manage Dynamic VLAN Registration

Understanding Multiple VLAN Registration Protocol (MVRP) for Dynamic VLAN Registration | 793

Configuring Multiple VLAN Registration Protocol (MVRP) on Switches | 797

Verifying That MVRP Is Working Correctly | 851

Understanding Multiple VLAN Registration Protocol (MVRP) for Dynamic VLAN Registration | 793

native-vlan-id

Syntax

```
native-vlan-id vlan-id;
```

Hierarchy Level (QFX Series and EX4600)

For platforms without ELS:

```
[edit interfaces (QFX Series) interface-name unit 0 family ethernet-switching]
```

For platforms with ELS:

```
[edit interfaces (QFX Series) interface-name]
```

Hierarchy Level (ACX Series, EX Series, SRX Series, M Series, MX Series, and T Series)

```
[edit interfaces ge-fpc/pic/port],
[edit interfaces interface-name]
```

Hierarchy Level (SRX Series)

```
[edit interfacesinterface-name ]
```

Release Information

Statement introduced in Junos OS Release 8.3.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 9.5 for SRX Series.
Statement introduced in Junos OS Release 11.1 for the QFX Series.
Statement introduced in Junos OS Release 12.2 for ACX Series Universal Metro Routers.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.
Statement introduced in Junos OS Release 13.2X51-D20 for the QFX Series.

Description

Configure the VLAN identifier to associate with untagged packets received on the physical interface of a trunk mode interface for the following:

- QFX Series and EX4600
- M Series routers with Gigabit Ethernet IQ PICs with SFP and Gigabit Ethernet IQ2 PICs with SFP configured for 802.1Q flexible VLAN tagging

- MX Series routers with Gigabit Ethernet DPCs and MICs, Tri-Rate Ethernet DPCs and MICs, and 10-Gigabit Ethernet DPCs and MICs and MPCs configured for 802.1Q flexible VLAN tagging
- T4000 routers with 100-Gigabit Ethernet Type 5 PIC with CFP
- EX Series switches with Gigabit Ethernet, 10-Gigabit Ethernet, 40-Gigabit Ethernet, and aggregated Ethernet interfaces

The logical interface on which untagged packets are received must be configured with the same VLAN ID as the native VLAN ID configured on the physical interface, otherwise the untagged packets are dropped. To configure the logical interface, include the **vlan-id** statement (matching the **native-vlan-id** statement on the physical interface) at the **[edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level**.

When the **native-vlan-id** statement is included with the **flexible-vlan-tagging** statement, untagged packets are accepted on the same mixed VLAN-tagged port and on the interfaces that are configured for Q-in-Q tunneling.

When the **native-vlan-id** statement is combined with the **interface-mode** statement, untagged packets are accepted and forwarded within the bridge domain or VLAN that is configured with the matching VLAN ID.

To configure the logical interface, include the **vlan-id** statement (matching the **native-vlan-id** statement on the physical interface) at the **[edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level**.

> **NOTE:** Starting in Junos OS Release 17.1R1, you can send untagged traffic without a native VLAN ID to the remote end of the network. To do this, remove the native VLAN ID from the untagged traffic configuration by setting the **no-native-vlan-insert** statement. If you do not configure this statement, the native VLAN ID is added to the untagged traffic.

Default

By default, the untagged packets are dropped. That is, if you do not configure the **native-vlan-id** option, the untagged packets are dropped.

Options

vlan-id—Numeric identifier of the VLAN.

Range: 1 through 4094

number—VLAN ID number.

Range: (ACX Series routers, SRX Series devices and EX Series switches) 0 through 4094.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring Gigabit Ethernet Interfaces (CLI Procedure)

Configuring Gigabit Ethernet Interfaces (J-Web Procedure)

Understanding Bridging and VLANs on Switches | **168**

Enabling VLAN Tagging

Configuring Access Mode on a Logical Interface | **322**

Configuring the Native VLAN Identifier on Switches With ELS Support | **290**

Understanding Interfaces

Understanding Q-in-Q Tunneling and VLAN Translation | **887**

no-native-vlan-insert | **1288**

Sending Untagged Traffic Without VLAN ID to Remote End | **302**

show ethernet-switching interfaces | **1485**

show vlans | **1648**

flexible-vlan-tagging | **1171**

Junos OS Network Interfaces Configuration Guide

next-hop (Static MAC-Based VLANs)

Syntax

```
next-hop interface-name;
```

Hierarchy Level

```
[edit ethernet-switching-options static vlan vlan-name mac mac-address]
```

Release Information

Statement introduced in Junos OS Release 11.1 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description

Specify the next hop for the indicated Ethernet node.

Options

interface-name—Name of the next-hop interface.

Required Privilege Level

system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Adding a Static MAC Address Entry to the Ethernet Switching Table | 113

no-attribute-length-in-pdu

Syntax

```
no-attribute-length-in-pdu;
```

Hierarchy Level

```
[edit protocols mvrp]
```

Release Information

Statement introduced in Junos OS Release 13.2X50-D10 for EX Series switches.
Statement introduced in Junos OS Release 15.1X49-D70 for SRX Series devices.

Description

Include an extra byte in protocol data units (PDUs) sent by the Multiple VLAN Registration Protocol (MVRP). You can disable the extra byte to address a compatibility issue between MVRP in Junos OS Releases 13.2 and later for EX Series switches with support for the Enhanced Layer 2 Software (ELS), which includes the extra byte, and MVRP in Junos OS Releases 11.3 and later for EX Series switches that do not support ELS, which does not include the extra byte. If this compatibility issue arises, the ELS version of MVRP does not recognize PDUs without the extra byte sent by the non-ELS version of MVRP.

You can recognize an MVRP version compatibility issue by observing the switch running the ELS version of MVRP. Because a switch running the ELS version of MVRP cannot interpret an unmodified PDU from a switch running the non-ELS version of MVRP, the switch will not add VLANs from the non-ELS version of MVRP. When you execute the command **show mvrp statistics** in the ELS version of MVRP, the values for **Received Join Empty** and **Received Join In** will incorrectly display zero, even though the value for the **Received MVRP PDUs without error** has been increased. Another indication that MVRP is having a version compatibility issue is that unexpected VLAN activity, such as multiple VLAN creation, takes place on the switch running the ELS version of MVRP.

Required Privilege Level

routing—To view this statement in the configuration.
routing control—To add this statement to the configuration.

RELATED DOCUMENTATION

- Configuring Multiple VLAN Registration Protocol (MVRP) on Switches | 797
- Understanding Multiple VLAN Registration Protocol (MVRP) | 787
- Understanding Multiple VLAN Registration Protocol (MVRP) for Dynamic VLAN Registration | 793

no-dynamic-vlan

Syntax

```
no-dynamic-vlan;
```

Hierarchy Level

[edit protocols **mvrp**]

[edit routing-instances *routing-instance-name* protocols **mvrp**] (for virtual switch instance type)

Release Information
Statement introduced in Junos OS Release 10.0 for EX Series switches.
Statement introduced in Junos OS Release 15.1X49-D70 for SRX Series devices.

Description
Disable the dynamic creation of VLANs using Multiple VLAN Registration Protocol (MVRP) for interfaces participating in MVRP.

Dynamic VLAN configuration can be enabled on an interface independent of MVRP. The MVRP dynamic VLAN configuration setting does not override the interface configuration dynamic VLAN configuration setting. If dynamic VLAN creation is disabled on the interface in the interface configuration, no dynamic VLANs are created on the interface, including dynamic VLANs created using MVRP.

This option can be applied globally; it cannot be applied per interface.

Default
If MVRP is enabled, the dynamic creation of VLANs as a result of MVRP protocol exchange messages is enabled.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring Multiple VLAN Registration Protocol (MVRP) on Switches | **797**
Understanding Multiple VLAN Registration Protocol (MVRP) for Dynamic VLAN Registration | **793**

no-gratuitous-arp-request

Syntax

```
no-gratuitous-arp-request;
```

Hierarchy Level

```
[edit interfaces interface-name]
```

Release Information

Statement introduced in Junos OS Release 9.6 for EX Series switches.
Statement introduced in Junos OS Release 12.2 for ACX Series Universal Metro Routers.

Description

For Ethernet interfaces and pseudowire logical interfaces, do not respond to gratuitous ARP requests.

Default

Gratuitous ARP responses are enabled on all Ethernet interfaces.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring Gratuitous ARP | 349

no-local-switching

Syntax

```
no-local-switching
```

Hierarchy Level

```
[edit vlans vlan-name]
```

Release Information

Statement introduced in Junos OS Release 9.3 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description

Specify that access ports in this VLAN domain do not forward packets to each other. You use this statement with primary VLANs and isolated secondary VLANs.

Required Privilege Level

system—To view this statement in the configuration.
system–control—To add this statement to the configuration.
routing—To view this statement in the configuration.
routing–control—To add this statement to the configuration.

RELATED DOCUMENTATION

Creating a Private VLAN on a Single QFX Switch | 475
Creating a Private VLAN Spanning Multiple QFX Series Switches | 479

no-mac-learning

Syntax

```
no-mac-learning;
```

QFX Series and EX4600

For QFX Series and EX4600 platforms without ELS:

```
[edit ethernet-switching-options interfaces interface-name]
```

For QFX Series and EX4600 platforms with ELS:

```
[edit vlans vlan-name switch-options]
```

QFX Series per VLAN

```
[edit vlans vlan-name]
```

```
[edit vlans vlan-name switch-options]
```

EX Series Q-in-Q Interfaces

```
[edit ethernet-switching-options interfaces interface-name]
```

EX Series and SRX Series Q-inQ Vlans

```
[edit vlans vlan-name]
```

ACX Series, MX Series, EX Series with ELS support, M Series, T Series

```
[edit bridge-domains bridge-domain-name bridge-options],
[edit bridge-domains bridge-domain-name bridge-options interface interface-name],
[edit logical-systems logical-system-name bridge-domains bridge-domain-name bridge-options],
[edit logical-systems logical-system-name bridge-domains bridge-domain-name bridge-options interface interface-name],
[edit logical-systems logical-system-name routing-instances routing-instance-name bridge-domains bridge-domain-name
    bridge-options],
[edit logical-systems logical-system-name routing-instances routing-instance-name bridge-domains bridge-domain-name
    bridge-options interface interface-name],
```

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* switch-options],
[edit logical-systems *logical-system-name* switch-options],
[edit **bridge-domains** *bridge-domain-name* bridge-options interface *interface-name*],
[edit routing-instances *routing-instance-name* **bridge-domains** *bridge-domain-name* bridge-options],
[edit routing-instances *routing-instance-name* **bridge-domains** *bridge-domain-name* bridge-options interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols evpn],
[edit routing-instances *routing-instance-name* protocols evpn interface *interface-name*],
[edit routing-instances *routing-instance-name* switch-options],
[edit switch-options],
[edit switch-options],
[edit switch-options interface *interface-name*],
[set vlans *vlan-name* switch-options]

Release Information

Statement introduced in Junos OS Release 8.4.
Support for the **switch-options** statement added in Junos OS Release 9.2.
Support for top-level configuration for the **virtual-switch** type of routing instance added in Junos OS Release 9.2. In Junos OS Release 9.1 and earlier, the routing instances hierarchy supported this statement only for a VPLS instance or bridge domain configured within a virtual switch.
Statement introduced in Junos OS Release 9.5 for EX Series switches.
Support for logical systems added in Junos OS Release 9.6.
Statement introduced in Junos OS Release 11.1 for the QFX Series.
[edit switch-options], **[edit switch-options interface *interface-name*]**, **[edit vlans *vlan-name* switch-options]**, and **[edit vlans *vlan-name* switch-options interface *interface-name*]** hierarchy levels introduced in Junos OS Release 12.3 R2 for EX Series switches.
Support for EVPNs added in Junos OS Release 13.2 for MX 3D Series routers.
Hierarchy levels **[edit switch-options interface *interface-name*]** and **[edit vlans *vlan-name* switch-options]** introduced in Junos OS Release 13.2X50-D10 for EX Series switches.

Description

For QFX Series, EX Series switches and SRX Series devices, disables MAC address learning for the specified VLAN.

For QFX Series and EX4600, disable MAC address learning for the specified interface. Disabling MAC address learning on an interface disables learning for all the VLANs of which that interface is a member.

For EX Series switches' Q-in-Q interfaces, disables MAC address learning for the specified interface. Disabling MAC address learning on an interface disables learning for all the VLANs of which that interface is a member.

For MX Series routers and EX Series switches with ELS support, disables MAC learning for a virtual switch, for a bridge domain or VLAN, for a specific logical interface in a bridge domain or VLAN, or for a set of bridge domains or VLANs associated with a Layer 2 trunk port. On platforms that support EVPNs, you can disable MAC learning on an EVPN.

> **NOTE:** When MAC learning is disabled for a VPLS routing instance, traffic is not load-balanced and only one of the equal-cost next hops is used.

Default

MAC learning is enabled.

Required Privilege Level

system—To view this statement in the configuration.
system–control—To add this statement to the configuration.
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- *Configuring EVPN Routing Instances*
- *Configuring EVPN Routing Instances on EX9200 Switches*
- *Understanding Layer 2 Learning and Forwarding for Bridge Domains*
- Layer 2 Learning and Forwarding for VLANs Overview | **78**
- *Understanding Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports*
- Understanding Bridging and VLANs on Switches | **168**
- Understanding Q-in-Q Tunneling and VLAN Translation | **887**
- Understanding Q-in-Q Tunneling and VLAN Translation | **887**
- Configuring Q-in-Q Tunneling on EX Series Switches | **910**

no-native-vlan-insert

Syntax

```
no-native-vlan-insert;
```

Hierarchy Level

[edit interfaces *interface-name*]

Release Information

Statement introduced in Junos OS Release 17.1R1.

Description

Send traffic without the native VLAN ID (**native-vlan-id**) to the remote end of the network if untagged traffic is received.

If this statement is not configured, then **native-vlan-id** is added to untagged traffic. But if this statement is configured, then **native-vlan-id** is not added to untagged traffic.

NOTE:

- This feature works only on MX Series routers with MPCs/MICs. Configuring this statement on MX Series routers with DPCs results is no behavioral change. However, if you configure the statement on aggregated Ethernet (ae) interfaces with logical interfaces across MPCs/MICs and DPCs, then the MPCs/MICs and DPCs behave differently.
- In the egress direction, this feature is disrupted by VLAN normalization. Because of normalization, the egress interface cannot distinguish between untagged traffic and tagged traffic. And untagged traffic is sent out with **native-vlan-id**. Consider this while configuring both VLAN normalization and new **native-vlan-id** statement.

  There will be a problem with ingress firewall filter if filter term includes **native-vlan-id**. With **no-native-vlan-insert** statement configured, **native-vlan-id** will not be inserted to untagged traffic. So, firewall filter term will not match with untagged traffic. But if incoming traffic have VLAN ID which is equal to **native-vlan-id**, then firewall filter term will match and firewall will work.
- When this feature is used with AE, all sub-interfaces of AE should be in same type of FPC.

Default

By default, **native-vlan-id** is inserted to untagged traffic. That is, if this statement is not configured, then **native-vlan-id** is inserted to untagged traffic.

Required Privilege Level

interface—To view this statement in the configuration.

interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring Mixed Tagging Support for Untagged Packets

Configuring Access Mode on a Logical Interface | **322**

Configuring the Native VLAN Identifier on Switches With ELS Support | **290**

Understanding Bridging and VLANs on Switches | **168**

flexible-vlan-tagging | **1171**

native-vlan-id | **1276**

Understanding Q-in-Q Tunneling and VLAN Translation | **887**

Sending Untagged Traffic Without VLAN ID to Remote End | **302**

node-id

Syntax

```
node-id mac-address;
```

Hierarchy Level

```
[edit protocols protection-group ethernet-ring ring-name]
```

Release Information

Statement introduced in Junos OS Release 9.4.

Description

For EX Series switches and QFX Series switches, node-id is not configurable.

For MX Series routers, optionally specify the MAC address of a node in the protection group. If this statement is not included, the router assigns the node's MAC address.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Ethernet Ring Protection Switching Overview

Example: Configuring Ethernet Ring Protection Switching on EX Series Switches | 855

Example: Configuring Ethernet Ring Protection Switching on QFX Series and EX Series Switches Supporting ELS | 874

notification-interval

Syntax

```
notification-interval seconds;
```

Hierarchy Level

```
[edit ethernet-switching-options mac-notification]
[edit switch-options mac-notification]
```

Release Information

Statement introduced in Junos OS Release 9.6 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.
Hierarchy level **[edit switch-options]** added in Junos OS Release 14.1X53-D10 for EX Series and QFX Series.

Description

Configure the MAC notification interval for a switch.

The MAC notification interval is the amount of time the switch waits before sending learned or unlearned MAC address SNMP notifications to the network management server. For instance, if the MAC notification interval is set to 10, all of the MAC address addition and removal SNMP notifications will be sent to the network management system every 10 seconds.

Options

seconds—The MAC notification interval, in seconds.

Range: 1 through 60

Default: 30

Required Privilege Level

system—To view this statement in the configuration.
system-control—To add this statement to the configuration.
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring Non-ELS MAC Notification | 130

Configuring MAC Notification on Switches with ELS Support | 129

num-65-127-prefix

Syntax

num-65-127-prefix *number*;

Hierarchy Level

[edit chassis (QFX Series) **forwarding-options** *profile-name*]

Release Information

Statement introduced in Junos OS Release 13.2 for QFX Series switches.
Support for QFX5200 Series switches introduced in Junos OS Release 15.1X53-D30.

Description

For the Unified Forwarding Table (UFT) feature, specify how much forwarding table memory to allocate for IPv6 entries with prefix lengths in the range of /65 through /127. The ability to allocate flexibly the memory for IPv6 entries with prefixes in this range extends the use of this memory space to accommodate the appropriate mix of longest-prefix match (LPM) entries that best suits your network. The LPM table stores IPv4 unicast prefixes, IPv6 prefixes with lengths equal to or less than 64, and IPv6 prefixes with lengths from 65 through 127. With this option, you can increase, decrease, or allocate no memory for IPv6 prefixes with lengths from 65 through 127, depending on which version of Junos OS you are using.

NOTE: This statement is supported only for the following forwarding table memory profiles: **l2-profile-one**, **l2-profile-three**, **l2-profile-two**, and **l3-profile**. Do not use this statement with the **custom-profile** or the **lpm-profile** statements.

NOTE: The values you can configure are different depending on the version of Junos OS you are using.

Options

number—Specify a numerical value.

Range: (Junos OS Release 13.2x51-D10 only) 1 through 128. Each increment represents 16 IPv6 prefixes with lengths in the range of /65 through /127, for a total maximum of 2,058 prefixes (16 x 128 = 2,048).

Default: 1 (16 IPv6 prefixes with lengths in the range of /65 through /127).

Range: (Junos OS Release 13.2X51-D15 or later) 0 through 4. Each increment allocates memory for 1,000 IPv6 prefixes with lengths in the range of /65 through /127, for a maximum of 4,000 such IPv6 prefixes.

Default: 1 (1,000 IPv6 prefixes with lengths in the range of /65 through /127).

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring the Unified Forwarding Table on Switches | 94

output-vlan-map

Syntax

```
output-vlan-map {
    (pop | pop-pop | pop-swap | push | push-push | swap | swap-push | swap-swap);
    inner-tag-protocol-id tpid;
    inner-vlan-id number;
    tag-protocol-id tpid;
    vlan-id number;
}
```

Hierarchy Level

[edit interfaces *interface-name* unit *logical-unit-number*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]

Release Information

Statement introduced before Junos OS Release 7.4.
pop-pop, **pop-swap**, **push-push**, **swap-push**, and **swap-swap** statements added in Junos OS Release 8.1.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.

Description

For EX Series switches, defines the rewrite operation to be applied to outgoing frames.

For MX Series routers and NFX Series devices' Gigabit Ethernet IQ and 10-Port 10-Gigabit Ethernet SFPP interfaces only, defines the rewrite operation to be applied to outgoing frames on this logical interface.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Stacking and Rewriting Gigabit Ethernet VLAN Tags | 382

input-vlan-map | 1195

packet-action

Syntax

packet-action *action*;

Hierarchy Level

[edit **bridge-domains** *bridge-domain-name* bridge-options interface *interface-name* **interface-mac-limit** *limit*],
[edit **bridge-domains** *bridge-domain-name* bridge-options **interface-mac-limit** *limit*],
[edit logical-systems *logical-system-name* **bridge-domains** *bridge-domain-name* bridge-options interface *interface-name* **interface-mac-limit** *limit*],
[edit logical-systems *logical-system-name* **bridge-domains** *bridge-domain-name* bridge-options **interface-mac-limit** *limit*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* **bridge-domains** *bridge-domain-name* bridge-options interface *interface-name* **interface-mac-limit** *limit*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* **bridge-domains** *bridge-domain-name* bridge-options **interface-mac-limit** *limit*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* switch-options interface *interface-name* **interface-mac-limit** *limit*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* switch-options **interface-mac-limit** *limit*],
[edit logical-systems *logical-system-name* switch-options **interface-mac-limit** *limit*],
[edit protocols **l2-learning** global-mac-limit *limit*],
[edit routing-instances *routing-instance-name* **bridge-domains** *bridge-domain-name* bridge-options interface *interface-name* **interface-mac-limit** *limit*],
[edit routing-instances *routing-instance-name* **bridge-domains** *bridge-domain-name* bridge-options **interface-mac-limit** *limit*],
[edit routing-instances *routing-instance-name* protocols evpn interface-mac-limit (VPLS)],
[edit routing-instances *routing-instance-name* protocols evpn interface *interface-name* interface-mac-limit (VPLS)],
[edit routing-instances *routing-instance-name* protocols evpn mac-table-size *limit*],
[edit routing-instances *routing-instance-name* switch-options interface *interface-name* **interface-mac-limit** *limit*],
[edit routing-instances *routing-instance-name* switch-options **interface-mac-limit** *limit*],
[edit switch-options **interface-mac-limit** *limit*],
[edit switch-options interface *interface-name* **interface-mac-limit** *limit*],
[edit switch-options **interface-mac-limit** *limit*],
[edit switch-options interface *interface-name* **interface-mac-limit** *limit*],
[edit switch-options **interface-mac-limit** *limit*],
[edit switch-options **mac-table-size** *limit*],
[edit switch-options interface *interface-name* **interface-mac-limit** *limit*],
[edit vlans *vlan-name* switch-options interface *interface-name* **interface-mac-limit** *limit*],
[edit vlans *vlan-name* switch-options **interface-mac-limit** *limit*],
[edit vlans *vlan-name* switch-options **mac-table-size** *limit*]

[edit vlans *vlan-name* switch-options **interface-mac-limit** *limit*],
[edit vlans *vlan-name* switch-options interface *interface-name* **interface-mac-limit** *limit*],
[edit vlans *vlan-name* switch-options **mac-table-size** *limit*]

Release Information

Statement introduced in Junos OS Release 8.4.
Support for the **switch-options** statement added in Junos OS Release 9.2.
Support for top-level configuration for the **virtual-switch** type of routing instance added in Junos OS Release 9.2. In Junos OS Release 9.1 and earlier, the routing instances hierarchy supported this statement only for a VPLS instance or a bridge domain configured within a virtual switch.
Support for logical systems added in Junos OS Release 9.6.
[edit switch-options interface *interface-name* interface-mac-limit *limit*], [edit switch-options interface-mac-limit *limit*], [edit switch-options mac-table-size *limit*], [edit vlans *vlan-name* switch-options interface *interface-name* interface-mac-limit *limit*], [edit vlans *vlan-name* switch-options interface-mac-limit *limit*], and [edit vlans *vlan-name* switch-options mac-table-size *limit*] hierarchy levels introduced in Junos OS Release 12.3R2 for EX Series switches.
Support for EVPNs introduced in Junos OS Release 13.2 on MX Series 5G Universal Routing Platforms.
Support at the **[edit switch-options interface *interface-name* interface-mac-limit *limit*]** hierarchy level and hierarchy levels under **[edit vlans *vlan-name*]** introduced in Junos OS Release 13.2X50-D10 for EX Series switches and Junos OS Release 13.2 for the QFX Series.

Description

Specify the action taken when packets with new source MAC addresses are received after the MAC address limit is reached. If this statement is not configured, packets with new source MAC addresses are forwarded by default.

NOTE: The **packet-action** statement is not supported on the QFX10002-60C switch.

Default

NOTE: On a QFX Series Virtual Chassis, if you include the **shutdown** option at the **[edit vlans *vlan-name* switch-options interface *interface-name* interface-mac-limit packet-action]** hierarchy level and issue the **commit** operation, the system generates a commit error. The system does not generate an error if you include the **shutdown** option at the **[edit switch-options interface *interface-name* interface-mac-limit packet-action]** hierarchy level.

Disabled. The default is for packets for new source MAC addresses to be forwarded after the MAC address limit is reached.

Options

drop—Drop packets with new source MAC addresses, and do not learn the new source MAC addresses.

> **NOTE:** On QFX10000 switches, if you include the drop option, you cannot configure unicast reverse-path forwarding (URFP) on integrated routing and bridging (IRB) and MAC limiting on the same interface. If you have an MC-LAG configuration, you cannot configure MAC limiting on the interchassis link (ICL) interface.

drop-and-log—(EX Series switches and QFX Series only) Drop packets with new source MAC addresses, and generate an alarm, an SNMP trap, or a system log entry.

log—(EX Series switches and QFX Series only) Hold packets with new source MAC addresses, and generate an alarm, an SNMP trap, or a system log entry.

none—(EX Series switches and QFX Series only) Forward packets with new source MAC addresses, and learn the new source MAC address.

shutdown—(EX Series switches and QFX Series only) Disable the specified interface, and generate an alarm, an SNMP trap, or a system log entry.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- *Configuring EVPN Routing Instances*
- *Configuring EVPN Routing Instances on EX9200 Switches*
- Configuring MAC Limiting (ELS) | **110**
- *Configuring Persistent MAC Learning (ELS)*
- *Understanding Layer 2 Learning and Forwarding for Bridge Domains*
- Layer 2 Learning and Forwarding for VLANs Overview | **78**
- *Understanding Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports*
- Layer 2 Learning and Forwarding for VLANs Overview | **78**
- Layer 2 Learning and Forwarding for VLANs Acting as a Switch for a Layer 2 Trunk Port | **80**

passive (MVRP)

Syntax

```
passive;
```

Hierarchy Level

```
[edit protocols mvrp],
[edit protocols mvrp interface(all | interface-name)]
```

Release Information

Statement introduced in Junos OS Release 13.1 for the QFX Series.

Description

Configure an MVRP interface to not announce its membership in a VLAN or send any VLAN declarations (updates) unless it receives registration for that VLAN from a peer (server).

Default

Passive mode is disabled by default.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Example: Configuring Automatic VLAN Administration on QFX Switches Using MVRP | 808

peer-selection-service

Syntax

```
peer-selection-service {
    command binary-file-path;
    disable;
    failover (alternate-media | other-routing-engine);
}
```

Hierarchy Level

```
[edit system processes]
```

Release Information

Statement introduced in Junos OS Release 8.5.

Description

Enable the peer selection service process.

Options

- **command *binary-file-path***—Path to the binary process.
- **disable**—Disable the peer selection service process.
- **failover**—Configure the device to reboot if the software process fails four times within 30 seconds, and specify the software to use during the reboot.
  - **alternate-media**—Configure the device to switch to backup media that contains a version of the system if a software process fails repeatedly.
  - **other-routing-engine**—Instruct the secondary Routing Engine to take mastership if a software process fails. If this statement is configured for a process, and that process fails four times within 30 seconds, then the device reboots from the secondary Routing Engine.

Required Privilege Level

system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Interfaces User Guide for Security Devices

pgcp-service

Syntax

```
pgcp-service {
    command binary-file-path;
    disable;
    failover (alternate-media | other-routing-engine);
}
```

Hierarchy Level

```
[edit system processes]
```

Release Information

Statement introduced in Junos OS Release 8.5.

Description

Specify the Packet Gateway Control Protocol (PGCP) that is required for the border gateway function (BGF) feature.

Options

- **command *binary-file-path***—Path to the binary process.
- **disable**—Disable the Packet Gateway Control Protocol (PGCP) process.
- **failover**—Configure the device to reboot if the software process fails four times within 30 seconds, and specify the software to use during the reboot.
  - **alternate-media**—Configure the device to switch to backup media that contains a version of the system if a software process fails repeatedly.
  - **other-routing-engine**—Instruct the secondary Routing Engine to take mastership if a software process fails. If this statement is configured for a process, and that process fails four times within 30 seconds, the device reboots from the secondary Routing Engine.

Required Privilege Level

system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

restart (Reset)

point-to-point (MVRP)

Syntax

```
point-to-point;
```

Hierarchy Level

```
[edit logical-systems logical-system-name protocols mvrp interface (all | interface-name)],
[edit logical-systems logical-system-name routing-instances routing-instance-name protocols mvrp interface (all |
    interface-name)] (for virtual switch instance type),
```

```
[edit protocols mvrp interface (all | interface-name)],
```

```
[edit routing-instances routing-instance-name protocols mvrp interface (all | interface-name)] (for virtual switch
    instance type)
```

Release Information

Statement introduced in Junos OS Release 10.1 for MX Series routers.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.
Statement introduced in Junos OS Release 15.1X49-D70 for SRX Series devices.

Description

(Optional) For Multiple VLAN Registration Protocol (MVRP) configurations, configure an interface to be recognized as a point-to-point connection. If specified, a point-to-point subset of the MRP state machine is used to provide a simpler and more efficient method to accelerate convergence on the network. Point-to-point must be enabled after enabling MVRP for the interface to be recognized as a point-to-point connection.

Default

MVRP is disabled by default.

point-to-point is disabled by default.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Example: Configuring Automatic VLAN Administration Using MVRP on MX Series Routers

Configuring Multiple VLAN Registration Protocol (MVRP) to Manage Dynamic VLAN Registration

Verifying That MVRP Is Working Correctly | **851**

Understanding Multiple VLAN Registration Protocol (MVRP) for Dynamic VLAN Registration | **793**

Understanding Multiple VLAN Registration Protocol (MVRP) for Dynamic VLAN Registration | **793**

Understanding Multiple VLAN Registration Protocol (MVRP) | **787**

join-timer | **1227**

leaveall-timer | **1240**

leave-timer | **1238**

registration | **1340**

pop

Syntax

```
pop;
```

Hierarchy Level

[edit interfaces *interface-name* unit *logical-unit-number* **input-vlan-map**],
[edit interfaces *interface-name* unit *logical-unit-number* **output-vlan-map**],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* input-vlan-map],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* output-vlan-map]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.
Statement introduced in Junos OS Release 13.2X51-D20 for the QFX Series.

Description

NOTE: On EX4300 switches, **pop** is not supported at the [edit interfaces *interface-name* unit *logical-unit-number* input-vlan-map] hierarchy level.

For Gigabit Ethernet IQ, 10-Gigabit Ethernet IQ2, and IQ2-E interfaces; 10-Gigabit Ethernet LAN/WAN PIC; aggregated Ethernet interfaces using Gigabit Ethernet IQ interfaces; 100-Gigabit Ethernet Type 5 PIC with CFP; and Gigabit Ethernet, 10-Gigabit Ethernet, 40-Gigabit Ethernet, and aggregated Ethernet interfaces, specify the VLAN rewrite operation to remove a VLAN tag from the top of the VLAN tag stack. The outer VLAN tag of the frame is removed.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Removing a VLAN Tag | **395**
Configuring Q-in-Q Tunneling on EX Series Switches with ELS Support | **900**

pop-pop

Syntax

```
pop-pop;
```

Hierarchy Level

[edit interfaces *interface-name* unit *logical-unit-number* **input-vlan-map**],
[edit interfaces *interface-name* unit *logical-unit-number* **output-vlan-map**],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* **input-vlan-map**],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* **output-vlan-map**]

Release Information

Statement introduced in Junos OS Release 8.1.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.

Description

For Gigabit Ethernet IQ, IQ2 and IQ2-E interfaces, 10-Gigabit Ethernet LAN/WAN PIC, for aggregated Ethernet interfaces using Gigabit Ethernet IQ2 and IQ2-E or 10-Gigabit Ethernet PICs on MX Series routers, and 100-Gigabit Ethernet Type 5 PIC with CFP, and for 10-Gigabit Ethernet SFP interfaces on EX Series switches, specify the VLAN rewrite operation to remove both the outer and inner VLAN tags of the frame.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Removing the Outer and Inner VLAN Tags | **395**

pop-swap

Syntax

```
pop-swap;
```

Hierarchy Level

[edit interfaces *interface-name* unit *logical-unit-number* **input-vlan-map**],
[edit interfaces *interface-name* unit *logical-unit-number* **output-vlan-map**],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* **input-vlan-map**],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* **output-vlan-map**]

Release Information

Statement introduced in Junos OS Release 8.1.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.

Description

Specify the VLAN rewrite operation to remove the outer VLAN tag of the frame, and replace the inner VLAN tag of the frame with a user-specified VLAN tag value. The inner tag becomes the outer tag in the final frame.

You can use this statement on Gigabit Ethernet IQ, IQ2, IQ2-E interfaces, 10-Gigabit Ethernet LAN/WAN PIC, on aggregated Ethernet interfaces using Gigabit Ethernet IQ2 and IQ2-E or 10-Gigabit Ethernet PICs on MX Series routers, and 100-Gigabit Ethernet Type 5 PIC with CFP.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Removing the Outer VLAN Tag and Rewriting the Inner VLAN Tag | 396

port-mode

Syntax

```
port-mode (access | tagged-access | trunk);
```

Hierarchy Level

[edit interfaces *interface-name* unit *logical-unit-number* family **ethernet-switching**]

Release Information

Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description

> **NOTE:** This statement does not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see **interface-mode**. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 50.

Configure whether an interface on the switch operates in access, tagged access, or trunk mode.

Default

All switch interfaces are in access mode.

Options

access—Have the interface operate in access mode. In this mode, the interface can be in a single VLAN only. Access interfaces typically connect to network devices, such as PCs, printers, IP telephones, and IP cameras.

tagged-access—Have the interface operate in tagged-access mode. In this mode, the interface can be in multiple VLANs. Tagged access interfaces typically connect to network devices, such as PCs, printers, IP telephones, and IP cameras.

trunk—Have the interface operate in trunk mode. In this mode, the interface can be in multiple VLANs and can multiplex traffic between different VLANs. Trunk interfaces typically connect to other switches and to routers on the LAN.

NOTE: The number of VLANs supported per switch varies for each model. Use the configuration-mode command **set vlans id vlan-id ?** to determine the maximum number of VLANs allowed on a switch. You cannot exceed this VLAN limit because each VLAN is assigned an ID number when it is created. You can, however, exceed the recommended VLAN member maximum. To determine the maximum number of VLAN members allowed on a switch, multiply the VLAN maximum for the switch times 8 (vmember limit = vlan max * 8).

If a switch configuration exceeds the recommended VLAN member maximum, you see a warning message when you commit the configuration. If you ignore the warning and commit such a configuration, the configuration succeeds but you run the risk of crashing the Ethernet switching process (**eswd**) due to memory allocation failure.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- *Example: Connecting an EX Series Access Switch to a Distribution Switch*
- *Configuring Gigabit Ethernet Interfaces (CLI Procedure)*
- Configuring VLANs for EX Series Switches | **183**
- *Junos OS Ethernet Interfaces Configuration Guide*

preempt-cutover-timer

Syntax

```
preempt-cutover-timer seconds;
```

Hierarchy Level

- For platforms with ELS:

```
[edit switch-options redundant-trunk-group group name]
[edit interfaces name aggregated-ether-options lacp link-protection rtg-config]
[edit interfaces name aggregated-ether-options link-protection rtg-config]
```

- For platforms without ELS:

```
[edit ethernet-switching-optionsredundant-trunk-group group name]
```

Release Information

Statement introduced in Junos OS Release 11.1 for EX Series switches.
Hierarchy level **[edit switch-options]** introduced in Junos OS Release 13.2X50-D10 (ELS). (See "Using the Enhanced Layer 2 Software CLI" on page 50 for information about ELS.)
Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series.

Description

Change the length of time that a re-enabled primary link waits to take over from an active secondary link in a redundant trunk group (RTG).

Default

If you do not change the time with the **preempt-cutover-timer** statement, a re-enabled primary link takes over from the active secondary link after 1 second.

Options

seconds—Number of seconds that the primary link waits to take over from the active secondary link.

Range: 1 through 600 seconds

Required Privilege Level

admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

RELATED DOCUMENTATION

prefix-65-127-disable

Syntax

```
prefix-65-127-disable;
```

Hierarchy Level

[edit chassis (QFX Series) **forwarding-options** lpm-profile]

Release Information

Statement introduced in Junos OS Release 13.2X51-D15 for QFX Series switches.
Support introduced in Junos OS Release 15.1X53-D30 for QFX5200 Series switches.
Support introduced in Junos OS Release 18.1R1 for QFX5200-48C and QFX5210 switches.

Description

For the Unified Forwarding Table (UFT) feature, specify not to allocate any memory for IPv6 prefixes with lengths in the range /65 through /127 for longest-prefix-match (LPM) entries. Doing so increases the memory available for LPM entries for IPv4 unicast prefixes and IPv6 prefixes with lengths equal to or less than 64. The maximum default value for LPM entries is 16,000 IPv6 prefixes of all lengths.

In an environment where the switch is being used in the core of the network, for example, it might not need to store IPv6 prefixes with lengths in the range /65 through /127. IPv6 prefixes of this type are not typically used in the core.

NOTE: When using this statement, IPv6 prefixes within the range /65 through /127 will still appear in the routing table, but will *not* be installed in the forwarding table; therefore, matching traffic will be dropped. Note further that if a default route is configured, traffic will be forwarded, though it will be sent through the RE and rate-limited.

NOTE: On QFX5100 switches, when you configure this statement, the maximum number of LPM IPv6 entries with prefix lengths equal to or less than 64 increases to 128,000. On the QFX5200 switch, when you configure this statement, the maximum number of IPv6 entries with prefix lengths equal to or less 64 that are allocated in the LPM table increases to 98,0000.

NOTE: This statement is supported only with the **lpm-profile**. No other profile is supported.

The effects of this statement can be seen on a QFX5100 as follows:

```
[edit]
```

```
user@host# set chassis forwarding-options lpm-profile prefix-65-127-disable
```

```
[edit]
```

```
user@host# commit
```

```
configuration check succeeds
commit complete
```

```
[edit]
```

```
user@host# run show chassis forwarding-options
```

```
fpc0:
--------------------------------------------------------------------------
Current UFT Configuration:
lpm-profile. (MAC: 32K L3-host: 16K LPM: 128K)
prefix-65-127 = disable
```

```
[edit]
```

```
user@host# run show pfe route summary hw
```

```
Slot 0
================ fpc0 ================

Unit: 0
Profile active: lpm-profile
Type            Max       Used      Free      % free
-----------------------------------------------------
IPv4 Host       16384     20        16354     99.82
IPv4 LPM        131072    5         131065    99.99
IPv4 Mcast      8192      0         8177      99.82

IPv6 Host       8192      5         8177      99.82
IPv6 LPM(< 64)  131072    2         131065    99.99
```

```
IPv6 LPM(> 64) 0      0      0      0.00
IPv6 Mcast       4096       0          4089       99.83
```

Options

None—This statement has no options.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring the Unified Forwarding Table on Switches | 94

Understanding the Unified Forwarding Table

primary-vlan

Syntax

```
primary-vlan vlan-name;
```

Hierarchy Level

```
[edit vlans vlan-name]
```

Release Information

Statement introduced in Junos OS Release 9.3 for EX Series switches.
Statement updated with enhanced **?** (CLI completion feature) functionality in Junos OS Release 9.5 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description

For a private VLAN (PVLAN), configure the primary VLAN. The primary VLAN is always tagged.

- If the PVLAN is configured on a single switch, do not assign a tag to the community VLANs.
- If the PVLAN is configured to span multiple switches, you must assign tags to the community VLANs also.

For a community VLAN, configure the primary VLAN. The primary VLAN must be tagged, and the community VLAN must be untagged.

If you want to create a community VLAN, you must configure the primary VLAN to be private using the **pvlan** statement.

> **TIP:** To display a list of all configured VLANs on the system, including VLANs that are configured but not committed, type **?** after **vlan** or **vlans** in your configuration mode command line. Note that only one VLAN is displayed for a VLAN range.

Required Privilege Level

system—To view this statement in the configuration.
system–control—To add this statement to the configuration.
routing—To view this statement in the configuration.
routing–control—To add this statement to the configuration.

RELATED DOCUMENTATION

private-vlan

Syntax

```
private-vlan (isolated | community) vlan-id number;
```

Hierarchy Level

```
[edit vlans vlan-name]
```

Release Information

Statement introduced in Junos OS Release 14.1X53-D10 for EX Series switches.
Statement introduced in Junos OS Release 14.1X53-D15 for QFX Series switches.

Description

Configure a secondary VLAN (either an isolated VLAN or a community VLAN) within a private VLAN (PVLAN) and specify a VLAN ID for that secondary VLAN. This statement essentially converts a VLAN into a PVLAN, by carving out discrete subdomains (secondary VLANs) within the primary VLAN. You must specify a VLAN ID for each secondary PVLAN.

> **NOTE:** After you have configured the secondary VLAN, you must also configure its association with a specific primary VLAN. See **isolated-vlan** and **community-vlan** for additional information.

Options

- **isolated** — The VLAN specified by *vlan-name* is defined as an *isolated* VLAN and a VLAN-ID is assigned to it. An isolated VLAN receives packets only from the primary VLAN and forwards frames upstream to the primary VLAN. The VLAN name is optional. The VLAN ID is required.
- **community** — The VLAN specified by *vlan-name* is defined as community VLAN and a VLAN-ID is assigned to it. A *community* VLAN used to transport frames among members of a community, which is a subset of users within the VLAN, and to forward frames upstream to the primary VLAN. The VLAN name is optional. The VLAN ID is required.

Required Privilege Level

system—To view this statement in the configuration.
system–control—To add this statement to the configuration.

RELATED DOCUMENTATION

profile (Access)

Syntax

```
profile profile-name  {
    accounting {
        accounting-stop-on-access-deny;
        accounting-stop-on-failure;
        duplication;
        duplication-attribute-format;
        duplication-filter;
        duplication-vrf;
        order [accounting-method];
        statistics (time | volume-time);
    }
    address-assignment {
        inet6-pool inet6-pool-name;
        pool pool-name;
    }
    authentication-order (ldap | none | password | radius | s6a | securid);
    charging-service-list;
    client client-name  {
        chap-secret  chap-secret;
        client-group [ group-names ];
        firewall-user {
            password password;
        }
        no-rfc2486;
        pap-password pap-password;
        x-auth ip-address;
    }
    client-name-filter {
        count number;
        domain-name domain-name;
        separator special-character;
    }
    domain-name-server name;
    domain-name-server-inet name;
    domain-name-server-inet6 name;
    jsrc;
    ldap-options {
        assemble {
            common-name common-name;
        }
```

```
        base-distinguished-name base-distinguished-name;
        revert-interval seconds;
        search {
            admin-search {
                distinguished-name distinguished-name;
                password password;
            }
            search-filter search-filter-name;
        }
    }
    ldap-server server-address {
        port port-number;
        retry attempts;
        routing-instance routing-instance-name;
        source-address source-address;
        timeout seconds;
    }
    provisioning-order (gx-plus | jsrc);
    radius;
    radius-options;
    radius-server;
    session-limit-per-username;
    session-options {
        client-group [group-name];
        client-idle-timeout minutes;
        client-session-timeout minutes;
    }
    subscriber;
    wins-server;
}
```

Hierarchy Level

```
[edit access]
```

Release Information

Statement introduced in Junos OS Release 10.4.

inet6-pool and **none** options are introduced in Junos OS Release 20.2R1.

Description

Create a profile containing a set of attributes that define device management access.

Options

name—Profile name

accounting—Specifies the accounting options

address-assignment—Specify the address assignment pool

authentication-order—Order in which authentication mechanisms are used

Values:

- ldap—Light weight directory access protocol
- none—No authentication performed
- password—Locally configured password in access profile
- radius—Remote authentication dial-in user service
- s6a—S6a authentication
- securid—RSA secure ID authentication

charging-service-list—List of used 3gpp charging services

Values:

- ocs—Online charging service

client—Entity requesting access

client-name-filter—Restrictions on client names authenticated on this server

domain-name-server—Default DNS server's IPv4 address

domain-name-server-inet—DNS server's IPv4 address

domain-name-server-inet6—DNS server's IPv6 address

jsrc—Set of JSRC configurations

ldap-options—Light weight directory access protocol options

ldap-server—Light weight directory access protocol server

preauthentication-order—Order in which pre authentication mechanisms are used

Values:

- radius—Remote Authentication Dial-In User Service

radius—Set of RADIUS configurations

radius-options—RADIUS options

radius-server—RADIUS server configuration

session-limit-per-username—Maximum number of sessions allowed per username

Range: 1 through 16

session-options—Options for an authenticated client's session

subscriber—Locally authenticated subscriber configuration

wins-server—Default WINS server's IPv4 address

Required Privilege Level

access—To view this statement in the configuration.

access-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Understanding Interfaces

Understanding User Authentication for Security Devices

Ethernet Switching and Layer 2 Transparent Mode Overview | **41**

promiscuous

Syntax

```
promiscuous;
```

Hierarchy Level

```
[edit vlans vlan-name interface interface-name]
```

Release Information

Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description

Configure an access or trunk port to be promiscuous.

Required Privilege Level

system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Creating a Private VLAN on a Single QFX Switch | 475

Creating a Private VLAN Spanning Multiple QFX Series Switches | 479

protection-group

Syntax

```
protection-group {
    ethernet-ring ring-name {
      data-channel {
        vlan number
      }
      east-interface {
        control-channel channel-name {
          vlan number;
          interface name interface-name
        }
      }
      guard-interval number;
      node-id mac-address;
      restore-interval number;
      ring-protection-link-owner;
      non-revertive;
      wait-to-block-interval number;
      major-ring-name name;
      propagate-tc;
      compatibility-version (1|2);
      ring-id number;
      non-vc-mode;
      dot1p-priority number;
      west-interface {
        control-channel channel-name {
          vlan number;
          interface name interface-name
        }
        virtual-control-channel {
          west-interface name;
          east-interface name;
        }
      }
    }
  control-vlan (vlan-id | vlan-name);
      east-interface {
        node-id mac-address;
        control-channel channel-name {
          vlan number;
          interface name interface-name
```

```
            }
            interface-none
            ring-protection-link-end;
        }
    }
        control-channel channel-name {
            vlan number;
            interface name interface-name
        }
    }
    data-channel {
        vlan number
    }
    guard-interval number;
    node-id mac-address;
    restore-interval number;
    ring-protection-link-owner;
        west-interface {
            node-id mac-address;
            control-channel channel-name {
                vlan number;
                interface name interface-name
            }
            interface-none
            ring-protection-link-end;
        }
        control-channel channel-name {
            vlan number;
            interface name interface-name
        }
    }
}
guard-interval number;
restore-interval number;
traceoptions {
    file filename <no-stamp> <world-readable | no-world-readable> <replace> <size size>;
    flag flag;
}
}
```

Hierarchy Level

```
[edit protocols]
```

Release Information

Statement introduced in Junos OS Release 9.4.
Statement introduced in Junos OS Release 12.1 for EX Series switches.

Description

Configure Ethernet ring protection switching.

The statements are explained separately. All statements apply to MX Series routers. EX Series switches do not assign **node-id** and use **control-vlan** instead of **control-channel**.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- *Ethernet Ring Protection Switching Overview*
- *Ethernet Ring Protection Using Ring Instances for Load Balancing*
- *Example: Configuring Load Balancing Within Ethernet Ring Protection for MX Series Routers*
- *Configuring Ethernet Ring Protection Switching on Switches (CLI Procedure)*
- Example: Configuring Ethernet Ring Protection Switching on EX Series Switches | **855**
- Example: Configuring Ethernet Ring Protection Switching on QFX Series and EX Series Switches Supporting ELS | **874**

protocol

List of Syntax

Syntax (MX Series Routers)

```
protocol (cdp | pvstp | stp | vtp);
```

Syntax (ACX Series Routers)

```
protocol (cdp | elmi | ieee8021x | ieee8023ah | lacp | lldp | mmrp | mvrp | stp | vtp);
```

Syntax (EX2300, EX3400, EX4300, EX4300 Multigigabit Model, EX4600, EX4650, and QFX Series Switches)

```
protocol (cdp | elmi | gvrp | ieee8021x | ieee8023ah | lacp | lldp | mmrp | mvrp | stp | udld | vstp | vtp);
```

Syntax (EX2300 Multigigabit Model Switches)

```
protocol (cdp | gvrp | ieee8023ah | lacp | lldp | mvrp | stp | vstp | vtp);
```

Syntax (EX9200 Switches)

```
protocol (cdp | elmi | gvrp | ieee8021x | ieee8023ah | lacp | lldp | mmrp | mvrp | pvstp | stp | udld | vtp);
```

Hierarchy Level

[edit logical-systems *name* protocols layer2-control **mac-rewrite interface** *interface-name*],
[edit protocols **layer2-control mac-rewrite interface** *interface-name*]

Release Information

Statement introduced in Junos OS Release 9.1.
Statement introduced in Junos OS Releases 12.3X52-D10 and 13.2R1 for ACX Series Routers.
Statement introduced in Junos OS Release 13.2X50-D10 for EX Series switches.

Support for PVST/PVST+ introduced in Junos OS Release 13.3 for MX Series routers and EX9200 switches.
Statement introduced in Junos OS Release 14.1X53-D10 and 17.3R1 for EX4300 switches
Statement introduced in Junos OS Release 15.1X53-D55 and 18.2R1 for EX2300 and EX3400 switches.
Support for E-LMI, IEEE 802.1X, MMRP, and UDLD introduced in Junos OS Release 17.3R1 for EX4300 switches.
Support for E-LMI, GVRP, IEEE 802.1x, IEEE 802.3AH, LACP, LLDP, MMRP, MVRP, and UDLD introduced in Junos OS Release 17.3R1 for EX9200 switches.
Statement introduced in Junos OS Release 17.4R1 for EX4600 switches.
Support for E-LMI, IEEE 802.1X, MMRP, and UDLD introduced in Junos OS Release 18.2R1 for EX2300 and EX3400 switches.
Statement introduced in Junos OS Release 19.1R1 for QFX Series switches.
Statement introduced in Junos OS Release 19.2R1 for EX4300 multigigabit switches.

Description

Configure the protocol to be tunneled on an interface using Layer 2 protocol tunneling (L2PT). To enable tunneling multiple protocols, include multiple **protocol** statements.

You can tunnel different protocols listed in the Options section on different types of devices. The Syntax and Release Information sections list the available options for the protocols that different devices can tunnel as of a particular Junos OS release (for devices that support L2PT).

When a service provider edge (PE) port configured for L2PT receives a control packet for a supported protocol, the device rewrites the multicast destination MAC address with the predefined multicast tunneling MAC address 01:00:0c:cd:cd:d0. The packet travels across the provider network transparently to the other end of the tunnel, and the destination device restores the original multicast destination MAC address to deliver the packet at its destination.

Options

cdp—Tunnel the Cisco Discovery Protocol (CDP).

elmi—Tunnel Ethernet Local Management Interface (E-LMI) packets.

gvrp—Tunnel Generic Attribute Registration Protocol (GARP) VLAN Registration Protocol (GVRP) packets.

ieee8021x—Tunnel IEEE 802.1X authentication packets.

ieee8023ah—Tunnel IEEE 802.3AH Operation, Administration, and Maintenance (OAM) link fault management (LFM) packets.

lacp—Tunnel Link Aggregation Control Protocol (LACP) packets.

lldp—Tunnel Link Layer Discovery Protocol (LLDP) packets.

mmrp—Tunnel Multiple MAC Registration Protocol (MMRP) packets.

mvrp—Tunnel Multiple VLAN Registration Protocol (MVRP) packets.

pvstp—Tunnel VLAN Spanning Tree Protocol (VSTP), Per-VLAN Spanning Tree (PVST), and Per-VLAN Spanning Tree Plus (PVST+) Protocol packets.

stp—Tunnel packets for all versions of Spanning-Tree Protocols.

udld—Tunnel Unidirectional Link Detection (UDLD) packets.

vstp—Tunnel VLAN Spanning Tree Protocol (VSTP) packets.

vtp—Tunnel VLAN Trunking Protocol (VTP) packets.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Understanding Layer 2 Protocol Tunneling | 684

Configuring Layer 2 Protocol Tunneling | 694

protocols (Fabric)

Syntax

```
protocols {
    fabric-control {
        graceful-restart {
            restart-timeseconds;
            stale-routes-time seconds;
        }
    }
}
```

Hierarchy Level

```
[edit fabric]
```

Release Information

Statement introduced in Junos OS Release 13.2X52-D10 for the QFX Series.

Description

Specify attributes for the fabric control protocol.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- *Understanding Routing Engines in the QFabric System*

proxy-arp

Syntax

```
proxy-arp (restricted | unrestricted);
```

Hierarchy Level

[edit interfaces *interface-name* unit *logical-unit-number*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.6 for EX Series switches.
restricted added in Junos OS Release 10.0 for EX Series switches.
Statement introduced in Junos OS Release 12.2 for the QFX Series.

Description

For Ethernet interfaces only, configure the router or switch to respond to any ARP request, as long as the router or switch has an active route to the ARP request's target address.

> **NOTE:** You must configure the IP address and the **inet** family for the interface when you enable proxy ARP.

Default

Proxy ARP is not enabled. The router or switch responds to an ARP request only if the destination IP address is its own.

Options

- **none**—The router or switch responds to any ARP request for a local or remote address if the router or switch has a route to the target IP address.
- **restricted**—(Optional) The router or switch responds to ARP requests in which the physical networks of the source and target are different and does not respond if the source and target IP addresses are in the same subnet. The router or switch must also have a route to the target IP address.
- **unrestricted**—(Optional) The router or switch responds to any ARP request for a local or remote address if the router or switch has a route to the target IP address.

Default: unrestricted

Required Privilege Level

interface—To view this statement in the configuration.

interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- Configuring Restricted and Unrestricted Proxy ARP | **346**
- Configuring Proxy ARP on Switches | **958**
- *Example: Configuring Proxy ARP on an EX Series Switch*
- Configuring Gratuitous ARP | **349**

push

Syntax

```
push;
```

Hierarchy Level

[edit interfaces *interface-name* unit *logical-unit-number* **input-vlan-map**],
[edit interfaces *interface-name* unit *logical-unit-number* **output-vlan-map**],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* **input-vlan-map**],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* output-vlan-map]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.
Statement introduced in Junos OS Release 13.2X51-D20 for the QFX Series.

Description

NOTE: On EX4300 switches, **push** is not supported at the [edit interfaces *interface-name* unit *logical-unit-number* output-vlan-map] hierarchy level.

Specify the VLAN rewrite operation to add a new VLAN tag to the top of the VLAN stack. An outer VLAN tag is pushed in front of the existing VLAN tag.

You can use this statement on Gigabit Ethernet IQ and 10-Gigabit Ethernet IQ2 and IQ2-E interfaces; 10-Gigabit Ethernet LAN/WAN PIC; aggregated Ethernet interfaces using Gigabit Ethernet IQ interfaces; 100-Gigabit Ethernet Type 5 PIC with CFP; and Gigabit Ethernet, 10-Gigabit Ethernet, 40-Gigabit Ethernet, and aggregated Ethernet interfaces.

If you include the **push** statement in the configuration, you must also include the **pop** statement at the **[edit interfaces *interface-name* unit *logical-unit-number* output-vlan-map]** hierarchy level.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Stacking a VLAN Tag | 393

Configuring Q-in-Q Tunneling on EX Series Switches with ELS Support | 900

push-push

Syntax

```
push-push;
```

Hierarchy Level

[edit interfaces *interface-name* unit *logical-unit-number* **input-vlan-map**],
[edit interfaces *interface-name* unit *logical-unit-number* **output-vlan-map**],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* **input-vlan-map**],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* **output-vlan-map**]

Release Information

Statement introduced in Junos OS Release 8.1.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.

Description

Specify the VLAN rewrite operation to push two VLAN tags in front of the frame.

You can use this statement on Gigabit Ethernet IQ, IQ2 and IQ2-E interfaces, 10-Gigabit Ethernet LAN/WAN PIC, on aggregated Ethernet interfaces using Gigabit Ethernet IQ2 and IQ2-E or 10-Gigabit Ethernet PICs on MX Series routers, and 100-Gigabit Ethernet Type 5 PIC with CFP.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Stacking Two VLAN Tags | 394

pvlan

Syntax

```
pvlan;
```

Hierarchy Level

```
[edit vlans vlan-name]
```

Release Information

Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description

Specify that the VLAN is private and access ports in the VLAN do not forward packets to each other. You use this statement with primary VLANs and isolated secondary VLANs.

Options

none

Required Privilege Level

system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Creating a Private VLAN on a Single QFX Switch | 475

Creating a Private VLAN Spanning Multiple QFX Series Switches | 479

pvlan-trunk

Syntax

```
pvlan-trunk;
```

Hierarchy Level

```
[edit vlans vlan-name vlan-id number interface interface-name]
```

Release Information

Statement introduced in Junos OS Release 10.4 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description

Configure an interface to be the trunk port, connecting switches that are configured with a private VLAN (PVLAN) across these switches.

Required Privilege Level

routing—To view this statement in the configuration.
routing–control—To add this statement to the configuration.
system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- Creating a Private VLAN Spanning Multiple EX Series Switches (CLI Procedure) | 484
- Creating a Private VLAN on a Single QFX Switch | 475
- Creating a Private VLAN Spanning Multiple QFX Series Switches | 479

recovery-timeout

Syntax

```
recovery-timeout seconds;
```

Hierarchy Level

```
[edit interfaces interface-name unit 0 family ethernet-switching]
```

Release Information

Statement introduced in Junos OS Release 15.1X49-D70.

Description

Configure an interface to be temporarily disabled when MAC limiting is in effect with the action **shutdown**. This enables the affected interface to recover automatically from the error condition after the specified period of time:

- If you configure MAC limiting with the **shutdown** option and you enable **recovery-timeout**, the interface is temporarily disabled when the MAC address limit is reached. The interface will recover automatically after the number of seconds specified.

> **NOTE:** The **recovery-timeout** configuration does not apply to preexisting error conditions. It impacts only error conditions that are detected after the **recovery-timeout** statement is configured and committed. To clear a preexisting error condition and restore the interface to service, use the operational mode commands **clear ethernet-switching recovery-timeout** .

Default

The interface does not automatically recover from an error condition.

Options

seconds— Number of seconds that the interface remains in a disabled state due to a port error prior to automatic recovery.

Range: 10 through 3600

Required Privilege Level

system—To view this statement in the configuration.
system–control—To add this statement to the configuration.

RELATED DOCUMENTATION

- clear ethernet-switching recovery-timeout | 1443
- *Understanding MAC Limiting*
- *Example: Configuring MAC Limiting on a Security Device*
- *Configuring Autorecovery From the Disabled State on Secure Interfaces of a Security Device (CLI Procedure)*

redundancy-group (Interfaces)

Syntax

```
redundancy-group number ;
```

Hierarchy Level

```
[edit interfaces interface-name redundant-ether-options]
```

Release Information

Statement introduced in Junos OS Release 9.0.

Description

Specify the redundancy group that a redundant Ethernet interface belongs to.

Options

number —Number of the redundancy group that the redundant interface belongs to. Failover properties of the interface are inherited from the redundancy group.

Range: 1 through 255

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- Interfaces User Guide for Security Devices

redundant-trunk-group

Syntax

```
redundant-trunk-group {
    group name {
        interface interface-name <primary>;
        interface interface-name;
        preempt-cutover-timer seconds;
    }
}
```

Hierarchy Level

- For platforms with ELS:

  ```
  [edit switch-options]
  ```

- For platforms without ELS:

  ```
  [edit ethernet-switching-options]
  ```

Release Information

Statement introduced in Junos OS Release 9.0 for EX Series switches.
Hierarchy level **[edit switch-options]** introduced in Junos OS Release 13.2X50-D10 (ELS). (See "Using the Enhanced Layer 2 Software CLI" on page 50 for information about ELS.)
Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series.

Description

Configure a primary link and secondary link on trunk ports. If the primary link fails, the secondary link automatically takes over without waiting for normal spanning-tree protocol convergence.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

system—To view this statement in the configuration.
system–control—To add this statement to the configuration.

RELATED DOCUMENTATION

Example: Configuring Redundant Trunk Links for Faster Recovery on EX Series Switches | 946

Example: Configuring Redundant Trunk Links for Faster Recovery on Devices with ELS Support | 940

Understanding Redundant Trunk Links (Legacy RTG Configuration) | 937

reflective-relay

Syntax

```
reflective-relay;
```

Hierarchy Level

```
[edit interfaces interface-name unit logical-unit-number family ethernet-switching]
```

Release Information

Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D35 for the EX Series.

Description

Configure a switch interface to return packets back to a device on the same interface that was used to deliver the packets.

Default

Switch interfaces are not configured for reflective relay.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Example: Configuring Reflective Relay for Use with VEPA Technology on QFX Switches | 1044

Configuring Reflective Relay on Switches | 1043

registration

Syntax

```
registration (forbidden | normal | restricted);
```

Hierarchy Level

```
[edit protocols mvrp interface (all | interface-name)],
```

```
[edit routing-instances routing-instance-name protocols mvrp interface (all | interface-name)] (for virtual switch
    instance type),
```

```
[edit logical-systems logical-system-name protocols mvrp interface (all | interface-name)],
[edit logical-systems logical-system-name routing-instances routing-instance-name protocols mvrp interface (all |
    interface-name)] (for virtual switch instance type)
```

Release Information

Statement introduced in Junos OS Release 10.0 for EX Series switches.
Statement introduced in Junos OS Release 10.1 for MX Series routers.
Statement introduced in Junos OS Release 15.1X49-D70 for SRX Series devices.

Description

Specifies the Multiple VLAN Registration Protocol (MVRP) registration mode for the interface if MVRP is enabled.

Default

normal—The interface or interfaces accept MVRP messages and participate in MVRP.

Options

forbidden—The interface or interfaces do not register and do not participate in MVRP.

normal—The interface or interfaces accept MVRP messages and participate in MVRP.

restricted—The interface or interfaces ignore all MVRP JOIN messages received for VLANs that are not statically configured for MVRP on the interface.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring Multiple VLAN Registration Protocol (MVRP) on Switches | 797

Example: Configuring Automatic VLAN Administration Using MVRP on MX Series Routers

Understanding Multiple VLAN Registration Protocol (MVRP) for Dynamic VLAN Registration | 793

Configuring Multiple VLAN Registration Protocol (MVRP) to Manage Dynamic VLAN Registration

Verifying That MVRP Is Working Correctly | 851

join-timer (MVRP) | 1227

leaveall-timer (MVRP) | 1240

leave-timer (MVRP) | 1238

point-to-point | 1301

ring-protection-link-end

Syntax

```
ring-protection-link-end;
```

Hierarchy Level

[edit protocols **protection-group ethernet-ring** *ring-name* (**east-interface** | **west-interface**)]

Release Information

Statement introduced in Junos OS Release 9.4.
Statement introduced in Junos OS Release 12.1 for EX Series switches.

Description

Specify that the port is one side of a ring protection link (RPL) by setting the RPL end flag.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Ethernet Ring Protection Switching Overview

Example: Configuring Ethernet Ring Protection Switching on EX Series Switches | **855**

Example: Configuring Ethernet Ring Protection Switching on QFX Series and EX Series Switches Supporting ELS | **874**

Configuring Ethernet Ring Protection Switching on Switches (CLI Procedure)

ring-protection-link-owner

Syntax

```
ring-protection-link-owner;
```

Hierarchy Level

[edit protocols **protection-group ethernet-ring** *ring-name*]

Release Information

Statement introduced in Junos OS Release 9.4.
Statement introduced in Junos OS Release 12.1 for EX Series switches.
Statement introduced in Junos OS Release 14.153-D10 for QFX Series switches.

Description

Specify the ring protection link (RPL) owner flag in the Ethernet protection ring. Include this statement only once for each ring (only one node can function as the RPL owner).

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Ethernet Ring Protection Switching Overview

Example: Configuring Ethernet Ring Protection Switching on EX Series Switches | **855**

Example: Configuring Ethernet Ring Protection Switching on QFX Series and EX Series Switches Supporting ELS | **874**

routing-instances

Syntax

```
routing-instances routing-instance-name {
    instance-type virtual-router;
    interface interface-name;
}
```

Hierarchy Level

```
[edit]
```

Release Information

Statement introduced in Junos OS Release 9.2 for EX Series switches.

Description

Configure a virtual routing entity.

Options

routing-instance-name—Name for this routing instance.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Example: Using Virtual Routing Instances to Route Among VLANs on EX Series Switches | 711

Configuring Virtual Routing Instances on EX Series Switches | 710

security-zone

Syntax

```
security-zone zone-name {
    address-book {
        address address-name {
            ip-prefix {
                description text;
            }
            description text;
            dns-name domain-name {
                ipv4-only;
                ipv6-only;
            }
            range-address lower-limit to upper-limit;
            wildcard-address ipv4-address/wildcard-mask;
        }
        address-set address-set-name {
            address address-name;
            address-set address-set-name;
            description text;
        }
    }
    advance-policy-based-routing;
    application-tracking;
    description text;
    enable-reverse-reroute;
    host-inbound-traffic {
        protocols protocol-name {
            except;
        }
        system-services service-name {
            except;
        }
    }
    interfaces interface-name {
        host-inbound-traffic {
            protocols protocol-name {
                except;
            }
            system-services service-name {
                except;
            }
```

```
    }
  }
  screen screen-name;
  tcp-rst;
}
```

Hierarchy Level

```
[edit security zones]
```

Release Information

Statement introduced in Junos OS Release 8.5. Support for wildcard addresses added in Junos OS Release 11.1. The **description** option added in Junos OS Release 12.1.

Description

Define a security zone, which allows you to divide the network into different segments and apply different security options to each segment.

Options

zone-name —Name of the security zone.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- *Security Zones Overview*
- *Example: Configuring Application Firewall Rule Sets Within a Security Policy*

service-id

Syntax

```
service-id number;
```

Hierarchy Level

```
[edit switch-options]
[edit vlans vlan-name]
```

Release Information

Statement introduced in Junos OS Release 12.3R2 for EX Series switches and MX Series routers.
Statement introduced in Junos OS Release 13.2 for the QFX Series.

Description

Specify a service identifier for each multichassis aggregated Ethernet interface that belongs to a link aggregation group (LAG).

Options

number—A number that identifies a particular service.

Range: 1 through 65535

Required Privilege Level

system—To view this statement in the configuration.
system control—To add this statement to the configuration.

shutdown-threshold

Syntax

```
shutdown-threshold number;
```

Hierarchy Level

```
[edit vlans vlan-name dot1q-tunneling layer2-protocol-tunneling (all | protocol-name)]
```

Release Information

Statement introduced in Junos OS Release 10.0 for EX Series switches.

Description

Specify the maximum number of Layer 2 PDUs of the specified protocol that can be received per second on the interfaces in a specified VLAN before the interface is disabled. Once an interface is disabled, you must explicitly reenable it using the **clear ethernet-switching layer2-protocol-tunneling error** command. Otherwise, the interface remains disabled.

The shutdown threshold value must be greater than or equal to the drop threshold value. If the shutdown threshold value is less than the drop threshold value, the drop threshold value has no effect.

You can specify a shutdown threshold value without specifying a drop threshold value.

Default

No shutdown threshold is specified.

Options

number—Maximum number of Layer 2 PDUs of the specified protocol that can be received per second on the interfaces in a specified VLAN before the interface is disabled.

Range: 1 through 1000

Required Privilege Level

system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

drop-threshold | **1125**

Example: Configuring Layer 2 Protocol Tunneling on EX Series Switches Without ELS Support | **701**

Configuring Layer 2 Protocol Tunneling on EX Series Switches Without ELS Support | **699**

source-address (Security Policies)

Syntax

```
source-address {
    [address];
    any;
    any-ipv4;
    any-ipv6;
}
```

Hierarchy Level

[edit security policies from-zone *zone-name* to-zone *zone-name* policy *policy-name* match]

[edit security policies global policy *policy-name* match]

Release Information

Statement introduced in Junos OS Release 8.5. Support for IPv6 addresses added in Junos OS Release 10.2. Support for IPv6 addresses in active/active chassis cluster configurations (in addition to the existing support of active/passive chassis cluster configurations) added in Junos OS Release 10.4. Support for wildcard addresses added in Junos OS Release 11.1.

Description

Define the matching criteria. You can specify one or more IP addresses, address sets, or wildcard addresses. You can specify wildcards **any**, **any-ipv4**, or **any-ipv6**.

Options

address—IP addresses, address sets, or wildcard addresses (represented as A.B.C.D/wildcard-mask). You can configure multiple addresses or address prefixes separated by spaces and enclosed in square brackets.

Required Privilege Level

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- *Security Policies Overview*
- *Understanding Security Policy Rules*
- *Understanding Security Policy Elements*

stacked-vlan-tagging

Syntax

```
stacked-vlan-tagging;
```

Hierarchy Level

```
[edit interfaces interface-name]
```

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.2 for ACX Series Universal Metro Routers.

Description

For Gigabit Ethernet IQ interfaces, Gigabit Ethernet, 10-Gigabit Ethernet LAN/WAN PIC, and 100-Gigabit Ethernet Type 5 PIC with CFP, enable stacked VLAN tagging for all logical interfaces on the physical interface.

For pseudowire subscriber interfaces, enable stacked VLAN tagging for logical interfaces on the pseudowire service.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Stacking and Rewriting Gigabit Ethernet VLAN Tags Overview | 381

stale-routes-time (Fabric Control)

Syntax

```
stale-routes-time seconds;
```

Hierarchy Level

```
[edit fabric protocols fabric-control graceful-restart]
```

Release Information

Statement introduced in Junos OS Release 13.2X52-D10 for the QFX Series.

Description

Set the length of time that the fabric control Routing Engine waits to receive messages from devices before declaring them down. Configure a stale routes time of 1800 seconds if the number of VLAN members (vmembers) exceeds 32k.

Options

seconds—Amount of time that the fabric control Routing Engine waits to receive messages from other devices before declaring them down.

Default: 900 seconds

Range: 900 to 1800 seconds

Required Privilege Level

admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Understanding Bridging and VLANs on Switches | 168

Understanding Routing Engines in the QFabric System

static-mac

Syntax

```
static-mac mac-address;
```

```
static-mac mac-address {
    vlan-id number;
}
```

Hierarchy Level

```
[edit vlans vlan-name switch-options interface interface-name]
```

```
[edit bridge-domains bridge-domain-name bridge-options interface interface-name],
```

```
[edit logical-systems logical-system-name bridge-domains bridge-domain-name bridge-options interface interface-name],
[edit logical-systems logical-system-name routing-instances routing-instance-name bridge-domains bridge-domain-name
    bridge-options interface interface-name],
```

```
[edit routing-instances routing-instance-name bridge-domains bridge-domain-name bridge-options interface
    interface-name],
[edit routing-instances routing-instance-name protocols evpn interface interface-name]
```

Release Information

Statement introduced in Junos OS Release 8.4.
Statement modified in Junos OS Release 9.5.
Support for logical systems added in Junos OS Release 9.6.
[edit vlans *vlan-name* switch-options interface *interface name*] hierarchy level introduced in Junos OS Release 12.3R2 for EX Series switches.
Statement introduced in Junos OS Release 13.2 for EX Series switches.
Statement introduced in Junos OS Release 13.2 for the QFX Series.
Support for EVPNs added in Junos OS Release 13.2 for MX 3D Series routers. The **vlan-id** option is not available for EVPNs.
[edit vlans *vlan-name* switch-options interface *interface name*] hierarchy level introduced in Junos OS Release 13.2 for the QFX Series.

Description

Configure a static MAC address for a logical interface in a bridge domain or VLAN.

The **vlan-id** option can be specified for **static-macs** only if **vlan-id all** is configured for the bridging domain or VLAN.

Options

mac-address—MAC address

vlan-id *number*—(Optional) VLAN identifier to associate with static MAC address.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring EVPN Routing Instances

Understanding Layer 2 Learning and Forwarding for Bridge Domains

Layer 2 Learning and Forwarding for VLANs Overview | **78**

Adding a Static MAC Address Entry to the Ethernet Switching Table on a Switch with ELS Support | **112**

swap

Syntax

```
swap;
```

Hierarchy Level

[edit interfaces *interface-name* unit *logical-unit-number* **input-vlan-map**],
[edit interfaces *interface-name* unit *logical-unit-number* **output-vlan-map**],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* **input-vlan-map**],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* **output-vlan-map**]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.
Statement introduced in Junos OS Release 13.2X51-D20 for the QFX Series.

Description

Specify the VLAN rewrite operation to replace a VLAN tag. The outer VLAN tag of the frame is overwritten with the user-specified VLAN tag information.

On MX Series routers, you can enter this statement on Gigabit Ethernet IQ and 10-Gigabit Ethernet IQ2 and IQ2-E interfaces, 10-Gigabit Ethernet LAN/WAN PIC, aggregated Ethernet using Gigabit Ethernet IQ interfaces, and 100-Gigabit Ethernet Type 5 PIC with CFP. On EX Series switches, you can enter this statement on Gigabit Ethernet, 10-Gigabit Ethernet, 40-Gigabit Ethernet, and aggregated Ethernet interfaces.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Rewriting the VLAN Tag on Tagged Frames | **369**

Configuring Q-in-Q Tunneling on EX Series Switches with ELS Support | **900**

swap-by-poppush

Syntax

```
swap-by-poppush;
```

Hierarchy Level

```
[edit interfaces interface-name unit logical-unit-number]
```

Release Information
Statement introduced in Junos OS Release 11.2

Description
For Gigabit Ethernet IQ and 10-Gigabit Ethernet IQ2 and IQ2-E interfaces, 10-Gigabit Ethernet LAN/WAN PIC, and aggregated Ethernet using Gigabit Ethernet IQ interfaces, specify the VLAN rewrite operation to replace a VLAN tag. Pop original tag, then push an entirely new tag. The swap operation is performed as pop followed by push.

Required Privilege Level
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

swap-push

Syntax

```
swap-push;
```

Hierarchy Level

[edit interfaces *interface-name* unit *logical-unit-number* **input-vlan-map**],
[edit interfaces *interface-name* unit *logical-unit-number* **output-vlan-map**],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* **input-vlan-map**],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* **output-vlan-map**]

Release Information

Statement introduced in Junos OS Release 8.1.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.

Description

Specify the VLAN rewrite operation to replace the outer VLAN tag of the frame with a user-specified VLAN tag value. A user-specified outer VLAN tag is pushed in front. The outer tag becomes an inner tag in the final frame.

You can use this statement on Gigabit Ethernet IQ, IQ2 and IQ2-E interfaces, 10-Gigabit Ethernet LAN/WAN PIC, and for aggregated Ethernet interfaces using Gigabit Ethernet IQ2 and IQ2-E or 10-Gigabit Ethernet PICs on MX Series routers, and 100-Gigabit Ethernet Type 5 PIC with CFP.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Rewriting a VLAN Tag and Adding a New Tag | **367**

swap-swap

Syntax

```
swap-swap;
```

Hierarchy Level

[edit interfaces *interface-name* unit *logical-unit-number* **input-vlan-map**],
[edit interfaces *interface-name* unit *logical-unit-number* **output-vlan-map**],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* **input-vlan-map**],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* **output-vlan-map**]

Release Information

Statement introduced in Junos OS Release 8.1.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.

Description

Specify the VLAN rewrite operation to replace both the inner and the outer VLAN tags of the frame with a user-specified VLAN tag value.

You can use this statement on Gigabit Ethernet IQ, IQ2 and IQ2-E interfaces, 10-Gigabit Ethernet LAN/WAN PIC, for aggregated Ethernet interfaces using Gigabit Ethernet IQ2 and IQ2-E or 10-Gigabit Ethernet PICs on MX Series routers, and for 100-Gigabit Ethernet Type 5 PIC with CFP.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Rewriting the Inner and Outer VLAN Tags | 368

switch-options (VLANs)

List of Syntax

Syntax (EX Series, MX Series, QFX Series and NFX Series)

```
switch-options {
    interface interface-name {
        interface-mac-limit limit {
            packet-action drop;
        }
        mac-pinning
        no-mac-learning;
        static-mac static-mac-address {
            vlan-id number;
        }
    }
    interface-mac-limit limit {
        packet-action drop;
    }
    mac-statistics;
    mac-ip-table-size number;
    mac-table-size limit {
        packet-action drop;
    }
    no-mac-learning;
    service-id number;
    vtep-source-interface
}
```

Syntax (SRX Series)

```
switch-options {
    interface interface-name {
        encapsulation-type;
        ignore-encapsulation-mismatch;
        pseudowire-status-tlv;
        static-mac mac-address {
            vlan-id vlan-id;
        }
    }
```

```
    mac-table-aging-time seconds;
    mac-table-size {
        number;
        packet-action drop;
    }
}
```

EX Series, MX Series, QFX Series and NFX Series

```
[edit ],
[edit logical-systems logical-system-name routing-instances routing-instance-name vlans vlan-name],
[edit routing-instances routing-instance-name vlans vlan-name],
[edit vlans vlan-name]
```

SRX Series

```
[edit vlans vlans-name]
```

Release Information

Statement modified in Junos OS Release 9.5.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches and MX Series routers.
Statement introduced in Junos OS Release 13.2 for the QFX Series.
Statement (mac-pinning) introduced in Junos OS 16.2 for MX Series routers.
mac-ip-table-size statement introduced in Junos OS 17.4 Release for MX Series routers and EX9200 switches.

Description

Configure Layer 2 learning and forwarding properties for a VLAN or a virtual switch.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Ethernet Switching and Layer 2 Transparent Mode Overview | 41

system-services (Security Zones Interfaces)

Syntax

```
system-services service-name {
    except;
}
```

Hierarchy Level

```
[edit security zones security-zone zone-name interfaces interface-name host-inbound-traffic]
```

Release Information

Statement introduced in Junos OS Release 8.5.

Description

Specify the types of traffic that can reach the device on a particular interface.

Options

- ***service-name*** —Service for which traffic is allowed. The following services are supported:
  - **all**—Enable all possible system services available on the Routing Engine (RE).
  - **any-service**—Enable services on entire port range.
  - **bootp**—Enable traffic destined to BOOTP and DHCP relay agents.
  - **dhcp**—Enable incoming DHCP requests.
  - **dhcpv6**—Enable incoming DHCP requests for IPv6.
  - **dns**—Enable incoming DNS services.
  - **finger**—Enable incoming finger traffic.
  - **ftp**—Enable incoming FTP traffic.
  - **http**—Enable incoming J-Web or clear-text Web authentication traffic.
  - **https**—Enable incoming J-Web or Web authentication traffic over Secure Sockets Layer (SSL).
  - **ident-reset**—Enable the access that has been blocked by an unacknowledged identification request.
  - **ike**—Enable Internet Key Exchange traffic.
  - **netconf SSH**—Enable incoming NetScreen Security Manager (NSM) traffic over SSH.
  - **ntp**—Enable incoming Network Time Protocol (NTP) traffic.
  - **ping**—Allow the device to respond to ICMP echo requests.

- **r2cp**—Enable incoming Radio Router Control Protocol traffic.
- **reverse-ssh**—Reverse SSH traffic.
- **reverse-telnet**—Reverse Telnet traffic.
- **rlogin**—Enable incoming **rlogin** (remote login) traffic.
- **rpm**—Enable incoming real-time performance monitoring (RPM) traffic.
- **rsh**—Enable incoming Remote Shell (**rsh**) traffic.
- **snmp**—Enable incoming SNMP traffic (UDP port 161).
- **snmp-trap**—Enable incoming SNMP traps (UDP port 162).
- **ssh**—Enable incoming SSH traffic.
- **telnet**—Enable incoming Telnet traffic.
- **tftp**—Enable TFTP services.
- **traceroute**—Enable incoming traceroute traffic (UDP port 33434).
- **xnm-clear-text**—Enable incoming Junos XML protocol traffic for all specified interfaces.
- **xnm-ssl**— Enable incoming Junos XML protocol-over-SSL traffic for all specified interfaces.

- **except**—(Optional) except can only be used if all has been defined.

Required Privilege Level
security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Security Zones Overview

Supported System Services for Host Inbound Traffic

tag-protocol-id (TPIDs Expected to Be Sent or Received)

Syntax

```
tag-protocol-id [tpids];
```

Hierarchy Level

```
[edit interfaces interface-name gigether-options ethernet-switch-profile],
[edit interfaces interface-name aggregated-ether-options ethernet-switch-profile],
[edit interfaces interface-name aggregated-ether-options ethernet-switch-profile],
[edit interfaces interface-name ether-options ethernet-switch-profile]
```

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.2 for ACX Series Universal Metro Routers.
Statement introduced in Junos OS Release 13.2X50-D15 for EX Series switches.
Statement introduced in Junos OS Release 14.1X53-D15 for the QFX Series.

Description

For Gigabit Ethernet IQ and 10-Gigabit Ethernet IQ2 and IQ2-E interfaces, aggregated Ethernet with Gigabit Ethernet IQ interfaces, and Gigabit Ethernet PICs with SFPs (except the 10-port Gigabit Ethernet PIC, and the built-in Gigabit Ethernet port on the M7i router), define the TPIDs expected to be sent or received on a particular VLAN. For each Gigabit Ethernet port, you can configure up to eight TPIDs using the **tag-protocol-id** statement; but only the first four TPIDs are supported on IQ2 and IQ2-E interfaces.

For 10-Gigabit Ethernet LAN/WAN PIC interfaces on T Series routers only the default TPID value (**0x8100**) is supported.

For Gigabit Ethernet, 10-Gigabit Ethernet, 40-Gigabit Ethernet, and aggregated Ethernet interfaces on EX Series switches, define the TPIDs expected to be sent or received on a particular VLAN. The default TPID value is **0x8100**. Other supported values are **0x88a8**, **0x9100**, and **0x9200**.

Options

tpids—TPIDs to be accepted on the VLAN. Specify TPIDs in hexadecimal.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring Frames with Particular TPIDs to Be Processed as Tagged Frames | 385

Configuring Q-in-Q Tunneling on EX Series Switches with ELS Support | 900

tag-protocol-id (TPID to Rewrite)

Syntax

```
tag-protocol-id tpid;
```

Hierarchy Level

```
[edit interfaces interface-name unit logical-unit-number input-vlan-map],
[edit interfaces interface-name unit logical-unit-number output-vlan-map],
[edit logical-systems logical-system-name interfaces interface-name unit logical-unit-number input-vlan-map],
[edit logical-systems logical-system-name interfaces interface-name unit logical-unit-number output-vlan-map]
```

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.

Description

For Gigabit Ethernet IQ and 10-Gigabit Ethernet IQ2 and IQ2-E interfaces only, configure the outer TPID value. All TPIDs you include in input and output VLAN maps must be among those you specify at the **[edit interfaces *interface-name* gigether-options ethernet-switch-profile tag-protocol-id [*tpids*]]** hierarchy level.

For 10-Gigabit Ethernet LAN/WAN PIC interfaces on T Series routers the default TPID value (**0x8100**) is supported.

Default

If the **tag-protocol-id** statement is not configured, the TPID value is 0x8100.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring Inner and Outer TPIDs and VLAN IDs | 388

traceoptions

List of Syntax

Ethernet Switching Options

```
traceoptions {
    file filename <files number> <no-stamp> <replace> <size size> <world-readable | no-world-readable>;
    flag flag <disable>;
}
```

Ethernet Ring Protection

```
traceoptions {
    file filename <no-stamp> <world-readable | no-world-readable> <replace> <size size>;
    flag flag;
}
```

Edge Virtual Bridging

```
traceoptions {
    file filename <files number> <no-stamp> <replace> <size size> <world-readable | no-world-readable>;
    flag flag;
}
```

Hierarchy Level

```
[edit ethernet-switching-options]
```

```
[edit protocols protection-group]
```

```
[edit protocols edge-virtual-bridging]
```

Release Information

Statement introduced in Junos OS Release 11.1 for the QFX Series.
Statement introduced in Junos OS Release 12.1 for EX Series switches.
Statement introduced in Junos OS Release 14.153-D10 for QFX Series switches.

NOTE: The **traceoptions** statement is not supported on the QFX3000 QFabric system.

Description

Define global tracing operations for access security features on Ethernet switches.

Configure trace options for the protection group.

Define global tracing operations for edge virtual bridging (EVB) features on Ethernet switches.

Default

The Ethernet Switching Options **traceoptions** feature is disabled by default.

Edge Virtual Bridging tracing operations are disabled by default.

Ethernet Ring Protection trace options are not set by default. On some EX Series switches, logging of basic ERPS state transitions is set by default. You can configure trace options on those switches to obtain more details than are provided by the default log. See *Understanding Ethernet Ring Protection Switching Functionality* for additional information about default logging of the basic state transitions.

Options

For Ethernet Switching Options:

disable—(Optional) Disable the tracing operation. You can use this option to disable a single operation when you have defined a broad group of tracing operations, such as **all**.

file *filename* —Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory **/var/log**.

files *number* —(Optional) Maximum number of trace files. When a trace file named **trace-file** reaches its maximum size, it is renamed **trace-file.0**, then **trace-file.1**, and so on, until the maximum number of trace files is reached (***x*k** to specify KB, ***x*m** to specify MB, or ***x*g** to specify gigabytes), at which point the oldest trace file is overwritten. If you specify a maximum number of files, you also must specify a maximum file size with the **size** option.

Range: 2 through 1000

Default: 3 files

flag *flag* —Tracing operation to perform. To specify more than one tracing operation, include multiple flag statements. You can include the following flags:

- **access-security**—Trace access security events.
- **all**—All tracing operations.
- **analyzer**—Trace analyzer events.
- **config-internal**—Trace internal configuration operations.
- **filter**—Trace filter transaction events.
- **forwarding-database**—Trace forwarding database events.
- **general**—Trace general events.
- **interface**—Trace interface events.
- **krt**—Trace communications over routing sockets.
- **lib**—Trace library calls.
- **nexthop**—Trace next-hop events.
- **normal**—Trace normal events.
- **parse**—Trace reading of the configuration.
- **regex-parse**—Trace regular-expression parsing operations.
- **rtg**—Trace redundant trunk group events.
- **state**—Trace state transitions.
- **stp**—Trace spanning-tree events.
- **task**—Trace Ethernet-switching task processing.

- **timer**—Trace Ethernet-switching timer processing.
- **unknown-unicast-forwarding**—Trace unknown unicast forwarding events.
- **vlan**—Trace VLAN events.

no-stamp—(Optional) Do not timestamp the trace file.

Default: If you omit this option, timestamp information is placed at the beginning of each line of the tracing output.

no-world-readable—(Optional) Restrict file access to the user who created the file.

replace—(Optional) Replace an existing trace file if there is one rather than appending to it.

Default: If you do not include this option, tracing output is appended to an existing trace file.

size *size* —(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes. When a trace file named **trace-file** reaches its maximum size, it is renamed **trace-file.0**, then **trace-file.1**, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten. If you specify a maximum number of files, you also must specify a maximum file size with the **files** option.

Syntax: ***x*k** to specify KB, ***x*m** to specify MB, or ***x*g** to specify gigabytes

Range: 10 KB through 1 gigabyte

Default: 128 KB

world-readable—(Optional) Enable unrestricted file access.

For Ethernet Ring Protection:

file *filename*—Name of the file to receive the output of the tracing operation. All files are placed in the directory **/var/log**. You can include the following file options:

- **no-stamp**—(Optional) Do not timestamp trace file.
- **no-world-readable**—(Optional) Do not allow any user to read the log file.
- **replace**—(Optional) Replace the trace file rather than appending to it.
- **size**—(Optional) Maximum trace file size (10240..4294967295).
- **world-readable**—(Optional) Allow any user to read the log file.

flag *flag*—Tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements. The following flags are available for both EX Switches and MX Series routers:

- **all**—Trace all
- **config** —Trace configuration messages
- **debug**—Trace debug messages
- **events**—Trace events to the protocol state machine
- **normal**—Trace normal messages
- **pdu** —Trace RAPS PDU reception and transmission
- **periodic-packet-management**—Trace periodic packet management state and events
- **state-machine**—Trace RAPS state machine
- **timers**—Trace protocol timers

For Edge Virtual Bridging:

file *filename*—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory **/var/log**.

files *number*—(Optional) Maximum number of trace files. When a trace file named **trace-file** reaches its maximum size, it is renamed **trace-file.0**, then **trace-file.1**, and so on, until the maximum number of trace files is reached (***x***k to specify KB, ***x***m to specify MB, or ***x***g to specify gigabytes), at which point the oldest trace file is overwritten. If you specify a maximum number of files, you also must specify a maximum file size with the **size** option.

Range: 2 through 1000

Default: 3 files

no-world-readable—(Optional) Restrict file access to the user who created the file.

replace—(Optional) Replace an existing trace file if there is one rather than appending output to it.

Default: If you do not include this option, tracing output is appended to an existing trace file.

size *size*—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes. When a trace file named **trace-file** reaches its maximum size, it is renamed **trace-file.0**, then **trace-file.1**, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten. If you specify a maximum number of files, you also must specify a maximum file size with the **files** option.

Syntax: ***x*k** to specify KB, ***x*m** to specify MB, or ***x*g** to specify gigabytes

Range: 10 KB through 1 gigabyte

Default: 128 KB

world-readable—(Optional) Enable unrestricted file access.

flag *flag*—Tracing operation to perform. To specify more than one tracing operation, include multiple flag statements. You can include the following flags:

- **all**—Trace everything.
- **ecp**—Trace Edge Control Protocol (ECP) events.
- **evb-tlv**— Trace EVB type, length, and value (TLV) events.
- **parse**—Trace configuration parsing.
- **policy**—Trace policy events.
- **vdp**—Trace Virtual Station Interface (VSI) Discovery and Configuration Protocol (VDP) events.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.
system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Overview of Spanning-Tree Protocols

Configuring Ethernet Ring Protection Switching on Switches (CLI Procedure)

Example: Configuring Ethernet Ring Protection Switching on EX Series Switches | **855**

Example: Configuring Ethernet Ring Protection Switching on QFX Series and EX Series Switches Supporting ELS | **874**

Configuring Edge Virtual Bridging on an EX Series Switch | **1060**

traceoptions (LLDP)

Syntax

```
traceoptions {
    file filename <files number> <size maximum-file-size> <world-readable | no-world-readable>;
    flag flag <disable>;
}
```

Hierarchy Level

[edit protocols **lldp**],
[edit routing-instances *routing-instance-name* protocols **lldp**]

Release Information

Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 9.6 for MX Series.
Statement introduced in Junos OS Release 11.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for OCX Series switches.

Description

Define tracing operations for the Link Layer Discovery Protocol (LLDP). You can trace messages under LLDP for LLDP and physical topology SNMP MIBs.

NOTE: The traceoptions statement is not supported on the QFX3000 QFabric system.

Default

The default LLDP protocol-level trace options are inherited from the global **traceoptions** statement.

Options

disable—(Optional) Disable the tracing operation. One use of this option is to disable a single operation when you have defined a broad group of tracing operations, such as **all**.

file *filename*—Name of the file to receive the output of the tracing operation. Enclose the name in quotation marks. All files are placed in the directory **/var/log**. We recommend that you place spanning-tree protocol tracing output in the file **/var/log/stp-log**.

files *number*—(Optional) Maximum number of trace files. When a trace file named ***trace-file*** reaches its maximum size, it is renamed ***trace-file*.0**, then ***trace-file*.1**, and so on, until the maximum number of trace files is reached. Then, the oldest trace file is overwritten.

If you specify a maximum number of files, you must also specify a maximum file size with the **size** option.

Range: 2 through 1000 files

Default: 1 trace file only

flag—Specify a tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements.

Values: The following are the LLDP-specific tracing options:

- **all**—Trace all operations.
- **configuration**—Log configuration events.
- **interface**—Trace interface update events.
- **packet**—Trace packet events.
- **protocol**—Trace protocol information.
- **rtsock**—Trace socket events.
- **snmp**—Trace SNMP configuration operations.
- **vlan**—Trace VLAN update events.

The following are the global tracing options:

- **all**—All tracing operations.
- **config-internal**—Trace configuration internals.
- **general**—Trace general events.
- **normal**—All normal events. This is the default. If you do not specify this option, only unusual or abnormal operations are traced.
- **parse**—Trace configuration parsing.
- **policy**—Trace policy operations and actions.
- **regex-parse**—Trace regular-expression parsing.
- **route**—Trace routing table changes.
- **state**—Trace state transitions.
- **task**—Trace protocol task processing.
- **timer**—Trace protocol task timer processing.

no-world-readable—(Optional) Prevent any user from reading the log file. This is the default. If you do not include this option, tracing output is appended to an existing trace file.

size *maximum-file-size*—(Optional) Maximum size of each trace file, in kilobytes (KB) or megabytes (MB). When a trace file named ***trace-file*** reaches this size, it is renamed ***trace-file*.0**. When the ***trace-file*** again reaches its maximum size, ***trace-file*.0** is renamed ***trace-file*.1** and ***trace-file*** is renamed ***trace-file*.0**. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum file size, you must also specify a maximum number of trace files with the files option.

Syntax: ***x*k** to specify KB, ***x*m** to specify MB, or ***x*g** to specify GB

Range: 10 KB through the maximum file size supported on your system

Default: 1 MB

world-readable—(Optional) Allow any user to read the log file.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring LLDP-MED (CLI Procedure)

Understanding LLDP and LLDP-MED on EX Series Switches

Understanding LLDP

Tracing LLDP Operations | **681**

traceoptions (MVRP)

Syntax

```
traceoptions {
    file name <size size> <files number>  <(world-readable | no-world-readable)>;
    flag flag <flag-modifier> <disable>;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols **mvrp**],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols **mvrp**] (for virtual switch instance type),
[edit protocols **mvrp**],
[edit routing-instances *routing-instance-name* protocols **mvrp**] (for virtual switch instance type)

Release Information

Statement introduced in Junos OS Release 10.1 for MX Series routers.

Description

For Multiple VLAN Registration Protocol (MVRP), configure tracing options.

Default

Traceoptions is disabled.

Options

disable —(Optional) Disable the tracing operation. One use of this option is to disable a single operation when you have defined a broad group of tracing operations, such as **all**.

file *filename*—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. To include the **file** statement, you must specify a filename. Name of the file to receive the output of the tracing operation. Enclose the name in quotation marks. We recommend that you place MVRP tracing output in the file **/var/log/mvrp-log**.

files *number*—(Optional) Maximum number of trace files, in the range from 2 through 1000. The default is 1 trace file. When a trace file named **trace-file** reaches its maximum size, it is renamed **trace-file.0**, then **trace-file.1**, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum number of files, you also must specify a maximum file size with the **size** option.

flag *flag*—Specify which tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements. You can include the following flags:

- **all**—Enable all trace options flags.
- **error**—Trace all failure conditions.
- **events**—Trace process state change and cleanup events.
- **pdu**—Trace RAPS PDU reception and transmission.
- **socket**—Trace socket activity.
- **state-machine**—Trace information about the state machine.
- **timers**—Trace protocol timers.

no-world-readable—(Optional) Prevent any user from reading the log file.

size *size*—(Optional) Maximum size of each trace file, in kilobytes (KB) or megabytes (MB). When a trace file named ***trace-file*** reaches this size, it is renamed ***trace-file*.0.** When the ***trace-file*** again reaches its maximum size, ***trace-file*.0** is renamed ***trace-file*.1** and ***trace-file*** is renamed ***trace-file*.0** . This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten. The file size range is from 10240 through 4294967295. The default file size is 1 MB.

world-readable—(Optional) Allow any user to read the log file.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Example: Configuring Automatic VLAN Administration Using MVRP on MX Series Routers

Configuring Multiple VLAN Registration Protocol (MVRP) to Manage Dynamic VLAN Registration

Verifying That MVRP Is Working Correctly | **851**

Understanding Multiple VLAN Registration Protocol (MVRP) for Dynamic VLAN Registration | **793**

unconditional-src-learn

Syntax

```
unconditional-src-learn;
```

Hierarchy Level

[edit interfaces *interface-name* unit *logical-unit-number* family inet]

Release Information

Statement introduced in Junos OS Release 10.4R16.

Description

Enables the router to learn IP addresses from nonvalidated sources when proxy Address Resolution Protocol (ARP) is configured.

CAUTION: By default, the router learns IP addresses from validated sources only. When this statement is configured and proxy ARP is enabled on an unnumbered interface, the router responds to ARP requests from any IP address, which might lead to exploitable information disclosure. An attacker can poison the ARP cache and create a fake forwarding table entry for an IP address, effectively creating a denial of service for that subscriber or interface. Therefore, exercise caution when configuring this statement.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Understanding Proxy ARP | 954

Example: Configuring Proxy ARP on an EX Series Switch

unframed | no-unframed (Interfaces)

Syntax

```
(unframed | no-unframed);
```

Hierarchy Level

```
[edit interfaces interface-name  t3-options]
```

Release Information
Statement introduced in Junos OS Release 11.1.

Description
Enable or disable framing for the T3 interface on a 1-Port Clear Channel DS3/E3 GPIM on an SRX Series device. By default, unframed mode is enabled. Select no-unframed to enable framing. Select unframed to return to the default mode.

Required Privilege Level
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Example: Configuring a T3 Interface

unicast-in-lpm

Syntax

```
unicast-in-lpm;
```

Hierarchy Level

```
[edit chassis forwarding-options lpm-profile]
```

Release Information

Statement introduced in Junos OS Release 14.1x53-D30 for QFX Series switches.

Description

For the Unified Forwarding Table feature, specify to store all unicast IPv4 and IPv6 entries with prefixes with lengths equal to or less than 64 in the table for longest prefix match (LPM) entries, thereby freeing up space in the Layer 3 host table. Only unicast entries can be moved to the LPM table. Multicast entries must be stored in the Layer 3 host table.

You can also configure this statement in conjunction with the **prefix-65-127-disable** statement, which allocates no memory for IPv6 prefixes with lengths in the range /65 through /127. Together, these two statements allocate more space for unicast IPv4 and IPv6 entries with prefix lengths equal to or less than 64.

> **NOTE:** On QFX5110 switches running Junos OS Release 18.1R1 and higher, the **unicast-in-lpm** and **prefix-65-127-disable** statements cannot be configured at the same time.
>
> This statement is not supported on QFX5200 switches.

> **NOTE:** This statement is supported only on the **lpm-profile**.
>
> This statement is not supported on QFX5200 switches.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- *Understanding the Unified Forwarding Table*

unknown-unicast-forwarding

Syntax

```
unknown-unicast-forwarding {
   vlan (all | vlan-name){
      interface interface-name;
   }
}
```

Hierarchy Level

```
[edit ethernet-switching-options],
[edit switch-options]
```

Release Information

Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description

Configure the switch to forward all unknown unicast packets in a VLAN or on all VLANs to a particular interface.

> **NOTE:** Before you can configure unknown unicast forwarding within a VLAN, you must first configure that VLAN.

The remaining statements are explained separately. See CLI Explorer.

Default

Unknown unicast packets are flooded to all interfaces that belong to the same VLAN.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

show ethernet-switching table | 1520
show vlans | 1648

vlan

List of Syntax

Syntax (Ethernet and 802.1Q Tagging)

```
vlan {
   members [ (all | names | vlan-ids) ];
}
```

Syntax (Static MAC-based VLANs)

```
vlan vlan-name {
   mac mac-address {
        next-hop interface-name;
   }
}
```

Syntax (Unknown Unicast)

```
vlan (all | vlan-name) {
   interface interface-name;
}
```

Ethernet

[edit interfaces ge-*chassis*/*slot*/*port* unit *logical-unit-number* **ethernet-switching**],
[edit interfaces xe-*chassis*/*slot*/*port* unit *logical-unit-number* **ethernet-switching**]

802.1Q Tagging

[edit interfaces *interface-name* unit *logical-unit-number* family ethernet-switching]

Static MAC-based VLANs

[edit ethernet-switching-options static]

Unknown Unicast

[edit **ethernet-switching-options unknown-unicast-forwarding**]

Release Information

Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.
Statement introduced in Junos OS Release 11.1 for EX Series switches.

Description

For both Gigabit Ethernet and aggregated Ethernet interfaces and 802.1Q Tagging, assign an 802.1Q VLAN tag ID to a logical interface.

For Static MAC-based VLANs, specify the name of a VLAN to add to the Ethernet switching table.

For unknown unicast, specify a VLAN from which unknown unicast packets will be forwarded, or specify that the packets should be forwarded from *all* VLANS. Unknown unicast packets are forwarded from a VLAN to a specific trunk interface.

The remaining statements are explained separately. See CLI Explorer.

> **TIP:** To display a list of all configured VLANs on the system, including VLANs that are configured but not committed, type **?** after **vlan** or **vlans** in your configuration mode command line. Note that only one VLAN is displayed for a VLAN range.

Options

all—All VLANs.

vlan-name—Name of the VLAN to add to the Ethernet switching table.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.
system—To view this statement in the configuration.
system-control—To add this statement to the configuration.
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- Example: Setting Up Bridging with Multiple VLANs | **236**
- Adding a Static MAC Address Entry to the Ethernet Switching Table | **113**
- show ethernet-switching interfaces | **1485**
- show ethernet-switching table | **1520**
- show ethernet-switching interface | **1481**
- Example: Setting Up Bridging with Multiple VLANs for EX Series Switches | **265**
- *Configuring Routed VLAN Interfaces on Switches (CLI Procedure)*
- Configuring Integrated Routing and Bridging Interfaces on Switches (CLI Procedure) | **739**
- Understanding Bridging and VLANs on Switches | **168**
- show vlans | **1648**
- *Junos OS Network Interfaces Configuration Guide*

vlan-id

Syntax

```
vlan-id (all | none | number);
```

VLANs and Bridge Domain VLANs

For platforms without ELS:

```
[edit vlans vlan-name vlan-range]
```

For platforms without ELS and with ELS:

```
[edit vlans vlan-name]
```

For ELS platforms only:

```
[edit interfaces interface-name unit number]
[edit vlans vlan-name vlan-id-list]
```

```
[edit vlans vlan-name],
[edit logical-systems logical-system-name vlansvlan-name],
[edit logical-systems logical-system-name routing-instances routing-instance-name vlansvlan-name],
[edit routing-instances routing-instance-name vlans vlan--name]
```

802.1Q Tagging

```
[edit vlans vlan-name]
```

VLAN ID to Rewrite

```
[edit interfaces interface-name unit logical-unit-number input-vlan-map],
[edit interfaces interface-name unit logical-unit-number output-vlan-map],
[edit logical-systems logical-system-name interfaces interface-name unit logical-unit-number input-vlan-map],
[edit logical-systems logical-system-name interfaces interface-name unit logical-unit-number output-vlan-map]
```

VLAN Tagging and Layer 3 Subinterfaces

```
[edit interfaces interface-name unit logical-unit-number]
```

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 8.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 9.2 for EX Series switches VLAN tagging and Layer 3 subinterfaces.
Support for Layer 2 trunk ports added in Junos OS Release 9.2.
Support for SRX 5600, and SRX 5800 devices added in Junos OS Release 9.6.
Support for logical systems added in Junos OS Release 9.6.
Statement introduced in Junos OS Release 11.1 for the QFX Series.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.

Description

For VLANs, specify a VLAN identifier (VID) to include in the packets sent to and from the VLAN, or a VPLS routing instance.

> **NOTE:** When configuring a VLAN identifier for provider backbone bridge (PBB) routing instances, dual-tagged VIDs and the **none** option are not permitted.

For 802.1Q tagging, configure an 802.1Q tag to apply to all traffic that originates on the VLAN.

The number zero is reserved for priority tagging and the number 4095 is also reserved.

For VLAN ID to Rewrite Gigabit Ethernet IQ and 10-Gigabit Ethernet IQ2, 10-Gigabit Ethernet LAN/WAN PIC, and IQ2-E interfaces and aggregated Ethernet using Gigabit Ethernet IQ interfaces, specify the line VLAN identifiers to be rewritten at the input or output interface.

You cannot include the **vlan-id** statement with the **swap** statement, **swap-push** statement, **push-push** statement, or **push-swap** statement at the **[edit interfaces *interface-name* unit *logical-unit-number* output-vlan-map]** hierarchy level. If you include any of those statements in the output VLAN map, the VLAN ID in the outgoing frame is rewritten to the **vlan-id** statement that you include at the **[edit interfaces *interface-name* unit *logical-unit-number*]** hierarchy level.

Default

For 802.1Q Tagging on EX Series and SRX Series, If you use the default factory configuration, all traffic originating on the VLAN is untagged and has a VLAN identifier of **1**.

On a EX2300 switch, the maximum number of vlans supported is 2024. The vlans can be in the range of 1-4093.

On a EX3400 switch, the maximum number of vlans that can be created is 4093 including the default vlan with id 1. In a single **vlan-id-list** default **vlan1** is always inherited, so the valid configurable vlan range is 2-4093. You can use **vlan-id** up to and including 4094, but 4093 is the maximum number of vlans that can be configured.

For VLANs on a QFX3500 and QFX3500 switch, if you use the default factory configuration, all traffic originating on the VLAN is untagged and has a VLAN identifier of **1**. The number zero is reserved for priority tagging and the number 4093 is also reserved.

On a QFX5100 switch, if you use the default factory configuration, all traffic originating on the VLAN is untagged and has a VLAN identifier of **1**. The number zero is reserved for priority tagging and the number 4093 is also reserved.

> **NOTE:** You can only create up 4090 VLANs on a QFX5100 switch. If you create more than 4090 VLANs, the interfaces associated with the extra VLANs are not displayed in the **show vlans** command output. For example, if you create 4094 VLANs, the extra VLANs will not have interfaces associated with the VLANs. The order in which you configure the extra VLANs determines which interfaces are missing from the **show vlans** command output.

For VLAN tagging and Layer 3 subinterfaces, bind an 802.1Q VLAN tag ID to a logical interface.

> **NOTE:** The VLAN tag ID cannot be configured on logical interface unit **0**. The logical unit number must be **1** or higher.

Options

For VLANs:

number—A valid VLAN identifier. If you configure multiple VLANs with a valid VLAN identifier, you must specify a unique VLAN identifier for each. However, you can use the same VLAN identifier for VLANs that belong to different virtual switches. Use this option to send single tagged frames with the specified VLAN identifier over VPLS VT interfaces.

> **NOTE:** If you specify a VLAN identifier, you cannot also use the **all** option. They are mutually exclusive.

all—Specify that the VLAN spans all the VLAN identifiers configured on the member logical interfaces.

> **NOTE:** You cannot specify the **all** option if you include a routing interface in the VLAN.

none—Specify to enable shared VLAN learning or to send untagged frames over VPLS VT interfaces.

> **NOTE:** Multichassis link aggregation (MC-LAG) does not support the **none** option with the **vlan-id** statement with VLANs.

For 802.1Q Tagging:

number —VLAN tag identifier

Range:

- 1 through 4094 (all switches except EX8200 Virtual Chassis)
- 1 through 4092 (EX8200 Virtual Chassis only)

Default: 1

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.
system—To view this statement in the configuration.
system-control—To add this statement to the configuration.
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- Example: Connecting Access Switches with ELS Support to a Distribution Switch with ELS Support | **251**
- Example: Configuring a Private VLAN on a Single Switch with ELS Support | **486**
- Creating a Private VLAN on a Single Switch with ELS Support (CLI Procedure) | **472**
- Creating a Private VLAN Spanning Multiple EX Series Switches with ELS Support (CLI Procedure) | **481**
- Example: Configuring VLANs on Security Devices | **187**
- *Example: Configuring Interfaces and Routing Instances for a User Logical Systems*
- Rewriting the VLAN Tag on Tagged Frames | **369**
- Binding VLAN IDs to Logical Interfaces | **309**
- vlan-tagging | **1399**
- *Example: Configuring Layer 3 Subinterfaces for a Distribution Switch and an Access Switch*
- *Configuring Gigabit Ethernet Interfaces (CLI Procedure)*
- *Configuring Gigabit Ethernet Interfaces for EX Series Switches with ELS support*
- *Configuring Gigabit Ethernet Interfaces (J-Web Procedure)*
- *Configuring a Layer 3 Subinterface (CLI Procedure)*
- Configuring Q-in-Q Tunneling on EX Series Switches with ELS Support | **900**
- *Junos OS Ethernet Interfaces Configuration Guide*

vlan-id-list

Syntax

```
vlan-id-list [ vlan-id-numbers ];
```

Hierarchy Level

[edit **bridge-domains** *bridge-domain-name*],
[edit logical-systems *logical-system-name* **bridge-domains** *bridge-domain-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* **bridge-domains** *bridge-domain-name*],
[edit routing-instances *routing-instance-name* **bridge-domains** *bridge-domain-name*],
[edit interfaces *interface-name* unit 0],
[edit interfaces *interface-name* unit *logical-unit-number*],
[edit vlans *vlan-name*]

Release Information

Statement introduced in Junos OS Release 9.4.
Support for logical systems added in Junos OS Release 9.6.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.
Statement introduced in Junos OS Release 13.2 for the QFX Series.

Description

Specify a VLAN identifier list to use for a bridge domain or VLAN in trunk mode. VLAN identifier list can be used on C-VLAN interfaces in Q–in–Q tunneling for EX and QFX Series switches.

Specify the **trunk** option in the **interface-mode** statement to accept packets with a VLAN ID that matches the list of VLAN IDs specified in the **vlan-id-list** statement to forward the packet within the bridge domain or VLAN configured with the matching VLAN ID. Specify the **access** option to accept packets with no VLAN ID to forward the packet within the bridge domain or VLAN configured with the VLAN ID that matches the VLAN ID specified in the **vlan-id** statement.

This statement also enables you to bind a logical interface to a list of VLAN IDs, thereby configuring the logical interface to receive and forward a frame with a tag that matches the specified VLAN ID list.

WARNING: On some EX and QFX Series switches, if VLAN identifier list (**vlan-id-list**) is used for Q-in-Q tunnelling, you can apply no more than eight VLAN identifier lists to a physical interface.

Options

vlan-id-numbers—Valid VLAN identifiers. You can combine individual numbers with range lists by including a hyphen.

Range: 0 through 4095

> **NOTE:** On EX Series switches and the QFX Series, the range is 0 through 4094.

Required Privilege Level

routing—To view this statement in the configuration.

routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring a Bridge Domain

Configuring a VLAN | **51**

Configuring VLAN Identifiers for Bridge Domains and VPLS Routing Instances

Configuring VLAN Identifiers for VLANs and VPLS Routing Instances | **780**

Configuring Q-in-Q Tunneling and VLAN Q-in-Q Tunneling and VLAN Translation | **887**

vlan-id-range

Syntax

vlan-id-range *vlan-id–vlan-id*

Hierarchy Level

[edit interfaces *interface-name* unit *logical-unit-number*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]

Release Information

Statement introduced in Junos OS Release 8.4.

Description

Bind a range of VLAN IDs to a logical interface.

Options

number—The first number is the lowest VLAN ID in the range the second number is the highest VLAN ID in the range.

Range: 1 through 4094

> **NOTE:** On SRX240, SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX650 devices, the VLAN range from 1 to 4094 on inet interfaces and the VLAN range from 1 to 3967 on Ethernet switching interfaces. On Ethernet switching interfaces, the VLAN range from 3968 to 4094 falls under the reserved VLAN address range, and the user is not allowed to configure VLANs in this range.

NOTE: Configuring **vlan-id-range** with the entire vlan-id range is an unnecessary waste of system resources and is not best practice. It should be used only when a subset of VLAN IDs (not the entire range) needs to be associated with a logical interface. If you specify the entire range (1-4094), it has the same result as not specifying a range; however, it consumes PFE resources such as VLAN lookup tables entries, and so on.

The following examples illustrate this further:

```
[edit interfaces interface-name]
vlan-tagging;
unit number {
    vlan-id-range 1-4094;
}
```

```
[edit interfaces interface-name]
unit 0;
```

VLAN ID 0 is reserved for tagging the priority of frames.

Required Privilege Level
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Understanding VLANs

vlan-id-range

Syntax

vlan-id-range *vlan-id–vlan-id*

Hierarchy Level

[edit interfaces *interface-name* unit *logical-unit-number*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]

Release Information

Statement introduced in Junos OS Release 8.4.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.

Description

Bind a range of VLAN IDs to a logical interface.

Options

number—The first number is the lowest VLAN ID in the range the second number is the highest VLAN ID in the range.

Range: 1 through 4094

NOTE: Configuring **vlan-id-range** with the entire vlan-id range is an unnecessary waste of system resources and is not best practice. It should be used only when a subset of VLAN IDs (not the entire range) needs to be associated with a logical interface. If you specify the entire range (1-4094), it has the same result as not specifying a range; however, it consumes PFE resources such as VLAN lookup tables entries, and so on.

The following examples illustrate this further:

```
[edit interfaces interface-name]
vlan-tagging;
unit number {
    vlan-id-range 1-4094;
}
```

```
[edit interfaces interface-name]
unit 0;
```

VLAN ID 0 is reserved for tagging the priority of frames.

Required Privilege Level
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Binding a Range of VLAN IDs to a Logical Interface

vlan-id-start

Syntax

```
vlan-id-start S-VLAN-ID;
```

Hierarchy Level

```
[edit vlans vlan-name interface interface-name mapping-range C-VLAN-range (push | swap)]
```

Release Information

Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description

Configure an access interface to map a range of C-VLANs to a range of S-VLANs. Use this statement instead of using multiple **set vlans *VLAN-name* interface *interface-name* mapping (push | swap)** statements to configure Q-in-Q tunneling or VLAN translation on a per-VLAN basis. This statement sets the start of the S-VLAN range that the C-VLANs are mapped to. If you omit this option, mapping begins with the first ID in the range of S-VLAN IDs (which you configure using the **set vlans vlan-range**statement).

Options

None

Required Privilege Level

system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring Q-in-Q Tunneling on QFX Series Switches | 899

Example: Setting Up Q-in-Q Tunneling on QFX Series Switches | 920

vlan-prune

Syntax

```
vlan-prune;
```

Hierarchy Level

```
[edit vlans vlan-name]
```

Release Information
Statement introduced in Junos OS Release 12.3 for EX Series switches.

Description
Prune the Virtual Chassis port (VCP) paths in a Virtual Chassis to ensure received broadcast, multicast, and unknown unicast traffic in a VLAN uses the shortest possible path through the Virtual Chassis to the egress VLAN interface.

By default, all broadcast, multicast, and unknown unicast traffic in a VLAN on an EX Series Virtual Chassis is broadcast to all member switches in the Virtual Chassis. This behavior unnecessarily consumes bandwidth within the Virtual Chassis because unneeded traffic is sent to all Virtual Chassis member switches.

Enabling this option allows you to conserve bandwidth within the Virtual Chassis. Broadcast, multicast, and unknown unicast traffic still enters and exits the Virtual Chassis within the same VLAN, without the added bandwidth consumption that results from broadcasting this traffic to all member switches.

Default
Disabled

Required Privilege Level
system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring VLANs for EX Series Switches | 183

vlan-range

Syntax

vlan-range *vlan-id-low-vlan-id-high*;

Hierarchy Level

[edit **vlans** *vlan-name*]

Release Information

Statement introduced in Junos OS Release 9.2 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description

Configure multiple VLANs. Each VLAN is assigned a VLAN ID number from the range.

Default

None.

Options

vlan-id-low-vlan-id-high —Specify the first and last VLAN ID number for the group of VLANs.

Required Privilege Level

system—To view this statement in the configuration.
system–control—To add this statement to the configuration.
routing—To view this statement in the configuration.
routing–control—To add this statement to the configuration.

RELATED DOCUMENTATION

- Configuring VLANs on Switches | **182**
- Configuring VLANs for EX Series Switches | **183**
- *Configuring VLANs for EX Series Switches (J-Web Procedure)*
- *Configuring Routed VLAN Interfaces on Switches (CLI Procedure)*
- Understanding Bridging and VLANs on Switches | **168**
- Configuring IRB Interfaces on Switches | **735**

vlan-rewrite

Syntax

```
vlan-rewrite translate (200 500 | 201 501)
```

Hierarchy Level

```
[edit interfaces interface-name unit number family bridge interface-mode trunk]
[edit interfaces interface-name unit number family ethernet-switching interface-mode trunk]
```

Release Information

Statement introduced in Junos OS Release 9.4.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.

Description

Translates an incoming VLAN to a bridge-domain VLAN, corresponding counter translation at egress. Supports translation of VLAN 200 to VLAN 500 and VLAN 201 to VLAN 501. Other valid VLANs pass through without translation.

Options

translate 200 500—Translates incoming packets with VLAN 200 to 500.

translate 201 501—Translates incoming packets with VLAN 201 to 501.

translate 202 502—Translates incoming packets with VLAN 202 to 502.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Rewriting a VLAN Tag and Adding a New Tag | 367

vlan-tagging

Syntax

```
vlan-tagging;
```

Syntax (QFX Series, NFX Series, and EX4600)

```
vlan-tagging;
```

Syntax (SRX Series Interfaces)

```
vlan-tagging native-vlan-id vlan-id;
```

Hierarchy Level

```
[edit interfaces interface-name],
[edit logical-systems logical-system-name interfaces interface-name]
```

QFX Series, NFX Series, and EX4600 Interfaces

```
[edit interfaces (QFX Series) interface-name ]
[edit interfaces (QFX Series) interface-range interface-range-name ]
```

SRX Series Interfaces

```
[edit interfaces interface ]
```

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 9.5.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 12.2 for ACX Series Universal Metro Routers.
Statement introduced in Junos OS Release 13.2 for PTX Series Routers.
Statement introduced in Junos OS Release 14.1X53-D10 for the QFX Series.

Description

For Fast Ethernet and Gigabit Ethernet interfaces, aggregated Ethernet interfaces configured for VPLS, and pseudowire subscriber interfaces, enable the reception and transmission of 802.1Q VLAN-tagged frames on the interface.

> **NOTE:** For QFX Series configure VLAN identifier for untagged packets received on the physical interface of a trunk mode interface. Enable VLAN tagging. The platform receives and forwards single-tag frames with 802.1Q VLAN tags.
>
> On EX Series switches except for EX4300 and EX9200 switches, the **vlan-tagging** and **family ethernet-switching** statements cannot be configured on the same interface. Interfaces on EX2200, EX3200, EX3300, EX4200, and EX4500 switches are set to **family ethernet-switching** by the default factory configuration. EX6200 and EX8200 switch interfaces do not have a default **family** setting.

Default

VLAN tagging is disabled by default.

Options

native-vlan-id— (SRX Series)Configures a VLAN identifier for untagged packets. Enter a number from 0 through 4094.

> **NOTE:** The **native-vlan-id** can be configured only when either **flexible-vlan-tagging** mode or **interface-mode** trunk is configured.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

802.1Q VLANs Overview | **295**

Configuring a Layer 3 Subinterface (CLI Procedure)

Configuring Tagged Aggregated Ethernet Interfaces

Example: Configuring Layer 3 Subinterfaces for a Distribution Switch and an Access Switch

vlan-id

Configuring a Layer 3 Logical Interface | **720**

Configuring VLAN Tagging

vlan-tags

Syntax

vlan-tags outer *number* inner *number*;

Hierarchy Level

[edit **bridge-domains** *bridge-domain-name*],
[edit logical-systems *logical-system-name* **bridge-domains** *bridge-domain-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* **bridge-domains** *bridge-domain-name*],
[edit routing-instances *routing-instance-name* **bridge-domains** *bridge-domain-name*]
[edit vlans *vlan-name*]

Release Information

Statement introduced in Junos OS Release 8.4.
Support for logical systems added in Junos OS Release 9.6.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.
Statement introduced in Junos OS Release 13.2X51-D10 for QFX Series switches.

Description

Specify dual VLAN identifier tags for a bridge domain, VLAN, or VPLS routing instance.

Options

outer *number*—A valid VLAN identifier.

inner *number*—A valid VLAN identifier.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring a Bridge Domain

Configuring a VLAN | **51**

Configuring VLAN Identifiers for Bridge Domains and VPLS Routing Instances

Configuring VLAN Identifiers for VLANs and VPLS Routing Instances | **780**

Configuring a Layer 2 Virtual Switch .

Configuring a Layer 2 Virtual Switch on an EX Series Switch | **670**

vlan-tags

Syntax

```
vlan-tags outer [tpid].vlan-id [inner [tpid].vlan-id];
```

Hierarchy Level

```
[edit dynamic-profiles profile-name interfaces interface-name unit logical-unit-number]
```

Release Information

Statement introduced in Junos OS Release 9.5.
VLAN demux interface support introduced in Junos OS Release 10.2.

Description

For Gigabit Ethernet IQ and IQE interfaces only, binds TPIDs and 802.1Q VLAN tag IDs to a logical interface. You must include the **stacked-vlan-tagging** statement at the **[edit interfaces *interface-name*]** hierarchy level.

> **NOTE:** The **inner-range *vid1–vid2*** option is supported on IQE PICs only.

Options

inner [*tpid*].*vlan-id*—A TPID (optional) and a valid VLAN identifier in the format *tpid*.*vlan-id*. When used in the **dynamic-profiles** hierarchy, specify the **$junos-vlan-id** predefined variable to dynamically obtain the VLAN ID.

> **NOTE:** On the network-to-network (NNI) or egress interfaces of provider edge (PE) routers, you cannot configure the **inner-range *tpid. vid1—vid2*** option with the **vlan-tags** statement for ISP-facing interfaces.

Range: For VLAN ID, 1 through 4094. VLAN ID 0 is reserved for tagging the priority of frames.

outer [*tpid*].*vlan-id*—A TPID (optional) and a valid VLAN identifier in the format *tpid*.*vlan-id*. When used in the **dynamic-profiles** hierarchy, specify the **$junos-stacked-vlan-id** predefined variable.

Range: For VLAN ID, 1 through 511 for normal interfaces, and 512 through 4094 for VLAN CCC interfaces. VLAN ID 0 is reserved for tagging the priority of frames.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring Dual VLAN Tags | 388

vlan-tags (Dual-Tagged Logical Interface)

Syntax

```
vlan-tags inner-list [vlan-id vlan-id–vlan-id ] outer <tpid.>vlan-id;
```

Hierarchy Level

```
[edit interfaces interface-name unit logical-unit-number],
[edit logical-systems logical-system-name interfaces interface-name unit logical-unit-number]
```

Release Information

Statement introduced in Junos OS Release 9.5.

Description

(MX Series routers only) Binds a dual-tag logical interface to a list of VLAN IDs. Configures the logical interface to receive and forward any dual-tag frame whose inner VLAN ID tag matches the list of VLAN IDs you specify.

NOTE:

To create a circuit cross-connect (CCC) using VLAN-bundled dual-tag logical interfaces on Layer 2 VPN routing instances, you must include the **encapsulation-type** statement and specify the value **ethernet-vlan** at the one of the following hierarchy levels:

- **[edit routing-instances *routing-instance-name* protocols l2vpn]**
- **[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn]**

For more information about the **encapsulation-type** configuration statement and the Layer 2 encapsulation types **ethernet** and **ethernet-vlan**, see the *Junos OS VPNs Library for Routing Devices*.

Options

inner-list [*vlan-id vlan-id vlan-id-vlan-id*]—A list of valid VLAN ID numbers. Specify the VLAN IDs individually by using a space to separate each ID, as an inclusive list by separating the starting VLAN ID and ending VLAN ID with a hyphen, or as a combination of both.

Range: 1 through 4094. VLAN ID 0 is reserved for tagging the priority of frames.

outer <*tpid.*>*vlan-id*—An optional Tag Protocol ID (TPID) and a valid VLAN ID.

Range: For TPID, specify a hexadecimal value in the format **0x*nnnn***.

Range: For VLAN ID, 1 through 4094. VLAN ID 0 is reserved for tagging the priority of frames.

NOTE: Configuring **inner-list** with the entire vlan-id range is an unnecessary waste of system resources and is not best practice. It should be used only when a subset of VLAN IDs of inner tag (not the entire range) needs to be associated with a logical interface. If you specify the entire range (1 through 4094), it has the same result as not specifying a range; however, it consumes PFE resources such as VLAN lookup tables entries, and so on.

The following examples illustrate this further:

```
[edit interfaces interface-name]
vlan-tagging;
unit number {
    vlan-tags outer vid inner-list 1-4094;
}
```

```
[edit interfaces interface-name]
vlan-tagging;
unit number {
    vlan-id vid;
}
```

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- Binding VLAN IDs to Logical Interfaces | **309**
- *encapsulation (Logical Interface)*
- encapsulation | **1131**
- encapsulation-type (Layer 2 VPN routing instance), see the *Junos OS VPNs Library for Routing Devices*.
- flexible-vlan-tagging | **1171**
- *vlan-id-list (Ethernet VLAN Circuit)*
- vlan-tagging | **1399**

vlan-tags (Stacked VLAN Tags)

Syntax

```
vlan-tags inner tpid.vlan-id inner-list value inner-range vid1—vid2 outer tpid.vlan-id;
```

Hierarchy Level

```
[edit interfaces interface-name unit logical-unit-number],
[edit logical-systems logical-system-name interfaces interface-name unit logical-unit-number]
```

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.1X48 for PTX Series Packet Transport Routers.

Description

Bind TPIDs and 802.1Q VLAN tag IDs to a logical interface. TPID fields are used to identify the frame as an IEEE 802.1Q-tagged frame.

Options

inner *tpid.vlan-id*—A TPID and a valid VLAN identifier. TPID is a 16-bit field set to a value of 0x8100 in order to identify the frame as an IEEE 802.1Q-tagged frame.

Range: (most routers) For VLAN ID, 1 through 4094. VLAN ID 0 is reserved for tagging the priority of frames. For PTX Series, VLAN ID 0 is not supported.

inner-list *value*— List or a set of VLAN identifiers.

> **NOTE:** This is supported on MX Series routers with Trio-based FPCs.

inner-range *tpid. vid1–vid2*—Specify a TPID and a range of VLAN IDs where vid1 is the start of the range and vid2 is the end of the range.

> **NOTE:** On the network-to-network (NNI) or egress interfaces of provider edge (PE) routers, you cannot configure the **inner-range *tpid. vid1—vid2*** option with the **vlan-tags** statement for ISP-facing interfaces.

Range: For VLAN ID, 1 through 4094. VLAN ID 0 is reserved for tagging the priority of frames.

outer *tpid.vlan-id*—A TPID and a valid VLAN identifier.

Range: (most routers) For VLAN ID, 1 through 511 for normal interfaces, and 512 through 4094 for VLAN CCC interfaces. VLAN ID 0 is reserved for tagging the priority of frames. For PTX Series, VLAN ID 0 is not supported.

> **NOTE:** Configuring **inner-range** with the entire vlan-id range consumes system resources and is not a best practice. The **inner-range** must be used only when a subset of VLAN IDs of inner tag (not the entire range) needs to be associated with a logical interface. If you specify the entire range (1 through 4094), it has the same result as not specifying a range; however, it consumes Packet Forwarding Engine resources such as VLAN lookup table entries, and so on.
>
> The following examples illustrate this further:

```
[edit interfaces  interface-name]
stacked-vlan-tagging;
unit number {
    vlan-tags outer vid inner-range 1-4094;
}
```

```
[edit interfaces interface-name]
vlan-tagging;
unit number {
    vlan-id vid;
}
```

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- Configuring Dual VLAN Tags | 388
- Configuring Flexible VLAN Tagging on PTX Series Packet Transport Routers | 303
- stacked-vlan-tagging | 1350

vlan members (VLANs)

Syntax

```
vlan members [vlan-id];
```

Hierarchy Level

```
[edit vlans vlan-name]
```

Release Information
Statement modified in Junos OS Release 9.5.

Description
Specify multiple VLAN identifiers to create a VLAN for each VLAN identifier.

Options
vlan-id—A list of valid VLAN identifiers. A VLAN is created for each VLAN identifier in the list.

> **NOTE:** If you specify a VLAN identifier list, you cannot configure an IRB interface in the VLAN.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Example: Configuring VLANs on Security Devices | 187

vlans

List of Syntax

Syntax (QFX Series, QFabric, NFX Series and EX4600)

```
vlans {
    vlan-name {
        description text-description;
        dot1q-tunneling {
            customer-vlans (id | range);
        }
        filter input filter-name;
        filter output filter-name;
        interface interface-name {
            isolated;
            mapping (policy | tag push | native push);
            promiscuous;
        }
        isolation-vlan-id;
        l3-interface vlan.logical-interface-number;
        mac-limit number;
        no-local-switching;
        no-mac-learning;
        primary-vlan vlan-name;
        pvlan extend-secondary-vlan-id vlan-id;
        vlan-id number;
        vlan-range vlan-id-low-vlan-id-high;
    }
}
```

Syntax (QFX Series, NFX Series and EX4600)

```
vlans {
    vlan-name {
        description text-description;
        domain-type bridge;
        forwarding-options {
```

```
dhcp-security {
    arp-inspection;
    group group-name {
        interface interface-name {
            static-ip ip-address {
                mac mac-address;
            }
        }
        overrides {
            no-option82;
            trusted;
            untrusted;
        }
    }
    ip-source-guard;
    no-dhcp-snooping;
    option-82 {
        circuit-id {
            prefix {
                host-name;
                logical-system-name;
                routing-instance-name;
            }
            use-interface-description (device | logical);
            use-vlan-id;
        }
        remote-id {
            host-name hostname;
            use-interface-description (device | logical);
            use-string string;
        }
        vendor-id {
            use-string string;
        }
    }
}
```

```
    fip-security {
        examine-vn2vf;
        examine-vn2vn {
            beacon-period milliseconds;
        }
        fc-map fc-map-value;
        interface interface-name {
            (fcoe-trusted | no-fcoe-trusted;)
        }
    }
}
l3-interface irb.logical-unit-number;
```

```
multicast-snooping-options {
    flood-groups [group-names];
    forwarding-cache {
        threshold {
            reuse threshold;
            suppress threshold;
        }
    }
    graceful-restart {
        disable;
        restart-duration duration;
    }
    host-outbound-traffic {
        dot1p bits;
        forwarding-class forwarding-class;
    }
    multichassis-lag-replicate-state;
    nexthop-hold-time time;
    options {
        syslog {
            level level;
            mark interval;
            upto level;
        }
    }
    traceoptions {
        file filename {
            files number;
            no-world-readable;
            size file-size;
            world-readable;
        }
        flag flag {
            disable;
        }
    }
}
```

```
    switch-options {
      interface interface-name {
        interface-mac-limit limit {
          packet-action action;
        }
        static-mac mac-address;
      }
      interface-mac-limit limit {
        packet-action action;
      }
      mac-move-limit limit {
        packet-action action;
      }
      mac-table-size limit {
        packet-action drop;
      }
      no-mac-learning;
      }
    }
    vlan-id number;
    vlan-id-list [vlan-id | vlan-id–vlan-id];
    vlan-tags
      inner value;
      outer value;
    }
    vxlan {
      ingress-node-replication
      ovsdb-managed
      }
    }
  }
}
```

Syntax (SRX Series and EX Series)

```
vlans {
    vlan-name {
        description text-description;
        dot1q-tunneling {
            customer-vlans (id | range)
            layer2-protocol-tunneling all | protocol-name {
                drop-threshold number;
                shutdown-threshold number;
            }
        }
        filter input filter-name;
        filter output filter-name;
        interface interface-name {
            egress;
            ingress;
            mapping (native (push | swap) | policy | tag (push | swap));
            pvlan-trunk;
        }
        isolation-id id-number;
        l3-interface l3-interface-name.logical-interface-number;
        l3-interface-ingress-counting layer-3-interface-name;
        mac-limit limit action action;
        mac-table-aging-time seconds;
        no-local-switching;
        no-mac-learning;
        primary-vlan vlan-name;
        vlan-id number;
        vlan-prune;
        vlan-range vlan-id-low-vlan-id-high;
    }
}
```

Syntax (SRX Series)

```
vlans {
    vlan name {
        (vlan-id (1..3967) | vlan-id-list [ vlan-id-numbers]);
        description;
        forwarding-options {
            dhcp-security {
                arp-inspection;
                dhcpv6-options {
                    option-16 {
                        use-string use-string;
                    }
                    option-18 {
                        prefix {
                            host-name;
                            logical-system-name;
                            routing-instance-name;
                            vlan-id;
                            vlan-name;
                        }
                        use-interface-description (device | logical);
                        use-interface-index (device | logical);
                        use-interface-mac;
                        use-interface-name (device | logical);
                        use-string use-string;
                    }
                    option-37 {
                        prefix {
                            host-name;
                            logical-system-name;
                            routing-instance-name;
                            vlan-id;
                            vlan-name;
                        }
                        use-interface-description (device | logical);
                        use-interface-index (device | logical);
                        use-interface-mac;
                        use-interface-name (device | logical);
                        use-string use-string;
                    }
                }
                group group-name {
                    interface interface-name {
                        static-ip {
```

```
            ip-address {
            mac-address;
            }
        }
        static-ipv6 {
            ip-address {
            mac-address;
            }
        }
    }
    overrides {
        no-dhcpv6-options;
        no-option16;
        no-option18;
        no-option37;
        no-option82;
        trusted;
        untrusted;
    }
}
ip-source-guard;
ipv6-source-guard;
neighbor-discovery-inspection;
no-dhcp-snooping;
no-dhcpv6-snooping;
```

```
        option-82 {
            circuit-id {
                prefix {
                    host-name;
                    logical-system-name;
                    routing-instance-name;
                }
                use-interface-description (device | logical);
                use-vlan-id;
            }
            remote-id {
                host-name;
                mac;
                use-interface-description (device | logical);
                use-string use-string;
            }
            vendor-id {
                use-string use-string;
            }
        }
    }
    filter {
        input filter-name;
    }
    flood {
        input filter-name;
    }
}
interface interface-name;
l3-interface l3-interface-name;
mcae-mac-flush;
mcae-mac-synchronize;
service-id service-id;
```

```
    switch-options {
      interface name {
        action-priority action-priority;
        encapsulation-type (ethernet | ethernet-vlan);
      ignore-encapsulation-mismatch;
      interface-mac-limit {
        limit;
        packet-action (drop | drop-and-log | log | none | shutdown);
      }
      no-mac-learning;
      pseudowire-status-tlv;
      static-mac mac-address {
        vlan-id value;
      }
      }
      interface-mac-limit {
        limit;
        packet-action (drop | drop-and-log | log | none | shutdown);
      }
      mac-table-aging-time seconds;
      mac-table-size {
        limit;
        packet-action {
        drop;
        }
      }
      no-mac-learning;
      static-rvtep-mac {
        mac mac_addr {
        remote-vtep;
      }
      }
    }
  }
}
```

Syntax (vSRX)

```
vlans {
    vlan name {
        (vlan-id (all | none | number) | vlan-id-list [ vlan-id-numbers] | vlan-tags <inner number> outer number);
        description;
        forwarding-options {
            dhcp-security {
                arp-inspection;
                dhcpv6-options {
                    option-16 {
                        use-string use-string;
                    }
                    option-18 {
                        prefix {
                            host-name;
                            logical-system-name;
                            routing-instance-name;
                            vlan-id;
                            vlan-name;
                        }
                        use-interface-description (device | logical);
                        use-interface-index (device | logical);
                        use-interface-mac;
                        use-interface-name (device | logical);
                        use-string use-string;
                    }
                    option-37 {
                        prefix {
                            host-name;
                            logical-system-name;
                            routing-instance-name;
                            vlan-id;
                            vlan-name;
                        }
                        use-interface-description (device | logical);
                        use-interface-index (device | logical);
                        use-interface-mac;
                        use-interface-name (device | logical);
                        use-string use-string;
                    }
                }
                group group-name {
                    interface interface-name {
                        static-ip {
```

```
            ip-address;
          }
          static-ipv6 {
            ip-address;
          }
        }
        overrides {
          no-dhcpv6-options;
          no-option16;
          no-option18;
          no-option37;
          no-option82;
          trusted;
          untrusted;
        }
      }
      ip-source-guard;
      ipv6-source-guard;
      light-weight-dhcpv6-relay;
      neighbor-discovery-inspection;
      no-dhcp-snooping;
      no-dhcpv6-snooping;
      option-82 {
        circuit-id {
          prefix {
            host-name;
            logical-system-name;
            routing-instance-name;
          }
          use-interface-description (device | logical);
          use-vlan-id;
        }
        remote-id {
          host-name;
          mac;
          use-interface-description (device | logical);
          use-string use-string;
        }
        vendor-id {
          use-string use-string;
        }
      }
    }
```

```
    filter {
        input filter-name;
    }
    flood {
        input filter-name;
    }
}
interface interface-name;
l3-interface l3-interface-name;
mcae-mac-synchronize;
no-irb-layer-2-copy;
service-id service-id;
```

```
    switch-options {
        interface name {
            action-priority action-priority;
            encapsulation-type (ethernet | ethernet-vlan);
        ignore-encapsulation-mismatch;
        interface-mac-limit {
            disable;
            limit;
            packet-action (drop | drop-and-log | log | none | shutdown);
        }
        mac-pinning;
        no-mac-learning;
        pseudowire-status-tlv;
        static-mac mac-address {
            vlan-id value;
        }
        }
        interface-mac-limit {
            limit;
            packet-action (drop | drop-and-log | log | none | shutdown);
        }
        mac-statistics;
        mac-table-aging-time seconds;
        mac-table-size {
            limit;
            packet-action {
            drop;
            }
        }
        no-mac-learning;
        static-rvtep-mac {
            mac mac_addr {
            remote-vtep;
        }
        }
    }
  }
}
```

Hierarchy Level

[edit]

[edit routing-instances *routing-instance-name*]

Release Information

Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.
Statements for private VLANs and Q-in-Q tunneling introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 13.2 for the QFX Series.
Statement introduced in Junos OS Release 15.1X49-D10.

Description

Configure VLAN properties.

On EX Series switches and SRX Series devices (including vSRX), the following configuration guidelines apply:

- Only private VLAN (PVLAN) firewall filters can be used when the VLAN is enabled for Q-in-Q tunneling.
- An S-VLAN tag is added to the packet if the VLAN is Q-in-Q-tunneled and the packet is arriving from an access interface.
- You cannot use a firewall filter to assign an integrated routing and bridging (IRB) interface or a routed VLAN interface (RVI) to a VLAN.
- VLAN assignments performed using a firewall filter override all other VLAN assignments.

Default

If you use the default factory configuration, all switch interfaces become part of the VLAN **default**.

Options

vlan-name—Name of the VLAN. The name can include letters, numbers, hyphens (-), and periods (.) and can contain up to 255 characters long.

The remaining statements are explained separately. See CLI Explorer.

The remaining statements are described separately.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.
system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

- Configuring IRB Interfaces on Switches | **735**
- Understanding Bridging and VLANs on Switches | **168**
- Configuring VLANs on Switches with Enhanced Layer 2 Support | **179**
- Configuring VLANs for EX Series Switches | **183**
- *Configuring VLANs for EX Series Switches with ELS Support (CLI Procedure)*
- Configuring Q-in-Q Tunneling on EX Series Switches with ELS Support | **900**
- *Configuring Q-in-Q Tunneling on Security Devices*
- Configuring Integrated Routing and Bridging Interfaces on Switches (CLI Procedure) | **739**
- Understanding Bridging and VLANs on Switches | **168**

vrf-mtu-check

Syntax

```
vrf-mtu-check;
```

Hierarchy Level

```
[edit chassis]
```

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 11.1 for EX Series switches.

Description

On M Series routers (except the M120 and M320 router) and on EX Series 8200 switches, configure path maximum transmission unit (MTU) checks on the outgoing interface for unicast traffic routed on a virtual private network (VPN) routing and forwarding (VRF) instance.

Default

Disabled.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring Path MTU Checks for VPN Routing Instances

Configure Path MTU Discovery

vsi-discovery

Syntax

```
vsi-discovery {
    interface interface-name
    vsi-policy vsi-policy-name
}
```

Hierarchy Level

[edit protocols **edge-virtual-bridging**]

Description

Configure Virtual Station Interface (VSI) Discovery and Configuration Protocol (VDP). VDP is used to program policies for each individual station interface (VSI).

Default

VDP is disabled by default.

Options

interface-name—Name of the interface on which VDP is configured. The remaining statement is explained separately.

Required Privilege Level

system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Example: Configuring Edge Virtual Bridging for Use with VEPA Technology on an EX Series Switch | **1062**

Configuring Edge Virtual Bridging on an EX Series Switch | **1060**

vsi-policy

Syntax

vsi-policy *vsi-policy-name*from vsi-manager *vsi-manager-id* vsi-type *vsi-type* vsi-version *vsi-version* vsi-instance *instance-number*;

Hierarchy Level

[edit policy-options]

Description

Define and apply the named VSI policy to the edge virtual bridging (EVB) configuration. For use with edge virtual bridging, each virtual machine (VM) on the server is uniquely identified by following four parameters, which are contained in a VSI policy:

- vsi-manager-id
- vsi-type
- vsi-version
- vsi-instance-id

The vsi-policy command manually configures these four parameters on the EX switch for the successful association of VM-VSI. VDP protocol helps determine the parameters defined for the virtual machines on the server and configure them on the switch. Use policy options to define the VM-VSI parameters. Configure a firewall filter for each of the VM profiles and use it in this statement.

Required Privilege Level

system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Example: Configuring Edge Virtual Bridging for Use with VEPA Technology on an EX Series Switch | **1062**

Configuring Edge Virtual Bridging on an EX Series Switch | **1060**

west-interface

Syntax

```
west-interface {
    node-id mac-address;
    control-channel channel-name {
        vlan number;
        interface name interface-name
    }
    interface-none
    ring-protection-link-end;
    virtual-control-channel {
        west-interface name;
        east-interface name;
    }
}
```

Hierarchy Level

```
[edit protocols protection-group ethernet-ring ring-name]
```

Release Information

Statement introduced in Junos OS Release 9.5.
Statement introduced in Junos OS Release 12.1 for EX Series switches.
Statement introduced in Junos OS Release 14.153-D10 for QFX Series switches.

Description

Define one of the two interface ports for Ethernet ring protection, the other being defined by the **east-interface** statement at the same hierarchy level. The interface must use the control channel's logical interface name. The control channel is a dedicated VLAN channel for the ring port.

NOTE: Always configure this port second, after configuring the **east-interface** statement.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Ethernet Ring Protection Switching Overview

Ethernet Ring Protection Using Ring Instances for Load Balancing

east-interface | **1127**

ethernet-ring | **1148**

Example: Configuring Ethernet Ring Protection Switching on EX Series Switches | **855**

Example: Configuring Ethernet Ring Protection Switching on QFX Series and EX Series Switches Supporting ELS | **874**

Configuring Ethernet Ring Protection Switching on Switches (CLI Procedure)

41

CHAPTER

Operational Commands

clear dot1x

Syntax

```
clear dot1x
interface <interface-name>
mac-address <static-mac-address>
statistics <interface interface-name>
```

Release Information

Command introduced in Junos OS Release 15.1X49-D80.

Description

Reset the authentication state of an interface or delete 802.1X statistics from the device. When you reset an interface using the **interface** or **mac-address** options, reauthentication on the interface is also triggered. The device sends out a multicast message on the interface to restart the authentication of all connected supplicants. If a MAC address is reset, then the device sends out a unicast message to that specific MAC address to restart authentication.

Options

interface <[*interface-name*]>—Reset the authentication state of all the supplicants (also, clear all the authentication bypassed clients) connected to the specified interface (when the interface is an authenticator) or reset the authentication state for the interface itself (when the interface is a supplicant).

mac-address [*mac-addresses*]—Reset the authentication state of the specified MAC addresses.

statistics <interface *interface-name*>—Clear 802.1X statistics on all 802.1X-enabled interfaces. If the **interface** option is specified, clear 802.1X firewall statistics for that interface or interfaces.

Required Privilege Level

view

RELATED DOCUMENTATION

dot1x | **1122**

List of Sample Output

Sample Output

clear dot1x interface

```
user@host> clear dot1x interface ge-0/0/1
```

clear dot1x mac-address

```
user@host> clear dot1x mac-address 00:04:ae:cd:23:5f
```

clear dot1x statistics interface

```
user@host> clear dot1x statistics interface ge-0/0/1
```

clear edge-virtual-bridging

Syntax

```
clear edge-virtual-bridging
   <edge-control-protocol-statistics>
   <firewall <interface interface-name>
<vsi-profiles <interface interface-name>
```

Release Information

Statement introduced in Junos OS Release 12.1 for EX Series switches.

Description

Clear edge-virtual-bridging (EVB).

Options

none—Clear EVB.

edge-control-protocol-statistics—(Optional) Clear Edge Control Protocol (ECP) statistics.

firewall <interface *interface-name*>—(Optional) Clear EVB implicit filter counters on all interfaces or on a specific interface.

Required Privilege Level

system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Example: Configuring Edge Virtual Bridging for Use with VEPA Technology on an EX Series Switch | **1062**

Configuring Edge Virtual Bridging on an EX Series Switch | **1060**

clear error mac-rewrite

Syntax

```
clear error mac-rewrite
<interface interface-name>
```

Release Information

Command introduced in Junos OS Release 9.1.
Command introduced in Junos OS Release 14.1X53-D10 and 17.3R1 for EX4300 switches.
Command introduced in Junos OS Release 15.1X53-D55 and 18.2R1 for EX2300 and EX3400 switches.
Command introduced in Junos OS Release 17.4R1 for EX4600 switches.
Command introduced in Junos OS Release 19.1R1 for QFX Series switches.

Description

Clear a MAC rewrite error condition on an interface receiving tunneled Layer 2 protocol packets.

On interfaces with L2PT configured, customer-facing ports should not receive packets with the L2PT MAC address as the destination address unless there is a network tolopogy or configuration error. The device sets the status of such interfaces to “Disabled”. Use this command to clear the error and re-enable the interface.

Options

interface *interface-name*—(Optional) Clear the MAC rewrite error condition for the specified interface.

Required Privilege Level

clear

RELATED DOCUMENTATION

- Configuring Layer 2 Protocol Tunneling | **694**
- Clearing a MAC Rewrite Error on an Interface with Layer 2 Protocol Tunneling | **697**
- show mac-rewrite interface | **1572**

List of Sample Output

clear error mac-rewrite interface on page 1438

Output Fields

When you enter this command, the device returns feedback on the status of the request.

Sample Output

clear error mac-rewrite interface

```
user@host> clear error mac-rewrite interface ge-1/0/1
```

clear ethernet-switching layer2-protocol-tunneling error

Syntax

```
clear ethernet-switching layer2-protocol-tunneling error
<interface interface-name>
```

Release Information

Command introduced in Junos OS Release 10.0 for EX Series switches.
Command introduced in Junos OS Release 12.1 for the QFX Series.

Description

Clear Layer 2 protocol tunneling (L2PT) errors on one or more interfaces. If an interface has been disabled because the amount of Layer 2 protocol traffic exceeded the shutdown threshold or because the switch has detected an error in the network topology or configuration, use this command to reenable the interface.

Options

none—Clears L2PT errors on all interfaces.

interface *interface-name*—(Optional) Clear L2PT errors on the specified interface.

Required Privilege Level

view

RELATED DOCUMENTATION

Example: Configuring Layer 2 Protocol Tunneling on EX Series Switches Without ELS Support | 701

Configuring Layer 2 Protocol Tunneling on EX Series Switches Without ELS Support | 699

List of Sample Output

clear ethernet-switching layer2-protocol-tunneling error on page 1439
clear ethernet-switching layer2-protocol-tunneling error interface xe-0/0/1.0 on page 1440

Sample Output

clear ethernet-switching layer2-protocol-tunneling error

```
user@switch> clear ethernet-switching layer2-protocol-tunneling error
```

clear ethernet-switching layer2-protocol-tunneling error interface xe-0/0/1.0

```
user@switch> clear ethernet-switching layer2-protocol-tunneling error interface xe-0/0/1.0
```

clear ethernet-switching layer2-protocol-tunneling statistics

Syntax

```
clear ethernet-switching layer2-protocol-tunneling statistics
<interface interface-name>
<vlan vlan-name>
```

Release Information

Command introduced in Junos OS Release 10.0 for EX Series switches.
Command introduced in Junos OS Release 12.1 for the QFX Series.

Description

Clear Layer 2 protocol tunneling (L2PT) statistics on one or more interfaces or VLANs.

Options

none—Clear L2PT statistics on all interfaces and VLANs.

interface *interface-name*—(Optional) Clear L2PT statistics on the specified interface.

vlan *vlan-name*—(Optional) Clear L2PT statistics on the specified VLAN.

Required Privilege Level

view

RELATED DOCUMENTATION

- show ethernet-switching layer2-protocol-tunneling statistics | **1497**
- Example: Configuring Layer 2 Protocol Tunneling on EX Series Switches Without ELS Support | **701**
- Configuring Layer 2 Protocol Tunneling on EX Series Switches Without ELS Support | **699**

List of Sample Output

Sample Output

clear ethernet-switching layer2-protocol-tunneling statistics

```
user@switch> clear ethernet-switching layer2-protocol-tunneling statistics
```

clear ethernet-switching layer2-protocol-tunneling error interface ge-0/1/1.0

```
user@switch> clear ethernet-switching layer2-protocol-tunneling statistics interface xe-0/1/1.0
```

clear ethernet-switching layer2-protocol-tunneling error vlan v2

```
user@switch> clear ethernet-switching layer2-protocol-tunneling statistics vlan v2
```

clear ethernet-switching recovery-timeout

Syntax

```
clear ethernet-switching recovery-timeout
<interface interface-name>
```

Release Information

Command introduced in Junos OS Release 15.1X49-D70

Description

Clear all MAC limiting errors from all the Ethernet switching interfaces on the device or from the specified interface, and restore the interfaces or the specified interface to service.

Options

interface *interface-name*—(Optional) Clear all MAC limiting errors from the specified interface and restore the interface to service.

Required Privilege Level

clear

RELATED DOCUMENTATION

- *Understanding MAC Limiting*
- *Example: Configuring MAC Limiting on a Security Device*
- *Configuring Autorecovery From the Disabled State on Secure Interfaces of a Security Device (CLI Procedure)*

clear ethernet-switching table

Syntax

```
clear ethernet-switching table
<interface interface-name>
<mac mac-address>
<management-vlan>
<persistent-mac <interface | mac-address>>
<vlan vlan-name>
```

Syntax (QFX Series)

```
clear ethernet-switching table
<interface interface-name>
<mac mac-address>
<persistent-mac <interface | mac-address>>
<vlan vlan-name>
```

Release Information

Command introduced in Junos OS Release 9.3 for EX Series switches.
Command introduced in Junos OS Release 11.1 for the QFX Series.

Description

> **NOTE:** On a QFabric system, using this command on an FCoE-enabled VLAN when FCoE sessions are active can cause traffic flooding and FCoE traffic drop. The FCoE sessions are not terminated and the traffic reconverges after a short period of time.

Clear learned entries, which are media access control (MAC) addresses, in the Ethernet switching table (also called the forwarding database table).

Options

none—Clear learned entries in the Ethernet switching table, except for persistent MAC addresses.

interface *interface-name*—(Optional) Clear all learned MAC addresses for the specified interface from the Ethernet switching table.

mac *mac-address*—(Optional) Clear the specified learned MAC address from the Ethernet switching table.

management-vlan—(Optional) Clear all MAC addresses learned for the management VLAN from the Ethernet switching table. Note that you do not specify a VLAN name because only one management VLAN exists.

persistent-mac *<interface | mac-address>*—(Optional) Clear all MAC addresses, including persistent MAC addresses. Use the **interface** option to clear all MAC addresses on an interface, or use the **mac-address** option to clear all entries for a specific MAC address.

Use this command whenever you move a device in your network that has a persistent MAC address on the switch. If you move the device to another port on the switch and do not clear the persistent MAC address from the original port it was learned on, then the new port will not learn the MAC address and the device will not be able to connect. If the original port is down when you move the device, then the new port will learn the MAC address and the device can connect—however, unless you cleared the MAC address on the original port, when the port comes back up, the system reinstalls the persistent MAC address in the forwarding table for that port. If this occurs, the address is removed from the new port and the device loses connectivity.

vlan *vlan-name*—(Optional) Clear all MAC addresses learned for the specified VLAN from the Ethernet switching table.

Required Privilege Level
view

RELATED DOCUMENTATION

show ethernet-switching table | **1520**

List of Sample Output
clear ethernet-switching table on page 1445

Output Fields
This command produces no output.

Sample Output

clear ethernet-switching table

```
user@switch> clear ethernet-switching table
```

clear interfaces statistics swfabx

Syntax

```
clear interfaces statistics <swfab0 | swfab1>
```

Release Information

Command introduced in Junos OS Release 11.1.

Description

Clear interface statistics for the specified swfab interface.

Required Privilege Level

clear

RELATED DOCUMENTATION

show interfaces swfabx

List of Sample Output

clear interfaces statistics <swfab0 | swfab1> on page 1446

Output Fields

When you enter this command, interface statistics for swfab0 and swfab1 are cleared.

Sample Output

clear interfaces statistics <swfab0 | swfab1>

```
user@host> clear interfaces statistics <swfab0 | swfab1>
```

clear lldp neighbors

Syntax

```
clear lldp neighbor
<interface interface-name>
```

Release Information

Command introduced in Junos OS Release 9.6.

Description

Clear information regarding all Link Layer Discovery Protocol (LLDP) neighbors or LLDP neighbors of the specified interface.

For information about interface names, see *Interface Naming Overview*. For information about interface names for TX Matrix routers, see *TX Matrix Router Chassis and Interface Names*. For information about FPC numbering on TX Matrix routers, see *Routing Matrix with a TX Matrix Router FPC Numbering*.

For information about interface names in the Junos Fusion technology, see *Understanding Junos Fusion Ports* .

Options

interface *interface-name*—(Optional) Clear the LLDP neighbors on the specified interface.

Required Privilege Level

clear

RELATED DOCUMENTATION

clear lldp statistics | **1449**

List of Sample Output

clear lldp neighbors on page 1448
clear lldp neighbors interface ge-0/1/1.0 on page 1448

Output Fields

When you enter this command, you are provided no feedback on the status of your request. You can enter the **show lldp neighbors** command before and after clearing the LLDP neighbors to verify the clear operation.

Sample Output

clear lldp neighbors

```
user@switch> clear lldp neighbors
```

clear lldp neighbors interface ge-0/1/1.0

```
user@switch> clear lldp neighbors interface ge-0/1/1.0
```

clear lldp statistics

Syntax

```
clear lldpp neighbor
<interface interface-name>
```

Release Information
Command introduced in Junos OS Release 9.6.

Description
Clear all Link Layer Discovery Protocols (LLDP) statistics or LLDP statistics associated with the specified interface.

For information about interface names, see *Interface Naming Overview*. For information about interface names for TX Matrix routers, see *TX Matrix Router Chassis and Interface Names*. For information about FPC numbering on TX Matrix routers, see *Routing Matrix with a TX Matrix Router FPC Numbering*.

For information about interface names in the Junos Fusion technology, see *Understanding Junos Fusion Ports* .

Options
interface *interface-name*—(Optional) Clear LLDP statistics on the specified interface.

Required Privilege Level
clear

RELATED DOCUMENTATION

clear lldp neighbors | 1447

List of Sample Output
clear lldp statistics on page 1450
clear lldp statistics interface ge-0/1/1.0 on page 1450

Output Fields
When you enter this command, you are provided no feedback on the status of your request. You can enter the **show lldp statistics** command before and after clearing the LLDP statistics to verify the clear operation.

Sample Output

clear lldp statistics

```
user@switch> clear lldp statistics
```

clear lldp statistics interface ge-0/1/1.0

```
user@switch> clear lldp statistics interface ge-0/1/1.0
```

clear mvrp statistics

List of Syntax

Syntax (EX Series)

```
clear mvrp statistics <interface interface-name>
```

Syntax (SRX1500, SRX300, SRX550M, SRX345, SRX340 and SRX320)

```
clear mvrp statistics
<interface interface-name>
<routing-instance routing-instance-name>
```

Release Information

Command introduced in Junos OS Release 10.0 for EX Series switches.
Statement introduced in Junos OS Release 15.1X49-D70.

Description

Clear all Multiple VLAN Registration Protocol (MVRP) interface and, for SRX devices, routing instances statistics.

Options

none—Clear all MVRP statistics.

interface *interface-name*—Clear the MVRP statistics on the specified interface.

routing-instance name—Clear the MVRP statistics on the specified SRX Series device's named routing instance. If you do not specify a routing instance name, statistics are cleared for the default routing instance.

Required Privilege Level

clear

RELATED DOCUMENTATION

show mvrp statistics | 1589
Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches | 832
show mvrp | 1574

List of Sample Output

clear mvrp statistics on page 1452
clear mvrp statistics interface ge-0/0/1.0 on page 1452

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear mvrp statistics

```
user@switch> clear mvrp statistics
```

clear mvrp statistics interface ge-0/0/1.0

```
user@switch> clear mvrp statistics interface ge-0/0/1.0
```

show chassis forwarding-options

Syntax

```
show chassis forwarding-options
```

Release Information

Command introduced in Junos OS Release 13.2
Support added to QFX5200 switches in Junos OS Release 15.1X53-D30

Description

Display the configuration for the Unified Forwarding Table.

Options

There are no options for this command.

> **NOTE:** Starting in Junos OS Releases 17.3R2, for QFX5200 Virtual Chassis, information about memory banks are displayed only for the Master, not for the other members. Values remain the same across all members. All configuration changes for the Unified Forwarding Table are made through the Master.

Required Privilege Level

view

RELATED DOCUMENTATION

Configuring the Unified Forwarding Table on Switches | 94
Example: Configuring a Unified Forwarding Table Custom Profile | 90

List of Sample Output

Output Fields

Table 137 on page 1454 lists the output fields for the **show chassis forwarding-options** command. Output fields are listed in the approximate order in which they appear.

Table 137: show chassis forwarding-options Output Fields

Field Name	Field Description
profile name	Name of profile configured: • **custom-profile** (QFX5200 only) • **l2-profile-one** • **l2-profile-three** (default) • **l2-profile-two** • **l3-profile** • **lpm-profile**
MAC	Maximum amount of memory allocated for Layer 2 entries.
L3-host	Maximum amount of memory allocated for Layer 3 host entries.
LPM	Maximum amount of memory allocated for longest match prefix (LPM) entries.
num-65-127-prefix	Maximum amount of memory allocated in LPM table for IP prefixes with lengths in the range /65 through /127.
Total scale(K)	(QFX5200 only) Maximum amount of memory allocated for each address type. This amount includes the amount configured plus the amount allocated through the dedicated hash table.
Bank details for various types of entries	(QFX5200 only) Maximum amount of memory configured by address type for each of the four shared memory banks and the dedicated hash table.
Entry type	(QFX5200 only) Type of forwarding-table entry: **L2(mac)**; **L3 (unicast and multicast)**; **Exact Match**; and **Longest Prefix Match (lpm)**
Dedicated bank size(K)	(QFX5200 only) Maximum amount of memory allocated for each address type in the dedicated hash table.
Shared bank size(K)	(QFX5200 only) Default Maximum amount of memory allocated for each address type in the shared memory banks.

Sample Output

show chassis forwarding-options (l2-profile-three)

```
user@host> show chassis forwarding-options
```

```
UFT Configuration:
l2-profile-three. (MAC: 160K L3-host: 144K LPM: 16K) (default)
num-65-127-prefix = none

{master:0}
```

show chassis forwarding-options (custom-profile on QFX5200 Series switch)

`user@host>` **show chasiss forwarding-options**

```
UFT Configuration:
custom-profile
Configured custom scale:
Entry type                Total scale(K)
L2(mac)                        8
L3 (unicast & multicast)       72
Exact Match                    0
Longest Prefix Match (lpm)     80
num-65-127-prefix = 1K
-------------Bank details for various types of entries------------
Entry type                  Dedicated Bank Size(K)      Shared Bank Size(K)
L2 (mac)                    8                         32 * num shared banks
L3 (unicast & multicast     8                         32 * num shared banks
Exact match                 0                         16 * num shared banks
Longest Prefix match(lpm)   16                        32 * num shared banks
```

show chassis forwarding-options (QFX5200 Virtual Chassis)

`user@host>` **show chassis forwarding-options**

```
localre:
-
UFT Configuration:
l2-profile-three.(default)
num-65-127-prefix = 1K
-Bank details for various types of entries-
Entry type                       Dedicated Bank Size(K)     Shared Bank Size(K)
L2(mac)                         8                         32 * num shared banks
L3(unicast & multicast)          8                         32 * num shared banks
Exact Match                     0                         16 * num shared banks
Longest Prefix Match(lpm)        16                        32 * num shared banks

fpc1:
```

```
-
UFT Configurtion:
l2-profile-three.(default)
num-65-127-prefix = 1K
```

show dot1x authentication-bypassed-users

Syntax

```
show dot1x authentication-bypassed-users
```

Release Information

Command introduced in Junos OS Release 15.1X49-D80.

Description

Display the supplicants (users) that have bypassed 802.1X authentication.

Required Privilege Level

view

RELATED DOCUMENTATION

show dot1x authentication-failed-users | 1459

dot1x | 1122

List of Sample Output

show dot1x authentication-bypassed-users on page 1458

Output Fields

Table 138 on page 1457 lists the output fields for the show dot1x authentication-bypassed-users command. Output fields are listed in the approximate order in which they appear.

Table 138: show dot1x authentication-bypassed-users Output Fields

Field Name	Field Description	Level of Output
Interface	The MAC address configured to bypass 802.1X authentication.	**all**
MAC address	The MAC address configured statically on the interface.	**all**
VLAN	The VLAN that is configured to bypass 802.1X authentication.	**all**

Sample Output

show dot1x authentication-bypassed-users

user@host> **show dot1x authentication-bypassed-users**

```
MAC address           Interface      VLAN
00:50:56:85:66:0f     ge-0/0/0.0     vlan6
00:50:56:9e:56:42     ge-0/0/1.0     vlan6
```

show dot1x authentication-failed-users

Syntax

```
show dot1x authentication-failed-users
```

Release Information

Command introduced in Junos OS Release 15.1X49-D80.

Description

Display the supplicants (users) that have failed 802.1X authentication.

Required Privilege Level

view

RELATED DOCUMENTATION

show dot1x authentication-bypassed-users | 1457

dot1x | 1122

List of Sample Output

show dot1x authentication-failed-users on page 1460

Output Fields

Table 139 on page 1459 lists the output fields for the **show dot1x authentication-failed-users** command. Output fields are listed in the approximate order in which they appear.

Table 139: show dot1x authentication-failed-users Output Fields

Field Name	Field Description	Level of Output
Interface	The MAC address configured to bypass 802.1X authentication.	**all**
MAC address	The MAC address configured statically on the interface.	**all**
User	The user that is configured on the RADIUS server and that has failed 802.1X authentication.	**all**
Failure Count	The number of times that 802.1X authentication has failed on the interface.	**all**

Sample Output

show dot1x authentication-failed-users

```
user@host> show dot1x authentication-failed-users

Interface     MAC address          User            Failure Count
ge-0/0/0.0    00:50:56:85:66:0f    00505685660f    1
ge-0/0/1.0    00:50:56:9e:56:42    0050569e5642    1
```

show dot1x interface

Syntax

```
show dot1x interface interface-name
<brief | detail | extensive>
```

Release Information

Command introduced in Junos OS Release 15.1X49-D80.

extensive option introduced in Junos OS Release 19.4R1 to display the additional fields when compared to **brief** option. The additional fields are **authentication method** and **vlan-id**.

Description

Display the current operational state of all ports with the list of connected users.

This command displays the list of connected supplicants received from the RADIUS authentication server regardless of the session state—that is, for both authenticated supplicants and for supplicants that attempted authentication.

Options

none—Display information for all authenticator ports.

brief | detail | extensive—(Optional) Display the specified level of output.

interface *interface-name*—(Optional) Display information for the specified interface with a list of connected supplicants.

Required Privilege Level

view

RELATED DOCUMENTATION

show dot1x authentication-bypassed-users | 1457

dot1x | 1122

List of Sample Output

show dot1x interface brief on page 1466
show dot1x interface detail on page 1466
show dot1x interface extensive on page 1467

Output Fields

Table 140 on page 1462 lists the output fields for the **show dot1x interface** command. Output fields are listed in the approximate order in which they appear.

Table 140: show dot1x interface Output Fields

Field Name	Field Description	Level of Output
Interface	Name of a port.	**all**
MAC address	The MAC address of the connected supplicant on the port.	**all**
Role	The 802.1X authentication role of the interface. When 802.1X is enabled on an interface, the role is **Authenticator**. As **Authenticator**, the interface blocks LAN access until a supplicant is authenticated through 802.1X or MAC RADIUS authentication.	**brief**, **detail**
State	The state of the port: • **Authenticated**—The supplicant has been authenticated through the RADIUS server or has been permitted access through server fail fallback. • **Authenticating**—The supplicant is authenticating through the RADIUS server. • **Held**—An action has been triggered through server fail fallback during a RADIUS server timeout. A supplicant is denied access, permitted access through a specified VLAN, or maintains the authenticated state granted to it before the RADIUS server timeout occurred.	**brief**, **extensive**
User	The username of the connected supplicant.	**brief**, **extensive**
Administrative state	The administrative state of the port: • **auto**—Traffic is allowed through the port based on the authentication result (by default). • **force-authorize**—All traffic flows through the port irrespective of the authentication result. This state is not allowed on an interface whose VLAN membership has been set to **dynamic**. • **force-unauthorize**—All traffic drops on the port irrespective of the authentication result. This state is not allowed on an interface whose VLAN membership has been set to **dynamic**.	**detail**

Table 140: show dot1x interface Output Fields *(continued)*

Field Name	Field Description	Level of Output
Supplicant	The mode for the supplicant: • **single**—Only the first supplicant is authenticated. All other supplicants that connect later to the port are allowed full access without any further authentication. They effectively *piggyback* on the first supplicant's authentication. • **single-secure**—Only one supplicant is allowed to connect to the port. No other supplicant is allowed to connect until the first supplicant logs out. • **multiple**—Multiple supplicants are allowed to connect to the port. Each supplicant is authenticated individually.	**detail**
Quiet period	The number of seconds the port waits before reattempting authentication after a failed authentication exchange with the supplicant.	**detail**
Transmit period	The number of seconds the port waits before retransmitting the initial EAPOL PDUs to the supplicant.	**detail**
MAC Radius	MAC RADIUS authentication: • **enabled**—The device sends an EAPOL request to the connecting host to attempt 802.1X authentication and if the connecting host is unresponsive, the device tries to authenticate the host by using the MAC address. • **disabled**—The default. The device does not attempt to authenticate the MAC address of the connecting host.	**detail**
MAC Radius authentication protocol	MAC RADIUS authentication protocol: • **EAP-MD5**—The EAP-MD5 protocol is used for MAC RADIUS authentication. EAP-MD5 is an authentication method belonging to the Extensible Authentication Protocol (EAP) authentication framework. EAP-MD5 is the default authentication protocol. • **PAP**—The Password Authentication Protocol (PAP) authentication protocol is used for MAC RADIUS authentication.	**detail**
MAC Radius restrict	The authentication method is restricted to MAC RADIUS. 802.1X authentication is not enabled.	**detail**
Reauthentication	The reauthentication state: • **disable**—Periodic reauthentication of the client is disabled. • **interval**—Sets the periodic reauthentication time interval.	**detail**

Table 140: show dot1x interface Output Fields *(continued)*

Field Name	Field Description	Level of Output
Supplicant timeout	The number of seconds the port waits for a response when relaying a request from the authentication server to the supplicant before resending the request.	**detail**
Server timeout	The number of seconds the port waits for a reply when relaying a response from the supplicant to the authentication server before timing out.	**detail**
Maximum EAPOL requests	The maximum number of times an EAPOL request packet is retransmitted to the supplicant before the authentication session times out.	**detail**
Number of clients bypassed because of authentication	The number of non-802.1X clients granted access to the LAN by means of static MAC bypass. The following fields are displayed: • **Client**—MAC address of the client. • **vlan**—The name of the VLAN to which the client is connected.	**detail**
Guest VLAN member	The VLAN to which a supplicant is connected when the supplicant is authenticated using a guest VLAN. If a guest VLAN is not configured on the interface, this field displays <**not configured**>.	**detail**
Number of connected supplicants	The number of supplicants connected to a port.	**detail**
Supplicant	The username and MAC address of the connected supplicant.	**detail**

Table 140: show dot1x interface Output Fields *(continued)*

Field Name	Field Description	Level of Output
Authentication method	The authentication method used for a supplicant: • **CWA Authentication**—A supplicant is authenticated by the central Web authentication (CWA) server. • **Guest VLAN**—A supplicant is connected to the LAN through the guest VLAN. • **MAC RADIUS**—A nonresponsive host is authenticated based on its MAC address. The MAC address is configured as permitted on the RADIUS server. The RADIUS server lets the device know that the MAC address is a permitted address, and the device opens LAN access to the nonresponsive host on the interface to which it is connected. • **RADIUS**—A supplicant is configured on the RADIUS server, the RADIUS server communicates this to the device, and the device opens LAN access on the interface to which the supplicant is connected. • **Server-fail**—One of the following fallback actions is in effect because the RADIUS server is unreachable. Indicates whether EAPOL block is in effect, and the amount of time remaining for EAPOL block (in seconds). ◦ **deny**—The supplicant is denied access to the LAN, preventing traffic from flowing from the supplicant through the interface. This is the default server fail fallback action. ◦ **permit**—The supplicant is permitted access to the LAN as if the supplicant had been successfully authenticated by the RADIUS server. ◦ **use-cache**—In the event that the RADIUS server times out when the supplicant is attempting reauthentication, the supplicant is reauthenticated only if it was previously authenticated; otherwise, the supplicant is denied LAN access. ◦ **VLAN**—The supplicant is configured to be moved to a specified VLAN if the RADIUS server is unavailable to reauthenticate the supplicant. (The VLAN must already exist on the device.)	**detail**, **extensive**
Authenticated VLAN	The VLAN to which the supplicant is connected.	**detail**, **extensive**
Dynamic filter	User policy filter sent by the RADIUS server.	**detail**
Session Reauth interval	The configured reauthentication interval.	**detail**
Reauthentication due in	The number of seconds in which reauthentication occurs again for the connected supplicant.	**detail**

Table 140: show dot1x interface Output Fields *(continued)*

Field Name	Field Description	Level of Output
Session Accounting Interim Interval	The number of seconds between interim RADIUS accounting messages.	**detail**
Accounting Update due in	The number of seconds until the next interim RADIUS accounting update is due.	**detail**
CWA Redirect URL	The URL used to redirect the supplicant to a central Web server for authentication.	**detail**

Sample Output

show dot1x interface brief

```
user@host> show dot1x interface brief
```

```
802.1X Information:
Interface     Role           State           MAC address              User
ge-0/0/1      Authenticator  Connecting      2001:db8:56:85:66:0F     00505685660f
ge-0/0/2      Authenticator  Authenticated   2001:db8:56:9E:56:42     0050569e5642
```

show dot1x interface detail

```
user@host> show dot1x interface detail
```

```
ge-0/0/0.0
 Role: Authenticator
 Administrative state: Auto
 Supplicant mode: Multiple
 Number of retries: 3
 Quiet period: 60 seconds
 Transmit period: 30 seconds
 Mac Radius: Enabled
 Mac Radius Restrict: Disabled
 Reauthentication: Enabled
 Configured Reauthentication interval: 30 seconds
 Supplicant timeout: 30 seconds
```

```
  Server timeout: 30 seconds
  Maximum EAPOL requests: 2
  Guest VLAN member: not configured
  Number of connected supplicants: 1
    Supplicant: 00505685660f, 00:50:56:85:66:0F
      Operational state: Authenticated
      Backend Authentication state: Idle
      Authentication method: Server-Reject Vlan
      Authenticated VLAN: visitor-vlan
      Session Reauth interval: 30 seconds
      Reauthentication due in 20 seconds
ge-0/0/1.0
  Role: Authenticator
  Administrative state: Auto
  Supplicant mode: Multiple
  Number of retries: 3
  Quiet period: 60 seconds
  Transmit period: 30 seconds
  Mac Radius: Enabled
  Mac Radius Restrict: Disabled
  Reauthentication: Enabled
  Configured Reauthentication interval: 30 seconds
  Supplicant timeout: 30 seconds
  Server timeout: 30 seconds
  Maximum EAPOL requests: 2
  Guest VLAN member: not configured
  Number of connected supplicants: 1
    Supplicant: 0050569e5642, 00:50:56:9E:56:42
      Operational state: Authenticated
      Backend Authentication state: Idle
      Authentication method: Server-Reject Vlan
      Authenticated VLAN: visitor-vlan
      Session Reauth interval: 30 seconds
      Reauthentication due in 24 seconds
```

show dot1x interface extensive

user@host> **show dot1x interface extensive**

```
802.1X Information:
Interface  State         MAC address          Method             Vlan User
ge-0/0/6.0 Authenticated 2001:db8:94:00:00:01 Server-Reject Vlan 1400 Test12345
```

show dot1x static-mac-address

Syntax

```
show dot1x static-mac-address <interface interface-name>
```

Release Information

Command introduced in Junos OS Release 15.1X49-D80.

Description

Display all the static MAC addresses of interfaces that are configured to bypass 802.1X authentication.

Options

none—Display static MAC addresses for all interfaces.

interface *interface-name*—(Optional) Display static MAC addresses for a specific interface.

Required Privilege Level

view

RELATED DOCUMENTATION

show dot1x authentication-bypassed-users | 1457

dot1x | 1122

List of Sample Output

show dot1x static-mac-address on page 1469

show dot1x static-mac-address interface (Specific Interface) on page 1469

Output Fields

Table 141 on page 1468 lists the output fields for the **show dot1x static-mac-address** command. Output fields are listed in the approximate order in which they appear.

Table 141: show dot1x static-mac-address Output Fields

Field Name	Field Description	Level of Output
MAC address/prefix	The MAC address of the device that is configured to bypass 802.1X authentication.	**all**
VLAN-Assignment	The name of the VLAN to which the device is assigned.	**all**

Table 141: show dot1x static-mac-address Output Fields *(continued)*

Field Name	Field Description	Level of Output
Interface	The name of the interface on which authentication is bypassed for a given MAC address.	**all**

Sample Output

show dot1x static-mac-address

```
user@host> show dot1x static-mac-address
```

```
MAC address/prefix          VLAN-Assignment          Interface
00:50:56:85:66:0f/48        vlan6                    ge-0/0/0.0
00:50:56:9e:56:42/48        vlan6                    ge-0/0/1.0
```

show dot1x static-mac-address interface (Specific Interface)

```
user@host> show dot1x static-mac-address interface ge-0/0/0
```

```
MAC address/prefix          VLAN-Assignment          Interface
00:50:56:85:66:0f/48        vlan6                    ge-0/0/0.0
```

show dot1x statistics

Syntax

```
show dot1x statistics interface <interface-name>
```

Release Information

Command introduced in Junos OS Release 15.1X49-D80.

Description

Display 802.1X statistics on this interface.

Options

interface *interface-name*—(Optional) Displays statistical information for the interface.

Required Privilege Level

view

RELATED DOCUMENTATION

dot1x | 1122

show dot1x authentication-bypassed-users | 1457

List of Sample Output

show dot1x statistics interface on page 1470

Sample Output

show dot1x statistics interface

```
user@host> show dot1x statistics interface ge-0/0/0

Interface: ge-0/0/0.0
  TxReqId = 4 TxReq = 0 TxTotal = 4
  RxStart = 0 RxLogoff = 0 RxRespId = 0 RxResp = 0
  RxInvalid = 0 RxLenErr = 0 RxTotal = 0
  LastRxVersion = 0 LastRxSrcMac = 00:50:56:85:66:0f
```

show edge-virtual-bridging

Syntax

```
show edge-virtual-bridging
<detail>
<edge-control-protocol statistics <interface interface-name>>
<firewall>
<interface interface-name>
vsi-profiles <interface interface-name>
```

Release Information

Command introduced in Junos OS Release 12.1 for EX Series switches.

Description

Display information about edge virtual bridging (EVB).

Options

none—Display EVB parameters for all interfaces configured with EVB.

detail—(Optional) Display EVB parameters and virtual station interface (VSI) profiles associated with each interface.

edge-control-protocol statistics <interface *<nterface-name>*—(Optional) Display Edge Control Protocol (ECP)l statistics for all configured EVB interfaces or for the specified interface.

firewall—Display the firewall filters created by EVB.

interface *interface-name*—(Optional) Display EVB parameters for the specified interface.

vsi-profiles <interface *interface-name*>—(Optional) Display VSI profiles associated on each interface or for the specified interface.

Required Privilege Level

view

RELATED DOCUMENTATION

Example: Configuring Edge Virtual Bridging for Use with VEPA Technology on an EX Series Switch | **1062**

List of Sample Output

show edge-virtual-bridging vsi-profiles on page 1473
show edge-virtual-bridging vsi-profiles interface on page 1473
show edge-virtual-bridging firewall on page 1474

Output Fields

Table 142 on page 1472 lists the output fields for the **show edge-virtual-bridging** command. Output fields are listed in the approximate order in which they appear.

Table 142: show edge-virtual-bridging Output Field Descriptions

Field Name	Field Description
Interface	Switch interface configured for EVB.
Interface input ECP Packets	Number of ECP packets received by the switch. ECP is a Layer 2 protocol that is used to carry VSI Discovery and Configuration Protocol (VDP) messages.
Interface output ECP Packets	Number of ECP packets sent by the switch. ECP is a Layer 2 protocol that is used to carry VDP messages.
Forwarding Mode	Mode by which packets are forwarded to their destination. The value for forwarding mode is either **Standard** (meaning the forwarding is done through 802.1Q) or **Reflective-relay**, meaning that both the source and destination addresses are located on the same VM server.
RTE	Retransmission timer exponent (RTE) is an EVB interface attribute used to calculate the minimum VDP protocol data unit (PDU) retransmission time.
Number of VSIs	Number of virtual station interfaces on the switch connected to the VEPA.
Protocols	EVB protocols currently enabled. The values can be **VDP**, **ECP** or **RTE**. Protocols are configured during the capabilities exchange via an EVB type, length, and value (TLV) carried by the Link Layer Discovery Protocol (LLDP) between the switch and the server.
VSI profile	EVB profile including parameters that uniquely identify each VSI entry (VSI manager, VSI type, VSI version, VSI instance, VSI state).
Filter Name	Name of the filter defined in the firewall stanza.
Counters	Number of packets and bytes that have satisfied the match conditions defined by the filter.

Sample Output

show edge-virtual-bridging

user@switch#**show edge-virtual-bridging**

```
Interface     Forwarding Mode      RTE   Number of VSIs  Protocols
ge-0/0/20.0   Reflective-relay     25    400             ECP, VDP, RTE
```

show edge-virtual-bridging interface

user@switch#**show edge-virtual-bridging interface ge-0/0/20.0**

```
Interface: ge-0/0/20.0, Forwarding mode: Reflective-relay RTE: 25, Number of VSIs:
 400, Protocols: ECP, VDP, RTE
VSI profiles:
 Manager: 97, Type: 997, Version: 3, VSI State: Associate
 Instance: 09b11c53-8b5c-4eeb-8f00-c84ebb0bb997
      MAC                      VLAN
      00:10:94:00:00:04
```

show edge-virtual-bridging edge-control-protocol statistics

user@switch#**show edge-virtual-bridging edge-control-protocol-statistics**

```
Interface: ge-0/0/20.0
      Input ECP packets: 302
      Output ECP packets: 303
```

show edge-virtual-bridging vsi-profiles

user@switch#**show edge-virtual-bridging vsi-profiles**

```
Interface: ge-0/0/20.0
Manager: 97, Type: 997, Version: 3, VSI State: Associate
 Instance: 09b11c53-8b5c-4eeb-8f00-c84ebb0bb997
      MAC                      VLAN
      00:10:94:00:00:04        3
```

show edge-virtual-bridging vsi-profiles interface

user@switch#**show edge-virtual-bridging vsi-profiles interface ge-0/0/20.0**

```
Interface: ge-0/0/20.0
Manager: 97, Type: 997, Version: 3, VSI State: Associate
 Instance: 09b11c53-8b5c-4eeb-8f00-c84ebb0bb997
          MAC
          VLAN   00:10:94:00:00:04         3
```

show edge-virtual-bridging firewall

user@switch#**show edge-virtual-bridging firewall**

```
Filter name: evb_filter_ge-0/0/20
Counters:
  Name: evb_filter_term_3_00:10:94:00:00:04_default
          Bytes: 0, Packets: 0
  Name: f3_accept__evb_filter_term_3_00:10:94:00:00:04-f3-t1
          Bytes: 1028, Packets: 14
```

show ethernet-switching flood

Syntax

```
show ethernet-switching flood
<brief | detail | extensive>
<event-queue>
<instance instance-name>
<logical-system logical-system-name>
<route (all-ce-flood | all ve-flood | alt-root-flood | bd-flood | mlp-flood | re-flood)>
<vlan-name vlan-name>
```

Release Information

Command introduced in Junos OS Release 12.3R2.
Command introduced in Junos OS Release 12.3R2 for EX Series switches.
Command introduced in Junos OS Release 17.4R1 for QFX Series switches.

Description

(EX Series switches and QFX Series switches only) Display Ethernet-switching flooding information.

Options

none—Display all Ethernet-switching flooding information for all VLANs.

brief | detail | extensive—(Optional) Display the specified level of output.

event-queue—(Optional) Display the queue of pending Ethernet-switching flood events.

instance *instance-name*—(Optional) Display Ethernet-switching flooding information for the specified routing instance.

logical-system *logical-system-name*—(Optional) Display Ethernet-switching flooding information for the specified logical system.

route (all-ce-flood | all ve-flood | alt-root-flood | bd-flood | mlp-flood | re-flood)—(Optional) Display the following:

- **all-ce-flood**—Display the route for flooding traffic to all customer edge routers or switches if **no-local-switching** is enabled.
- **all-ve-flood**—Display the route for flooding traffic to all VPLS edge routers or switches if **no-local-switching** is enabled.
- **alt-root-flood**—Display the Spanning Tree Protocol (STP) alt-root flooding route used for the interface.
- **bd-flood**—Display the route for flooding traffic of a VLAN if **no-local-switching** is not enabled.

- **mlp-flood**—Display the route for flooding traffic to MAC learning chips.
- **re-flood**—Display the route for Routing Engine flooding to all interfaces.

vlan-name *vlan-name*—(Optional) Display Ethernet-switching flooding information for the specified VLAN.

Required Privilege Level
view

List of Sample Output

Sample Output

show ethernet-switching flood

`user@host>` **show ethernet-switching flood**

```
Name: __juniper_private1__
CEs: 0
VEs: 0
Name: default-switch
CEs: 9
VEs: 0
VLAN Name: VLAN101
Flood Routes:
  Prefix    Type          Owner                NhType          NhIndex
  0x3057b/51 FLOOD_GRP_COMP_NH __all_ces__     comp            12866
  0x30004/51 FLOOD_GRP_COMP_NH __re_flood__    comp            12863
VLAN Name: VLAN102
Flood Routes:
  Prefix    Type          Owner                NhType          NhIndex
  0x3057c/51 FLOOD_GRP_COMP_NH __all_ces__     comp            12875
  0x30005/51 FLOOD_GRP_COMP_NH __re_flood__    comp            12872
VLAN Name: VLAN103
Flood Routes:
  Prefix    Type          Owner                NhType          NhIndex
  0x3057d/51 FLOOD_GRP_COMP_NH __all_ces__     comp            12884
  0x30006/51 FLOOD_GRP_COMP_NH __re_flood__    comp            12881
```

show ethernet-switching flood brief

```
user@host> show ethernet-switching flood brief
```

```
Name                Active CEs         Active VEs
__juniper_private1__ 0                 0
default-switch      9                  0
```

show ethernet-switching flood detail

```
user@host> show ethernet-switching flood detail
```

```
Name: __juniper_private1__
CEs: 0
VEs: 0
Name: default-switch
CEs: 9
VEs: 0
VLAN Name: VLAN101
Flood Routes:
  Prefix    Type          Owner                NhType          NhIndex
  0x3057b/51 FLOOD_GRP_COMP_NH __all_ces__     comp            12866
  0x30004/51 FLOOD_GRP_COMP_NH __re_flood__    comp            12863
VLAN Name: VLAN102
Flood Routes:
  Prefix    Type          Owner                NhType          NhIndex
  0x3057c/51 FLOOD_GRP_COMP_NH __all_ces__     comp            12875
  0x30005/51 FLOOD_GRP_COMP_NH __re_flood__    comp            12872
VLAN Name: VLAN103
Flood Routes:
  Prefix    Type          Owner                NhType          NhIndex
  0x3057d/51 FLOOD_GRP_COMP_NH __all_ces__     comp            12884
  0x30006/51 FLOOD_GRP_COMP_NH __re_flood__    comp            12881
```

show ethernet-switching flood extensive

```
user@host> show ethernet-switching flood extensive
```

```
Name: __juniper_private1__
CEs: 0
VEs: 0
Name: default-switch
CEs: 9
VEs: 0
```

```
VLAN Name: VLAN101
  Flood route prefix: 0x3057b/51
  Flood route type: FLOOD_GRP_COMP_NH
  Flood route owner: __all_ces__
  Flood group name: __all_ces__
  Flood group index: 1
  Nexthop type: comp
  Nexthop index: 12866
    Flooding to:
    Name             Type          NhType          Index
    __all_ces__      Group          comp            12860
        Composition: split-horizon
        Flooding to:
        Name             Type          NhType          Index
        ae20.0           CE             ucst            7605

  Flood route prefix: 0x30004/51
  Flood route type: FLOOD_GRP_COMP_NH
  Flood route owner: __re_flood__
  Flood group name: __re_flood__
  Flood group index: 65534
  Nexthop type: comp
  Nexthop index: 12863
    Flooding to:
    Name             Type          NhType          Index
    __all_ces__      Group          comp            12860
        Composition: split-horizon
        Flooding to:
        Name             Type          NhType          Index
        ae20.0           CE             ucst            7605
VLAN Name: VLAN102

  Flood route prefix: 0x3057c/51
  Flood route type: FLOOD_GRP_COMP_NH
  Flood route owner: __all_ces__
  Flood group name: __all_ces__
  Flood group index: 1
  Nexthop type: comp
  Nexthop index: 12875
    Flooding to:
    Name             Type          NhType          Index
    __all_ces__      Group          comp            12869
        Composition: split-horizon
        Flooding to:
```

```
        Name             Type          NhType          Index
        ae20.0           CE             ucst            7605

  Flood route prefix: 0x30005/51
  Flood route type: FLOOD_GRP_COMP_NH
  Flood route owner: __re_flood__
  Flood group name: __re_flood__
  Flood group index: 65534
  Nexthop type: comp
  Nexthop index: 12872
    Flooding to:
    Name             Type          NhType          Index
    __all_ces__      Group          comp           12869
        Composition: split-horizon
        Flooding to:
        Name             Type          NhType          Index
        ae20.0           CE             ucst            7605
VLAN Name: VLAN103

  Flood route prefix: 0x3057d/51
  Flood route type: FLOOD_GRP_COMP_NH
  Flood route owner: __all_ces__
  Flood group name: __all_ces__
  Flood group index: 1
  Nexthop type: comp
  Nexthop index: 12884
    Flooding to:
    Name             Type          NhType          Index
    __all_ces__      Group          comp           12878
        Composition: split-horizon
        Flooding to:
        Name             Type          NhType          Index
        ae20.0           CE             ucst            7605

  Flood route prefix: 0x30006/51
  Flood route type: FLOOD_GRP_COMP_NH
  Flood route owner: __re_flood__
  Flood group name: __re_flood__
  Flood group index: 65534
  Nexthop type: comp
  Nexthop index: 12881
    Flooding to:
    Name             Type          NhType          Index
    __all_ces__      Group          comp           12878
```

```
        Composition: split-horizon
        Flooding to:
        Name             Type          NhType          Index
        ae20.0           CE             ucst            7605
VLAN Name: VLAN104
```

show ethernet-switching interface

Syntax

```
show ethernet-switching interface
<brief | detail | extensive>
<interface-name>
```

Release Information

Command introduced in Junos OS Release 12.3R2.
Command introduced in Junos OS Release 12.3R2 for EX Series switches.
Command introduced in Junos OS Release 13.2x51 for QFX Series switches.

Description

Display Layer 2 learning information for all the interfaces.

Options

none—Display Ethernet-switching information for all interfaces.

brief | detail | extensive—(Optional) Display the specified level of output.

interface-name—(Optional) Display Ethernet-switching information for the specified interface.

Required Privilege Level

view

RELATED DOCUMENTATION

List of Sample Output

show ethernet switching interface (Specific Interface) on page 1482
show ethernet switching interface (Storm Control in Effect) on page 1483
show ethernet-switching interface detail on page 1484

Output Fields

Table 143 on page 1481 describes the output fields for the **show ethernet-switching interface** command. Output fields are listed in the approximate order in which they appear.

Table 143: show ethernet-switching interface Output Fields

Field Name	Field Description
Logical interface	Name of the logical interface.

Table 143: show ethernet-switching interface Output Fields *(continued)*

Field Name	Field Description
VLAN members	VLANs associated with this interface.
Tag	VLAN ID.
MAC limit	Number of MAC addresses that can be associated with the interface.
STP state	Spanning Tree protocol (STP) state.
Logical interface flags	Status of Layer 2 learning properties for each interface: • **DL**—MAC learning is disabled. • **LH**—MAC interface limit has been reached. • **AD**—Packets are dropped after the MAC interface limit is reached. • **DN**—The MAC interface is down. • **MMAS**—The MAC interface is disabled after a MAC address move. • **SCTL**—The MAC interface is disabled after a configured storm-control level is exceeded. **NOTE:** If the physical interface is shutdown due to storm control, all logical interfaces on the shutdown interface display the SCTL logical interface flag.
Tagging	Tagging state of the VLAN.

Sample Output

show ethernet switching interface (Specific Interface)

`user@host>` **show ethernet-switching inerface ae10.0**

```
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down)

Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
ae10.0                        8192                                   tagged
```

```
             VLAN70..   701   1024     Forwarding

             VLAN70..   702   1024     Forwarding

             VLAN70..   703   1024     Forwarding

             VLAN70..   704   1024     Forwarding

             VLAN70..   705   1024     Forwarding

             VLAN70..   706   1024     Forwarding

             VLAN70..   707   1024     Forwarding

             VLAN70..   708   1024     Forwarding

             VLAN70..   709   1024     Forwarding

             VLAN71..   710   1024     Forwarding

             VLAN71..   711   1024     Forwarding

             VLAN71..   712   1024     Forwarding

             VLAN71..   713   1024     Forwarding

             VLAN71..   714   1024     Forwarding

             VLAN71..   715
[...output truncated...]
```

show ethernet switching interface (Storm Control in Effect)

user@host> **show ethernet-switching interface ge-0/0/2.0**

```
Logical Interface flags (DL - disable learning, AD - packet action drop, LH -
                         MAC limit hit, DN - interface down, MMAS - Mac-move
                         action shutdown, AS - Autostate-exclude enabled, SCTL
                         - shutdown by Storm-control, MI - MAC+IP limit hit)

Logical      Vlan       TAG   MAC    MAC+IP STP         Logical          Tagging
interface    members          limit  limit  state       interface flags
ge-0/0/2.0                    524287 8192               DN,SCTL          untagged
```

```
            VLAN1      100   65535  1024   Forwarding                    untagged
```

show ethernet-switching interface detail

user@host> **show ethernet-switching interface detail**

```
Information for interface family:
Name: ge-1/0/3.0
  Type: IFF                              Handle: 0x8bba280
  Index: 331                             Generation: 159
                                         Flags: UP,
  IFD index: 141                         Routing/Vlan index: 4
  IFL index: 331                         Address family: 50
  Sequence number: 0                     MAC sequence number: 0
  MAC limit: 65535                       MACs learned: 0
  Static MACs learned: 0                 Non configured static MACs learned: 0
Name: ge-1/0/3.0
  Type: IFBD (static)                    Handle: 0x8bb6e00
  Index:                                 Generation: 129
  Trunk id: 0                            Flags: UP,
  IFD index:                             Routing/Vlan index: 2
  IFL index:                             Address family:
  Sequence number: 1                     MAC sequence number: 1
  MAC limit: 65535                       MACs learned: 0
  Static MACs learned: 0                 Non configured static MACs learned: 0
  VSTP index: 11                         Rewrite op:
Name: ge-1/0/3.0
  Type: IFBD (static)                    Handle: 0x8bb6f00
  Index:                                 Generation: 130
  Trunk id: 0                            Flags: UP,
  IFD index:                             Routing/Vlan index: 3
  IFL index:                             Address family:
  Sequence number: 1                     MAC sequence number: 1
  MAC limit: 65535                       MACs learned: 0
  Static MACs learned: 0                 Non configured static MACs learned: 0
  VSTP index: 11                         Rewrite op:
```

show ethernet-switching interfaces

Syntax

```
show ethernet-switching interfaces
<brief | detail | summary>
<interface interface-name>
```

Release Information

Command introduced in Junos OS Release 9.0 for EX Series switches.

In Junos OS Release 9.6 for EX Series switches, the following updates were made:

- **Blocking** field output was updated.
- The default view was updated to include information about 802.1Q tags.
- The **detail** view was updated to include information on VLAN mapping.

Command introduced in Junos OS Release 11.1 for the QFX Series.

In Junos OS Release 11.1 for EX Series switches, the **detail** view was updated to include reflective relay information.

Command introduced in Junos OS Release 14.1X53-D20 for OCX Series switches.

Description

Display information about switched Ethernet interfaces.

Options

none—(Optional) Display brief information for Ethernet-switching interfaces.

brief | detail | summary—(Optional) Display the specified level of output.

interface *interface-name*—(Optional) Display Ethernet-switching information for a specific interface.

Required Privilege Level

view

RELATED DOCUMENTATION

Troubleshooting Ethernet Switching | **1073**Understanding Bridging and VLANs on Switches | **168**

Example: Setting Up Basic Bridging and a VLAN on Switches | **203**

Example: Setting Up Bridging with Multiple VLANs | **236**

Understanding FCoE

Interfaces Overview for Switches

show ethernet-switching mac-learning-log | 1502

show ethernet-switching table | 1520

Configuring Autorecovery for Port Security Events

List of Sample Output

Output Fields

For QFX Series, QFabric, NFX Series, EX4600 and OCX1100:

Table 144 on page 1486 lists the output fields for the **show ethernet-switching interfaces** command on QFX Series, QFabric, NFX Series, EX4600 and OCX1100. Output fields are listed in the approximate order in which they appear.

Table 144: show ethernet-switching interfaces Output Fields

Field Name	Field Description	Level of Output
Interface	Name of a switching interface.	All levels
State	Interface state. Values are **up** or **down**.	none, **brief**, **detail**, **summary**
VLAN members	Name of a VLAN.	none, **brief**, **detail**, **summary**

Table 144: show ethernet-switching interfaces Output Fields *(continued)*

Field Name	Field Description	Level of Output
Blocking	Forwarding state of the interface: • **blocked**—Traffic is not being forwarded on the interface. • **unblocked**—Traffic is forwarded on the interface. • **MAC limit exceeded**—The interface is temporarily disabled because of a MAC limiting error. The disabled interface is automatically restored to service when the disable timeout expires. • **MAC move limit exceeded**—The interface is temporarily disabled because of a MAC move limiting error. The disabled interface is automatically restored to service when the disable timeout expires. • **Storm control in effect** —The interface is temporarily disabled because of a storm control error. The disabled interface is automatically restored to service when the disable timeout expires. • **Storm control shutdown in effect** —The interface is temporarily disabled because of a storm control shutdown error. The disabled interface is automatically restored to service when the disable timeout expires.	none, **brief**, **detail**, **summary**
Index	VLAN index internal to Junos OS software.	**detail**
untagged | tagged	Specifies whether the interface forwards IEEE802.1Q-tagged or untagged traffic.	**detail**

Output fields for EX Series:

Table 145 on page 1487 lists the output fields for the **show ethernet-switching interfaces** command on EX Series switches. Output fields are listed in the approximate order in which they appear.

Table 145: show ethernet-switching interfaces Output Fields

Field Name	Field Description	Level of Output
Interface	Name of a switching interface.	none, **brief**, **detail**, **summary**
Index	VLAN index internal to Junos OS.	**detail**
State	Interface state. Values are **up** and **down**.	none, **brief**, **detail**
Port mode	The **access** mode is the port mode default and works with a single VLAN. Port mode can also be **trunk**, which accepts tagged packets from multiple VLANs on other switches. The third port mode value is **tagged-access**, which accepts tagged packets from access devices.	**detail**

Table 145: show ethernet-switching interfaces Output Fields *(continued)*

Field Name	Field Description	Level of Output
Reflective Relay Status	Reflective relay allows packets to use the same interface for both upstream and downstream traffic. When reflective relay has been configured, the status displayed is always **enabled** . When reflective relay is not configured, this entry does not appear in the command output.	**detail**
Ether type for the interface	Ether type is a two-octet field in an Ethernet frame used to indicate which protocol is encapsulated in the payload of an incoming Ethernet packet. Both 802.1Q packets and Q-in-Q packets use this field. The output displayed for this particular field indicates the interface's Ether type, which is used to match the Ether type of incoming 802.1Q packets and Q-in-Q packets. The indicated Ether type field is also added to the interface's outgoing 802.1Q and Q-in-Q packets.	**detail**
VLAN membership	Names of VLANs that belong to this interface.	none, **brief**, **detail**,
Tag	Number of the 802.1Q tag.	none, **brief**, **detail**,
Tagging	Specifies whether the interface forwards 802.1Q **tagged** or **untagged** traffic.	none, **brief**, **detail**,
Blocking	The forwarding state of the interface: • **unblocked**—Traffic is forwarded on the interface. • **blocked**—Traffic is not being forwarded on the interface. • **Disabled by bpdu control**—The interface is disabled due to receiving BPDUs on a protected interface. If the **disable-timeout** statement has been included in the BPDU configuration, the interface automatically returns to service after the timer expires. • **blocked by RTG**—The specified redundant trunk group is disabled. • **blocked by STP**—The interface is disabled due to a spanning-tree protocol error. • **MAC limit exceeded**—The interface is temporarily disabled due to a MAC limit error. The disabled interface is automatically restored to service when the disable timeout expires. • **MAC move limit exceeded**—The interface is temporarily disabled due to a MAC move limit error. The disabled interface is automatically restored to service when the disable timeout expires. • **Storm control in effect**—The interface is temporarily disabled due to a storm control error. The disabled interface is automatically restored to service when the disable timeout expires.	none, **brief**, **detail**,

Table 145: show ethernet-switching interfaces Output Fields *(continued)*

Field Name	Field Description	Level of Output
Number of MACs learned on IFL	Number of MAC addresses learned by this interface.	**detail**
mapping	When mapping is configured, the status is one of the following C-VLAN to S-VLAN mapping types: • **dot1q-tunneled**—The interface maps all traffic to the S-VLAN (all-in-one bundling). • **native**—The interface maps untagged and priority tagged packets to the S-VLAN. • **push**—The interface maps packets to a firewall filter to an S-VLAN. • **policy-mapped**—The interface maps packets to a specifically defined S-VLAN. • *integer*—The interface maps packets to the specified S-VLAN. When mapping is not configured, this entry does not appear in the command output.	**detail**

Sample Output for QFX Series Switches, QFabric, NFX Series, EX4600 and OCX1100

show ethernet-switching interfaces

```
user@switch> show ethernet-switching interfaces
```

```
Interface   State    VLAN members          Blocking
xe-0/0/0.0  up       T1122                 unblocked
xe-0/0/1.0  down     default               — MAC limit exceeded
xe-0/0/2.0  down     default               — MAC move limit exceeded
xe-0/0/3.0  down     default               — Storm control in effect
xe-0/0/4.0  down     default               unblocked
xe-0/0/5.0  down     default               unblocked
xe-0/0/6.0  down     default               unblocked
xe-0/0/7.0  down     default               unblocked
xe-0/0/8.0  down     default               unblocked
xe-0/0/9.0  up       T111                  unblocked
xe-0/0/10.0 down     default               unblocked
xe-0/0/11.0 down     default               unblocked
```

```
xe-0/0/12.0 down     default               unblocked
xe-0/0/13.0 down     default               unblocked
xe-0/0/14.0 down     default               unblocked
xe-0/0/15.0 down     default               unblocked
xe-0/0/16.0 down     default               unblocked
xe-0/0/17.0 down     default               unblocked
xe-0/0/18.0 down     default               unblocked
xe-0/0/19.0 up       T111                  unblocked
xe-0/1/0.0  down     default               unblocked
xe-0/1/1.0  down     default               unblocked
xe-0/1/2.0  down     default               unblocked
xe-0/1/3.0  down     default               unblocked
```

show ethernet-switching interfaces summary

`user@switch>` **show ethernet-switching interfaces summary**

```
xe-0/0/0.0
xe-0/0/1.0
xe-0/0/2.0
xe-0/0/3.0
xe-0/0/8.0
xe-0/0/10.0
xe-0/0/11.0
```

show ethernet-switching interfaces brief

`user@switch>` **show ethernet-switching interfaces brief**

```
Interface   State    VLAN members          Blocking
xe-0/0/0.0  down     default               unblocked
xe-0/0/1.0  down     employee-vlan         unblocked
xe-0/0/2.0  down     employee-vlan         unblocked
xe-0/0/3.0  down     employee-vlan         unblocked
xe-0/0/8.0  down     employee-vlan         unblocked
xe-0/0/10.0 down     default               unblocked
xe-0/0/11.0 down     employee-vlan         unblocked
```

show ethernet-switching interfaces detail

`user@switch>` **show ethernet-switching interfaces detail**

```
Interface: xe-0/0/0.0 Index: 65
  State: down
  VLANs:
    default                untagged     unblocked

Interface: xe-0/0/1.0 Index: 66
  State: down
  VLANs:
    employee-vlan          untagged     unblocked

Interface: xe-0/0/2.0 Index: 67
  State: down
  VLANs:
    employee-vlan          untagged     unblocked

Interface: xe-0/0/3.0 Index: 68
  State: down
  VLANs:
    employee-vlan          untagged     unblocked

Interface: xe-0/0/8.0 Index: 69
  State: down
  VLANs:
    employee-vlan          untagged     unblocked

Interface: xe-0/0/10.0 Index: 70
  State: down
  VLANs:
    default                untagged     unblocked

Interface: xe-0/0/11.0 Index: 71
  State: down
  VLANs:
    employee-vlan          tagged       unblocked
```

show ethernet-switching interfaces interface-name

`user@switch>` **show ethernet-switching interfaces xe-0/0/0.0**

```
 Interface   State    VLAN members           Blocking
xe-0/0/0.0  down     default                unblocked
```

Sample Output for EX Series Switches

show ethernet-switching interfaces

```
user@switch> show ethernet-switching interfaces
```

```
Interface    State  VLAN members        Tag   Tagging  Blocking

ae0.0        up      default                  untagged unblocked
ge-0/0/2.0   up      vlan300           300    untagged blocked by RTG (rtggroup)
ge-0/0/3.0   up      default                           blocked by STP
ge-0/0/4.0   down    default                           MAC limit exceeded
ge-0/0/5.0   down    default                           MAC move limit exceeded
ge-0/0/6.0   down    default                           Storm control in effect
ge-0/0/7.0   down    default                           unblocked
ge-0/0/13.0  up      default                  untagged unblocked
ge-0/0/14.0  up      vlan100             100   tagged   unblocked
                     vlan200             200   tagged   unblocked
ge-0/0/15.0  up      vlan100             100   tagged   blocked by STP
                     vlan200             200   tagged   blocked by STP
ge-0/0/16.0  down    default                  untagged unblocked
ge-0/0/17.0  down    vlan100             100   tagged   Disabled by bpdu-control
                     vlan200             200   tagged   Disabled by bpdu-control
```

show ethernet-switching interfaces ge-0/0/15 brief

```
user@switch> show ethernet-switching interfaces ge-0/0/15 brief
```

```
Interface   State  VLAN members   Tag   Tagging        Blocking

ge-0/0/15.0  up     vlan100        100   tagged         blocked by STP
                    vlan200        200   tagged         blocked by STP
```

show ethernet-switching interfaces ge-0/0/2 detail (Blocked by RTG rtggroup)

```
user@switch> show ethernet-switching interfaces ge-0/0/2 detail
```

```
Interface: ge-0/0/2.0, Index: 65, State: up, Port mode: Access
Ether type for the interface: 0X8100
VLAN membership:
    vlan300, 802.1Q Tag: 300, untagged, msti-id: 0, blocked by RTG(rtggroup)
```

```
Number of MACs learned on IFL: 0
```

show ethernet-switching interfaces ge-0/0/15 detail (Blocked by STP)

`user@switch>` **show ethernet-switching interfaces ge-0/0/15 detail**

```
Interface: ge-0/0/15.0, Index: 70, State: up, Port mode: Trunk
Ether type for the interface: 0X8100
VLAN membership:
    vlan100, 802.1Q Tag: 100, tagged, msti-id: 0, blocked by STP
    vlan200, 802.1Q Tag: 200, tagged, msti-id: 0, blocked by STP

Number of MACs learned on IFL: 0
```

show ethernet-switching interfaces ge-0/0/17 detail (Disabled by bpdu-control)

`user@switch>` **show ethernet-switching interfaces ge-0/0/17 detail**

```
Interface: ge-0/0/17.0, Index: 71, State: down, Port mode: Trunk
Ether type for the interface: 0X8100
VLAN membership:
    vlan100, 802.1Q Tag: 100, tagged, msti-id: 1, Disabled by bpdu-control
    vlan200, 802.1Q Tag: 200, tagged, msti-id: 2, Disabled by bpdu-control
Number of MACs learned on IFL: 0
```

show ethernet-switching interfaces detail (C-VLAN to S-VLAN Mapping)

`user@switch>`**show ethernet-switching interfaces ge-0/0/6.0 detail**

```
Interface: ge-0/0/6.0, Index: 73, State: up, Port mode: Access
Ether type for the interface: 0X8100
VLAN membership:
    map, 802.1Q Tag: 134, Mapped Tag: native, push, dot1q-tunneled, unblocked
    map, 802.1Q Tag: 134, Mapped Tag: 20, push, dot1q-tunneled, unblocked
```

show ethernet-switching interfaces detail (Reflective Relay Is Configured)

`user@switch1>` **show ethernet-switching interfaces ge-7/0/2 detail**

```
Interface: ge-7/0/2, Index: 66, State: down, Port mode: Tagged-access
Ether type for the interface: 0X8100
Reflective Relay Status: Enabled
Ether type for the interface: 0x8100
VLAN membership:
    VLAN_Purple VLAN_Orange VLAN_Blue, 802.1Q Tag: 450, tagged, unblocked
Number of MACs learned on IFL: 0
```

show ethernet-switching layer2-protocol-tunneling interface

Syntax

```
show ethernet-switching-layer2-protocol-tunneling interface
<interface-name>
```

Release Information

Command introduced in Junos OS Release 10.0 for EX Series switches.
Command introduced in Junos OS Release 12.1 for the QFX Series.

Description

Display information about Layer 2 protocol tunneling (L2PT) on interfaces that have been configured for L2PT.

Options

none—Display L2PT information about all interfaces on which L2PT is enabled.

interface-name—(Optional) Display L2PT information for the specified interface.

Required Privilege Level

view

RELATED DOCUMENTATION

- show ethernet-switching layer2-protocol-tunneling statistics | **1497**
- show ethernet-switching layer2-protocol-tunneling vlan | **1500**
- Configuring Layer 2 Protocol Tunneling on EX Series Switches Without ELS Support | **699**
- show ethernet-switching layer2-protocol-tunneling statistics | **1497**
- show ethernet-switching layer2-protocol-tunneling vlan | **1500**

List of Sample Output

show ethernet-switching layer2-protocol-tunneling interface on page 1496
show ethernet-switching layer2-protocol-tunneling interface xe-0/0/0.0 on page 1496

Output Fields

Table 146 on page 1496 lists the output fields for the **show ethernet-switching layer2-protocol-tunneling interface** command. Output fields are listed in the approximate order in which they appear.

Table 146: show ethernet-switching layer2-protocol-tunneling interface Output Fields

Field Name	Field Description
Interface	Name of an interface on the switch.
Operation	Type of operation being performed on the interface. Values are **Encapsulation** and **Decapsulation**.
State	State of the interface. Values are **active** and **shutdown**.
Description	If the interface state is **shutdown**, displays why the interface is shut down. If the description says **Loop detected**, it means that the interface is an access interface that has received L2PT-enabled PDUs. Access interfaces should not receive L2PT-enabled PDUs. This scenario might mean that there is a loop in the network.

Sample Output

show ethernet-switching layer2-protocol-tunneling interface

```
user@switch> show ethernet-switching layer2-protocol-tunneling interface
```

```
Layer2 Protocol Tunneling information:
Interface     Operation       State      Description
xe-0/0/0.0    Encapsulation   Shutdown   Shutdown threshold exceeded
xe-0/0/1.0    Decapsulation   Shutdown   Loop detected
xe-0/0/2.0    Decapsulation   Active
```

show ethernet-switching layer2-protocol-tunneling interface xe-0/0/0.0

```
user@switch> show ethernet-switching layer2-protocol-tunneling interface xe-0/0/0.0
```

```
Layer2 Protocol Tunneling information:
Interface     Operation       State      Description
xe-0/0/0.0    Encapsulation   Shutdown   Shutdown threshold exceeded
```

show ethernet-switching layer2-protocol-tunneling statistics

Syntax

```
show ethernet-switching-layer2-protocol-tunneling statistics
<interface interface-name>
<vlan vlan-name>
```

Release Information

Command introduced in Junos OS Release 10.0 for EX Series switches.
Command introduced in Junos OS Release 12.1 for the QFX Series.

Description

Display Layer 2 protocol tunneling (L2PT) statistics for Layer 2 PDU packets received by the switch.

> **NOTE:** The **show ethernet-switching-layer2-protocol-tunneling statistics** command does not display L2PT statistics for Layer 2 PDU packets transmitted from the switch.

Options

none—Display L2PT statistics for all interfaces on which you enabled L2PT.

interface *interface-name*—(Optional) Display L2PT statistics for the specified interface.

vlan *vlan-name*—(Optional) Display L2PT statistics for the specified VLAN.

Required Privilege Level

view

RELATED DOCUMENTATION

- clear ethernet-switching layer2-protocol-tunneling statistics | 1441
- show ethernet-switching layer2-protocol-tunneling interface | 1495
- show ethernet-switching layer2-protocol-tunneling vlan | 1500
- Example: Configuring Layer 2 Protocol Tunneling on EX Series Switches Without ELS Support | 701
- Configuring Layer 2 Protocol Tunneling on EX Series Switches Without ELS Support | 699
- show vlans | 1648

List of Sample Output

Output Fields

Table 147 on page 1498 lists the output fields for the **show ethernet-switching layer2-protocol-tunneling statistics** command. Output fields are listed in the approximate order in which they appear.

Table 147: show ethernet-switching layer2-protocol-tunneling statistics Output Fields

VLAN	Field Description
VLAN	Name of a VLAN on which L2PT has been configured.
Interface	Name of an interface on which L2PT has been configured.
Protocol	Name of a protocol for which L2PT has been enabled. Values are **all**, **802.1x**, **802.3ah**, **cdp**, **e-lmi**, **gvrp**, **lacp**, **lldp**, **mmrp**, **mvrp**, **stp**, **udld**, **vstp**, and **vtp**.
Operation	Type of operation being performed on the interface. Values are **Encapsulation** and **Decapsulation**.
Packets	Number of packets that have been encapsulated or de-encapsulated.
Drops	Number of packets that have exceeded the drop threshold and have been dropped.
Shutdowns	Number of times that packets have exceeded the shutdown threshold and the interface has been shut down.

Sample Output

show ethernet-switching layer2-protocol-tunneling statistics

```
user@switch> show ethernet-switching layer2-protocol-tunneling statistics
```

```
Layer2 Protocol Tunneling Statistics:
VLAN   Interface  Protocol  Operation     Packets Drops      Shutdowns
v1     xe-0/0/0.0 mvrp      Encapsulation 0       0          0
v1     xe-0/0/1.0 mvrp      Decapsulation 0       0          0
v1     xe-0/0/2.0 mvrp      Decapsulation 60634   0          0
v2     xe-0/0/0.0 cdp       Encapsulation 0       0          0
```

```
v2     xe-0/0/0.0 gvrp      Encapsulation 0       0          0
v2     xe-0/0/0.0 lldp      Encapsulation 0       0          0
```

show ethernet-switching layer2-protocol-tunneling statistics interface xe-0/0/0.0

```
user@switch> show ethernet-switching layer2-protocol-tunneling statistics interface xe-0/0/0.0
```

```
Layer2 Protocol Tunneling Statistics:
VLAN    Interface  Protocol Operation      Packets  Drops     Shutdowns
v1      xe-0/0/0.0 mvrp     Encapsulation  0        0          0
v2      xe-0/0/0.0 cdp      Encapsulation  0        0          0
v2      xe-0/0/0.0 gvrp     Encapsulation  0        0          0
v2      xe-0/0/0.0 lldp     Encapsulation  0        0          0
v2      xe-0/0/0.0 mvrp     Encapsulation  0        0          0
v2      xe-0/0/0.0 stp      Encapsulation  0        0          0
v2      xe-0/0/0.0 vtp      Encapsulation  0        0          0
v2      xe-0/0/0.0 vstp     Encapsulation  0        0          0
```

show ethernet-switching layer2-protocol-tunneling statistics vlan v2

```
user@switch> show ethernet-switching layer2-protocol-tunneling statistics vlan v2
```

```
Layer2 Protocol Tunneling Statistics:
VLAN    Interface   Protocol Operation       Packets  Drops   Shutdowns
v2      xe-0/0/0.0  cdp      Encapsulation   0        0       0
v2      xe-0/0/0.0  gvrp     Encapsulation   0        0       0
v2      xe-0/0/0.0  lldp     Encapsulation   0        0       0
v2      xe-0/0/0.0  mvrp     Encapsulation   0        0       0
v2      xe-0/0/0.0  stp      Encapsulation   0        0       0
v2      xe-0/0/0.0  vtp      Encapsulation   0        0       0
v2      xe-0/0/0.0  vstp     Encapsulation   0        0       0
v2      xe-0/0/1.0  cdp      Decapsulation   0        0       0
v2      xe-0/0/1.0  gvrp     Decapsulation   0        0       0
v2      xe-0/0/1.0  lldp     Decapsulation   0        0       0
v2      xe-0/0/1.0  mvrp     Decapsulation   0        0       0
v2      xe-0/0/1.0  stp      Decapsulation   0        0       0
v2      xe-0/0/1.0  vtp      Decapsulation   0        0       0
```

show ethernet-switching layer2-protocol-tunneling vlan

Syntax

```
show ethernet-switching-layer2-protocol-tunneling vlan <vlan-name>
```

Release Information

Command introduced in Junos OS Release 10.0 for EX Series switches.
Command introduced in Junos OS Release 12.1 for the QFX Series.

Description

Display information about Layer 2 protocol tunneling (L2PT) on VLANs that have been configured for L2PT.

Options

none—Display information about L2PT for the VLANs on which you have configured L2PT.

vlan-name—(Optional) Display information about L2PT for the specified VLAN.

Required Privilege Level

view

RELATED DOCUMENTATION

- show ethernet-switching layer2-protocol-tunneling interface | **1495**
- show ethernet-switching layer2-protocol-tunneling statistics | **1497**
- Example: Configuring Layer 2 Protocol Tunneling on EX Series Switches Without ELS Support | **701**
- Configuring Layer 2 Protocol Tunneling on EX Series Switches Without ELS Support | **699**
- show vlans | **1648**

List of Sample Output

show ethernet-switching layer2-protocol-tunneling vlan on page 1501
show ethernet-switching layer2-protocol-tunneling vlan v2 on page 1501

Output Fields

Table 148 on page 1501 lists the output fields for the **show ethernet-switching layer2-protocol-tunneling vlan** command. Output fields are listed in the approximate order in which they appear.

Table 148: show ethernet-switching layer2-protocol-tunneling vlan Output Fields

Field Name	Field Description
VLAN	Name of the VLAN on which L2PT has been configured.
Protocol	Name of a protocol for which L2PT has been enabled. Values are **all**, **802.1x**, **802.3ah**, **cdp**, **e-lmi**, **gvrp**, **lacp**, **lldp**, **mmrp**, **mvrp**, **stp**, **vstp**, and **vtp**.
Drop Threshold	Maximum number of Layer 2 PDUs of the specified protocol that can be received per second on the VLAN before the switch begins dropping the Layer 2 PDUs.
Shutdown Threshold	Maximum number of Layer 2 PDUs of the specified protocol that can be received per second on the VLAN before the interface is disabled.

Sample Output

show ethernet-switching layer2-protocol-tunneling vlan

```
user@switch> show ethernet-switching layer2-protocol-tunneling vlan
```

```
Layer2 Protocol Tunneling VLAN information:
VLAN          Protocol      Drop       Shutdown
                            Threshold  Threshold
v1            mvrp          100        200
v2            cdp           0          0
v2            cdp           0          0
v2            gvrp          0          0
```

show ethernet-switching layer2-protocol-tunneling vlan v2

```
user@switch> show ethernet-switching layer2-protocol-tunneling vlan v2
```

```
Layer2 Protocol Tunneling VLAN information:
VLAN          Protocol      Drop       Shutdown
                            Threshold  Threshold
v2            cdp           0          0
v2            cdp           0          0
v2            gvrp          0          0
```

show ethernet-switching mac-learning-log

Syntax

```
show ethernet-switching mac-learning-log
```

Release Information

Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 9.5 for SRX Series devices.
Command introduced in Junos OS Release 11.1 for the QFX Series.

Description

Displays the event log of learned MAC addresses.

Required Privilege Level

view

RELATED DOCUMENTATION

- show ethernet-switching table | 1520
- show ethernet-switching interfaces | 1485
- show ethernet-switching table | 1520
- show ethernet-switching interfaces | 1485
- Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch | 226
- Example: Setting Up Bridging with Multiple VLANs for EX Series Switches | 265
- *Example: Connecting an EX Series Access Switch to a Distribution Switch*

List of Sample Output

show ethernet-switching mac-learning-log (EX Series switch) on page 1504
show ethernet-switching mac-learning-log (QFX Series Switches, QFabric, NFX Series Devices and EX4600) on page 1505
show ethernet-switching mac-learning-log (SRX Series devices) on page 1505

Output Fields

Output fields for EX Series switches:

The following table lists the output fields for the **show ethernet-switching mac-learning-log** command. Output fields are listed in the approximate order in which they appear.

Table 149: show ethernet-switching mac-learning-log Output Fields

Field Name	Field Description
Date and Time	Timestamp when the MAC address was added or deleted from the log.
vlan_name	VLAN name. A value defined by the user for all user-configured VLANs.
MAC	Learned MAC address.
Deleted | Added	MAC address deleted or added to the MAC learning log.
Blocking	The forwarding state of the interface: • **blocked**—Traffic is not being forwarded on the interface. • **unblocked**—Traffic is forwarded on the interface.
Flags	Displays the MAC address flags in which the MAC event occurred. This option is for debugging purposes.

Output fields for QFX Series switches, QFabric, NFX Series devices and EX4600:

Table 150 on page 1503 lists the output fields for the **show ethernet-switching mac-learning-log** command. Output fields are listed in the approximate order in which they appear.

Table 150: show ethernet-switching mac-learning-log Output Fields

Field Name	Field Description
Date and Time	Timestamp in UTC when the MAC operation occurred.
vlan_name	VLAN name. A value defined by the user for all user-configured VLANs. The name of the VLAN on which the MAC is learned.
MAC	Learned MAC address.
Event op	MAC address that are added, learned, deleted, changed or moved from one interface to another interface.
Interface Name	The name of the interface on which the MAC address is learned. When a MAC address is moved, there is another field with the name of the interface. The log displays the name of the interface from where the MAC address moved, and the name of the interface to where the MAC address moved.
Flags	Displays the MAC address flags in which the MAC event occurred. This option is for debugging purposes.

Output fields for SRX Series devices:

Table 151 on page 1504 lists the output fields for the show ethernet-switching mac-learning-log command on SRX Series devices. Output fields are listed in the approximate order in which they appear.

Table 151: show ethernet-switching-mac-learning-log Output Fields

Field Name	Field Description
Date and Time	Timestamp when the MAC address was added or deleted from the log.
VLAN-IDX	VLAN index. An internal value assigned by Junos OS for each VLAN.
MAC	Learned MAC address.
Deleted | Added	MAC address deleted or added to the MAC learning log.
Blocking	The forwarding state of the interface: • blocked—Traffic is not being forwarded on the interface. • unblocked—Traffic is forwarded on the interface.

Sample Output

show ethernet-switching mac-learning-log (EX Series switch)

`user@switch>` **show ethernet-switching mac-learning-log**

```
Mon Feb 25 08:07:05 2008
 vlan_name v1 mac 00:00:00:00:00:00 was deleted
Mon Feb 25 08:07:05 2008
 vlan_name v9 mac 00:00:00:00:00:00 was deleted
Mon Feb 25 08:07:05 2008
 vlan_name HR_vlan mac 00:00:00:00:00:00 was deleted
Mon Feb 25 08:07:05 2008
 vlan_name v3 mac 00:00:00:00:00:00 was deleted
Mon Feb 25 08:07:05 2008
 vlan_name v12 mac 00:00:00:00:00:00 was deleted
Mon Feb 25 08:07:05 2008
 vlan_name v13 mac 00:00:00:00:00:00 was deleted
Mon Feb 25 08:07:05 2008
 vlan_name sales_vlan mac 00:00:00:00:00:00 was deleted
Mon Feb 25 08:07:05 2008
 vlan_name employee1 mac 00:00:00:00:00:00 was deleted
```

```
Mon Feb 25 08:07:05 2008
 vlan_name employee2 mac 00:00:00:00:00:00 was deleted
Mon Feb 25 08:07:05 2008
 vlan_name v3 mac 00:00:00:00:00:00 was added
Mon Feb 25 08:07:05 2008
 vlan_name HR_vlan mac 00:00:00:00:00:00 was added
Mon Feb 25 08:07:05 2008
 vlan_name employee2 mac 00:00:00:00:00:00 was added
Mon Feb 25 08:07:05 2008
 vlan_name employee1 mac 00:00:00:00:00:00 was added
Mon Feb 25 08:07:05 2008
 vlan_name employee2 mac 00:00:05:00:00:05 was learned
Mon Feb 25 08:07:05 2008
 vlan_name employee1 mac 00:30:48:90:54:89 was learned
Mon Feb 25 08:07:05 2008
 vlan_name HR_vlan mac 00:00:5e:00:01:00 was learned
Mon Feb 25 08:07:05 2008
 vlan_name sales_vlan mac 00:00:5e:00:01:08 was learned
[output truncated]
```

show ethernet-switching mac-learning-log (QFX Series Switches, QFabric, NFX Series Devices and EX4600)

`user@switch>` **show ethernet-switching mac-learning-log**

```
Mon Jun 30 13:49:49 2014 vlan_name v11+11 mac 00:10:94:00:00:02 was learned on
ge-1/0/22.0 with flags: 0x2001f     << MAC address that as dynamically learned
Mon Jun 30 13:50:29 2014 vlan_name v11+11 mac 00:10:94:00:00:02 was deleted from
ge-1/0/22.0 with flags: 0x1080  << MAC address that was deleted
Mon Jun 30 13:51:28 2014 vlan_name v11+11 mac 00:00:00:01:01:01 was added to
ge-1/0/22.0 with flags: 0x2013f        << Static MAC address that was added
Mon Jun 30 13:51:46 2014 vlan_name v11+11 mac 00:00:00:01:01:01 was deleted from
ge-1/0/22.0 with flags: 0x1120   << delete of Static MAC address that was deleted
Mon Jun 30 13:52:03 2014 vlan_name v11+11 mac 00:10:94:00:00:02 was learned on
ge-1/0/22.0 with flags: 0x2001f    << MAC address that was dynamically learned
Mon Jun 30 13:52:11 2014 vlan_name v11+11 mac 00:10:94:00:00:02 was moved from
ge-1/0/22.0 to ge-1/0/21.0 with flags: 0x2101f    << MAC address that was moved
Mon Jun 30 13:54:24 2014 vlan_name v11+11 mac 00:10:94:00:00:02 was changed on
ge-1/0/21.0 with flags: 0x2113f   << MAC address that changed from a dynamic address
 to a static address
```

show ethernet-switching mac-learning-log (SRX Series devices)

`user@host>` **show ethernet-switching mac-learning-log**

```
Wed Mar 18 08:07:05 2009
vlan_idx 7 mac 00:00:5E:00:53:00 was deleted
Wed Mar 18 08:07:05 2009
vlan_idx 9 mac 00:00:5E:00:53:00 was deleted
Wed Mar 18 08:07:05 2009
vlan_idx 10 mac 00:00:5E:00:53:00 was deleted
Wed Mar 18 08:07:05 2009
vlan_idx 11 mac 00:00:5E:00:53:00 was deleted
Wed Mar 18 08:07:05 2009
vlan_idx 12 mac 00:00:5E:00:53:00 was deleted
Wed Mar 18 08:07:05 2009
vlan_idx 13 mac 00:00:5E:00:53:00 was deleted
Wed Mar 18 08:07:05 2009
vlan_idx 14 mac 00:00:5E:00:53:00 was deleted
Wed Mar 18 08:07:05 2009
vlan_idx 15 mac 00:00:5E:00:53:00 was deleted
Wed Mar 18 08:07:05 2009
vlan_idx 16 mac 00:00:5E:00:53:00 was deleted
Wed Mar 18 08:07:05 2009
vlan_idx 4 mac 00:00:5E:00:53:00 was added
Wed Mar 18 08:07:05 2009
vlan_idx 6 mac 00:00:5E:00:53:00 was added
Wed Mar 18 08:07:05 2009
vlan_idx 7 mac 00:00:5E:00:53:00 was added
Wed Mar 18 08:07:05 2009
vlan_idx 9 mac 00:00:5E:00:53:00 was added
Wed Mar 18 08:07:05 2009
vlan_idx 10 mac 00:00:5E:00:53:00 was added
Wed Mar 18 08:07:05 2009
vlan_idx 11 mac 00:00:5E:00:53:00 was added
Wed Mar 18 08:07:05 2009
vlan_idx 12 mac 00:00:5E:00:53:00 was added
Wed Mar 18 08:07:05 2009
vlan_idx 13 mac 00:00:5E:00:53:00 was added
Wed Mar 18 08:07:05 2009
vlan_idx 14 mac 00:00:5E:00:53:00 was added
Wed Mar 18 08:07:05 2009
vlan_idx 15 mac 00:00:5E:00:53:00 was added
Wed Mar 18 08:07:05 2009
vlan_idx 16 mac 00:00:5E:00:53:00 was added
Wed Mar 18 08:07:05 2009
vlan_idx 5 mac 00:00:5E:00:53:00 was added
Wed Mar 18 08:07:05 2009
vlan_idx 18 mac 00:00:5E:00:53:AA was learned
```

```
Wed Mar 18 08:07:05 2009
vlan_idx 5 mac 00:00:5E:00:53:AB was learned
Wed Mar 18 08:07:05 2009
vlan_idx 6 mac 00:00:5E:00:53:AC was learned
Wed Mar 18 08:07:05 2009
vlan_idx 16 mac 00:00:5E:00:53:AD was learned
Wed Mar 18 08:07:05 2009
vlan_idx 7 mac 00:00:5E:00:53:AE was learned
Wed Mar 18 08:07:05 2009
vlan_idx 8 mac 00:00:5E:00:53:AF was learned
Wed Mar 18 08:07:05 2009
vlan_idx 12 mac 00:00:5E:00:53:AG was learned
[output truncated]
```

show ethernet-switching statistics

Syntax

```
show ethernet-switching statistics
<instance instance-name>
<logical-system logical-system-name>
<vlan-name vlan-name>
```

Release Information

Command introduced in Junos OS Release 12.3R2.
Command introduced in Junos OS Release 12.3R2 for EX Series switches.

Description

(MX Series routers, QFX Series switches, and EX Series switches only) Display Ethernet-switching statistics.

Options

none—Display Ethernet-switching statistics for all VLANs in all routing instances.

instance *instance-name*—(Optional) Display statistics for the specified routing instance.

logical-system *logical-system-name*—(Optional) Display Ethernet-switching statistics information for the specified logical system.

vlan-name *vlan-name*—(Optional) Display statistics for the specified VLAN.

Required Privilege Level

view

List of Sample Output

show ethernet-switching statistics on page 1508

Sample Output

show ethernet-switching statistics

`user@host>` **show ethernet-switching statistics**

```
Local interface: ae1.0, Index: 1035
  Broadcast packets:                  220
  Broadcast bytes  :                13720
  Multicast packets:                  130
  Multicast bytes  :                11700
```

```
  Flooded packets  :                    0
  Flooded bytes    :                    0
  Unicast packets  :                    0
  Unicast bytes    :                    0
  Current MAC count:                    0 (Limit 1024)
Local interface: vt-3/3/10.1048576, Index: 1280
  Broadcast packets:                    0
  Broadcast bytes  :                    0
  Multicast packets:                    0
  Multicast bytes  :                    0
  Flooded packets  :                    2
  Flooded bytes    :                  128
  Unicast packets  :                  632
  Unicast bytes    :                39184
  Current MAC count:                    2
Local interface: ge-3/1/2.0, Index: 1258
  Broadcast packets:                  100
  Broadcast bytes  :                 6800
  Multicast packets:                  200
  Multicast bytes  :                18000
  Flooded packets  :                    0
  Flooded bytes    :                    0
  Unicast packets  :                  632
  Unicast bytes    :                39184
  Current MAC count:                    2 (Limit 1024)
Local interface: ae3.0, Index: 1043
  Broadcast packets:                    0
  Broadcast bytes  :                    0
  Multicast packets:                    0
  Multicast bytes  :                    0
  Flooded packets  :                    0
  Flooded bytes    :                    0
  Unicast packets  :                    0
  Unicast bytes    :                    0
  Current MAC count:                    0 (Limit 1024)
Local interface: ge-3/3/8.0, Index: 1276
  Broadcast packets:                    0
  Broadcast bytes  :                    0
  Multicast packets:                    0
  Multicast bytes  :                    0
  Flooded packets  :                    0
  Flooded bytes    :                    0
  Unicast packets  :                    0
  Unicast bytes    :                    0
```

```
  Current MAC count:                    0 (Limit 8192)
Local interface: ae5.0, Index: 1045
  Broadcast packets:                    0
  Broadcast bytes  :                    0
  Multicast packets:                    0
  Multicast bytes  :                    0
  Flooded packets  :                    0
  Flooded bytes    :                    0
  Unicast packets  :                    0
  Unicast bytes    :                    0
  Current MAC count:                    0 (Limit 8192)
Local interface: ae4.0, Index: 1044
  Broadcast packets:                  200
  Broadcast bytes  :                13600
  Multicast packets:                    0
  Multicast bytes  :                    0
  Flooded packets  :                    0
  Flooded bytes    :                    0
  Unicast packets  :                    0
  Unicast bytes    :                    0
  Current MAC count:                    0 (Limit 8192)
Local interface: ae26.0, Index: 1042
  Broadcast packets:                    0
  Broadcast bytes  :                    0
  Multicast packets:                    0
  Multicast bytes  :                    0
  Flooded packets  :                    0
  Flooded bytes    :                    0
  Unicast packets  :                    0
  Unicast bytes    :                    0
  Current MAC count:                    0 (Limit 8192)
Local interface: ae25.0, Index: 1041
  Broadcast packets:                  133
  Broadcast bytes  :                 7980
  Multicast packets:               369934
  Multicast bytes  :             59207572
  Flooded packets  :                    0
  Flooded bytes    :                    0
  Unicast packets  :                 1433
  Unicast bytes    :               119930
  Current MAC count:                    3 (Limit 8192)
Local interface: ae23.0, Index: 1040
  Broadcast packets:                  226
  Broadcast bytes  :                14464
```

```
  Multicast packets:               585668
  Multicast bytes  :            153464476
  Flooded packets  :                    0
  Flooded bytes    :                    0
  Unicast packets  :                26552
  Unicast bytes    :              1947627
  Current MAC count:                    7 (Limit 8192)
Local interface: ae20.0, Index: 1039
  Broadcast packets:                  115
  Broadcast bytes  :                 6900
  Multicast packets:               395113
  Multicast bytes  :             61622869
  Flooded packets  :                    0
  Flooded bytes    :                    0
  Unicast packets  :                 1419
  Unicast bytes    :               117924
  Current MAC count:                    4 (Limit 8192)
```

show ethernet-switching statistics aging

Syntax

```
show ethernet-switching statistics aging
```

Release Information

Command introduced in Junos OS Release 9.4 for EX Series switches.

Description

Display media access control (MAC) aging statistics.

Options

none—(Optional) Display MAC aging statistics.

brief | detail—(Optional) Display the specified level of output.

Required Privilege Level

view

RELATED DOCUMENTATION

show ethernet-switching statistics mac-learning | 1514

Configuring MAC Table Aging on Switches | 137

List of Sample Output

show ethernet-switching statistics aging on page 1513

Output Fields

Table 152 on page 1512 lists the output fields for the **show ethernet-switching statistics aging** command. Output fields are listed in the approximate order in which they appear.

Table 152: show ethernet-switching statistics aging Output Fields

Field Name	Field Description	Level of Output
Total age messages received	Total number of aging messages received from the hardware.	All levels
Immediate aging	Aging message indicating that the entry should be removed immediately.	All levels
MAC address seen	Aging message indicating that the MAC address has been detected by hardware and that the aging timer should be stopped.	All levels

Table 152: show ethernet-switching statistics aging Output Fields *(continued)*

Field Name	Field Description	Level of Output
MAC address not seen	Aging message indicating that the MAC address has not been detected by the hardware and that the aging timer should be started.	All levels
Error age messages	The received aging message contains the following errors: • **Invalid VLAN**—The VLAN of the packet does not exist. • **No such entry**—The MAC address and VLAN pair provided by the aging message does not exist. • **Static entry**—An unsuccessful attempt was made to age out a static MAC entry.	All levels

Sample Output

show ethernet-switching statistics aging

`user@switch>` **show ethernet-switching statistics aging**

```
Total age messages received: 0
  Immediate aging: 0, MAC address seen: 0, MAC address not seen: 0
Error age messages: 0
  Invalid VLAN: 0, No such entry: 0, Static entry: 0
```

show ethernet-switching statistics mac-learning

Syntax

```
show ethernet-switching statistics mac-learning
<brief | detail>
<interface interface-name>
```

Release Information

Command introduced in Junos OS Release 9.4 for EX Series switches.
Command introduced in Junos OS Release 11.1 for the QFX Series.

Description

Display media access control (MAC) learning statistics.

NOTE: For the QFX Series, this command is not supported in Enhanced Layer 2 Software (ELS).

Options

none—(Optional) Display MAC learning statistics for all interfaces.

brief | detail—(Optional) Display the specified level of output. The default is **brief**.

interface *interface-name*—(Optional) Display MAC learning statistics for the specified interface.

Required Privilege Level

view

RELATED DOCUMENTATION

- show ethernet-switching table | 1520
- show ethernet-switching interfaces | 1485
- show ethernet-switching mac-learning-log | 1502
- show ethernet-switching table | 1520
- show ethernet-switching interfaces | 1485
- Example: Setting Up Basic Bridging and a VLAN on Switches | 203

List of Sample Output

show ethernet-switching statistics mac-learning on page 1516

show ethernet-switching statistics mac-learning detail on page 1517
show ethernet-switching statistics mac-learning interface ge-0/0/28 detail on page 1517
show ethernet-switching statistics mac-learning interface on page 1518
show ethernet-switching statistics mac-learning detail (QFX Series) on page 1518

Output Fields

Table 153 on page 1515 lists the output fields for the **show ethernet-switching statistics mac-learning** command. Output fields are listed in the approximate order in which they appear.

Table 153: show ethernet-switching statistics mac-learning Output Fields

Field Name	Field Description	Level of Output
Interface	Name of the interface for which statistics are being reported. (Displayed in the output under the heading **Interface**.)	All levels
Learning message from local packets	MAC learning message generated due to packets coming in on the management interface. (Displayed in the output under the heading **Local pkts**.)	All levels
Learning message from transit packets	MAC learning message generated due to packets coming in on network interfaces. (Displayed in the output under the heading **Transit pkts**.)	All levels

Table 153: show ethernet-switching statistics mac-learning Output Fields *(continued)*

Field Name	Field Description	Level of Output
Learning message with error	MAC learning messages received with errors (Displayed under the heading **Error**): • **Invalid VLAN**—The VLAN of the packet does not exist. • **Invalid MAC**—The MAC address is either NULL or a multicast MAC address. • **Security violation**—The MAC address is not an allowed MAC address. • **Interface down**—The MAC address is learned on an interface that is down. • **Incorrect membership**—The MAC address is learned on an interface that is not a member of the VLAN. • **Interface limit**—The number of MAC addresses learned on the interface has exceeded the limit. • **MAC move limit**—This MAC address has moved among multiple interfaces too many times in a given interval. • **VLAN limit**—The number of MAC addresses learned on the VLAN has exceeded the limit. • **VLAN membership limit**—The number of MAC addresses learned on the interface as a member of the specified VLAN (VLAN membership MAC limit) has exceeded the limit. • **Invalid VLAN index**—The VLAN of the packet, although configured, does not yet exist in the kernel. • **Interface not learning**—The MAC address is learned on an interface that does not yet allow learning—for example, the interface is blocked. • **No nexthop**—The MAC address is learned on an interface that does not have a unicast next hop. • **MAC learning disabled**—The MAC address is learned on an interface on which MAC learning has been disabled. • **Others**—The message contains some other error.	All levels

Sample Output

show ethernet-switching statistics mac-learning

```
user@switch> show ethernet-switching statistics mac-learning
```

```
Learning stats: 0 learn msg rcvd, 0 error
  Interface         Local pkts        Transit pkts      Error
  ge-0/0/0.0        0                 0                 0
  ge-0/0/1.0        0                 0                 0
  ge-0/0/2.0        0                 0                 0
  ge-0/0/3.0        0                 0                 0
```

show ethernet-switching statistics mac-learning detail

user@switch> **show ethernet-switching statistics mac-learning detail**

```
Learning stats: 0 learn msg rcvd, 0 error

    Interface: ge-0/0/0.0
    Learning message from local packets:   0
    Learning message from transit packets: 1
    Learning message with error:           0
      Invalid VLAN:               0       Invalid MAC:                0
      Security violation:         0       Interface down:             0
      Incorrect membership:       0       Interface limit:            0
      MAC move limit:             0       VLAN limit:                 0
      Invalid VLAN index:         0       Interface not learning:     0
      No nexthop:                 0       MAC learning disabled:      0
      Others:                     0

    Interface: ge-0/0/1.0
    Learning message from local packets:   0
    Learning message from transit packets: 2
    Learning message with error:           0
      Invalid VLAN:               0       Invalid MAC:                0
      Security violation:         0       Interface down:             0
      Incorrect membership:       0       Interface limit:            0
      MAC move limit:             0       VLAN limit:                 0
      Invalid VLAN index:         0       Interface not learning:     0
      No nexthop:                 0       MAC learning disabled:      0
      Others:                     0
```

show ethernet-switching statistics mac-learning interface ge-0/0/28 detail

user@switch> **show ethernet-switching statistics mac-learning interface ge-0/0/28 detail**

```
    Interface: ge-0/0/28.0
    Learning message from local packets:   0
    Learning message from transit packets: 5
    Learning message with error:           0
      Invalid VLAN:               0       Invalid MAC:                  0
      Security violation:         0       Interface down:               0
      Incorrect membership:       0       Interface limit:              0
      MAC move limit:             0       VLAN limit:                   0
                                          VLAN membership limit:        20
      Invalid VLAN index:         0       Interface not learning:       0
      No nexthop:                 0       MAC learning disabled:        0
      Others:                     0
```

show ethernet-switching statistics mac-learning interface

`user@switch>` **show ethernet-switching statistics mac-learning interface ge-0/0/1**

```
Interface         Local pkts        Transit pkts      Error
ge-0/0/1.0        0                 1                 1
```

show ethernet-switching statistics mac-learning detail (QFX Series)

`user@switch>` **show ethernet-switching statistics mac-learning detail**

```
Learning stats: 0 learn msg rcvd, 0 error

    Interface: xe-0/0/0.0
    Learning message from local packets:   0
    Learning message from transit packets: 1
    Learning message with error:           0
      Invalid VLAN:               0       Invalid MAC:                  0
      Security violation:         0       Interface down:               0
      Incorrect membership:       0       Interface limit:              0
      MAC move limit:             0       VLAN limit:                   0
      Invalid VLAN index:         0       Interface not learning:       0
      No nexthop:                 0       MAC learning disabled:        0
      Others:                     0

    Interface: xe-0/0/1.0
    Learning message from local packets:   0
    Learning message from transit packets: 2
    Learning message with error:           0
```

```
Invalid VLAN:              0       Invalid MAC:                0
Security violation:        0       Interface down:             0
Incorrect membership:      0       Interface limit:            0
MAC move limit:            0       VLAN limit:                 0
Invalid VLAN index:        0       Interface not learning:     0
No nexthop:                0       MAC learning disabled:      0
Others:                    0
```

show ethernet-switching table

List of Syntax

Syntax (QFX Series, QFabric, NFX Series and EX4600)

```
show ethernet-switching table
<brief | detail | extensive | summary>
<interface interface-name>
<management-vlan>
<sort-by (name | tag)>
<vlan vlan-name>
```

Syntax (EX Series)

```
show ethernet-switching table
<brief | detail | extensive | summary>
<interface interface-name>
<management-vlan>
<persistent-mac <interface interface-name>>
<sort-by (name | tag)>
<vlan vlan-name>
```

Syntax (EX Series, MX Series and QFX Series)

```
show ethernet-switching table
<brief | count | detail | extensive | summary>
<address>
<instance instance-name>
<interface interface-name>
isid isid
<logical-system logical-system-name>
<persistent-learning (interface interface-name | mac mac-address)>
<address>
<vlan-id (all-vlan | vlan-id)>
<vlan-name (all | vlan-name)>
```

Syntax (SRX Series)

show ethernet-switching table (brief |detail |extensive) interface *interface-name*

Release Information

Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 9.5 for SRX Series.
Options **summary**, **management-vlan**, and **vlan *vlan-name*** introduced in Junos OS Release 9.6 for EX Series switches.
Option **sort-by** and field name **tag** introduced in Junos OS Release 10.1 for EX Series switches.
Command introduced in Junos OS Release 11.1 for the QFX Series.
Output for private VLANs introduced in Junos OS Release 12.1 for the QFX Series.
Option **persistent-mac** introduced in Junos OS Release 11.4 for EX Series switches.
Command introduced in Junos OS Release 12.3R2.
Command introduced in Junos OS Release 12.3R2 for EX Series switches.
Options **logical-system**, **persistent-learning**, and **summary** introduced in Junos OS Release 13.2X50-D10 (ELS).

Description

Displays the Ethernet switching table.

(MX Series routers,EX Series switches only) Displays Layer 2 MAC address information.

Options

For QFX Series, QFabric, NFX Series and EX4600:

none—(Optional) Display brief information about the Ethernet switching table.

brief | detail | extensive | summary—(Optional) Display the specified level of output.

interface *interface-name*—(Optional) Display the Ethernet switching table for a specific interface.

management-vlan—(Optional) Display the Ethernet switching table for a management VLAN.

persistent-mac <interface *interface-name*>—(Optional) Display the persistent MAC addresses learned for all interfaces or a specified interface. You can use this command to view entries that you want to clear for an interface that you intentionally disabled.

sort-by (*name* | *tag*)—(Optional) Display VLANs in ascending order of VLAN IDs or VLAN names.

vlan *vlan-name*—(Optional) Display the Ethernet switching table for a specific VLAN.

For EX Series, MX Series and QFX Series:

none—Display all learned Layer 2 MAC address information.

brief | count | detail | extensive | summary—(Optional) Display the specified level of output.

address—(Optional) Display the specified learned Layer 2 MAC address information.

instance *instance-name*—(Optional) Display learned Layer 2 MAC addresses for the specified routing instance.

interface *interface-name*—(Optional) Display learned Layer 2 MAC addresses for the specified interface.

isid *isid*—(Optional) Display learned Layer 2 MAC addresses for the specified ISID.

logical-system *logical-system-name*—(Optional) Display Ethernet-switching statistics information for the specified logical system.

persistent-learning (interface *interface-name* | mac *mac-address*)—(Optional) Display dynamically learned MAC addresses that are retained despite device restarts and interface failures for a specified interface, or information about a specified MAC address.

vlan-id (all-vlan | *vlan-id*)—(Optional) Display learned Layer 2 MAC addresses for all VLANs or for the specified VLAN.

vlan-name (all | *vlan-name*)—(Optional) Display learned Layer 2 MAC addresses for all VLANs or for the specified VLAN.

For SRX Series:

- **none**—(Optional) Display brief information about the Ethernet switching table.
- **brief** | **detail** | **extensive**—(Optional) Display the specified level of output.
- **interface-name**—(Optional) Display the Ethernet switching table for a specific interface.

Additional Information

When Layer 2 protocol tunneling is enabled, the tunneling MAC address 01:00:0c:cd:cd:d0 is installed in the MAC table. When the Cisco Discovery Protocol (CDP), Spanning Tree Protocol (STP), or VLAN Trunk Protocol (VTP) is configured for Layer 2 protocol tunneling on an interface, the corresponding protocol MAC address is installed in the MAC table.

Required Privilege Level

view

RELATED DOCUMENTATION

Example: Setting Up Basic Bridging and a VLAN on Switches | **203**

Example: Setting Up Bridging with Multiple VLANs | **236**

Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch | **226**

Example: Setting Up Bridging with Multiple VLANs for EX Series Switches | **265**

Example: Setting Up Q-in-Q Tunneling on EX Series Switches | **925**

clear ethernet-switching table | **1444**

show ethernet-switching mac-learning-log | 1502

List of Sample Output

Output Fields

For QFX Series, QFabric, NFX Series and EX4600:

The following table lists the output fields for the **show ethernet-switching table** command on QFX Series, QFabric, NFX Series and EX4600. Output fields are listed in the approximate order in which they appear.

Table 154: show ethernet-switching table Output Fields

Field Name	Field Description	Level of Output
VLAN	Name of a VLAN.	All levels
MAC address	MAC address associated with the VLAN.	All levels
Type	Type of MAC address: • **static**—The MAC address is manually created. • **learn**—The MAC address is learned dynamically from a packet's source MAC address. • **flood**—The MAC address is unknown and flooded to all members.	All levels

Table 154: show ethernet-switching table Output Fields *(continued)*

Field Name	Field Description	Level of Output
Age	Time remaining before the entry ages out and is removed from the Ethernet switching table.	All levels
Interfaces	Interface associated with learned MAC addresses or with the **All-members** option (flood entry).	All levels
Learned	For learned entries, the time at which the entry was added to the Ethernet switching table.	**detail**, **extensive**

For EX Series switches:

The following table lists the output fields for the **show ethernet-switching table** command on EX Series switches. Output fields are listed in the approximate order in which they appear.

Table 155: show ethernet-switching table Output Fields

Field Name	Field Description	Level of Output
VLAN	The name of a VLAN.	All levels
Tag	The VLAN ID tag name or number.	**extensive**
MAC or **MAC address**	The MAC address associated with the VLAN.	All levels
Type	The type of MAC address. Values are: • **static**—The MAC address is manually created. • **learn**—The MAC address is learned dynamically from a packet's source MAC address. • **flood**—The MAC address is unknown and flooded to all members. • **persistent**—The learned MAC addresses that will persist across restarts of the switch or interface-down events.	All levels except **persistent-mac**
Type	The type of MAC address. Values are: • **installed**—addresses that are in the Ethernet switching table. • **uninstalled**—addresses that could not be installed in the table or were uninstalled in an interface-down event and will be reinstalled in the table when the interface comes back up.	**persistent-mac**
Age	The time remaining before the entry ages out and is removed from the Ethernet switching table.	All levels

Table 155: show ethernet-switching table Output Fields *(continued)*

Field Name	Field Description	Level of Output
Interfaces	Interface associated with learned MAC addresses or **All-members** (flood entry).	All levels
Learned	For learned entries, the time which the entry was added to the Ethernet switching table.	**detail**, **extensive**
Nexthop index	The next-hop index number.	**detail**, **extensive**
persistent-mac	**installed** indicates MAC addresses that are in the Ethernet switching table and **uninstalled** indicates MAC addresses that could not be installed in the table or were uninstalled in an interface-down event (and will be reinstalled in the table when the interface comes back up).	

For EX Series, MX Series and QFX Series:

The table describes the output fields for the **show ethernet-switching table** command on EX Series, MX Series and QFX Series. Output fields are listed in the approximate order in which they appear.

Table 156: show ethernet-switching table Output fields

Field Name	Field Description
Routing instance	Name of the routing instance.
VLAN name	Name of the VLAN.
MAC address	MAC address or addresses learned on a logical interface.
MAC flags	Status of MAC address learning properties for each interface: • **S**—Static MAC address is configured. • **D**—Dynamic MAC address is configured. • **L**—Locally learned MAC address is configured. • **SE**—MAC accounting is enabled. • **NM**—Non-configured MAC. • **R**—Locally learned MAC address is configured.
Age	This field is not supported.
Logical interface	Name of the logical interface.
Active source	IP address of remote entity on which MAC address is learned.

Table 156: show ethernet-switching table Output fields *(continued)*

Field Name	Field Description
MAC count	Number of MAC addresses learned on the specific routing instance or interface.
Learning interface	Name of the logical interface on which the MAC address was learned.
Learning VLAN	VLAN ID of the routing instance or VLAN in which the MAC address was learned.
Layer 2 flags	Debugging flags signifying that the MAC address is present in various lists.
Epoch	Spanning-tree-protocolepoch number identifying when the MAC address was learned. Used for debugging.
Sequence number	Sequence number assigned to this MAC address. Used for debugging.
Learning mask	Mask of the Packet Forwarding Engines where this MAC address was learned. Used for debugging.
IPC generation	Creation time of the logical interface when this MAC address was learned. Used for debugging.

For SRX Series:

Table 157 on page 1526 lists the output fields for the **show ethernet-switching table** command. Output fields are listed in the approximate order in which they appear.

Table 157: show ethernet-switching table Output Fields

Field Name	Field Description
VLAN	The name of a VLAN.
MAC address	The MAC address associated with the VLAN.
Type	The type of MAC address. Values are: • static—The MAC address is manually created. • learn—The MAC address is learned dynamically from a packet's source MAC address. • flood—The MAC address is unknown and flooded to all members.

Table 157: show ethernet-switching table Output Fields *(continued)*

Field Name	Field Description
Age	The time remaining before the entry ages out and is removed from the Ethernet switching table.
Interfaces	Interface associated with learned MAC addresses or All-members (flood entry).
Learned	For learned entries, the time which the entry was added to the Ethernet switching table.

Sample Output

show ethernet-switching table (Enhanced Layer 2 Software on QFX Series, QFabric, NFX Series and EX460)

`user@switch>` **show ethernet-switching table**

```
MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent
static
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC, O
 - ovsdb MAC)


Ethernet switching table : 2 entries, 2 learned
Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags              interface
    vlan1               b0:c6:9a:ca:3c:01   D             -    ae1.0

    vlan1               b0:c6:9a:ca:3c:03   D             -    ae1.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent
static
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC, O
 - ovsdb MAC)

Ethernet switching table : 2 entries, 2 learned

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
```

```
    name                address             flags              interface
    vlan10              b0:c6:9a:ca:3c:01   D             -    ae1.0

    vlan10              b0:c6:9a:ca:3c:03   D             -    ae1.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent
static
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC, O
 - ovsdb MAC)


Ethernet switching table : 2 entries, 2 learned
Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags              interface
    vlan2               b0:c6:9a:ca:3c:01   D             -    ae1.0

    vlan2               b0:c6:9a:ca:3c:03   D             -    ae1.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent
static
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC, O
 - ovsdb MAC)


Ethernet switching table : 2 entries, 2 learned
Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags              interface
    vlan3               b0:c6:9a:ca:3c:01   D             -    ae1.0

    vlan3               b0:c6:9a:ca:3c:03   D             -    ae1.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent
static
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC, O
 - ovsdb MAC)


Ethernet switching table : 2 entries, 2 learned
Routing instance : default-switch
```

```
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags               interface
    vlan4               b0:c6:9a:ca:3c:01   D             -   ae1.0

    vlan4               b0:c6:9a:ca:3c:03   D             -   ae1.0
```

show ethernet-switching table (QFX Series, QFabric, NFX Series and EX460)

`user@switch>` **show ethernet-switching table**

```
Ethernet-switching table: 57 entries, 17 learned
  VLAN              MAC address       Type         Age Interfaces
  F2                *                 Flood          - All-members
  F2                00:00:05:00:00:03 Learn          0 xe-0/0/44.0
  F2                00:19:e2:50:7d:e0 Static         - Router
  Linux             *                 Flood          - All-members
  Linux             00:19:e2:50:7d:e0 Static         - Router
  Linux             00:30:48:90:54:89 Learn          0 xe-0/0/47.0
  T1                *                 Flood          - All-members
  T1                00:00:05:00:00:01 Learn          0 xe-0/0/46.0
  T1                00:00:5e:00:01:00 Static         - Router
  T1                00:19:e2:50:63:e0 Learn          0 xe-0/0/46.0
  T1                00:19:e2:50:7d:e0 Static         - Router
  T10               *                 Flood          - All-members
  T10               00:00:5e:00:01:09 Static         - Router
  T10               00:19:e2:50:63:e0 Learn          0 xe-0/0/46.0
  T10               00:19:e2:50:7d:e0 Static         - Router
  T111              *                 Flood          - All-members
  T111              00:19:e2:50:63:e0 Learn          0 xe-0/0/15.0
  T111              00:19:e2:50:7d:e0 Static         - Router
  T111              00:19:e2:50:ac:00 Learn          0 xe-0/0/15.0
  T2                *                 Flood          - All-members
  T2                00:00:5e:00:01:01 Static         - Router
  T2                00:19:e2:50:63:e0 Learn          0 xe-0/0/46.0
  T2                00:19:e2:50:7d:e0 Static         - Router
  T3                *                 Flood          - All-members
  T3                00:00:5e:00:01:02 Static         - Router
  T3                00:19:e2:50:63:e0 Learn          0 xe-0/0/46.0
  T3                00:19:e2:50:7d:e0 Static         - Router
  T4                *                 Flood          - All-members
  T4                00:00:5e:00:01:03 Static         - Router
  T4                00:19:e2:50:63:e0 Learn          0 xe-0/0/46.0
[output truncated]
```

show ethernet-switching table (Private VLANs on QFX Series, QFabric, NFX Series and EX460)

`user@switch>` **show ethernet-switching table**

```
Ethernet-switching table: 10 entries, 3 learned
  VLAN              MAC address       Type         Age Interfaces
  pvlan             *                 Flood          - All-members
  pvlan             00:10:94:00:00:02 Replicated        - xe-0/0/28.0
  pvlan             00:10:94:00:00:35 Replicated        - xe-0/0/46.0
  pvlan             00:10:94:00:00:46 Replicated        - xe-0/0/4.0
  c2                *                 Flood          - All-members
  c2                00:10:94:00:00:02 Learn          0 xe-0/0/28.0
  c1                *                 Flood          - All-members
  c1                00:10:94:00:00:46 Learn          0 xe-0/0/4.0
  __pvlan_pvlan_xe-0/0/46.0__ *       Flood          - All-members
  __pvlan_pvlan_xe-0/0/46.0__ 00:10:94:00:00:35 Learn        0 xe-0/0/46.0
```

show ethernet-switching table brief (QFX Series, QFabric, NFX Series and EX460)

`user@switch>` **show ethernet-switching table brief**

```
Ethernet-switching table: 57 entries, 17 learned
  VLAN              MAC address       Type         Age Interfaces
  F2                *                 Flood          - All-members
  F2                00:00:05:00:00:03 Learn          0 xe-0/0/44.0
  F2                00:19:e2:50:7d:e0 Static         - Router
  Linux             *                 Flood          - All-members
  Linux             00:19:e2:50:7d:e0 Static         - Router
  Linux             00:30:48:90:54:89 Learn          0 xe-0/0/47.0
  T1                *                 Flood          - All-members
  T1                00:00:05:00:00:01 Learn          0 xe-0/0/46.0
  T1                00:00:5e:00:01:00 Static         - Router
  T1                00:19:e2:50:63:e0 Learn          0 xe-0/0/46.0
  T1                00:19:e2:50:7d:e0 Static         - Router
  T10               *                 Flood          - All-members
  T10               00:00:5e:00:01:09 Static         - Router
  T10               00:19:e2:50:63:e0 Learn          0 xe-0/0/46.0
  T10               00:19:e2:50:7d:e0 Static         - Router
  T111              *                 Flood          - All-members
  T111              00:19:e2:50:63:e0 Learn          0 xe-0/0/15.0
  T111              00:19:e2:50:7d:e0 Static         - Router
  T111              00:19:e2:50:ac:00 Learn          0 xe-0/0/15.0
  T2                *                 Flood          - All-members
  T2                00:00:5e:00:01:01 Static         - Router
  T2                00:19:e2:50:63:e0 Learn          0 xe-0/0/46.0
```

```
  T2                00:19:e2:50:7d:e0 Static         - Router
  T3                *                 Flood          - All-members
  T3                00:00:5e:00:01:02 Static         - Router
  T3                00:19:e2:50:63:e0 Learn          0 xe-0/0/46.0
  T3                00:19:e2:50:7d:e0 Static         - Router
  T4                *                 Flood          - All-members
  T4                00:00:5e:00:01:03 Static         - Router
  T4                00:19:e2:50:63:e0 Learn          0 xe-0/0/46.0
[output truncated]
```

show ethernet-switching table detail (QFX Series, QFabric, NFX Series and EX460)

`user@switch>` **show ethernet-switching table detail**

```
Ethernet-switching table: 57 entries, 17 learned
  F2, *
    Interface(s): xe-0/0/44.0
    Type: Flood
    Nexthop index: 0

  F2, 00:00:05:00:00:03
    Interface(s): xe-0/0/44.0
    Type: Learn, Age: 0, Learned: 2:03:09
    Nexthop index: 0

  F2, 00:19:e2:50:7d:e0
    Interface(s): Router
    Type: Static
    Nexthop index: 0

  Linux, *
    Interface(s): xe-0/0/47.0
    Type: Flood
    Nexthop index: 0

  Linux, 00:19:e2:50:7d:e0
    Interface(s): Router
    Type: Static
    Nexthop index: 0

  Linux, 00:30:48:90:54:89
    Interface(s): xe-0/0/47.0
    Type: Learn, Age: 0, Learned: 2:03:08
    Nexthop index: 0
```

```
T1, *
  Interface(s): xe-0/0/46.0
  Type: Flood
  Nexthop index: 0

T1, 00:00:05:00:00:01
  Interface(s): xe-0/0/46.0
  Type: Learn, Age: 0, Learned: 2:03:07
  Nexthop index: 0

T1, 00:00:5e:00:01:00
  Interface(s): Router
  Type: Static
  Nexthop index: 0

T1, 00:19:e2:50:63:e0
  Interface(s): xe-0/0/46.0
  Type: Learn, Age: 0, Learned: 2:03:07
  Nexthop index: 0

T1, 00:19:e2:50:7d:e0
  Interface(s): Router
  Type: Static
  Nexthop index: 0

T10, *
  Interface(s): xe-0/0/46.0
  Type: Flood
  Nexthop index: 0

T10, 00:00:5e:00:01:09
  Interface(s): Router
  Type: Static
  Nexthop index: 0

T10, 00:19:e2:50:63:e0
  Interface(s): xe-0/0/46.0
  Type: Learn, Age: 0, Learned: 2:03:08
  Nexthop index: 0

T10, 00:19:e2:50:7d:e0
  Interface(s): Router
  Type: Static
```

```
    Nexthop index: 0

  T111, *
    Interface(s): xe-0/0/15.0
    Type: Flood
    Nexthop index: 0
[output truncated]
```

show ethernet-switching table extensive (QFX Series, QFabric, NFX Series and EX460)

`user@switch>` **show ethernet-switching table extensive**

```
Ethernet-switching table: 57 entries, 17 learned
  F2, *
    Interface(s): xe-0/0/44.0
    Type: Flood
    Nexthop index: 0

  F2, 00:00:05:00:00:03
    Interface(s): xe-0/0/44.0
    Type: Learn, Age: 0, Learned: 2:03:09
    Nexthop index: 0

  F2, 00:19:e2:50:7d:e0
    Interface(s): Router
    Type: Static
    Nexthop index: 0

  Linux, *
    Interface(s): xe-0/0/47.0
    Type: Flood
    Nexthop index: 0

  Linux, 00:19:e2:50:7d:e0
    Interface(s): Router
    Type: Static
    Nexthop index: 0

  Linux, 00:30:48:90:54:89
    Interface(s): xe-0/0/47.0
    Type: Learn, Age: 0, Learned: 2:03:08
    Nexthop index: 0

  T1, *
```

```
  Interface(s): xe-0/0/46.0
  Type: Flood
  Nexthop index: 0

T1, 00:00:05:00:00:01
  Interface(s): xe-0/0/46.0
  Type: Learn, Age: 0, Learned: 2:03:07
  Nexthop index: 0

T1, 00:00:5e:00:01:00
  Interface(s): Router
  Type: Static
  Nexthop index: 0

T1, 00:19:e2:50:63:e0
  Interface(s): xe-0/0/46.0
  Type: Learn, Age: 0, Learned: 2:03:07
  Nexthop index: 0

T1, 00:19:e2:50:7d:e0
  Interface(s): Router
  Type: Static
  Nexthop index: 0

T10, *
  Interface(s): xe-0/0/46.0
  Type: Flood
  Nexthop index: 0

T10, 00:00:5e:00:01:09
  Interface(s): Router
  Type: Static
  Nexthop index: 0

T10, 00:19:e2:50:63:e0
  Interface(s): xe-0/0/46.0
  Type: Learn, Age: 0, Learned: 2:03:08
  Nexthop index: 0

T10, 00:19:e2:50:7d:e0
  Interface(s): Router
  Type: Static
  Nexthop index: 0
```

```
  T111, *
    Interface(s): xe-0/0/15.0
    Type: Flood
    Nexthop index: 0
[output truncated]
```

show ethernet-switching table interface (QFX Series, QFabric, NFX Series and EX460)

`user@switch>` **show ethernet-switching table interface xe-0/0/1**

```
Ethernet-switching table: 1 unicast entries
  VLAN              MAC address       Type         Age Interfaces
  V1                *                 Flood          - All-members
  V1                00:00:05:00:00:05 Learn          0 xe-0/0/1.0
```

show ethernet-switching table (EX Series switches)

`user@switch>` **show ethernet-switching table**

```
Ethernet-switching table: 57 entries, 15 learned, 2 persistent
  VLAN              MAC address       Type         Age Interfaces
  F2                *                 Flood          - All-members
  F2                00:00:05:00:00:03 Learn          0 ge-0/0/44.0
  F2                00:19:e2:50:7d:e0 Static         - Router
  Linux             *                 Flood          - All-members
  Linux             00:19:e2:50:7d:e0 Static         - Router
  Linux             00:30:48:90:54:89 Learn          0 ge-0/0/47.0
  T1                *                 Flood          - All-members
  T1                00:00:05:00:00:01 Persistent     0 ge-0/0/46.0
  T1                00:00:5e:00:01:00 Static         - Router
  T1                00:19:e2:50:63:e0 Persistent     0 ge-0/0/46.0
  T1                00:19:e2:50:7d:e0 Static         - Router
  T10               *                 Flood          - All-members
  T10               00:00:5e:00:01:09 Static         - Router
  T10               00:19:e2:50:63:e0 Learn          0 ge-0/0/46.0
  T10               00:19:e2:50:7d:e0 Static         - Router
  T111              *                 Flood          - All-members
  T111              00:19:e2:50:63:e0 Learn          0 ge-0/0/15.0
  T111              00:19:e2:50:7d:e0 Static         - Router
  T111              00:19:e2:50:ac:00 Learn          0 ge-0/0/15.0
  T2                *                 Flood          - All-members
  T2                00:00:5e:00:01:01 Static         - Router
  T2                00:19:e2:50:63:e0 Learn          0 ge-0/0/46.0
```

```
  T2                00:19:e2:50:7d:e0 Static         - Router
  T3                *                 Flood          - All-members
  T3                00:00:5e:00:01:02 Static         - Router
  T3                00:19:e2:50:63:e0 Learn          0 ge-0/0/46.0
  T3                00:19:e2:50:7d:e0 Static         - Router
  T4                *                 Flood          - All-members
  T4                00:00:5e:00:01:03 Static         - Router
  T4                00:19:e2:50:63:e0 Learn          0 ge-0/0/46.0
[output truncated]
```

show ethernet-switching table brief (EX Series switches)

`user@switch>` **show ethernet-switching table brief**

```
Ethernet-switching table: 57 entries, 15 learned, 2 persistent entries
  VLAN              MAC address       Type         Age Interfaces
  F2                *                 Flood          - All-members
  F2                00:00:05:00:00:03 Learn          0 ge-0/0/44.0
  F2                00:19:e2:50:7d:e0 Static         - Router
  Linux             *                 Flood          - All-members
  Linux             00:19:e2:50:7d:e0 Static         - Router
  Linux             00:30:48:90:54:89 Learn          0 ge-0/0/47.0
  T1                *                 Flood          - All-members
  T1                00:00:05:00:00:01 Persistent     0 ge-0/0/46.0
  T1                00:00:5e:00:01:00 Static         - Router
  T1                00:19:e2:50:63:e0 Persistent     0 ge-0/0/46.0
  T1                00:19:e2:50:7d:e0 Static         - Router
  T10               *                 Flood          - All-members
  T10               00:00:5e:00:01:09 Static         - Router
  T10               00:19:e2:50:63:e0 Learn          0 ge-0/0/46.0
  T10               00:19:e2:50:7d:e0 Static         - Router
  T111              *                 Flood          - All-members
  T111              00:19:e2:50:63:e0 Learn          0 ge-0/0/15.0
  T111              00:19:e2:50:7d:e0 Static         - Router
  T111              00:19:e2:50:ac:00 Learn          0 ge-0/0/15.0
  T2                *                 Flood          - All-members
  T2                00:00:5e:00:01:01 Static         - Router
  T2                00:19:e2:50:63:e0 Learn          0 ge-0/0/46.0
  T2                00:19:e2:50:7d:e0 Static         - Router
  T3                *                 Flood          - All-members
  T3                00:00:5e:00:01:02 Static         - Router
  T3                00:19:e2:50:63:e0 Learn          0 ge-0/0/46.0
  T3                00:19:e2:50:7d:e0 Static         - Router
  T4                *                 Flood          - All-members
```

```
  T4                00:00:5e:00:01:03 Static         - Router
  T4                00:19:e2:50:63:e0 Learn          0 ge-0/0/46.0
[output truncated]
```

show ethernet-switching table detail (EX Series switches)

`user@switch>` **show ethernet-switching table detail**

```
Ethernet-switching table: 5 entries, 2 learned entries
 VLAN: default, Tag: 0, MAC: *, Interface: All-members
  Interfaces:
   ge-0/0/11.0, ge-0/0/20.0, ge-0/0/30.0, ge-0/0/36.0, ge-0/0/3.0
  Type: Flood
  Nexthop index: 1307

 VLAN: default, Tag: 0, MAC: 00:1f:12:30:b8:83, Interface: ge-0/0/3.0
   Type: Learn, Age: 0, Learned: 20:09:26
   Nexthop index: 1315

 VLAN: v1, Tag: 101, MAC: *, Interface: All-members
  Interfaces:
   ge-0/0/31.0
  Type: Flood
  Nexthop index: 1313

 VLAN: v1, Tag: 101, MAC: 00:1f:12:30:b8:89, Interface: ge-0/0/31.0
  Type: Learn, Age: 0, Learned: 20:09:25
  Nexthop index: 1312

 VLAN: v2, Tag: 102, MAC: *, Interface: All-members
  Interfaces:
   ae0.0
  Type: Flood
  Nexthop index: 1317
```

show ethernet-switching table extensive (EX Series switches)

`user@switch>` **show ethernet-switching table extensive**

```
Ethernet-switching table: 3 entries, 1 learned, 5 persistent entries

    VLAN: v1, Tag: 10, MAC: *, Interface: All-members
    Interfaces:
```

```
                ge-0/0/14.0, ge-0/0/1.0, ge-0/0/2.0, ge-0/0/3.0, ge-0/0/4.0,
                ge-0/0/5.0, ge-0/0/6.0, ge-0/0/7.0, ge-0/0/8.0, ge-0/0/10.0,
                ge-0/0/0.0
    Type: Flood
    Nexthop index: 567

    VLAN: v1, Tag: 10, MAC: 00:21:59:c6:93:22, Interface: Router
    Type: Static
    Nexthop index: 0

    VLAN: v1, Tag: 10, MAC: 00:21:59:c9:9a:4e, Interface: ge-0/0/14.0
    Type: Learn, Age: 0, Learned: 18:40:50
    Nexthop index: 564
```

show ethernet-switching table persistent-mac (EX Series switches)

`user@switch>` **show ethernet-switching table persistent-mac**

```
VLAN              MAC address       Type       Interface
default           00:10:94:00:00:02 installed    ge-0/0/42.0
default           00:10:94:00:00:03 installed    ge-0/0/42.0
default           00:10:94:00:00:04 installed    ge-0/0/42.0
default           00:10:94:00:00:05 installed    ge-0/0/42.0
default           00:10:94:00:00:06 installed    ge-0/0/42.0
default           00:10:94:00:05:02 uninstalled ge-0/0/16.0
default           00:10:94:00:06:03 uninstalled ge-0/0/16.0
default           00:10:94:00:07:04 uninstalled ge-0/0/16.0
```

show ethernet-switching table persistent-mac interface ge-0/0/16.0 (EX Series switches)

```
VLAN              MAC address       Type       Interface
default           00:10:94:00:05:02 uninstalled ge-0/0/16.0
default           00:10:94:00:06:03 uninstalled ge-0/0/16.0
default           00:10:94:00:07:04 uninstalled ge-0/0/16.0
```

show ethernet-switching table (EX Series, MX Series and QFX Series)

`user@host>` **show ethernet-switching table**

```
MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)
```

```
Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags               interface
    VLAN101             88:e0:f3:bb:07:f0   D             -   ae20.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags               interface
    VLAN102             88:e0:f3:bb:07:f0   D             -   ae20.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags               interface
    VLAN103             88:e0:f3:bb:07:f0   D             -   ae20.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags               interface
    VLAN104             88:e0:f3:bb:07:f0   D             -   ae20.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags               interface
    VLAN1101            00:1f:12:32:f5:c1   D             -   ae0.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)
```

```
Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags               interface
    VLAN1102            00:1f:12:32:f5:c1   D             -   ae0.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags               interface
    VLAN1103            00:1f:12:32:f5:c1   D             -   ae0.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags               interface
    VLAN1104            00:1f:12:32:f5:c1   D             -   ae0.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags               interface
    VLAN1105            00:1f:12:32:f5:c1   D             -   ae0.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags               interface
    VLAN1106            00:1f:12:32:f5:c1   D             -   ae0.0
[...output truncated...]
```

show ethernet-switching table brief

user@host> **show ethernet-switching table brief**

```
MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags              interface
    VLAN101             88:e0:f3:bb:07:f0   D             -    ae20.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags              interface
    VLAN102             88:e0:f3:bb:07:f0   D             -    ae20.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags              interface
    VLAN103             88:e0:f3:bb:07:f0   D             -    ae20.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags              interface
    VLAN104             88:e0:f3:bb:07:f0   D             -    ae20.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
```

```
    name                address             flags              interface
    VLAN1101            00:1f:12:32:f5:c1   D             -    ae0.0
[...output truncated...]
```

show ethernet-switching table count

user@host> **show ethernet-switching table count**

```
0 MAC address learned in routing instance default-switch VLAN VLAN1000
ae26.0:1000

1 MAC address learned in routing instance default-switch VLAN VLAN101
ae20.0:101

  MAC address count per learn VLAN within routing instance:
    Learn VLAN ID      MAC count      Static MAC count
           101              1                 0

1 MAC address learned in routing instance default-switch VLAN VLAN102
ae20.0:102

  MAC address count per learn VLAN within routing instance:
    Learn VLAN ID      MAC count      Static MAC count
           102              1                 0

1 MAC address learned in routing instance default-switch VLAN VLAN103
ae20.0:103

  MAC address count per learn VLAN within routing instance:
    Learn VLAN ID      MAC count      Static MAC count
           103              1                 0

1 MAC address learned in routing instance default-switch VLAN VLAN104
ae20.0:104

  MAC address count per learn VLAN within routing instance:
    Learn VLAN ID      MAC count      Static MAC count
           104              1                 0

0 MAC address learned in routing instance default-switch VLAN VLAN105
ae20.0:105

0 MAC address learned in routing instance default-switch VLAN VLAN106
ae20.0:106
```

```
0 MAC address learned in routing instance default-switch VLAN VLAN107
ae20.0:107

0 MAC address learned in routing instance default-switch VLAN VLAN108
ae20.0:108

0 MAC address learned in routing instance default-switch VLAN VLAN109
ae20.0:109

0 MAC address learned in routing instance default-switch VLAN VLAN110
ae20.0:110

1 MAC address learned in routing instance default-switch VLAN VLAN1101
ae0.0:1101

  MAC address count per learn VLAN within routing instance:
    Learn VLAN ID      MAC count      Static MAC count
          1101              1                0

1 MAC address learned in routing instance default-switch VLAN VLAN1102
ae0.0:1102

  MAC address count per learn VLAN within routing instance:
    Learn VLAN ID      MAC count      Static MAC count
          1102              1                0
[...output truncated...]
```

show ethernet-switching table extensive

`user@host>` **show ethernet-switching table extensive**

```
MAC address: 88:e0:f3:bb:07:f0
  Routing instance: default-switch
VLAN ID: 101
   Learning interface: ae20.0
   Layer 2 flags: in_hash,in_ifd,in_ifl,in_vlan,in_rtt,kernel,in_ifbd
   Epoch: 0                          Sequence number: 2
   Learning mask: 0x00000008

MAC address: 88:e0:f3:bb:07:f0
  Routing instance: default-switch
VLAN ID: 102
```

```
   Learning interface: ae20.0
   Layer 2 flags: in_hash,in_ifd,in_ifl,in_vlan,in_rtt,kernel,in_ifbd
   Epoch: 0                          Sequence number: 2
   Learning mask: 0x00000008

MAC address: 88:e0:f3:bb:07:f0
  Routing instance: default-switch
VLAN ID: 103
   Learning interface: ae20.0
   Layer 2 flags: in_hash,in_ifd,in_ifl,in_vlan,in_rtt,kernel,in_ifbd
   Epoch: 0                          Sequence number: 2
   Learning mask: 0x00000008

MAC address: 88:e0:f3:bb:07:f0
  Routing instance: default-switch
VLAN ID: 104
   Learning interface: ae20.0
   Layer 2 flags: in_hash,in_ifd,in_ifl,in_vlan,in_rtt,kernel,in_ifbd
   Epoch: 0                          Sequence number: 2
   Learning mask: 0x00000008

MAC address: 00:1f:12:32:f5:c1
  Routing instance: default-switch
VLAN ID: 1101
   Learning interface: ae0.0
   Layer 2 flags: in_hash,in_ifd,in_ifl,in_vlan,in_rtt,kernel,in_ifbd
   Epoch: 0                          Sequence number: 2
   Learning mask: 0x00000008

MAC address: 00:1f:12:32:f5:c1
  Routing instance: default-switch
VLAN ID: 1102
   Learning interface: ae0.0
   Layer 2 flags: in_hash,in_ifd,in_ifl,in_vlan,in_rtt,kernel,in_ifbd
   Epoch: 0                          Sequence number: 2
   Learning mask: 0x00000008

MAC address: 00:1f:12:32:f5:c1
  Routing instance: default-switch
VLAN ID: 1103
   Learning interface: ae0.0
   Layer 2 flags: in_hash,in_ifd,in_ifl,in_vlan,in_rtt,kernel,in_ifbd
   Epoch: 0                          Sequence number: 2
   Learning mask: 0x00000008
```

```
MAC address: 00:1f:12:32:f5:c1
  Routing instance: default-switch
VLAN ID: 1104
   Learning interface: ae0.0
   Layer 2 flags: in_hash,in_ifd,in_ifl,in_vlan,in_rtt,kernel,in_ifbd
   Epoch: 0                          Sequence number: 2
   Learning mask: 0x00000008
```

Sample Output

show ethernet-switching table detail (SRX Series)

`user@host>` **show ethernet-switching table detail**

```
Ethernet-switching table: 57 entries, 17 learned
F2, *
Interface(s): ge-0/0/44.0
Type: Flood
F2, 00:00:5E:00:53:AC
Interface(s): ge-0/0/44.0
Type: Learn, Age: 0, Learned: 2:03:09
F2, 00:00:5E:00:53:AA
Interface(s): Router
Type: Static
Linux, *
Interface(s): ge-0/0/47.0
Type: Flood
Linux, 00:00:5E:00:53:AB
Interface(s): Router
Type: Static
Linux, 00:00:5E:00:53:AC
Interface(s): ge-0/0/47.0
Type: Learn, Age: 0, Learned: 2:03:08
T1, *
Interface(s): ge-0/0/46.0
Type: Flood
T1, 00:00:5E:00:53:AD
Interface(s): ge-0/0/46.0
Type: Learn, Age: 0, Learned: 2:03:07
T1, 00:00:5E:00:53:AE
Interface(s): Router
```

```
Type: Static
T1, 00:00:5E:00:53:AF
Interface(s): ge-0/0/46.0
Type: Learn, Age: 0, Learned: 2:03:07
T1, 00:00:5E:00:53:AG
Interface(s): Router
Type: Static
T10, *
Interface(s): ge-0/0/46.0
Type: Flood
T10, 00:00:5E:00:53:AH
Interface(s): Router
Type: Static
T10, 00:00:5E:00:53:AI
Interface(s): ge-0/0/46.0
Type: Learn, Age: 0, Learned: 2:03:08
T10, 00:00:5E:00:53:AJ
Interface(s): Router
Type: Static
T111, *
Interface(s): ge-0/0/15.0
Type: Flood
[output truncated]
```

Sample Output

show ethernet-switching table extensive (SRX Series)

`user@host>` **show ethernet-switching table extensive**

```
Ethernet-switching table: 57 entries, 17 learned
F2, *
Interface(s): ge-0/0/44.0
Type: Flood
F2, 00:00:5E:00:53:AC
Interface(s): ge-0/0/44.0
Type: Learn, Age: 0, Learned: 2:03:09
F2, 00:00:5E:00:53:AA
Interface(s): Router
Type: Static
Linux, *
Interface(s): ge-0/0/47.0
```

```
Type: Flood
Linux, 00:00:5E:00:53:AB
Interface(s): Router
Type: Static
Linux, 00:00:5E:00:53:AC
Interface(s): ge-0/0/47.0
Type: Learn, Age: 0, Learned: 2:03:08
T1, *
Interface(s): ge-0/0/46.0
Type: Flood
T1, 00:00:5E:00:53:AD
Interface(s): ge-0/0/46.0
Type: Learn, Age: 0, Learned: 2:03:07
T1, 00:00:5E:00:53:AE
Interface(s): Router
Type: Static
T1, 00:00:5E:00:53:AF
Interface(s): ge-0/0/46.0
Type: Learn, Age: 0, Learned: 2:03:07
T1, 00:00:5E:00:53:AG
Interface(s): Router
Type: Static
T10, *
Interface(s): ge-0/0/46.0
Type: Flood
T10, 00:00:5E:00:53:AH
Interface(s): Router
Type: Static
T10, 00:00:5E:00:53:AI
Interface(s): ge-0/0/46.0
Type: Learn, Age: 0, Learned: 2:03:08
T10, 00:00:5E:00:53:AJ
Interface(s): Router
Type: Static
T111, *
Interface(s): ge-0/0/15.0
Type: Flood
[output truncated]
```

Sample Output

show ethernet-switching table interface ge-0/0/1 (SRX Series)

user@host> **show ethernet-switching table interface ge-0/0/1**

```
Ethernet-switching table: 1 unicast entries
VLAN     MAC address         Type     Age Interfaces
V1       *                   Flood    - All-members
V1       00:00:5E:00:53:AF   Learn    0 ge-0/0/1.0
```

show lldp

Syntax

```
show lldp
<detail>
```

Release Information

Command introduced in Junos OS Release 9.6.

Description

Display information about the Link Layer Discovery Protocol (LLDP).

Options

detail—(Optional) Display the detailed output level.

Required Privilege Level

view

List of Sample Output

show lldp on page 1551
show lldp detail on page 1551

Output Fields

Table 158 on page 1549 describes the output fields for the **show lldp** command. Output fields are listed in the approximate order in which they appear.

Table 158: show lldp Output Fields

Field Name	Field Description
LLDP	Status of LLDP: **Enabled** or **Disabled**.
Advertisement interval	Value of the advertisement interval parameter.
Transmit delay	Value of the transmit delay parameter.
Hold timer	Value of the hold timer parameter.
Notification interval	Value of the notification interval parameter.

Table 158: show lldp Output Fields *(continued)*

Field Name	Field Description
Config Trap Interval	Value of the configuration trap parameter.
Connection Hold timer	Value of the connection hold timer parameter.
Port ID TLV subtype	• *interface-name*—Indicates the interface name as the port information for the local device. • **locally-assigned**—Indicates that the sub-type for port ID TLV generation is locally assigned value of SNMP index of the interface.
Port Description TLV type	Following value used for port description TLV: • **interface-alias (ifAlias)**—Indicates that the *ifAlias* MIB object value is used to generate the port description TLV. • **interface-description (ifDescr)**—Indicates that the *ifDescr* MIB object value is used to generate the port description TLV.
Interface	Name of the interface for which LLDP configuration information is being reported For information about interface names, see *Interface Naming Overview*. For information about interface names for TX Matrix routers, see *TX Matrix Router Chassis and Interface Names*. For information about FPC numbering on TX Matrix routers, see *Routing Matrix with a TX Matrix Router FPC Numbering*.
Parent Interface	Name of the aggregated Ethernet interface, if any, to which the interface belongs.
LLDP	LLDP operating state. The state can be Enabled or Disabled.
LLDP-MED	LLDP-MED operating state. The state can be Enabled or Disabled.
Power Negotiation	LLDP power negotiation operating state. The state can be Enabled or Disabled.
LLDP basic TLVs supported	List of basic LLDP TLVs supported by this device (**detail** only).
LLDP 802 TLVs supported	List of IEEE 802.1 LLDP TLVs supported by this device (**detail** only).

Sample Output

show lldp

```
user@host> show lldp
```

```
LLDP                      : Enabled
Advertisement interval    : 30 seconds
Transmit delay            : 2 seconds
Hold timer                : 120 seconds
Notification interval     : 0 Second(s)
Config Trap Interval      : 0 seconds
Connection Hold timer     : 300 seconds
Port ID TLV subtype       : locally-assigned
Port Description TLV type : interface-description (ifDescr)

Interface      Parent Interface    LLDP        LLDP-MED       Power Negotiation
all            -                   Enabled
```

Sample Output

show lldp detail

```
user@host> show lldp detail
```

```
LLDP                      : Enabled
Advertisement interval    : 30 seconds
Transmit delay            : 2 seconds
Hold timer                : 120 seconds
Notification interval     : 0 Second(s)
Config Trap Interval      : 0 seconds
Connection Hold timer     : 300 seconds
Port ID TLV subtype       : locally-assigned
Port Description TLV type : interface-description (ifDescr)

Interface      Parent Interface    LLDP        LLDP-MED     Power Negotiation
 Neighbor count
all            -                   Enabled
 2

Interface      Parent Interface    Vlan-id     Vlan-name
xe-0/0/0       -                   4080        vlan-4080
```

```
xe-0/0/1       -                  4080        vlan-4080

Basic Management TLVs supported:
End Of LLDPDU, Chassis ID, Port ID, Time To Live, Port Description, System Name,
System Description, System Capabilities, Management Address

Organizationally Specific TLVs supported:
Port VLAN tag, VLAN Name, MAC/PHY Configuration/Status, Link Aggregation,Maximum
Frame Size
```

show lldp local-information

Syntax

```
show lldp local-information
```

Release Information

Command introduced in Junos OS Release 9.6.

Description

Display local Link Layer Discovery Protocol (LLDP) information.

Options

This command has no options.

Required Privilege Level

view

List of Sample Output

show lldp local-information(Management Information Address Subtype is IPv4) on page 1554
show lldp local-information(Management Information Address Subtype is IPv6) on page 1555

Output Fields

Table 159 on page 1553 describes the output fields for the **show lldp local-information** command. Output fields are listed in the approximate order in which they appear.

Table 159: show lldp local-information Output Fields

Field Name	Field Description
LLDP Local Information details	Information that follows pertains to the local system.
Chassis ID	List of chassis identifiers for local information.
System name	Local system name reported by LLDP.
System descr	Local system description reported by LLDP.
System Capabilities	Capabilities (such as **Bridge** or **Router**) that are **Supported** or **Enabled** by system on the interface.

Table 159: show lldp local-information Output Fields *(continued)*

Field Name	Field Description
Management Information	Listed by **Interface Name**, **Address Subtype** (such as **ipv4**, **ipv6**), **Address** (such as **192.168.168.229**, **1fd::1a10**), **Interface Number**, and **Interface Numbering Subtype**.
Interface Name	List of local interfaces. For information about interface names, see *Interface Naming Overview*. For information about interface names for TX Matrix routers, see *TX Matrix Router Chassis and Interface Names*. For information about FPC numbering on TX Matrix routers, see *Routing Matrix with a TX Matrix Router FPC Numbering*.
Parent Interface	Name of the **ae** interface to which the interface belongs
Interface ID	List of local interface identifiers.
Interface Description	List of local interface descriptions.
Status	List of interface conditions: **UP** or **DOWN**.

Sample Output

show lldp local-information(Management Information Address Subtype is IPv4)

`user@host>` **show lldp local-information**

```
LLDP Local Information details

Chassis ID   : 64:87:88:65:37:c0
System name  : apg-hp1
System descr : Juniper Networks, Inc. mx240 , version 14.1I20131231_0701_builder
[builder] Build date: 2013-12-31 07:13:42 UTC

System Capabilities
    Supported       : Bridge Router
    Enabled         : Bridge Router

Management Information
```

```
    Interface Name  : Unknown
    Address Subtype : IPv4(1)
    Address         : 10.216.97.103
    Interface Number               : 1
    Interface Numbering Subtype    : ifIndex(2)

Interface name   Parent Interface   Interface ID   Interface description   Status
fxp0                -                1             fxp0                    Up
me0                 -                33            me0                     Up
ge-2/0/0         ae0                 1475          ge-2/0/0                Up
ge-2/0/1         ae0                 1476          ge-2/0/1                Up
```

show lldp local-information(Management Information Address Subtype is IPv6)

`user@host>` **show lldp local-information**

```
LLDP Local Information details

Chassis ID   : ac:4b:c8:92:67:c0
System name  : apg-hp
System descr : Juniper Networks, Inc. mx240 , version 13.2-20131210.0 [builder]
               Build date: 2013-12-10 06:23:15 UTC

System Capabilities
    Supported       : Bridge Router
    Enabled         : Bridge Router

Management Information
    Interface Name  : fxp0
    Address Subtype : IPv6(2)
    Address         : 1fd::1a20
    Interface Number               : 1
    Interface Numbering Subtype    : ifIndex(2)

Interface name  Parent Interface   Interface ID    Interface description    Status
ge-1/2/4        -                  530             -                         Down
ge-1/2/5        -                  531             -                         Down
ge-1/2/2        -                  528             ge-1/2/2                  Up
ge-1/2/3        -                  529             ge-1/2/3                  Up
```

show lldp neighbors

Syntax

```
show lldp neighbors
<interface interface-name>
detail
```

Release Information
Command introduced in Junos OS Release 9.6.
detail option introduced in Junos OS Release 19.1R2.

Description

Display information about LLDP neighbors.

For information about interface names, see *Interface Naming Overview*. For information about interface names for TX Matrix routers, see *TX Matrix Router Chassis and Interface Names*. For information about FPC numbering on TX Matrix routers, see *Routing Matrix with a TX Matrix Router FPC Numbering*.

For information about extended port names in the Junos Fusion technology, see *Understanding Junos Fusion Ports* .

Options

interface *interface-name*—(Optional) Display the neighbor information about a particular physical interface.

> **NOTE:** Starting with Junos OS Release 14.2, you can also display LLDP neighbor details for management interfaces, such as **fxp** or **me**, on MX Series routers.

detail—(Optional) Display detailed information about LLDP neighbors.

Required Privilege Level
view

RELATED DOCUMENTATION

clear lldp neighbors | **1447**

List of Sample Output
show lldp neighbors interface ge-0/0/4 (Management Address is IPv4) on page 1560
show lldp neighbors interface ge-0/0/4 (Management Address is IPv6) on page 1561

show lldp neighbors (Management Ethernet Interfaces) on page 1563
show lldp neighbors detail on page 1563

Output Fields

Table 160 on page 1557 describes the output fields for the **show lldp neighbors** command. Output fields are listed in the approximate order in which they appear.

Table 160: show lldp neighbors Output Fields

Field Name	Field Description
LLDP Remote Devices Information	Information about remote devices.
LocalInterface	List of local interfaces for which neighbor information is available.
ChassisId	List of chassis identifiers for neighbors.
PortInfo	List of port information gathered from neighbors. This could be the port identifier or port description.
SysName	List of system names gathered from neighbors.
LLDP Neighbor Information	Information about both local and neighbor systems on the interface (appears when the **interface** option is used).
Local Information	Information about local systems on the interface (appears when the **interface** option is used).
Neighbor Information	Information about both local and neighbor system on the interface (appears when the **interface** option is used).
Index	Local interface index (appears when the **interface** option is used).
Time Mark	Date and timestamp of information (appears when the **interface** option is used).
Time To Live	Number of seconds for which this information is valid (appears when the **interface** option is used).
Local Interface	Name of the local physical interface (appears when the **interface** option is used).
Parent Interface	Name of the **ae** interface to which the interface belongs

Table 160: show lldp neighbors Output Fields *(continued)*

Field Name	Field Description
Local Port ID	Local port identifier (appears when the interface option is used).
Neighbor Information	Information about neighbor systems on the interface (appears when the **interface** option is used).
Chassis type	Type of chassis identifier supplied, such as **MAC address** (appears when the **interface** option is used).
Chassis ID	Chassis identifier of type listed (appears when the **interface** option is used).
Port type	Type of port identifier supplied, such as **local** (appears when the **interface** option is used).
Port ID	Port identifier of type listed (appears when the **interface** option is used).
Port description	Port description (appears when the **interface** option is used).
System name	Name supplied by the system on the interface (appears when the **interface** option is used).
System Description	Description supplied by the system on the interface (appears when the **interface** option is used).
System Capabilities	Capabilities (such as **bridge** or **router**) that are **Supported** or **Enabled** by the system on the interface (appears when the **interface** option is used).
Management address	Details of the management address: **Address Type** (such as **ipv4** and **ipv6**), **Address** (such as **10.204.34.35**, **1fd::1a10**), **Interface Number**, **Interface Subtype**, and Organization Identifier (**OID**) (appears when the **interface** option is used).
Organization Info	One or more entries listing remote information by Organizationally Unique Identifier (**OUI**), **Subtype**, **Index**, and **Info** (appears when the **interface** option is used).

Table 161 on page 1559 describes the output fields for the **show lldp neighbors detail** command. Output fields are listed in the approximate order in which they appear.

Table 161: show lldp neighbors detail Output Fields

Field Name	Field Description
LLDP Neighbor Information	Information about all neighbors in the system.
Local Information	Information about neighbors on a local interface.
Index	Local interface index.
Time To Live	Number of seconds for which neighbor information is valid for this interface.
Time Mark	Date and timestamp information.
Age	Age (in seconds) of the neighbor since the TLV is received from the neighbor.
Local Interface	Local Interface for which the neighbor detail is displayed.
Parent Interface	Name of the **ae** interface to which the interface belongs
Local Port ID	Local port identifier.
Ageout Count	Number of times the neighbor information has been aged out on this interface.
Neighbor Information	Information about the neighbor systems.
Chassis type	Type of chassis identifier supplied, such as **MAC address** .
ChassisId	List of chassis identifiers for all neighbors in the system.
Port type	Capabilities (such as **bridge** or **router**) that are **Supported** or **Enabled** by the system.
Port description	Port description.
System name	Name supplied by the system.
System Description	Description supplied by the system.

Table 161: show lldp neighbors detail Output Fields *(continued)*

Field Name	Field Description
System Capabilities	Capabilities (such as **bridge** or **router**) that are **Supported** or **Enabled** by the system.
Management address	Details about the management address and management interface of the neighbor.
Media endpoint class	LLDP MED device class value of the neighbor system.
Organization Info	One or more entries listing the organization-specific TLV sent by the neighbor.

Sample Output

show lldp neighbors interface ge-0/0/4 (Management Address is IPv4)

`user@host>` **show lldp neighbors interface ge-0/0/4**

```
LLDP Neighbor Information:
Local Information:
Index: 2 Time to live: 120 Time mark: Tue Dec 31 11:47:46 2013 Age: 15 secs
Local Interface    : ge-2/0/1
Parent Interface   : ae0
Local Port ID      : 1476
Ageout Count      : 0

Neighbour Information:
Chassis type       : Mac address
Chassis ID         : ac:4b:c8:92:67:c0
Port type          : Locally assigned
Port ID            : 529
Port description   : ge-1/2/3
System name        : apg-hp

System Description : Juniper Networks, Inc. mx240 , version 14.1-20131222.0
[builder] Build date: 2013-12-22 09:13:26 UTC


System capabilities
```

```
        Supported: Bridge Router
        Enabled  : Bridge Router

Management address
        Address Type      : IPv4(1)
        Address           : 10.216.98.57
        Interface Number  : 1
        Interface Subtype : ifIndex(2)
        OID               : 1.3.6.1.2.1.31.1.1.1.1.1.

Organization Info
       OUI      : IEEE 802.3 Private (0x00120f)
       Subtype  : MAC/PHY Configuration/Status (1)
       Info     : Autonegotiation [supported, enabled (0x3)], PMD Autonegotiation
 Capability (0x1d), MAU Type (0x0)
       Index    : 1

Organization Info
       OUI      : IEEE 802.3 Private (0x00120f)
       Subtype  : Link Aggregation (3)
       Info     : Aggregation Status (0x3), Aggregation Port ID (1694498816)
       Index    : 2

Organization Info
       OUI      : IEEE 802.3 Private (0x00120f)
       Subtype  : Maximum Frame Size (4)
       Info     : MTU Size (1518)
       Index    : 3
```

show lldp neighbors interface ge-0/0/4 (Management Address is IPv6)

`user@host>` **show lldp neighbors interface ge-0/0/4**

```
LLDP Neighbor Information:
Local Information:
Index: 1 Time to live: 120 Time mark: Thu Dec 12 07:19:45 2013 Age: 28 secs
Local Interface    : ge-1/2/2
Parent Interface   : -
Local Port ID      : 528
Ageout Count      : 0

Neighbour Information:
Chassis type       : Mac address
```

```
Chassis ID         : 64:87:88:65:37:c0
Port type          : Locally assigned
Port ID            : 1475
Port description   : ge-2/0/0
System name        : apg-hp1

System Description : Juniper Networks, Inc. mx240 , version 11.4R10 Build date:
2013-10-24 10:10:02 UTC


System capabilities
        Supported: Bridge Router
        Enabled  : Bridge Router

Management address
        Address Type      : IPv6(2)
        Address           : 1fd::1a10
        Interface Number  : 1
        Interface Subtype : ifIndex(2)
        OID               : 1.3.6.1.2.1.31.1.1.1.1.1.

Organization Info
       OUI      : IEEE 802.3 Private (0x00120f)
       Subtype  : MAC/PHY Configuration/Status (1)
       Info     : Autonegotiation [supported, enabled (0x3)], PMD Autonegotiation
 Capability (0x5), MAU Type (0x0)
       Index    : 1

Organization Info
       OUI      : IEEE 802.3 Private (0x00120f)
       Subtype  : Link Aggregation (3)
       Info     : Aggregation Status (0x1), Aggregation Port ID (0)
       Index    : 2

Organization Info
       OUI      : IEEE 802.3 Private (0x00120f)
       Subtype  : Maximum Frame Size (4)
       Info     : MTU Size (1518)
       Index    : 3

Organization Info
       OUI      : Ethernet Bridged (0x0080c2)
       Subtype  : VLAN Name (3)
```

```
       Info     : VLAN ID (100), VLAN Name (vlan-100)
       Index    : 4
```

show lldp neighbors (Management Ethernet Interfaces)

user@host> **show lldp neighbors**

```
Local Interface    Parent Interface    Chassis Id          Port info
System Name
fxp0               -                   78:fe:3d:ee:4e:00   151
x2-sw35
xe-0/0/0           -                   a8:d0:e5:50:26:c0   512
sitara
xe-0/0/1           -                   a8:d0:e5:50:26:c0   513
sitara
```

show lldp neighbors detail

user@host> **show lldp neighbors detail**

```
LLDP Neighbor Information:
Local Information:
Index: 1 Time to live: 120 Time mark: Wed Apr  3 08:25:57 2019 Age: 8 secs
Local Interface    : me0
Parent Interface   : -
Local Port ID      : 33
Ageout Count       : 0

Neighbour Information:
Chassis type       : Mac address
Chassis ID         : 1a:22:23:dc:d9:50
Port type          : Locally assigned
Port ID            : 517
Port description   : ge-0/0/7.0
System name        : test

System Description : Juniper Networks, Inc. ex3300-48t , version 12.3R Build date:
 2014-03-13 07:02:54 UTC


System capabilities
```

```
        Supported: Bridge Router
        Enabled  : Bridge Router

Management address
        Address Type      : IPv4(1)
        Address           : 10.221.0.111
        Interface Number  : 34
        Interface Subtype : ifIndex(2)
        OID               : 1.3.6.1.2.1.31.1.1.1.1.34.
Media endpoint class: Network Connectivity

Organization Info
       OUI      : IEEE 802.3 Private (0x00120f)
       Subtype  : MAC/PHY Configuration/Status (1)
       Info     : Autonegotiation [supported, enabled (0x3)], PMD Autonegotiation
 Capability (0x6c11), MAU Type (0x0)
       Index    : 1

Organization Info
       OUI      : IEEE 802.3 Private (0x00120f)
       Subtype  : Link Aggregation (3)
      Info     : Aggregation Status [supported, disabled (0x1)], Aggregation Port
 ID (0)
       Index    : 2

Organization Info
       OUI      : IEEE 802.3 Private (0x00120f)
       Subtype  : Maximum Frame Size (4)
       Info     : MTU Size (1514)
       Index    : 3

Organization Info
       OUI      : Ethernet Bridged (0x0080c2)
       Subtype  : Port Vid (1)
       Info     : VLAN ID (52),
       Index    : 4

Organization Info
       OUI      : Juniper Specific (0x009069)
       Subtype  : Chassis Serial Type (1)
       Info     : Juniper Slot Serial [GA0215270535]
       Index    : 5

Organization Info
```

```
       OUI      : Ethernet Bridged (0x0080c2)
       Subtype  : VLAN Name (3)
       Info     : VLAN ID (52), VLAN Name (vlan52)
       Index    : 6
       Index    : 7

LLDP Neighbor Information:
Local Information:
Index: 2 Time to live: 120 Time mark: Wed Apr  3 08:25:56 2019 Age: 9 secs
Local Interface    : ge-0/0/2
Parent Interface   : -
Local Port ID      : 512
Ageout Count       : 0

Neighbour Information:
Chassis type       : Mac address
Chassis ID         : 1a:22:23:dc:d9:50
Port type          : Locally assigned
Port ID            : 511
Port description   : ge-0/0/2

System Description : Juniper Networks, Inc. ex4300-24p Ethernet Switch, kernel
JUNOS 17.1, Build date: 2019-03-26 08:12:13 UTC Copyright (c) 1996-2019 Juniper
Networks, Inc.


System capabilities
        Supported: Bridge Router
        Enabled  : Bridge Router

Management address
        Address Type      : IPv4(1)
        Address           : 10.204.39.247
        Interface Number  : 33
        Interface Subtype : ifIndex(2)
        OID               : 1.3.6.1.2.1.31.1.1.1.1.33.
Media endpoint class: Network Connectivity

Organization Info
       OUI      : IEEE 802.3 Private (0x00120f)
       Subtype  : MAC/PHY Configuration/Status (1)
       Info     : Autonegotiation [supported, enabled (0x3)], PMD Autonegotiation
 Capability (0x1), MAU Type (0x0)
       Index    : 1
```

```
Organization Info
       OUI      : IEEE 802.3 Private (0x00120f)
       Subtype  : MDI Power (2)
       Info     : MDI Power Support [PSE bit set, supported, disabled, CONTROL bit
 not set (0x3)], MDI Power Pair  [signal], MDI Power Class [class1 (1)]
       Index    : 2

Organization Info
       OUI      : IEEE 802.3 Private (0x00120f)
       Subtype  : Link Aggregation (3)
       Info     : Aggregation Status [supported, enabled (0x3)], Aggregation Port
 ID (556)
       Index    : 3

Organization Info
       OUI      : IEEE 802.3 Private (0x00120f)
       Subtype  : Maximum Frame Size (4)
       Info     : MTU Size (6500)
       Index    : 4

Organization Info
       OUI      : Juniper Specific (0x009069)
       Subtype  : Chassis Serial Type (1)
       Info     : Juniper Slot Serial [MS3112240002]
       Index    : 5

Organization Info
       OUI      : Ethernet Bridged (0x0080c2)
       Subtype  : VLAN Name (3)
       Info     : VLAN ID (100), VLAN Name (vlan-100)
       Index    : 6
       Index    : 7
       Index    : 8
```

show lldp remote-global-statistics

Syntax

```
show lldp remote-global-statistics
```

Release Information

Command introduced in Junos OS Release 9.6.

Description

Display remote Link Layer Discovery Protocol (LLDP) global statistics.

Options

This command has no options.

Required Privilege Level

view

List of Sample Output

show lldp remote-global-statistics on page 1568

Output Fields

Table 162 on page 1567 describes the output fields for the **show lldp remote-global-statistics** command. Output fields are listed in the approximate order in which they appear.

Table 162: show lldp remote-global-statistics Output Fields

Field Name	Field Description
LLDP Remote Database Table Counters	Information about remote database table counters.
LastchangeTime	Time elapsed between LLDP agent startup and the last change to the remote database table information.
Inserts	Number of insertions made in the remote database table.
Deletes	Number of deletions made in the remote database table.
Drops	Number of LLDP frames dropped from the remote database table because of errors.
Ageouts	Number of remote database table entries that have aged out of the table.

Sample Output

show lldp remote-global-statistics

user@host> **show lldp remote-global-statistics**

```
user@host> show lldp remote-global-statistics
   LLDP Remote Database Table Counters
   LastchangeTime          Inserts    Deletes    Drops    Ageouts
   00:00:76 (76 sec)       192        0          0        0
```

show lldp statistics

Syntax

```
show lldp statistics
<interface interface-name>
```

Release Information

Command introduced in Junos OS Release 9.6.

Description

Display information about Link Layer Discovery Protocol (LLDP) statistics.

Options

interface *interface-name*—(Optional) Display the statistics about a particular physical interface.

> **NOTE:** Starting with Junos OS Release 14.2, you can also display LLDP statistical details for management interfaces, such as **fxp** or **me**, on MX Series routers.

Required Privilege Level

view

RELATED DOCUMENTATION

clear lldp statistics | **1449**

List of Sample Output

show lldp statistics on page 1570
show lldp statistics interface ge-0/1/1 on page 1571

Output Fields

Table 163 on page 1570 describes the output fields for the **show lldp statistics** command. Output fields are listed in the approximate order in which they appear.

Table 163: show lldp statistics Output Fields

Field Name	Field Description
Interface	Interface name. For information about interface names, see *Interface Naming Overview*. For information about interface names for TX Matrix routers, see *TX Matrix Router Chassis and Interface Names*. For information about FPC numbering on TX Matrix routers, see *Routing Matrix with a TX Matrix Router FPC Numbering*. For information about extended port names in the Junos Fusion technology, see *Understanding Junos Fusion Ports* .
Received	Number of LLDP frames received on this interface.
Transmitted	Number of LLDP frames sent on this interface.
Unknown-TLVs	Number of LLDP frames with unsupported content received on this interface.
With-Errors	Number of LLDP frames with errors received on this interface.
Discarded	Number of LLDP frames received on this interface that were discarded because of problems.
Transmitted	Total number of LLDP frames that were transmitted on an interface.
Untransmitted	Total number of LLDP frames that were untransmitted on an interface.

Sample Output

show lldp statistics

```
user@host> show lldp statistics
```

```
Interface    Parent Interface  Received  Unknown TLVs  With Errors
xe-3/0/0.0   ae31.0            1564      0             0
xe-3/0/1.0   ae31.0            1564      0             0
xe-3/0/2.0   ae31.0            1565      0             0
xe-3/0/3.0   ae31.0            1566      0             0
xe-3/0/4.0   ae31.0            1598      0             0
```

```
xe-3/0/5.0   ae31.0           1598      0            0
xe-3/0/6.0   ae31.0           1596      0            0
xe-3/0/7.0   ae31.0           1597      0            0
xe-5/0/6.0   -                0         0            0
xe-5/0/7.0   -                0         0            0


Discarded TLVs  Transmitted  Untransmitted
0               3044         1
0               3044         1
0               3044         1
0               3044         1
0               3075         1
0               3075         1
0               3075         1
0               3075         1
0               17312        0
0               17312        0
```

Sample Output

show lldp statistics interface ge-0/1/1

`user@host>` **show lldp statistics interface ge-0/1/1**

```
Interface Received Transmitted Unknown-TLVs With-Errors Discarded
--------- -------- ----------- ------------ ----------- ---------
ge-0/1/1  544      540         0            0           0
```

show mac-rewrite interface

Syntax

```
show mac-rewrite interface
<brief | detail>
<interface-name>
```

Release Information

Command introduced in Junos OS Release 9.1.
Command introduced in Junos OS Release 14.1X53-D10 and 17.3R1 for EX4300 switches.
Command introduced in Junos OS Release 15.1X53-D55 and 18.2R1 for EX2300 and EX3400 switches.
Command introduced in Junos OS Release 17.4R1 for EX4600 switches.
Command introduced in Junos OS Release 19.1R1 for QFX Series switches.

Description

Display Layer 2 protocol tunneling (L2PT) information.

Options

brief | detail—(Optional) Display the specified level of output.

interface *interface-name*—(Optional) Display L2PT information for the specified interface.

Required Privilege Level

view

RELATED DOCUMENTATION

- layer2-control | **1234**
- mac-rewrite | **1257**
- protocol | **1326**
- Understanding Layer 2 Protocol Tunneling | **684**
- Configuring Layer 2 Protocol Tunneling | **694**

List of Sample Output

show mac-rewrite interface on page 1573
show mac-rewrite interface (EX Series Switches) on page 1573

Output Fields

Table 164 on page 1573 lists the output fields for the **show mac-rewrite interface** command. Output fields are listed in the approximate order in which they appear.

Table 164: show mac-rewrite interface Output Fields

Field Name	Field Description	Level of Output
Interface	Name of the interface on which L2PT is configured.	brief detail
Protocols	Layer 2 protocols being tunneled on this interface. All devices that support L2PT can tunnel the following protocols: Cisco Discovery Protocol (CDP), Spanning Tree Protocol (STP), or VLAN Trunk Protocol (VTP). The following Layer 2 protocols can also be tunneled on some devices that support L2PT: E-LMI, GVRP, IEEE 802.1X, IEEE 802.3AH, LACP, LLDP, MMRP, MVRP, PVSTP+, UDLD, or VSTP. See protocol for more information on the supported protocols for tunneling on different devices.	brief detail

Sample Output

show mac-rewrite interface

```
user@host> show mac-rewrite interface
```

```
Interface       Protocols

 ge-1/0/5       STP VTP CDP PVSTP+
```

show mac-rewrite interface (EX Series Switches)

```
user@switch> show mac-rewrite interface
```

```
Interface       Protocols

 ge-0/0/1       802.3AH LLDP STP
```

show mvrp

Syntax

```
show mvrp
```

Release Information

Command introduced in Junos OS Release 10.0 for EX Series switches.
Command introduced in Junos OS Release 10.1 for MX Series routers.
Command introduced in Junos OS Release 13.2X50-D10 for EX Series switches.
Command introduced in Junos OS Release 15.1X49-D70 for SRX Series devices.

Description

Display Multiple VLAN Registration Protocol (MVRP) configuration information.

Required Privilege Level

view

RELATED DOCUMENTATION

- Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches | 832
- Verifying That MVRP Is Working Correctly on Switches | 847
- show mvrp statistics | 1589
- show mvrp applicant-state | 1578
- show mvrp dynamic-vlan-memberships | 1581
- show mvrp interface | 1584
- show mvrp registration-state | 1586
- show mvrp statistics | 1589
- show mvrp applicant-state | 1578
- show mvrp dynamic-vlan-memberships | 1581
- show mvrp interface | 1584
- show mvrp registration-state | 1586

List of Sample Output

show mvrp (EX Series switches and MX Series routers) on page 1575
show mvrp (SRX1500, SRX300, SRX550M, SRX345, SRX340 and SRX320) on page 1576
show mvrp (EX Series switches) on page 1576

Output Fields

Table 165 on page 1575 lists the output fields for the **show mvrp** command. Output fields are listed in the approximate order in which they appear.

Table 165: show mvrp Output Fields

Field Name	Field Description
MVRP dynamic VLAN creation	Displays whether global MVRP dynamic VLAN creation is **Enabled** or **Disabled**.
Global MVRP configuration	Displays global MVRP information: • **MVRP status**—Displays whether MVRP is **Enabled** or **Disabled**. • **MVRP dynamic vlan creation**—Displays whether global MVRP dynamic VLAN creation is **Dnabled** or **Disabled**.
MVRP BPDU MAC address	Displays the multicast media access control (MAC) address for MVRP. If configured, the provider MVRP multicast MAC address is used; otherwise, the customer MVRP multicast MAC address is used.
MVRP timers (ms)	Displays MVRP timer information: • **Interface**—The interface on which MVRP is configured. • **Join**—The maximum number of milliseconds the interfaces must wait before sending VLAN advertisements. • **Leave**—The number of milliseconds an interface must wait after receiving a Leave message to remove the interface from the VLAN specified in the message. • **LeaveAll**— The interval at which LeaveAll messages are sent on interfaces. LeaveAll messages maintain current MVRP VLAN membership information in the network.
Interface based configuration	Displays interface-specific MVRP information: • **Interface**—The interface on which MVRP is configured. • **Status**—Displays whether MVRP is **Enabled** or **Disabled**. • **Registration**—Displays whether registration for the interface is **Forbidden** or **Normal**. • **Dynamic VLAN Creation**—Displays whether interface dynamic VLAN creation is **Enabled** or **Disabled**.

Sample Output

show mvrp (EX Series switches and MX Series routers)

```
user@host> show mvrp
```

```
MVRP configuration for routing instance 'default-switch'
MVRP dynamic VLAN creation : Enabled
MVRP BPDU MAC address      : Customer bridge group (01-80-C2-00-00-21)
MVRP timers (ms)
  Interface       Join   Leave  LeaveAll
  ge-11/2/8        200     800     10000
  ge-11/0/9        200     800     10000
  ge-11/3/0        200     800     10000
```

Sample Output

show mvrp (SRX1500, SRX300, SRX550M, SRX345, SRX340 and SRX320)

```
user@host> show mvrp
```

```
MVRP configuration for routing instance 'default-switch'
MVRP dynamic VLAN creation : Enabled
MVRP BPDU MAC address      : Customer bridge group (00-00-5E-00-53-00)
MVRP timers (ms)
  Interface       Join   Leave  LeaveAll
  ge-0/0/1        200     800     60
```

Sample Output

show mvrp (EX Series switches)

```
user@switch> show mvrp
```

```
Global MVRP configuration
  MVRP status               : Enabled
  MVRP dynamic vlan creation: Enabled
  MVRP Timers (ms):
   Interface        Join   Leave   LeaveAll
   --------------   ----   -----   --------
   all              200    600     10000
   xe-0/1/1.0       200    600     10000

  Interface based configuration:
```

```
Interface        Status     Registration   Dynamic VLAN Creation
--------------   --------   ------------   ---------------------
all       Disabled   Normal    Enabled
xe-0/1/1.0     Enabled    Normal         Enabled
```

show mvrp applicant-state

Syntax

```
show mvrp applicant-state
```

Release Information

Command introduced in Junos OS Release 10.1.
Command introduced in Junos OS Release 15.1X49-D70 for SRX Series devices.

Description

For MX Series routers, EX Series switches, SRX1500, SRX300, SRX550M, SRX345, SRX340, and SRX320, display Multiple VLAN Registration Protocol (MVRP) applicant state information.

Required Privilege Level

view

RELATED DOCUMENTATION

- show mvrp | 1574
- show mvrp interface | 1584
- show mvrp registration-state | 1586
- show mvrp statistics | 1589
- show mvrp interface | 1584
- show mvrp registration-state | 1586

List of Sample Output

show mvrp applicant-state (EX Series and MX Series) on page 1579
show mvrp applicant-state on page 1580

Output Fields

Table 166 on page 1578 lists the output fields for the **show mvrp applicant-state** command. Output fields are listed in the approximate order in which they appear.

Table 166: show mvrp applicant-state Output Fields

Field Name	Field Description
VLAN Id	Displays the VLAN ID number.
Interface	Displays the interface number associated with the VLAN ID.

Table 166: show mvrp applicant-state Output Fields *(continued)*

Field Name	Field Description
State	Displays one of the following MVRP registrar states: • VO— Very anxious observer. • VP —Very anxious passive. • VA —Very anxious new. • AN —Anxious new. • AA —Anxious active. • QA —Quiet active. • LA —Leaving active. • AO —Anxious observer. • QO —Quiet observer. • LO —Leaving observer. • AP —Anxious passive. • QA —Quiet passive.

Sample Output (EX Series and MX Series)

show mvrp applicant-state (EX Series and MX Series)

`user@host>` **show mvrp applicant-state**

```
MVRP applicant state for routing instance 'default-switch'
 (VO) Very anxious observer, (VP) Very anxious passive, (VA) Very anxious new,
 (AN) Anxious new, (AA) Anxious active, (QA) Quiet active, (LA) Leaving active,
 (AO) Anxious observer, (QO) Quiet observer, (LO) Leaving observer,
 (AP) Anxious passive, (QP) Quiet passive

VLAN Id     Interface          State
    100     ge-11/3/0          Declaring (QA)
    200     ge-11/3/0          Declaring (QA)
    300     ge-11/3/0          Declaring (QA)
```

Sample Output (SRX1500, SRX300, SRX550M, SRX345, SRX340, and SRX320)

show mvrp applicant-state

```
user@host> show mvrp applicant-state
```

```
MVRP applicant state for routing instance 'default-switch'
 (VO) Very anxious observer, (VP) Very anxious passive, (VA) Very anxious new,
 (AN) Anxious new, (AA) Anxious active, (QA) Quiet active, (LA) Leaving active,
 (AO) Anxious observer, (QO) Quiet observer, (LO) Leaving observer,
 (AP) Anxious passive, (QP) Quiet passive

VLAN Id     Interface          State
    1       ge-0/0/1          Idle (VO)
    30      ge-0/0/1          Idle (VO)
    40      ge-0/0/1          Idle (VO)
    50      ge-0/0/1          Idle (VO)
    100     ge-0/0/1          Idle (VO)
```

show mvrp dynamic-vlan-memberships

Syntax

```
show mvrp dynamic-vlan-memberships
```

Release Information

Command introduced in Junos OS Release 10.0 for EX Series switches.
Command introduced in Junos OS Release 10.1 for MX Series routers.
Command introduced in Junos OS Release 13.2X50-D10 for EX Series switches.
Command introduced in Junos OS Release 15.1X49-D70 for SRX Series devices.

Description

Display all VLANs that have been created dynamically using Multiple VLAN Registration Protocol (MVRP) on the router, switch, or SRX Series device.

Required Privilege Level

clear

RELATED DOCUMENTATION

- Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches | 832
- Verifying That MVRP Is Working Correctly on Switches | 847
- show mvrp | 1574
- show mvrp applicant-state | 1578
- show mvrp interface | 1584
- show mvrp registration-state | 1586
- show mvrp registration-state | 1586
- show mvrp statistics | 1589

List of Sample Output

Output Fields

Table 167 on page 1582 lists the output fields for the **show mvrp dynamic-vlan-memberships** command on MX Series routers and EX Series switches. Output fields are listed in the approximate order in which they appear.

Table 167: show mvrp dynamic-vlan-memberships Output Fields

Field Name	Field Description
VLAN Id	The VLAN ID of the dynamically created VLAN.
Interfaces	The interface or interfaces that are bound to the dynamically created VLAN.

Sample Output (MX Series Routers and EX Series Switches)

show mvrp dynamic-vlan-memberships (MX Series and EX Series)

```
user@host> show mvrp dynamic-vlan-memberships
```

```
MVRP dynamic vlans for routing instance 'default-switch'
 (s) static vlan, (f) fixed registration

VLAN Id            Interfaces
    100 (s)        ge-11/3/0
    200 (s)        ge-11/3/0
    300 (s)
```

Sample Output (EX Series Switches)

show mvrp dynamic-vlan-memberships (EX Series)

```
user@switch> show mvrp dynamic-vlan-memberships
```

```
VLAN Name             Interfaces
-------------------       ----------------
__mvrp_100__              xe-0/1/1.0
                          xe-0/1/0.0
__mvrp_200__              xe-0/1/1.0
                          xe-0/1/0.0
__mvrp_300__              xe-0/1/1.0
```

Sample Output (SRX1500, SRX300, SRX550M, SRX345, SRX340, SRX320)

show mvrp dynamic-vlan-memberships

```
user@host> show mvrp dynamic-vlan-memberships
```

```
MVRP dynamic vlans for routing instance 'default-switch'
 (s) static vlan, (f) fixed registration

VLAN Id            Interfaces
    1  (s)
    30 (s)
    40 (s)        ge-0/0/1
    50 (s)        ge-0/0/1
   100 (s)        ge-0/0/1 (f)
```

show mvrp interface

Syntax

```
show mvrp interface
```

Release Information

Command introduced in Junos OS Release 10.1.
Command introduced in Junos OS Release 15.1X49-D70 for SRX Series devices.

Description

Display Multiple VLAN Registration Protocol (MVRP) interface-specific information.

Required Privilege Level

view

RELATED DOCUMENTATION

- show mvrp | 1574
- show mvrp applicant-state | 1578
- show mvrp dynamic-vlan-memberships | 1581
- show mvrp registration-state | 1586
- show mvrp registration-state | 1586
- show mvrp statistics | 1589

List of Sample Output

show mvrp interface on page 1585
show mvrp interface on page 1585

Output Fields

Table 168 on page 1584 lists the output fields for the **show mvrp interface** command. Output fields are listed in the approximate order in which they appear.

Table 168: show mvrp interface Output Fields

Field Name	Field Description
Interface	Interface on which MVRP is configured.
Status	Status of the MVRP: **Enabled** or **Disabled**.

Table 168: show mvrp interface Output Fields *(continued)*

Field Name	Field Description
Registration Mode	Registration for the interface: **Fixed**, **Forbidden**, or **Normal**.
Applicant Mode	Applicant mode.

Sample Output (MX Series Routers and SX Series Switches)

show mvrp interface

```
user@host> show mvrp interface
```

```
MVRP interface information for routing instance 'default-switch'

Interface       Status       Registration      Applicant
                             Mode              Mode
ge-11/2/8       Enabled      Normal            Normal
ge-11/0/9       Enabled      Normal            Normal
ge-11/3/0       Enabled      Normal            Normal
```

Sample Output (SRX1500, SRX300, SRX550M, SRX345, SRX340 and SRX320)

show mvrp interface

```
user@host> show mvrp interface
```

```
MVRP interface information for routing instance 'default-switch'

Interface       Status       Registration      Applicant
                Mode         Mode              Mode
ge-0/0/1        Enabled      Normal            Normal
```

show mvrp registration-state

Syntax

```
show mvrp registration-state
```

Release Information

Command introduced in Junos OS Release 10.1.
Command introduced in Junos OS Release 15.1X49-D70 for SRX Series devices.

Description

For MX Series routers, EX Series switches and SRX1500, SRX300, SRX550M, SRX345, SRX340 and SRX320, display Multiple VLAN Registration Protocol (MVRP) registration state information.

Required Privilege Level

view

RELATED DOCUMENTATION

- show mvrp | 1574
- show mvrp dynamic-vlan-memberships | 1581
- show mvrp interface | 1584
- show mvrp statistics | 1589

List of Sample Output

show mvrp registration-state (EX Series and MX Series) on page 1587
show mvrp registration-state (SRX1500, SRX300, SRX550M, SRX345, SRX340 and SRX320) on page 1587

Output Fields

Table 169 on page 1586 lists the output fields for the **show mvrp registration-state** command. Output fields are listed in the approximate order in which they appear.

Table 169: show mvrp registration-state Output Fields

Field Name	Field Description
VLAN Id	Displays the VLAN ID number.
Interface	Displays the interface number associated with the VLAN ID.
Registrar State	Displays whether the registrar state is Registered or Empty.

Table 169: show mvrp registration-state Output Fields *(continued)*

Field Name	Field Description
Forced State	Displays whether the forced state is Registered or Empty.
Managed State	Displays one of the following states: • **fixed**—VLANs always stay in a registered state and are declared as such on all other forwarding ports. • **normal** —VLANs participate in the MVRP protocol and honor incoming join requests normally. • **forbidden** —VLANs ignore the incoming join requests and always stay in an unregistered state.
STP State	Displays whether the Spanning Tree Protocol (STP) is Blocking or Forwarding.

Sample Output

show mvrp registration-state (EX Series and MX Series)

```
user@host> show mvrp registration-state
```

```
MVRP registration state for routing instance 'default-switch'

VLAN Id   Interface   Registrar   Forced      Managed    STP
                      State       State       State      State
    100   ge-11/2/8   Empty       Registered  Fixed      Forwarding
          ge-11/0/9   Empty       Empty       Normal     Forwarding
          ge-11/3/0   Registered  Registered  Normal     Forwarding
    101   ge-11/2/8   Empty       Registered  Fixed      Forwarding
          ge-11/0/9   Empty       Empty       Normal     Forwarding
          ge-11/3/0   Registered  Registered  Normal     Forwarding
```

Sample Output

show mvrp registration-state (SRX1500, SRX300, SRX550M, SRX345, SRX340 and SRX320)

```
user@host> show mvrp registration-state
```

```
MVRP registration state for routing instance 'default-switch'

VLAN Id   Interface   Registrar   Forced      Managed    STP
                      State       State       State      State
    1     ge-0/0/1    Empty       Empty       Normal     Forwarding
    30    ge-0/0/1    Empty       Empty       Normal     Forwarding
    40    ge-0/0/1    Registered  Registered  Normal     Forwarding
    50    ge-0/0/1    Registered  Registered  Normal     Forwarding
   100    ge-0/0/1    Empty       Registered  Fixed      Forwarding
```

show mvrp statistics

List of Syntax

Syntax (EX Series Switches)

```
show mvrp statistics
<interface interface-name>
```

Syntax (Switches with ELS Support)

```
show mvrp statistics
<interface interface-name>
<routing-instance routing-instance-name>
```

Syntax (SRX Devices)

```
show mvrp statistics
```

Release Information

Command introduced in Junos OS Release 10.0 for EX Series switches.
Command introduced in Junos OS Release 13.2X50-D10 (ELS).
Command introduced in Junos OS Release 15.1X49-D70 for SRX Series devices.

Description

Display Multiple VLAN Registration Protocol (MVRP) statistics in the form of Multiple Registration Protocol data unit (MRPDU) messages.

Options

none—Show MVRP statistics for all interfaces on the switch.

interface *interface-name*—(Optional) Show MVRP statistics for the specified interface.

Required Privilege Level

view

RELATED DOCUMENTATION

show mvrp | 1574
clear mvrp statistics | 1451
Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches | 832
Verifying That MVRP Is Working Correctly on Switches | 847
Verifying That MVRP Is Working Correctly on EX Series Switches with ELS Support | 849

List of Sample Output

show mvrp statistics interface xe-0/1/1.0 on page 1593
show mvrp statistics on page 1593
show mvrp statistics (SRX Devices) on page 1594

Output Fields

Table 167 on page 1582 lists the output fields for the **show mvrp statistics** command on EX Series switches. Output fields are listed in the approximate order in which they appear.

Table 170: show mvrp statistics Output Fields

Field Name	Field Description
MRPDU received	Number of MRPDU messages received on the switch.
Invalid PDU received	Number of invalid MRPDU messages received on the switch.
New received	Number of new messages received on the switch.
Join Empty received	Number of MRP JoinEmpty messages received on the switch. Either this value or the value for *JoinIn received* should increase when the value for *MRPDU received* increases. If this value is not incrementing when it should, you might have a Junos OS release version compatibility issue. To fix a version compatibility issue, see "Configuring Multiple VLAN Registration Protocol (MVRP) on Switches" on page 797.
Join In received	Number of MRP JoinIn messages received on the switch. Either this value or the value for *JoinEmpty received* should increase when the value for *MRPDU received* increases. If this value is not incrementing when it should, you might have a Junos OS release version compatibility issue. To fix a version compatibility issue, see "Configuring Multiple VLAN Registration Protocol (MVRP) on Switches" on page 797.
Empty received	Number of MRP Empty messages received on the switch.
In received	Number of MRP In messages received on the switch.
Leave received	Number of MRP Leave messages received on the switch.

Table 170: show mvrp statistics Output Fields *(continued)*

Field Name	Field Description
LeaveAll received	Number of LeaveAll messages received on the switch.
MRPDU transmitted	Number of MRPDU messages transmitted from the switch.
MRPDU transmit failures	Number of MRPDU transmit failures from the switch.
New transmitted	Number of new messages transmitted from the switch.
Join Empty transmitted	Number of JoinEmpty messages sent from the switch.
Join In transmitted	Number of MRP JoinIn messages sent from the switch.
Empty transmitted	Number of MRP Empty messages sent from the switch.
In transmitted	Number of MRP In messages sent from the switch.
Leave transmitted	Number of MRP Leave Empty messages sent from the switch.
LeaveAll transmitted	Number of MRP LeaveAll messages sent from the switch.

Table 171 on page 1591 lists the output fields for the **show mvrp statistics** command on SRX devices. Output fields are listed in the approximate order in which they appear.

Table 171: show mvrp statistics Output Fields

Field Name	Field Description
Interface name	Interface for which MVRP statistics are displayed.
VLAN IDs registered	Number of Virtual LAN (VLAN) IDs registered.
Sent MVRP PDUs	Number of MRPDU messages transmitted from the switch.
Received MVRP PDUs without error	Number of MRPDU messages received on the switch.

Table 171: show mvrp statistics Output Fields *(continued)*

Field Name	Field Description
Received MVRP PDUs with error	Number of invalid MRPDU messages received on the switch.
Transmitted Join Emply	Number of JoinEmpty messages sent from the switch.
Transmitted Leave All	Number of MRP LeaveAll messages sent from the switch.
Received Join In	Number of MRP JoinIn messages received on the switch. Either this value or the value for **Received Join Empty** should increase when the value for **Received MVRP PDUs without error** increases. If this value is not incrementing when it should, you might have a Junos OS release compatibility issue. To resolve the issue, see "Configuring Multiple VLAN Registration Protocol (MVRP) on Switches" on page 797.
Transmitted Join In	Number of MRP JoinIn messages sent from the switch.
Transmitted Empty	Number of MRP Empty messages sent from the switch.
Transmitted Leave	Number of MRP LeaveEmpty messages sent from the switch.
Transmitted In	Number of MRP In messages sent from the switch.
Transmitted New	Number of New messages transmitted from the switch.
Received Leave All	Number of LeaveAll messages received on the switch.
Received Leave	Number of MRP Leave messages received on the switch.
Received In	Number of MRP In messages received on the switch.
Received Empty	Number of MRP Empty messages received on the switch.
Received Join Empty	Number of MRP JoinEmpty messages received on the switch. Either this value or the value for **Received Join In** should increase when the value for **Received MVRP PDUs without error** increases. If this value is not incrementing when it should, you might have a Junos OS release compatibility issue. To resolve the issue, see "Configuring Multiple VLAN Registration Protocol (MVRP) on Switches" on page 797.
Received New	Number of New messages received on the switch.

Sample Output

show mvrp statistics interface xe-0/1/1.0

```
user@switch> show mvrp statistics interface xe-0/1/1.0
```

```
MVRP statistics
  MRPDU received                : 3342
  Invalid PDU received          : 0
  New received                  : 2
  Join Empty received           : 1116
  Join In received              : 2219
  Empty received                : 2
  In received                   : 2
  Leave received                : 1
  LeaveAll received             : 1117
  MRPDU transmitted             : 3280
  MRPDU transmit failures       : 0
  New transmitted               : 0
  Join Empty transmitted        : 1114
  Join In transmitted           : 2163
  Empty transmitted             : 1
  In transmitted                : 1
  Leave transmitted             : 1
  LeaveAll transmitted          : 1111
```

show mvrp statistics

```
user@host> show mvrp statistics
```

```
MVRP statistics for routing instance 'default-switch'

Interface name                  : xe-0/1/1
VLAN IDs registered             : 117
Sent MVRP PDUs                  : 118824
Received MVRP PDUs without error: 118848
Received MVRP PDUs with error   : 0
Transmitted Join Empty          : 5229
Transmitted Leave All           : 2
Recieved Join In                : 11884924
Transmitted Join In             : 1835
Transmitted Empty               : 93606408
Transmitted Leave               : 888
```

```
Transmitted In                  : 13780024
Transmitted New                 : 2692
Received Leave All              : 118761
Received Leave                  : 97
Received In                     : 3869
Received Empty                  : 828
Received Join Empty             : 2020152
Received New                    : 224
...
```

show mvrp statistics (SRX Devices)

user@host> **show mvrp statistics**

```
MVRP statistics for routing instance 'default-switch'

Interface name                  : ge-0/0/1
VLAN IDs registered             : 2
Sent MVRP PDUs                  : 41
Received MVRP PDUs without error: 28
Received MVRP PDUs with error   : 0
Transmitted Join Empty          : 0
Transmitted Leave All           : 20
Received Join In                : 0
Transmitted Join In             : 0
Transmitted Empty               : 114
Transmitted Leave               : 0
Transmitted In                  : 10
Transmitted New                 : 0
Received Leave All              : 1
Received Leave                  : 0
Received In                     : 0
Received Empty                  : 67
Received Join Empty             : 24
Received New                    : 0
```

show protection-group ethernet-ring aps

Syntax

```
show protection-group ethernet-ring aps
```

Release Information

Command introduced in Junos OS Release 9.4.
Command introduced in Junos OS Release 12.1 for EX Series switches.
Command introduced in Junos OS Release 18.1 for EX2300 and EX3400 switches.

Description

Display the status of the Automatic Protection Switching (APS) and Ring APS (RAPS) messages on an Ethernet ring.

Options

This command has no options.

Required Privilege Level

view

RELATED DOCUMENTATION

- show protection-group ethernet-ring data-channel | **1609**
- show protection-group ethernet-ring interface | **1612**
- show protection-group ethernet-ring node-state | **1617**
- show protection-group ethernet-ring statistics | **1624**
- show protection-group ethernet-ring vlan | **1631**

List of Sample Output

show protection-group ethernet-ring aps detail (MX Series router) on page 1598
show protection-group ethernet-ring aps (MX Interconnection Ring Node as RPL owner of major ring, rings in IDLE state) on page 1599
show protection-group ethernet-ring aps detail (EX2300 and EX3400 Switches) on page 1599

Output Fields

Table 172 on page 1596 lists the output fields for the **show protection-group ethernet-ring aps** command. Output fields are listed in the approximate order in which they appear.

Table 172: show protection-group ethernet-ring aps Output Fields

Field Name	Field Description
Ethernet Ring	Name configured for the Ethernet ring.
Request/State	Status of the Ethernet ring RAPS messages. • **NR**—Indicates that there is no request for APS on the ring. • **SF**—Indicates that there is a signal failure on the ring. • **FS**—Indicates that there are active forced-switch requests in the ring. • **MS**—Indicates that there are active manual-switch requests in the ring. NOTE: Both **FS** and **MS** values are valid only when G.8032v2 is supported.
Ring Protection Link Blocked	Blocking on the ring protection link: **Yes** or **No**.
No Flush	Indicates the value of the Do Not Flush (DNF) flag in the received RAPS PDU. If the value is Yes, then FDB flush is not triggered as part of processing of the received RAPS PDU.
Blocked Port Reference	This parameter is the reference to the blocked ring port. If the east ring port is blocked, the Blocked Port Reference (BPR) value is 0. If the west ring port is blocked, the BPR value is 1. If both ring ports are blocked, this parameter can take any value. If both east and west ports are blocked or not blocked, the value would be 0. This field is valid only when G.8032v2 is supported.
Blocked Port Reference	Reference of the ring port on which traffic is blocked.
Originator	Indicates whether the node is the originator of the RAPS messages.
Remote Node ID	Identifier (in MAC address format) of the remote node.

Sample Output

show protection-group ethernet-ring aps (EX Switches)

user@switch>**show protection-group ethernet-ring aps**

```
Ring Name    Request/state  No Flush  RPL Blocked  Originator  Remote Node ID erp1
         NR             No        Yes          No          00:1F:12:30:B8:81
```

Sample Output

show protection-group ethernet-ring aps (Owner Node, Normal Operation on ACX and MX Routers)

user@host> **show protection-group ethernet-ring aps**

```
Ethernet Ring   Request/state  RPL Blocked  No Flush  BPR  Originator  Remote Node
 ID
Erp_1           NR             Yes          No        1    No
00:00:00:02:00:01
```

Sample Output

show protection-group ethernet-ring aps detail (Owner Node, Normal Operation on ACX and MX Routers)

user@host> **show protection-group ethernet-ring aps detail**

```
Ethernet-Ring name             : Erp_1
Request/State                  : NR
Ring Protection Link blocked   : Yes
No Flush Flag                  : No
Blocked Port Reference         : 1
Originator                     : No
Remote Node ID                 : 00:00:00:02:00:01
```

show protection-group ethernet-ring aps (MX RPL Owner Ring Node, Failure condition on non-RPL link of the ring)

user@host>**show protection-group ethernet-ring aps**

```
Ethernet Ring   Request/state   RPL Blocked  No Flush
pg101           SF              No           No

Originator  Remote Node ID
No          00:01:02:00:00:01
```

show protection-group ethernet-ring aps (MX Interconnection Ring Node, Failure condition in major ring on non-RPL link of the ring)

user@host>**show protection-group ethernet-ring aps**

```
Ethernet Ring   Request/state  RPL Blocked  No Flush  BPR
pg_major        SF             No           No        0
pg_subring      NR             Yes          Yes       0

Originator  Remote Node ID
No          00:01:00:00:00:01
No          00:02:00:00:00:02
```

show protection-group ethernet-ring aps (MX Series router)

user@host>**show protection-group ethernet-ring aps**

```
Ethernet Ring   Request/state  RPL Blocked  No Flush  BPR  Originator  Remote Node
 ID
Inst_Vlans_1-15 NR             Yes          Yes       1    Yes         NA

Inst_Vlans_16-30 NR            Yes          Yes       0    No
00:00:00:03:00:02
```

show protection-group ethernet-ring aps detail (MX Series router)

user@host>**show protection-group ethernet-ring aps**

```
Ethernet-Ring name             : Inst_Vlans_1-15
Request/State                  : NR
Ring Protection Link blocked   : Yes
No Flush Flag                  : Yes
Blocked Port Reference         : 1
Originator                     : Yes
Remote Node ID                 : NA
```

```
Ethernet-Ring name             : Inst_Vlans_16-30
Request/State                  : NR
Ring Protection Link blocked   : Yes
No Flush Flag                  : Yes
Blocked Port Reference         : 0
Originator                     : No
Remote Node ID                 : 00:00:00:03:00:02
```

show protection-group ethernet-ring aps (MX Interconnection Ring Node as RPL owner of major ring, rings in IDLE state)

user@host>**show protection-group ethernet-ring aps detail**

```
Ethernet-Ring name             : pg_major
Request/State                  : NR
Ring Protection Link blocked   : Yes
No Flush Flag                  : Yes
Blocked Port Reference         : 0
Originator                     : Yes
Remote Node ID                 : NA

Ethernet-Ring name             : pg_subring
Request/State                  : NR
Ring Protection Link blocked   : Yes
No Flush Flag                  : Yes
Blocked Port Reference         : 0
Originator                     : No
Remote Node ID                 : 00:00:03:00:00:03
```

show protection-group ethernet-ring aps detail (EX2300 and EX3400 Switches)

user@switch>**show protection-group ethernet-ring aps detail**

```
Ethernet-Ring name             : pg1001
Request/State                  : NR
Ring Protection Link blocked   : Yes
No Flush Flag                  : Yes
Blocked Port Reference         : 0
Originator                     : Yes
Remote Node ID                 : NA
```

show protection-group ethernet-ring configuration

Syntax

```
show protection-group ethernet-ring configuration
```

Release Information

Command introduced in Junos OS Release 12.1 for EX Series switches.
Command introduced in Junos OS Release 14.1 for MX Series routers.
Command introduced in Junos OS Release 18.1 for EX2300 and EX3400 switches.

Description

Display the configuration of Ethernet ring protection group on EX Switches and MX Series routers.

Required Privilege Level

view

RELATED DOCUMENTATION

- show protection-group ethernet-ring aps | 1595
- show protection-group ethernet-ring data-channel | 1609
- show protection-group ethernet-ring interface | 1612
- show protection-group ethernet-ring node-state | 1617
- show protection-group ethernet-ring statistics | 1624
- show protection-group ethernet-ring vlan | 1631

List of Sample Output

show protection-group ethernet-ring configuration (EX Switch) on page 1603
show protection-group ethernet-ring configuration detail (MX Series Router) on page 1604
show protection-group ethernet-ring configuration (MX Series Router) on page 1604
show protection-group ethernet-ring configuration detail (MX Series Router) on page 1605
show protection-group ethernet-ring configuration detail (MX Series Router) on page 1605
show protection-group ethernet-ring configuration (MX Series Router) on page 1606
show protection-group ethernet-ring configuration detail (MX Series Router) on page 1607

Output Fields

Table 173 on page 1601 lists the output fields for the **show protection-group ethernet-ring configuration** command. Output fields are listed in the approximate order in which they appear.

Table 173: show protection-group ethernet-ring configuration Output Fields

Output Fields	Field Description
G8032 Compatability Version	This is the compatibility version mode of ERP. This parameter always takes the value 1 in the case of G8032v1. This parameter is valid only for MX Series routers.
East Interface	One of the two switch interfaces that participates in a ring link. When Junos supports G8032v2, this interface is treated as interface 0.
West Interface	One of the two interfaces in a switch that participates in a ring link. When Junos supports G8032v2, this interface is treated as interface 1.
Restore Interval	Configured interval of wait time after a link is restored. When a link goes down, the RPL link is activated. When the down link becomes active again, the RPL owner receives a notification. The RPL owner waits for the restore interval before issuing a block on the RPL link. The configured restore interval can be 5 through 12 minutes for ERPv1 and 1 through 12 minutes for ERPv2. This configuration is a global configuration and applies to all Ethernet rings if the Ethernet ring does not have a more specific configuration for this value. If no parameter is configured at the protection group level, the global configuration of this parameter uses the default value. **NOTE:** Wait to Restore (WTR) configuration values on EX2300 and EX3400 switches must be 5-12 minutes.
Wait to Block Interval	Configured interval of wait time for link restoration when a manual command (manual switch or force switch) is cleared. On clearing the manual command, the RPL owner receives NR messages, which starts a timer with interval 'Wait to Block' to restore the RPL link after its expiration. This delay timer is set to be 5 seconds longer than the guard timer. The configured number can be from 5 seconds through 10 seconds. The parameter is valid only for G.8032v2. **NOTE:** The Wait To Block Timer (WTB) is always disabled on EX2300 and EX3400 switches because it is not supported in ERPSv1. Any configuration you make to the WTB setting has no effect. The output from the CLI command 'show protection-group ethernet-ring node-state detail' lists a WTB setting but that setting has no effect.

Table 173: show protection-group ethernet-ring configuration Output Fields *(continued)*

Output Fields	Field Description
Guard Interval	Configured number of milliseconds (in 10 millisecond intervals, 10 milliseconds through 2000 milliseconds) that the node does not process any Ethernet ring protection protocol data units (PDUs). This configuration is a global configuration and applies to all Ethernet rings if the Ethernet ring does not have a more specific configuration for this value. If no parameter is configured at the protection group level, the global configuration of this parameter uses the default value.
Hold off interval	This is the interval at which the link is held down even before declaring that the link is down. Because the parameter is not supported at present, its value is always considered 0. This parameter is valid only for MX Series routers.
Node ID	Node ID for the switch or router. If the node ID is not configured, it is assigned by default. For EX Series switches, the Node ID value cannot be configured, whereas for MX Series routers, it can be configured.
Ring ID	In G8032v2, the ring ID can be within the range 1–239. All the nodes in a ring should have the same ring ID. In the case of G8032v1, the value of the ring ID is always 1. This parameter is valid only for MX Series routers.
Node Role	Indicates whether the ring node is operating as a normal ring-node or RPL-owner or RPL-neighbor. For G8032v1 RPL-neighbor role is not supported. This parameter is valid only for MX Series routers.
Revertive Mode of Operation	This parameter indicates whether the ring is operating in revertive mode or nonrevertive mode. In nonrevertive mode of operation, when all links in the ring and Ethernet Ring Nodes have recovered and no external requests are active, the Ethernet Ring does not automatically revert. G8032v1 supports only revertive mode of operation. This parameter is valid only for MX Series routers.
RAPS Tx Dot1p priority	The RAPS Tx Dot1p priority is a parameter with which the RAPS is transmitted from the ring node. For G8032v1, the value of this parameter is always 0. For G8032v2, the value of this parameter can be within the range 0–7. This parameter is valid only for MX Series routers.

Table 173: show protection-group ethernet-ring configuration Output Fields *(continued)*

Output Fields	Field Description
Node type	Indicates whether ring node is a normal ring node having two ring-links or a open ring-node having only a single ring-link or a interconnection ring-node. An interconnection ring node can be connected to major ring in non virtual-channel mode or in virtual channel mode. Ring interconnection is not supported for G8032v1. This parameter is valid only for MX Series routers.
Major ring name	If the node type is interconnection in the ring, this parameter takes the name of the major ring to which the sub-ring node is connected. This parameter is valid only for MX Series routers.
Interconnection mode	Indicates the interconnection mode if the type of the node is interconnection. An interconnection ring node can be connected to major ring in non-virtual channel mode or in virtual channel mode. This parameter is valid only for MX Series routers.
Propagate Topology Change event	When Propagate Topology Change event is set to 1, the change in the topology of sub-ring is propagated to the major ring, enabling the transmission of EVENT FLUSH RAPS PDU in the major ring. When the parameter is set to 0, the topology change in the sub-ring is not propagated to the major ring blocking EVENT FLUSH RAPS PDU transmission in the major ring. This parameter is valid only for MX Series routers.
Control Vlan	The VLAN that transfers ERP PDUs from one node to another.
Physical Ring	Physical ring if the east and west interfaces are nontrunk ports. For MX Series routers, the ring is termed a physical ring if no data channels are defined for the ring and the entire physical port forwarding is controlled by ERP.
Data Channel VLAN(s)	Data VLANs for which forwarding behavior is controlled by the ring instance.

Sample Output

show protection-group ethernet-ring configuration (EX Switch)

```
user@switch>show protection-group ethernet-ring configuration
```

```
Ethernet ring configuration parameters for protection group erp1
East Interface   : ge-0/0/3.0
West Interface   : ge-0/0/9.0
Restore Interval : 5 minutes
Guard Interval   : 500 ms
Node Id          : 00:1F:12:30:B8:81
Control Vlan     : 101
Physical Ring    : yes
```

show protection-group ethernet-ring configuration detail (MX Series Router)

user@switch>**show protection-group ethernet-ring configuration detail**

```
Ethernet Ring configuration information for protection group pg_101
G8032 Compatibility Version                  : 2
East interface (interface 0)                 : xe-2/3/0.1
West interface (interface 1)                 : xe-2/2/1.1
Restore interval                             : 5 minutes
Wait to Block interval                       : 5 seconds
Guard interval                               : 500 ms
Hold off interval                            : 0 ms
Node ID                                      : 64:87:88:65:37:D0
Ring ID (1 ... 239)                          : 1
Node role (normal/rpl-owner/rpl-neighbour)   : normal
Revertive mode of operation                  : 1
RAPS Tx Dot1p priority (0 .. 7)              : 0
Node type (normal/open/interconnection)      : Normal
Control Vlan                                 : 100
Physical Ring                                : No
Data Channel Vlan(s)                         : 200,300
```

show protection-group ethernet-ring configuration (MX Series Router)

user@switch>**show protection-group ethernet-ring configuration**

```
Ethernet Ring configuration information for protection group pg_101
G8032 Compatibility Version                  : 2
East interface (interface 0)                 : xe-2/3/0.1
West interface (interface 1)                 : xe-2/2/1.1
Restore interval                             : 5 minutes
Wait to Block interval                       : 5 seconds
Guard interval                               : 500 ms
Hold off interval                            : 0 ms
```

```
Node ID                                      : 64:87:88:65:37:D0
Ring ID (1 ... 239)                          : 1
Node role (normal/rpl-owner/rpl-neighbour)   : rpl-neighbour
Node RPL end                                 : east-port
Revertive mode of operation                  : 1
RAPS Tx Dot1p priority (0 .. 7)              : 0
Node type (normal/open/interconnection)      : Normal
Control Vlan                                 : 100
Physical Ring                                : No
Data Channel Vlan(s)                         : 200,300
```

show protection-group ethernet-ring configuration detail (MX Series Router)

user@switch>**show protection-group ethernet-ring configuration detail**

```
Ethernet Ring configuration information for protection group pg_101
G8032 Compatibility Version                  : 2
East interface (interface 0)                 : xe-2/3/0.1
West interface (interface 1)                 : xe-2/2/1.1
Restore interval                             : 5 minutes
Wait to Block interval                       : 5 seconds
Guard interval                               : 500 ms
Hold off interval                            : 0 ms
Node ID                                      : 64:87:88:65:37:D0
Ring ID (1 ... 239)                          : 1
Node role (normal/rpl-owner/rpl-neighbour)   : rpl-owner
Node RPL end                                 : east-port
Revertive mode of operation                  : 1
RAPS Tx Dot1p priority (0 .. 7)              : 0
Node type (normal/open/interconnection)      : Normal
Control Vlan                                 : 100
Physical Ring                                : No
Data Channel Vlan(s)                         : 200,300
```

show protection-group ethernet-ring configuration detail (MX Series Router)

user@switch>**show protection-group ethernet-ring configuration detail**

```
Ethernet Ring configuration information for protection group pg_101
G8032 Compatibility Version                  : 2
East interface (interface 0)                 : xe-2/3/0.1
West interface (interface 1)                 : (no erp)
Restore interval                             : 5 minutes
```

```
Wait to Block interval                        : 5 seconds
Guard interval                                : 500 ms
Hold off interval                             : 0 ms
Node ID                                       : 64:87:88:65:37:D0
Ring ID (1 ... 239)                           : 1
Node role (normal/rpl-owner/rpl-neighbour)    : rpl-owner
Node RPL end                                  : east-port
Revertive mode of operation                   : 1
RAPS Tx Dot1p priority (0 .. 7)               : 0
Node type (normal/open/interconnection)       : Open
Control Vlan                                  : 100
Physical Ring                                 : No
Data Channel Vlan(s)                          : 200,300
```

show protection-group ethernet-ring configuration (MX Series Router)

user@switch>**show protection-group ethernet-ring configuration**

```
Ethernet Ring configuration information for protection group pg_major
G8032 Compatibility Version                   : 2
East interface (interface 0)                  : xe-2/3/0.1
West interface (interface 1)                  : xe-2/2/1.1
Restore interval                              : 5 minutes
Wait to Block interval                        : 5 seconds
Guard interval                                : 500 ms
Hold off interval                             : 0 ms
Node ID                                       : 64:87:88:65:37:D0
Ring ID (1 ... 239)                           : 1
Node role (normal/rpl-owner/rpl-neighbour)    : rpl-owner
Node RPL end                                  : east-port
Revertive mode of operation                   : 1
RAPS Tx Dot1p priority (0 .. 7)               : 0
Node type (normal/open/interconnection)       : Normal
Control Vlan                                  : 100
Physical Ring                                 : No
Data Channel Vlan(s)                          : 200,300
```

```
Ethernet Ring configuration information for protection group pg_subring
G8032 Compatibility Version                   : 2
East interface (interface 0)                  : ge-2/0/0.1
West interface (interface 1)                  : (no erp)
Restore interval                              : 5 minutes
Wait to Block interval                        : 5 seconds
```

```
Guard interval                                : 500 ms
Hold off interval                             : 0 ms
Node ID                                       : 64:87:88:65:37:D0
Ring ID (1 ... 239)                           : 2
Node role (normal/rpl-owner/rpl-neighbour)    : normal
Revertive mode of operation                   : 1
RAPS Tx Dot1p priority (0 .. 7)               : 0
Node type (normal/open/interconnection)       : Non-VC-Interconnection
Major ring name                               : pg_major
Interconnection mode (VC/Non-VC)              : Non-VC mode
Propagate Topology Change event               : 0
Control Vlan                                  : 101
Physical Ring                                 : No
Data Channel Vlan(s)                          : 200,300
```

show protection-group ethernet-ring configuration detail (MX Series Router)

user@switch>**show protection-group ethernet-ring configuration detail**

```
Ethernet Ring configuration information for protection group pg_major
G8032 Compatibility Version                   : 2
East interface (interface 0)                  : xe-2/3/0.1
West interface (interface 1)                  : xe-2/2/1.1
Restore interval                              : 5 minutes
Wait to Block interval                        : 5 seconds
Guard interval                                : 500 ms
Hold off interval                             : 0 ms
Node ID                                       : 64:87:88:65:37:D0
Ring ID (1 ... 239)                           : 1
Node role (normal/rpl-owner/rpl-neighbour)    : rpl-owner
Node RPL end                                  : east-port
Revertive mode of operation                   : 1
RAPS Tx Dot1p priority (0 .. 7)               : 0
Node type (normal/open/interconnection)       : Normal
Control Vlan                                  : 100
Physical Ring                                 : No
Data Channel Vlan(s)                          : 200,300
```

```
Ethernet Ring configuration information for protection group pg_subring
G8032 Compatibility Version                   : 2
East interface (interface 0)                  : ge-2/0/0.1
West interface (interface 1)                  : (no erp)
Restore interval                              : 5 minutes
```

```
Wait to Block interval                          : 5 seconds
Guard interval                                  : 500 ms
Hold off interval                               : 0 ms
Node ID                                         : 64:87:88:65:37:D0
Ring ID (1 ... 239)                             : 2
Node role (normal/rpl-owner/rpl-neighbour)      : normal
Revertive mode of operation                     : 1
RAPS Tx Dot1p priority (0 .. 7)                 : 0
Node type (normal/open/interconnection)         : Non-VC-Interconnection
Major ring name                                 : pg_major
Interconnection mode (VC/Non-VC)                : Non-VC mode
Propagate Topology Change event                 : 0
Control Vlan                                    : 101
Physical Ring                                   : No
Data Channel Vlan(s)                            : 200,300
```

show protection-group ethernet-ring data-channel

Syntax

```
show protection-group ethernet-ring data-channel
<brief | detail>
<group-name group-name>
```

Release Information

Command introduced in Junos OS Release 10.2.
Command introduced in Junos OS Release 18.1 for EX2300 and EX3400 switches.

Description

Display the configuration of Ethernet ring protection group on EX Switches and MX Series routers.

Options

brief | detail—(Optional) Display the specified level of output.

group-name—(Optional) Protection group for which to display statistics. If you omit this optional field, all protection group statistics for configured groups will be displayed.

Required Privilege Level

view

RELATED DOCUMENTATION

- show protection-group ethernet-ring aps | **1595**
- show protection-group ethernet-ring interface | **1612**
- show protection-group ethernet-ring node-state | **1617**
- show protection-group ethernet-ring statistics | **1624**
- show protection-group ethernet-ring vlan | **1631**

List of Sample Output

show protection-group ethernet-ring data-channel on page 1610
show protection-group ethernet-ring data-channel detail on page 1610
show protection-group ethernet-ring data-channel detail (EX2300 and EX3400 Switches) on page 1611

Output Fields

Table 174 on page 1610 lists the output fields for the **show protection-group ethernet-ring data-channel** command. Output fields are listed in the approximate order in which they appear.

Table 174: show protection-group ethernet-ring data-channel Output Fields

Field Name	Field Description
Interface	Name of the interface configured for the Ethernet ring.
STP index	The Spanning Tree Protocol (STP) index number used by each interface in an Ethernet ring. The STP index controls the forwarding behavior for a set of VLANs on a data channel on an Ethernet ring port. For multiple Ethernet ring instances on an physical ring port, there are multiple STP index numbers. Different ring instances will have different STP index numbers and may have different forwarding behavior.
Forward State	Forwarding state on the Ethernet ring. • **fowarding**—Indicates packets are being forwarded. • **discarding**—Indicates packets are being discarded.

Sample Output

show protection-group ethernet-ring data-channel

```
user@host> show protection-group ethernet-ring data-channel
```

```
Ethernet ring data channel information for protection group pg301

Interface    STP index  Forward State
xe-5/0/2     78         forwarding
xe-2/2/0     79         discarding

Ethernet ring data channel parameters for protection group pg302

Interface    STP index  Forward State
xe-5/0/2     80         forwarding
xe-2/2/0     81         forwarding
```

show protection-group ethernet-ring data-channel detail

```
user@host> show protection-group ethernet-ring data-channel detail
```

```
Ethernet ring data channel parameters for protection group pg301

Interface name                : xe-5/0/2
STP index                     : 78
Forward State                 : forwarding

Interface name                : xe-2/2/0
STP index                     : 79
Forward State                 : discarding


Ethernet ring data channel parameters for protection group pg302

Interface name                : xe-5/0/2
STP index                     : 80
Forward State                 : forwarding

Interface name                : xe-2/2/0
STP index                     : 81
Forward State                 : forwarding
```

show protection-group ethernet-ring data-channel detail (EX2300 and EX3400 Switches)

user@switch>**show protection-group ethernet-ring data-channel detail**

```
Ethernet ring data channel parameters for protection group pg1001

Interface name                : ge-0/0/42
STP index                     : 52
Forward State                 : discarding

Interface name                : ge-0/0/38
STP index                     : 53
Forward State                 : forwarding
```

show protection-group ethernet-ring interface

Syntax

```
show protection-group ethernet-ring interface
```

Release Information

Command introduced in Junos OS Release 9.4.
Command introduced in Junos OS Release 12.3X54 for ACX Series routers.
Command introduced in Junos OS Release 18.1 for EX2300 and EX3400 switches.

Description

Displays the status of the Automatic Protection Switching (APS) interfaces on an Ethernet ring.

Options

This command has no options.

Required Privilege Level

view

RELATED DOCUMENTATION

- show protection-group ethernet-ring data-channel | 1609
- show protection-group ethernet-ring aps | 1595
- show protection-group ethernet-ring node-state | 1617
- show protection-group ethernet-ring statistics | 1624
- show protection-group ethernet-ring vlan | 1631

List of Sample Output

Output Fields

Table 175 on page 1613 lists the output fields for both the EX Series switch, and the ACX Series and MX Series router **show protection-group ethernet-ring interface** commands. Output fields are listed in the approximate order in which they appear.

Table 175: MX Series Routers show protection-group ethernet-ring interface Output Fields

Field Name	Field Description
Ethernet ring port parameters for protection group *group-name*	Output is organized by configured protection group.
Interface	Physical interfaces configured for the Ethernet ring. This can be an aggregated Ethernet link also.
Control Channel	(MX Series router only) Logical unit configured on the physical interface.
Direction	Direction of the traffic.
Forward State	State of the ring forwarding on the interface: **discarding** or **forwarding**.
Ring Protection Link End	Whether this interface is the end of the ring: **Yes** or **No**.
Signal Failure	Whether there a signal failure exists on the link: **Clear** or **Set**.
Admin State	State of the interface: For EX switches, **ready**, **ifl ready**, or **waiting**. For MX routers, **IFF ready** or **IFF disabled**.

Sample Output

show protection-group ethernet-ring interface (EX Series Switch Owner Node)

`user@host>` **show protection-group ethernet-ring interface**

```
Ethernet ring port parameters for protection group pg101

Interface      Forward State  RPL End  Signal Failure  Admin State

ge-0/0/3.0     discarding     Yes        Clear          ready
ge-0/0/9.0     forwarding     No         Clear          ready
```

show protection-group ethernet-ring interface (Owner Node MX Series Router)

```
user@host> show protection-group ethernet-ring interface
```

```
Ethernet ring port parameters for protection group pg101

Interface  Control Channel  Direction  Forward State  RPL End  SF     Admin State
ge-1/2/0   ge-1/2/0.100     east       forwarding     No       Clear  IFF ready
ge-1/2/2   ge-1/2/2.100     west       forwarding     No       Clear  IFF ready
```

show protection-group ethernet-ring interface detail (Owner Node MX Series Router)

```
user@host> show protection-group ethernet-ring interface detail
```

```
Ethernet ring port parameters for protection group pg101

Interface name                 : ge-1/2/0
Control channel name           : ge-1/2/0.100
Interface direction            : east
Ring Protection Link End       : No
Signal Failure                 : Clear
Forward State                  : forwarding
Interface Admin State          : IFF ready

Interface name                 : ge-1/2/2
Control channel name           : ge-1/2/2.100
Interface direction            : west
Ring Protection Link End       : No
Signal Failure                 : Clear
Forward State                  : forwarding
Interface Admin State          : IFF ready
```

show protection-group ethernet-ring interface (EX Series Switch Ring Node)

```
user@host> show protection-group ethernet-ring interface
```

```
Ethernet ring port parameters for protection group pg102

Ethernet ring port parameters for protection group pg101

Interface      Forward State  RPL End  Signal Failure  Admin State

ge-0/0/3.0     discarding     Yes        Clear          ready
```

```
ge-0/0/9.0     forwarding     No         Clear          ready
```

show protection-group ethernet-ring interface detail (ACX Series and MX Series)

`user@host>` **show protection-group ethernet-ring interface detail**

```
Ethernet ring port parameters for protection group Erp_1

Interface name                : xe-0/0/0
Control channel name          : xe-0/0/0.1
Interface direction           : east
Ring Protection Link End      : No
Signal Failure                : Clear
Forward State                 : forwarding
Interface Admin State         : IFF ready

Interface name                : et-0/0/48
Control channel name          : et-0/0/48.1
Interface direction           : west
Ring Protection Link End      : No
Signal Failure                : Clear
Forward State                 : forwarding
Interface Admin State         : IFF ready
```

show protection-group ethernet-ring interface detail (EX2300 and EX3400 Switches)

`user@switch>`**show protection-group ethernet-ring interface detail**

```
Ethernet ring port parameters for protection group pg1001

Interface name                : ge-0/0/14
Control channel name          : ge-0/0/14.0
Interface direction           : east
Ring Protection Link End      : No
Signal Failure                : Clear
Forward State                 : forwarding
Interface Admin State         : IFF ready

Interface name                : ge-0/0/18
Control channel name          : ge-0/0/18.0
Interface direction           : west
Ring Protection Link End      : No
```

```
Signal Failure                : Clear
Forward State                 : forwarding
Interface Admin State         : IFF ready
```

show protection-group ethernet-ring interface detail (EX2300 and EX3400 Switches)

user@switch>**show protection-group ethernet-ring interface detail**

```
Ethernet ring port parameters for protection group pg1001

Interface name                : ge-0/0/42
Control channel name          : ge-0/0/42.0
Interface direction           : east
Ring Protection Link End      : Yes
Signal Failure                : Clear
Forward State                 : discarding
Interface Admin State         : IFF ready

Interface name                : ge-0/0/38
Control channel name          : ge-0/0/38.0
Interface direction           : west
Ring Protection Link End      : No
Signal Failure                : Clear
Forward State                 : forwarding
Interface Admin State         : IFF ready
```

show protection-group ethernet-ring node-state

Syntax

```
show protection-group ethernet-ring node-state
```

Release Information

Command introduced in Junos OS Release 9.4 for MX Series routers.
Command introduced in Junos OS Release 12.1 for EX Series switches.
Command introduced in Junos OS Release 12.3X54 for ACX Series routers.
Command introduced in Junos OS Release 18.1 for EX2300 and EX3400 switches.

Description

Display the status of the Automatic Protection Switching (APS) nodes on an Ethernet ring.

Options

This command has no options.

Required Privilege Level

view

RELATED DOCUMENTATION

- show protection-group ethernet-ring data-channel | 1609
- show protection-group ethernet-ring aps | 1595
- show protection-group ethernet-ring interface | 1612
- show protection-group ethernet-ring statistics | 1624
- show protection-group ethernet-ring vlan | 1631

List of Sample Output

show protection-group ethernet-ring node-state detail (MX Series Router with WTB Timer) on page 1622
show protection-group ethernet-ring node-state detail (EX2300 and EX3400 Switches) on page 1622

Output Fields

Table 176 on page 1618 lists the output fields for the **show protection-group ethernet-ring node-state** command. Output fields are listed in the approximate order in which they appear.

Table 176: show protection-group ethernet-ring node-state Output Fields

Field Name	Field Description
Ring Name/Ethernet Ring	Name configured for the Ethernet ring.
APS State	State of the Ethernet ring APS. • **idle**—Indicates that the ring is working in normal condition and there is no active or pending protection-switching request in the ring. When the ring is in idle state, it is blocked at the RPL link. • **protected**—Indicates that there is a protection switch on the ring because of a signal failure condition on the ring link. • **MS**—Indicates that the manual switch command is active in the ring. • **FS**—Indicates that the forced switch command is active in the ring. • **pending**—Indicates that the ring is in pending state.

Table 176: show protection-group ethernet-ring node-state Output Fields *(continued)*

Field Name	Field Description
Event	Events on the ring. • **NR-RB**—Indicates that there is no APS request and the ring link is blocked on the ring owner node. • **NR**—Indicates that there is no APS request pending in the ring. • **local SF**—Indicates that there is signal failure on one or both of the ring links of the node. • **remote SF**—Indicates that there is signal failure on one or more ring links of any other node of the ring. • **local FS**—Indicates that there is a forced switched command active on one or both of the ring links of the node. • **remote FS**—Indicates that there is a forced switch command active on one or more ring links of any other node of the ring. • **local MS**—Indicates that there is a manual switch command active on one of the ring links of the node. • **remote MS**—Indicates that there is a manual switch command active on one or more ring links of any other node of the ring. • **WTR running**—Indicates that the wait to restore timer is running on the RPL owner. • **WTB running**—Indicates that the wait to block timer is running on the RPL owner.
RPL Owner / Ring Protection Link Owner	Whether this node is the ring owner: **Yes** or **No**.
WTR Timer / Restore Timer	Restoration timer: **running** or **disabled**.
WTB Timer / Wait to block timer	Wait to block timer: **running** or **disabled**. **NOTE:** The Wait To Block Timer (WTB) is always disabled on EX2300 and EX3400 switches because it is not supported in ERPSv1. Any configuration you make to the WTB setting has no effect. The output from the CLI command 'show protection-group ethernet-ring node-state detail' lists a WTB setting but that setting has no effect.

Table 176: show protection-group ethernet-ring node-state Output Fields *(continued)*

Field Name	Field Description
Wait to block timer (WTB Timer)	Wait to block interval. **NOTE:** The Wait To Block Timer (WTB) is always disabled on EX2300 and EX3400 switches because it is not supported in ERPSv1. Any configuration you make to the WTB setting has no effect. The output from the CLI command 'show protection-group ethernet-ring node-state detail' lists a WTB setting but that setting has no effect.
Guard Timer	Guard timer: **running** or **disabled**.
Op State / Operational State	State of the node: **Operational** or **any internal wait state.**.

Sample Output

show protection-group ethernet-ring node-state (MX Series Router - RPL Owner Node, Normal Operation)

```
user@host> show protection-group ethernet-ring node-state
```

```
Ethernet ring   APS State  Event      RPL Owner  WTR Timer  WTB Timer  Guard Timer
  Operation state
pg101           idle       NR-RB       Yes        disabled   disabled   disabled
    operational
pg102           idle       NR-RB       No         disabled   disabled   disabled
    operational
```

show protection-group ethernet-ring node-state (MX Series Router - Normal Ring Node, Normal Operation)

```
user@host> show protection-group ethernet-ring node-state
```

```
Ethernet ring    APS State    Event         RPL Owner
pg102            idle         NR-RB         No

WTR Timer  WTB Timer   Guard Timer  Operation state
disabled   disabled    disabled      operational
```

show protection-group ethernet-ring node-state (MX Series Router - RPL Owner Node, Remote Failure Condition)

```
user@host> show protection-group ethernet-ring node-state
```

```
Ethernet ring    APS State    Event         RPL Owner
pg101            protected    remote SF         Yes

WTR Timer   WTB Timer Guard Timer  Operation state
disabled    disabled   disabled     operational
```

show protection-group ethernet-ring node-state detail (ACX Series and MX Series Router)

```
user@host> show protection-group ethernet-ring node-state detail
```

```
Ethernet-Ring name             : Erp_1
APS State                      : idle
Event                          : NR-RB
Ring Protection Link Owner     : No
Wait to Restore Timer          : disabled
Wait to Block Timer            : disabled
Guard Timer                    : disabled
Operation state                : operational
```

show protection-group ethernet-ring node-state detail (MX Series Router - RPL Owner Node, Normal Operation)

```
user@host> show protection-group ethernet-ring node-state detail
```

```
Ethernet-Ring name             : pg101
APS State                      : idle
Event                          : NR-RB
Ring Protection Link Owner     : Yes
Wait to Restore Timer          : disabled
Wait to Block Timer            : disabled
Guard Timer                    : disabled
Operation state                : operational

Ethernet-Ring name             : pg102
APS State                      : idle
Event                          : NR-RB
Ring Protection Link Owner     : No
Wait to Restore Timer          : disabled
Wait to Block Timer            : disabled
Guard Timer                    : disabled
Operation state                : operational
```

show protection-group ethernet-ring node-state detail (MX Series Router with WTR Timer)

`user@host>` **show protection-group ethernet-ring node-state detail**

```
Ethernet-Ring name             : pg_major
APS State                      : pending
Event                          : WTR running
Ring Protection Link Owner     : Yes
Wait to Restore Timer          : running (time to expire: 269 sec)
Wait to Block Timer            : disabled
Guard Timer                    : disabled
Operation state                : operational

Ethernet-Ring name             : pg_subring
APS State                      : pending
Event                          : NR
Ring Protection Link Owner     : No
Wait to Restore Timer          : disabled
Wait to Block Timer            : disabled
Guard Timer                    : disabled
Operation state                : operational
```

show protection-group ethernet-ring node-state detail (MX Series Router with WTB Timer)

`user@host>` **show protection-group ethernet-ring node-state detail**

```
Ethernet-Ring name             : Pg-2
APS State                      : pending
Event                          : WTB running
Ring Protection Link Owner     : Yes
Wait to Restore Timer          : disabled
Wait to Block Timer            : running (time to expire: 2 sec)
Guard Timer                    : disabled
Operation state                : operational
```

show protection-group ethernet-ring node-state detail (EX2300 and EX3400 Switches)

`user@switch>`**show protection-group ethernet-ring node-state detail**

```
Ethernet-Ring name             : pg1001
APS State                      : idle
Event                          : NR-RB
Ring Protection Link Owner     : Yes
Wait to Restore Timer          : disabled
```

```
Wait to Block Timer          : disabled    <-field not supported. Always disabled.
Guard Timer                    : disabled
Operation state                : operational
```

show protection-group ethernet-ring statistics

Syntax

```
show protection-group ethernet-ring statistics group-name group-name
```

<brief | detail>

Release Information

Command introduced in Junos OS Release 9.4.
Command introduced in Junos OS Release 12.1 for EX Series switches.
Command introduced in Junos OS Release 12.3X54 for ACX Series routers.

Description

Display statistics regarding Automatic Protection Switching (APS) protection groups on an Ethernet ring.

Options

group-name—Display statistics for the protection group. If you omit this option, protection group statistics for all configured groups are displayed.

brief—Display brief statistics for the protection group.

detail—Display detailed statistics for the protection group.

Required Privilege Level

view

RELATED DOCUMENTATION

- show protection-group ethernet-ring data-channel | 1609
- show protection-group ethernet-ring aps | 1595
- show protection-group ethernet-ring node-state | 1617
- show protection-group ethernet-ring interface | 1612
- show protection-group ethernet-ring vlan | 1631

List of Sample Output

show protection-group ethernet-ring statisitics (Ring Node, Failure Condition on ACX and MX Router) on page 1629
show protection-group ethernet-ring statistics detail (EX2300 and EX3400 Switches) on page 1629
show protection-group ethernet-ring statistics detail (EX2300 and EX3400 Switches) on page 1630

Output Fields

Table 177 on page 1625 lists the output fields for the **show protection-group ethernet-ring statistics** command.

Table 177: show protection-group ethernet-ring statistics Output Fields

Field Name	Field Description
Ethernet Ring Statistics for PG	Name of the protection group for which statistics are displayed.
RAPS event sent	Number of times Ring Automatic Protection Switching (RAPS) message transmission event occurred locally. This field is applicable only to MX Series routers.
RAPS event received	Number of RAPS messages received and processed by ERP state-machine and which resulted in state transition. This field is applicable only to MX Series routers.
Local SF	Number of times a signal failure has occurred locally.
Remote SF	Number of times a signal failure has occurred anywhere else on the ring.
NR event	Number of times a No Request event has occurred on the ring. This field is applicable only to EX Series switches.
NR event sent	Number of times a No Request event has occurred locally. This field is applicable only to MX Series routers.
NR event received	Number of times a No Request event has occurred anywhere else on the ring. This field is applicable only to MX Series routers.
NR-RB event	Number of times a No Request, Ring Blocked event has occurred on the ring. This field is applicable only to EX Series switches.
NR-RB event sent	Number of times a No Request, Ring Blocked event has occurred locally. This field is applicable only to MX Series routers.
NR-RB event received	Number of times a No Request, Ring Blocked event has occurred anywhere else on the ring. This field is applicable only to MX Series routers.

Table 177: show protection-group ethernet-ring statistics Output Fields *(continued)*

Field Name	Field Description
Flush event sent	Number of times flush-event RAPS message transmission event occurred locally. This field is applicable only to MX Series routers.
Flush event received	Number of flush-event RAPS messages received and processed by the ring instance control process. This field is applicable only to MX Series routers.
Local FS event sent	Number of times a forced switch event has occurred locally. This field is applicable only to MX Series routers.
Remote FS event received	Number of times a forced switch event has occurred anywhere else on the ring. This field is applicable only to MX Series routers.
Local MS event sent	Number of times a manual switch event has occurred locally. This field is applicable only to MX Series routers.
Remote MS event received	Number of times a manual switch event has occurred anywhere else on the ring. This field is applicable only to MX Series routers.

Table 178 on page 1626 lists the output fields for the **show protection-group ethernet-ring statistics** command when the **detail** option is used. These fields are valid only for MX Series routers.

Table 178: show protection-group ethernet-ring statistics detail Output Fields (for MX Series Routers)

Field Name	Field Description
Total number of FDB flush	Number of times forwarding database (FDB) flush has happened for the ring instance.
Flush-logic triggered flush	Number of times FDB flush has happened because of flush-logic based on node ID and Blocked Port Reference (BPR).
Remote RAPS PDU received	Number of valid RAPS PDU messages received. This counter counts only RAPS messages generated by other devices on the ring.
Remote RAPS dropped due to guard-timer	Number of RAPS messages dropped by the device because the guard timer is running.
Invalid remote RAPS PDU dropped	Number of RAPS messages dropped by the device because the messages are invalid.
RAPS dropped due to miscellaneous errors	Number of RAPS messages dropped because of any other reason. For example, messages dropped because of unsupported functionality.

Table 178: show protection-group ethernet-ring statistics detail Output Fields (for MX Series Routers) *(continued)*

Field Name	Field Description
Local received RAPS PDU dropped	Number of self-generated RAPS messages received and dropped.

Sample Output

show protection-group ethernet-ring statistics (EX Series Switch)

```
user@switch> show protection-group ethernet-ring statistics
```

```
Ring Name Local SF Remote SF NR Event NR-RB Event
erp1       2       1       2        3
```

show protection-group ethernet-ring statistics (MX Series Router)

```
user@host> show protection-group ethernet-ring statistics
```

```
Ethernet Ring statistics for PG Pg-1
RAPS event sent                 : 1
RAPS event received             : 1152
Local SF happened:              : 0
Remote SF happened:             : 428
NR event sent:                  : 1
NR event received:              : 133
NR-RB event sent:               : 0
NR-RB event received:           : 591
Flush event sent                : 0
Flush event received:           : 0
Local FS event sent:            : 0
Remote FS event received:       : 0
Local MS event sent:            : 0
Remote MS event received:       : 0
```

show protection-group ethernet-ring statistics detail (Specific Group)(MX Series Router)

```
user@host> show protection-group ethernet-ring statistics detail
```

```
Ethernet Ring statistics for PG Pg-1
RAPS event sent                          : 1
RAPS event received                      : 0
Local SF happened                        : 0
Remote SF happened                       : 0
NR event sent                            : 1
NR event received                        : 0
NR-RB event sent                         : 0
NR-RB event received                     : 0
Flush event sent                         : 0
Flush event received                     : 0
Local FS event sent                      : 0
Remote FS event received                 : 0
Local MS event sent                      : 0
Remote MS event received                 : 0
Total number of FDB flush                : 0
Flush-logic triggered flush              : 0
Remote raps PDU received                 : 0
Remote raps dropped due to guard-timer   : 0
Invalid remote raps PDU dropped          : 0
Raps dropped due to miscellaneous errors : 0
Local received raps PDU dropped          : 0
```

show protection-group ethernet-ring statistics (Owner Node, Failure Condition on ACX and MX Router)

`user@host>` **show protection-group ethernet-ring statistics group-name pg101**

```
Ethernet Ring statistics for PG pg101
RAPS sent                        : 1
RAPS received                    : 0
Local SF happened:               : 0
Remote SF happened:              : 0
NR event happened:               : 0
NR-RB event happened:            : 1
NR event sent:                   : 0
NR event received:               : 0
NR-RB event sent:                : 1
NR-RB event received:            : 0
Flush event sent                 : 0
Flush event received:            : 0
Local FS event sent:             : 0
Remote FS event received:        : 0
Local MS event sent:             : 0
Remote MS event received:        : 0
```

show protection-group ethernet-ring statisitics (Ring Node, Failure Condition on ACX and MX Router)

`user@host>` **show protection-group ethernet-ring statistics group-name pg102**

```
Ethernet Ring statistics for PG pg102
RAPS sent                       : 1
RAPS received                   : 0
Local SF happened:              : 0
Remote SF happened:             : 0
NR event happened:              : 0
NR-RB event happened:           : 1
NR event sent:                  : 0
NR event received:              : 0
NR-RB event sent:               : 1
NR-RB event received:           : 0
Flush event sent                : 0
Flush event received:           : 0
Local FS event sent:            : 0
Remote FS event received:       : 0
Local MS event sent:            : 0
Remote MS event received:       : 0
```

show protection-group ethernet-ring statistics detail (EX2300 and EX3400 Switches)

`user@switch>`**show protection-group ethernet-ring statistics detail**

```
Ethernet Ring statistics for PG pg1001
RAPS event sent                          : 1
RAPS event received                      : 1
Local SF happened                        : 0
Remote SF happened                       : 0
NR event sent                            : 1
NR event received                        : 0
NR-RB event sent                         : 0
NR-RB event received                     : 1
Flush event sent                         : 0
Flush event received                     : 0
Local FS event sent                      : 0
Remote FS event received                 : 0
Local MS event sent                      : 0
Remote MS event received                 : 0
Total number of FDB flush                : 0
Flush-logic triggered flush              : 0
Remote raps PDU received                 : 145
Remote raps dropped due to guard-timer   : 0
```

```
Invalid remote raps PDU dropped          : 0
Raps dropped due to miscellaneous errors : 0
Local received raps PDU dropped          : 0
```

show protection-group ethernet-ring statistics detail (EX2300 and EX3400 Switches)

user@switch>**show protection-group ethernet-ring statistics detail**

```
Ethernet Ring statistics for PG pg1001
RAPS event sent                          : 2
RAPS event received                      : 0
Local SF happened                        : 0
Remote SF happened                       : 0
NR event sent                            : 1
NR event received                        : 0
NR-RB event sent                         : 1
NR-RB event received                     : 0
Flush event sent                         : 0
Flush event received                     : 0
Total number of FDB flush                : 0
Remote raps PDU received                 : 211
Remote raps dropped due to guard-timer   : 0
Invalid remote raps PDU dropped          : 0
Raps dropped due to miscellaneous errors : 0
Local received raps PDU dropped          : 91
```

show protection-group ethernet-ring vlan

Syntax

```
show protection-group ethernet-ring vlan
<brief | detail>
<group-name group-name>
```

Release Information

Command introduced in Junos OS Release 10.2.
Command introduced in Junos OS Release 18.1 for EX2300 and EX3400 switches.

Description

On MX Series routers, display all data channel logical interfaces and the VLAN IDs controlled by a ring instance data channel.

Options

brief | detail—(Optional) Display the specified level of output.

group-name—(Optional) Protection group for which to display details such as data channel interfaces, vlan, and bridge-domain. If you omit this optional field, details for all configured protection groups will be displayed.

Required Privilege Level

view

RELATED DOCUMENTATION

- show protection-group ethernet-ring aps | 1595
- show protection-group ethernet-ring data-channel | 1609
- show protection-group ethernet-ring interface | 1612
- show protection-group ethernet-ring node-state | 1617
- show protection-group ethernet-ring statistics | 1624

List of Sample Output

show protection-group ethernet-ring vlan on page 1632
show protection-group ethernet-ring vlan brief on page 1633
show protection-group ethernet-ring vlan detail on page 1634
show protection-group ethernet-ring vlan group-name vkm01 on page 1635

show protection-group ethernet-ring vlan detail (EX2300 and EX3400 Switches) on page 1636

Output Fields

Table 179 on page 1632 lists the output fields for the **show protection-group ethernet-ring vlan** command. Output fields are listed in the approximate order in which they appear.

Table 179: show protection-group ethernet-ring vlan Output Fields

Field Name	Field Description
Interface	Name of the interface configured for the Ethernet protection ring.
Vlan	Name of the VLAN associated with the interface configured for the Ethernet protection ring.
STP index	The Spanning Tree Protocol (STP) index number used by each interface in an Ethernet ring. The STP index controls the forwarding behavior for a set of VLANs on a data channel on an Ethernet ring port. For multiple Ethernet ring instances on an physical ring port, there are multiple STP index numbers. Different ring instances will have different STP index numbers and may have different forwarding behavior.
Bridge Domain	Name of the bridge domain that is associated with the VLAN configured for the Ethernet protection ring.

Sample Output

show protection-group ethernet-ring vlan

```
user@host> show protection-group ethernet-ring vlan
```

```
Ethernet ring IFBD parameters for protection group vkm01

Interface    Vlan    STP Index   Bridge Domain
xe-5/0/2     1       78          default-switch/bd1
xe-2/2/0     1       79          default-switch/bd1
xe-5/0/2     2       78          default-switch/bd2
xe-2/2/0     2       79          default-switch/bd2
xe-5/0/2     3       78          default-switch/bd3
xe-2/2/0     3       79          default-switch/bd3
xe-5/0/2     4       78          default-switch/bd4
xe-2/2/0     4       79          default-switch/bd4
xe-5/0/2     5       78          default-switch/bd5
```

```
xe-2/2/0     5       79          default-switch/bd5
xe-5/0/2     6       78          default-switch/bd6
xe-2/2/0     6       79          default-switch/bd6
xe-5/0/2     7       78          default-switch/bd7
xe-2/2/0     7       79          default-switch/bd7
xe-5/0/2     8       78          default-switch/bd8
xe-2/2/0     8       79          default-switch/bd8
xe-5/0/2     9       78          default-switch/bd9
xe-2/2/0     9       79          default-switch/bd9
xe-5/0/2     10      78          default-switch/bd10
xe-2/2/0     10      79          default-switch/bd10
xe-5/0/2     11      78          default-switch/bd11
xe-2/2/0     11      79          default-switch/bd11
xe-5/0/2     12      78          default-switch/bd12
xe-2/2/0     12      79          default-switch/bd12
xe-5/0/2     13      78          default-switch/bd13
xe-2/2/0     13      79          default-switch/bd13
xe-5/0/2     14      78          default-switch/bd14
xe-2/2/0     14      79          default-switch/bd14
xe-5/0/2     15      78          default-switch/bd15
xe-2/2/0     15      79          default-switch/bd15
```

show protection-group ethernet-ring vlan brief

`user@host>` **show protection-group ethernet-ring vlan brief**

```
Ethernet ring IFBD parameters for protection group vkm01

Interface    Vlan    STP Index   Bridge Domain
xe-5/0/2     1       78          default-switch/bd1
xe-2/2/0     1       79          default-switch/bd1
xe-5/0/2     2       78          default-switch/bd2
xe-2/2/0     2       79          default-switch/bd2
xe-5/0/2     3       78          default-switch/bd3
xe-2/2/0     3       79          default-switch/bd3
xe-5/0/2     4       78          default-switch/bd4
xe-2/2/0     4       79          default-switch/bd4
xe-5/0/2     5       78          default-switch/bd5
xe-2/2/0     5       79          default-switch/bd5
xe-5/0/2     6       78          default-switch/bd6
xe-2/2/0     6       79          default-switch/bd6
xe-5/0/2     7       78          default-switch/bd7
xe-2/2/0     7       79          default-switch/bd7
xe-5/0/2     8       78          default-switch/bd8
```

```
xe-2/2/0     8       79          default-switch/bd8
xe-5/0/2     9       78          default-switch/bd9
xe-2/2/0     9       79          default-switch/bd9
xe-5/0/2     10      78          default-switch/bd10
xe-2/2/0     10      79          default-switch/bd10
xe-5/0/2     11      78          default-switch/bd11
xe-2/2/0     11      79          default-switch/bd11
xe-5/0/2     12      78          default-switch/bd12
xe-2/2/0     12      79          default-switch/bd12
xe-5/0/2     13      78          default-switch/bd13
xe-2/2/0     13      79          default-switch/bd13
xe-5/0/2     14      78          default-switch/bd14
xe-2/2/0     14      79          default-switch/bd14
xe-5/0/2     15      78          default-switch/bd15
xe-2/2/0     15      79          default-switch/bd15
```

show protection-group ethernet-ring vlan detail

`user@host>` **show protection-group ethernet-ring vlan detail**

```
Ethernet ring IFBD parameters for protection group vkm01

Interface name                : xe-5/0/2
Vlan                          : 1
STP index                     : 78
Bridge Domain                 : default-switch/bd1

Interface name                : xe-2/2/0
Vlan                          : 1
STP index                     : 79
Bridge Domain                 : default-switch/bd1

Interface name                : xe-5/0/2
Vlan                          : 2
STP index                     : 78
Bridge Domain                 : default-switch/bd2

Interface name                : xe-2/2/0
Vlan                          : 2
STP index                     : 79
Bridge Domain                 : default-switch/bd2

Interface name                : xe-5/0/2
Vlan                          : 3
```

```
STP index                       : 78
Bridge Domain                   : default-switch/bd3
```

show protection-group ethernet-ring vlan group-name vkm01

```
user@host> show protection-group ethernet-ring vlan vkm01

Ethernet ring IFBD parameters for protection group vkm01

Interface    Vlan    STP Index   Bridge Domain
xe-5/0/2     16      80          default-switch/bd16
xe-2/2/0     16      81          default-switch/bd16
xe-5/0/2     17      80          default-switch/bd17
xe-2/2/0     17      81          default-switch/bd17
xe-5/0/2     18      80          default-switch/bd18
xe-2/2/0     18      81          default-switch/bd18
xe-5/0/2     19      80          default-switch/bd19
xe-2/2/0     19      81          default-switch/bd19
xe-5/0/2     20      80          default-switch/bd20
xe-2/2/0     20      81          default-switch/bd20
xe-5/0/2     21      80          default-switch/bd21
xe-2/2/0     21      81          default-switch/bd21
xe-5/0/2     22      80          default-switch/bd22
xe-2/2/0     22      81          default-switch/bd22
xe-5/0/2     23      80          default-switch/bd23
xe-2/2/0     23      81          default-switch/bd23
xe-5/0/2     24      80          default-switch/bd24
xe-2/2/0     24      81          default-switch/bd24
xe-5/0/2     25      80          default-switch/bd25
xe-2/2/0     25      81          default-switch/bd25
xe-5/0/2     26      80          default-switch/bd26
xe-2/2/0     26      81          default-switch/bd26
xe-5/0/2     27      80          default-switch/bd27
xe-2/2/0     27      81          default-switch/bd27
xe-5/0/2     28      80          default-switch/bd28
xe-2/2/0     28      81          default-switch/bd28
xe-5/0/2     29      80          default-switch/bd29
xe-2/2/0     29      81          default-switch/bd29
xe-5/0/2     30      80          default-switch/bd30
xe-2/2/0     30      81          default-switch/bd30
```

show protection-group ethernet-ring vlan detail (EX2300 and EX3400 Switches)

user@switch>**show protection-group ethernet-ring vlan detail**

```
Ethernet ring IFBD parameters for protection group pg1001

Interface name                : ge-0/0/42
Vlan                          : 2001
STP index                     : 52
Bridge Domain                 : default-switch/vlan2001

Interface name                : ge-0/0/38
Vlan                          : 2001
STP index                     : 53
Bridge Domain                 : default-switch/vlan2001
```

show redundant-trunk-group

Syntax

```
show redundant-trunk-group <group-name group-name>
```

Release Information

Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series.

Description

Display information about redundant trunk groups.

Options

group-name *group-name*—Display information about the specified redundant trunk group.

Required Privilege Level

view

RELATED DOCUMENTATION

- Example: Configuring Redundant Trunk Links for Faster Recovery on EX Series Switches | 946
- Example: Configuring Redundant Trunk Links for Faster Recovery on Devices with ELS Support | 940
- Understanding Redundant Trunk Links (Legacy RTG Configuration) | 937

List of Sample Output

show redundant-trunk-group group-name Group1 on page 1638

Output Fields

Table 180 on page 1637 lists the output fields for the **show redundant-trunk-group** command. Output fields are listed in the approximate order in which they appear.

Table 180: show redundant-trunk-group Output Fields

Field Name	Field Description
Group name	Name of the redundant trunk port group.
Interface	Name of an interface belonging to the trunk port group.

Table 180: show redundant-trunk-group Output Fields *(continued)*

Field Name	Field Description
State	Operating state of the interface. • **Up** denotes the interface is up. • **Down** denotes the interface is down. • **Pri** denotes a primary interface. • **Act** denotes an active interface.
Time of last flap	Date and time at which the advertised link became unavailable, and then, available again.
Flap count	Total number of flaps since the last switch reboot.

Sample Output

show redundant-trunk-group group-name Group1

```
user@switch> show redundant-trunk-group group-name Group1
```

```
Group      Interface       State          Time of last flap        Flap
name                                                               Count

Group1     ge-0/0/45.0     UP/Pri/Act     Never                    0
           ge-0/0/47.0     UP             Never                    0
```

show system statistics arp

List of Syntax

Syntax

```
show system statistics arp
```

Syntax (EX Series Switches)

```
show system statistics arp
<all-members>
<local>
<member member-id>
```

Syntax (TX Matrix Router)

```
show system statistics arp
<all-chassis | all-lcc | lcc number | scc>
```

Syntax (TX Matrix Plus Router)

```
show system statistics arp
<all-chassis | all-lcc | lcc number | sfc number>
```

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
sfc option introduced for the TX Matrix Plus router in Junos OS Release 9.6.

Description

Display system-wide Address Resolution Protocol (ARP) statistics.

Options

none—Display system-wide ARP statistics.

all-chassis—(TX Matrix routers and TX Matrix Plus routers only) (Optional) Display ARP statistics for all the routers in the chassis.

all-lcc—(TX Matrix routers and TX Matrix Plus routers only) (Optional) On a TX Matrix router, display system-wide ARP statistics for all T640 routers connected to the TX Matrix router. On a TX Matrix Plus router, display system-wide ARP statistics for all routers connected to the TX Matrix Plus router

all-members—(EX4200 switches only) (Optional) Display ARP statistics for all members of the Virtual Chassis configuration.

lcc *number*—(TX Matrix routers and TX Matrix Plus routers only) (Optional) On a TX Matrix router, display ARP statistics for a specific T640 router that is connected to the TX Matrix router. On a TX Matrix Plus router, display ARP statistics for a specific router that is connected to the TX Matrix Plus router.

Replace *number* with the following values depending on the LCC configuration:

- 0 through 3, when T640 routers are connected to a TX Matrix router in a routing matrix.
- 0 through 3, when T1600 routers are connected to a TX Matrix Plus router in a routing matrix.
- 0 through 7, when T1600 routers are connected to a TX Matrix Plus router with 3D SIBs in a routing matrix.
- 0, 2, 4, or 6, when T4000 routers are connected to a TX Matrix Plus router with 3D SIBs in a routing matrix.

local—(EX4200 switches only) (Optional) Display ARP statistics for the local Virtual Chassis member.

member *member-id*—(EX4200 switches only) (Optional) Display ARP statistics for the specified member of the Virtual Chassis configuration. Replace ***member-id*** with a value from 0 through 9.

scc—(TX Matrix routers only) (Optional) Display ARP statistics for the TX Matrix router (or switch-card chassis).

sfc *number*—(TX Matrix Plus routers only) (Optional) Display ARP statistics for the TX Matrix Plus router. Replace ***number*** with 0.

Additional Information

By default, when you issue the **show system statistics arp** command on the master Routing Engine of a TX Matrix router or a TX Matrix Plus router, the command is broadcast to all the master Routing Engines of the LCCs connected to it in the routing matrix. Likewise, if you issue the same command on the backup Routing Engine of a TX Matrix or a TX Matrix Plus router, the command is broadcast to all backup Routing Engines of the LCCs that are connected to it in the routing matrix.

Required Privilege Level

view

RELATED DOCUMENTATION

Routing Matrix with a TX Matrix Plus Router Solutions Page

Example: Configuring Proxy ARP on an EX Series Switch

Verifying That Proxy ARP Is Working Correctly | 959

List of Sample Output

Sample Output

show system statistics arp

`user@host>` **show system statistics arp**

```
arp:
         184710 datagrams received
         2886 ARP requests received
         684 ARP replies received
         0 resolution request  received
         0 unrestricted proxy requests
         0 restricted proxy requests
         0 received proxy requests
         0 unrestricted proxy requests not proxied
         0 restricted proxy requests not proxied
         0 datagrams with bogus interface
         0 datagrams with incorrect length
         0 datagrams for non-IP protocol
         0 datagrams with unsupported op code
         0 datagrams with bad protocol address length
         0 datagrams with bad hardware address length
         0 datagrams with multicast source address
         0 datagrams with multicast target address
         0 datagrams with my own hardware address
         0 datagrams for an address not on the interface
         0 datagrams with a broadcast source address
         0 datagrams with source address duplicate to mine
         181140 datagrams which were not for me
         0 packets discarded waiting for resolution
         4 packets sent after waiting for resolution
         703 ARP requests sent
         2886 ARP replies sent
         0 requests for memory denied
```

```
        0 requests dropped on entry
        0 requests dropped during retry
        0 requests dropped due to interface deletion
        0 requests on unnumbered interfaces
        0 new requests on unnumbered interfaces
        0 replies for from unnumbered interfaces
        0 requests on unnumbered interface with non-subnetted donor
        0 replies from unnumbered interface with non-subnetted donor
```

show system statistics arp (EX Series Switches)

`user@host>` **show system statistics arp**

```
arp:
        186423 datagrams received
        88 ARP requests received
        88 ARP replies received
        0 resolution request  received
        0 unrestricted proxy requests
        0 restricted proxy requests
        0 received proxy requests
        0 proxy requests not proxied
        0 restricted proxy requests not proxied
        0 datagrams with bogus interface
        0 datagrams with incorrect length
        0 datagrams for non-IP protocol
        0 datagrams with unsupported op code
        0 datagrams with bad protocol address length
        0 datagrams with bad hardware address length
        0 datagrams with multicast source address
        0 datagrams with multicast source address
        0 datagrams with my own hardware address
        164 datagrams for an address not on the interface
        0 datagrams with a broadcast source address
        0 datagrams with source address duplicate to mine
        186075 datagrams which were not for me
        0 packets discarded waiting for resolution
        0 packets sent after waiting for resolution
        50 ARP requests sent
        88 ARP replies sent
        0 requests for memory denied
        0 requests dropped on entry
        0 requests dropped during retry
        0 requests dropped due to interface deletion
```

```
         0 requests on unnumbered interfaces
         0 new requests on unnumbered interfaces
         0 replies for from unnumbered interfaces
         0 requests on unnumbered interface with non-subnetted donor
         0 replies from unnumbered interface with non-subnetted donor
```

show system statistics arp (TX Matrix Plus Router)

`user@host>` **show system statistics arp**

```
sfc0-re0:
--------------------------------------------------------------------------
arp:
        487 datagrams received
        8 ARP requests received
        438 ARP replys received
        438 resolution requests received
        0 unrestricted proxy requests
        0 restricted proxy requests
        0 received proxy requests
        0 proxy requestss not proxied
        0 restricted-proxy requestss not proxied
        0 with bogus interface
        0 with incorrect length
        0 for non-IP protocol
        0 with unsupported op code
        0 with bad protocol address length
        0 with bad hardware address length
        0 with multicast source address
        0 with multicast target address
        0 with my own hardware address
        0 for an address not on the interface
        0 with a broadcast source address
        0 with source address duplicate to mine
        41 which were not for me
        0 packets discarded waiting for resolution
        438 packets sent after waiting for resolution
        1282 ARP requests sent
        8 ARP replys sent
        0 requests for memory denied
        0 requests dropped on entry
        0 requests dropped during retry
        0 requests dropped due to interface deletion
        0 requests on unnumbered interfaces
```

```
        0 new requests on unnumbered interfaces
        0 replies for from unnumbered interfaces
        0 requests on unnumbered interface with non-subnetted donor
        0 replies from unnumbered interface with non-subnetted donor

lcc0-re0:
--------------------------------------------------------------------------
arp:
        19 datagrams received
        0 ARP requests received
        1 ARP reply  received
        0 resolution requests received
        0 unrestricted proxy requests
        0 restricted proxy requests
        0 received proxy requests
        0 proxy requestss not proxied
        0 restricted-proxy requestss not proxied
        0 with bogus interface
        0 with incorrect length
        0 for non-IP protocol
        0 with unsupported op code
        0 with bad protocol address length
        0 with bad hardware address length
        0 with multicast source address
        0 with multicast target address
        0 with my own hardware address
        0 for an address not on the interface
        0 with a broadcast source address
        0 with source address duplicate to mine
        18 which were not for me
        0 packets discarded waiting for resolution
        0 packets sent after waiting for resolution
        8 ARP requests sent
        0 ARP replys sent
        0 requests for memory denied
        0 requests dropped on entry
        0 requests dropped during retry
        0 requests dropped due to interface deletion
        0 requests on unnumbered interfaces
        0 new requests on unnumbered interfaces
        0 replies for from unnumbered interfaces
        0 requests on unnumbered interface with non-subnetted donor
        0 replies from unnumbered interface with non-subnetted donor
```

```
lcc1-re0:
--------------------------------------------------------------------------
arp:
        17 datagrams received
        0 ARP requests received
        1 ARP reply  received
        0 resolution requests received
        0 unrestricted proxy requests
        0 restricted proxy requests
        0 received proxy requests
        0 proxy requestss not proxied
        0 restricted-proxy requestss not proxied
        0 with bogus interface
        0 with incorrect length
        0 for non-IP protocol
        0 with unsupported op code
        0 with bad protocol address length
        0 with bad hardware address length
        0 with multicast source address
        0 with multicast target address
        0 with my own hardware address
        0 for an address not on the interface
        0 with a broadcast source address
        0 with source address duplicate to mine
        16 which were not for me
        0 packets discarded waiting for resolution
        0 packets sent after waiting for resolution
        9 ARP requests sent
        0 ARP replys sent
        0 requests for memory denied
        0 requests dropped on entry
        0 requests dropped during retry
        0 requests dropped due to interface deletion
        0 requests on unnumbered interfaces
        0 new requests on unnumbered interfaces
        0 replies for from unnumbered interfaces
        0 requests on unnumbered interface with non-subnetted donor
        0 replies from unnumbered interface with non-subnetted donor

lcc2-re0:
--------------------------------------------------------------------------
arp:
        18 datagrams received
        1 ARP request  received
```

```
        1 ARP reply  received
        0 resolution requests received
        0 unrestricted proxy requests
        0 restricted proxy requests
        0 received proxy requests
        0 proxy requestss not proxied
        0 restricted-proxy requestss not proxied
        0 with bogus interface
        0 with incorrect length
        0 for non-IP protocol
        0 with unsupported op code
        0 with bad protocol address length
        0 with bad hardware address length
        0 with multicast source address
        0 with multicast target address
        0 with my own hardware address
        0 for an address not on the interface
        0 with a broadcast source address
        0 with source address duplicate to mine
        16 which were not for me
        0 packets discarded waiting for resolution
        0 packets sent after waiting for resolution
        9 ARP requests sent
        1 ARP reply  sent
        0 requests for memory denied
        0 requests dropped on entry
        0 requests dropped during retry
        0 requests dropped due to interface deletion
        0 requests on unnumbered interfaces
        0 new requests on unnumbered interfaces
        0 replies for from unnumbered interfaces
        0 requests on unnumbered interface with non-subnetted donor
        0 replies from unnumbered interface with non-subnetted donor

lcc3-re0:
--------------------------------------------------------------------------
arp:
        13 datagrams received
        0 ARP requests received
        1 ARP reply  received
        0 resolution requests received
        0 unrestricted proxy requests
        0 restricted proxy requests
        0 received proxy requests
```

```
0 proxy requestss not proxied
0 restricted-proxy requestss not proxied
0 with bogus interface
0 with incorrect length
0 for non-IP protocol
0 with unsupported op code
0 with bad protocol address length
0 with bad hardware address length
0 with multicast source address
0 with multicast target address
0 with my own hardware address
0 for an address not on the interface
0 with a broadcast source address
0 with source address duplicate to mine
12 which were not for me
0 packets discarded waiting for resolution
0 packets sent after waiting for resolution
8 ARP requests sent
0 ARP replys sent
0 requests for memory denied
0 requests dropped on entry
0 requests dropped during retry
0 requests dropped due to interface deletion
0 requests on unnumbered interfaces
0 new requests on unnumbered interfaces
0 replies for from unnumbered interfaces
0 requests on unnumbered interface with non-subnetted donor
0 replies from unnumbered interface with non-subnetted donor
```

show vlans

List of Syntax

Syntax (EX Series and QFX Series Switches)

```
show vlans
<brief | detail | extensive>
<dot1q-tunneling>
<management-vlan>
<sort-by (tag | name)>
<vlan-range-name>
<summary>
<vlan-name>
<vlan-range-name>
```

Syntax (EX Series with ELS Switches and MX Routers)

```
show vlans
<brief | detail | extensive>
<instance instance-name>
<logical-system logical-system-name>
<operational>
<vlan-name)>
<interface interface-name>
```

Syntax (SRX Devices)

```
show vlans
<brief | detail | extensive>
<interface interface-name>
<logical-system (logical-system | all)>
<operational>
```

Release Information

Command introduced in Junos OS Release 8.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 11.1 for the QFX Series.
Option **dot1q-tunneling** added in Junos OS Release 12.1 for the QFX Series.

Command introduced in Junos OS Release 12.3R2 for EX Series switches.
Option **interface** introduced in Junos OS Release 13.2X50-D10 (ELS).

Description

Display information about VLANs configured on bridged Ethernet interfaces. For interfaces configured to support a VoIP VLAN and a data VLAN, the **show vlans** command displays both tagged and untagged membership for those VLANs.

> **NOTE:** When a series of VLANs is created using the **vlan-range** statement, such VLAN names are preceded and followed by a double underscore. For example, a series of VLANs using the VLAN range 1 through 3 and the base VLAN name **marketing** would be displayed as **__marketing_1__**, **__marketing_2__**, and **__marketing_3__**.

> **NOTE:** To display an 802.1X supplicant successfully authenticated in multiple-supplicant mode with dynamic VLAN movement, use the **show vlans *vlan-name* extensive** operational mode command, where ***vlan-name*** is the dynamic VLAN.

Options

For EX Series and QFX Series switches:

none—Display information for all VLANs. VLAN information is displayed by VLAN name in ascending order.

brief | detail | extensive—(Optional) Display the specified level of output.

dot1q-tunneling—(Optional) Display VLANs with the Q-in-Q tunneling feature enabled.

management-vlan—(Optional) Display management VLANs.

sort-by (tag | name)—(Optional) Display VLANs in ascending order of VLAN IDs or VLAN names.

vlan-range-name—(Optional) Display VLANs in ascending order of VLAN range names.

summary—(Optional) Display the total number of VLANs and counts of VLANs by type—for example, the number of dynamic, 802.1Q-tagged, and Q-in-Q tunneled VLANs.

vlan-range-name—(Optional) Display information for the specified VLAN range. To display information for all members of the VLAN range, specify the base VLAN name—for example, **employee** for a VLAN range that includes **__employee_1__** through **__employee_10__**.

For EX Series with ELS Switches and MX Routers:

none—Display information for all VLANs.

brief | detail | extensive—(Optional) Display the specified level of output.

instance *instance-name*—(Optional) Display information for the specified routing instance.

logical-system *logical-system-name*—(Optional) Display Ethernet-switching statistics information for the specified logical system.

operational—(Optional) Display information for the operational routing instances.

***vlan-name*)**— (Optional) Display information about the specified VLAN.

interface *interface-name*—(Optional) Display information about the specified interface.

For SRX devices:

none—Display information for all VLANs.

brief | detail | extensive—(Optional) Display the specified level of output.

interface *interface-name*— (Optional) Display information about a specific interface.

logical system—(Optional) Display name of the logical system or all.

operational—(Optional) Display information for the operational switching instances.

Required Privilege Level
view

RELATED DOCUMENTATION

- Example: Setting Up Basic Bridging and a VLAN on Switches | **203**
- Example: Setting Up Bridging with Multiple VLANs | **236**
- Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch | **226**
- Example: Setting Up Bridging with Multiple VLANs for EX Series Switches | **265**
- Example: Configuring a Private VLAN on a Single EX Series Switch | **498**
- Example: Configuring a Private VLAN Spanning Multiple EX Series Switches | **545**
- Example: Setting Up Q-in-Q Tunneling on EX Series Switches | **925**
- Understanding Bridging and VLANs on Switches | **168**
- show ethernet-switching interfaces | **1485**

List of Sample Output

show vlans (EX Series and QFX Series) on page 1654
show vlans (Private VLANs on EX and QFX Series) on page 1655
show vlans brief (EX and QFX Series) on page 1655

Output Fields

Table 163 on page 1570 lists the output fields for the **show vlans** command. Output fields are listed in the approximate order in which they appear.

Table 181: show vlans Output Fields

Field Name	Field Description	Level of Output
Name	Name of a VLAN.	none, **brief**
Tag	802.1Q tag applied to this VLAN. If **none** is displayed, no tag is applied.	All levels
Interfaces	Interface associated with learned MAC addresses or **All-members** option (flood entry). An asterisk (*) beside the interface indicates that the interface is **UP**.	All levels
Address	IP address.	none, **brief**
Ports Active /Total	Number of interfaces associated with a VLAN: **Active** indicates interfaces that are **UP**, and **Total** indicates interfaces that are active and inactive.	**brief**

Table 181: show vlans Output Fields *(continued)*

Field Name	Field Description	Level of Output
VLAN	Name of a VLAN.	**detail**, **extensive**
Admin state	State of the interface. Values are: **enabled**—The interface is turned on, and the physical link is operational and can pass packets.	**detail**,**extensive**
MAC learning Status	Indicates if MAC learning is disabled.	**detail**, **extensive**
Description	Description for the VLAN.	**detail**,**extensive**
Primary IP	Primary IP address associated with a VLAN.	**detail**
Number of interfaces	Number of interfaces associated with a VLAN. Both the total number of interfaces and the number of active interfaces associated with a VLAN are displayed.	**detail**, **extensive**
STP	Spanning tree associated with a VLAN.	**detail**,**extensive**
Tagged interfaces	Tagged interfaces with which a VLAN is associated.	**detail**,**extensive**
Untagged interfaces	Untagged interfaces with which a VLAN is associated.	**detail**. **extensive**
Dot1q Tunneling Status	Indicates if Q-in-Q tunneling is enabled.	**extensive**
Customer VLAN ranges	List of customer VLAN (C-VLAN) ranges associated with this service VLAN (S-VLAN).	**extensive**
Private VLAN Mode	The private VLAN mode for this VLAN. Values include **Primary**, **Isolated**, and **Community**.	**extensive**
Primary VLAN	Primary VLAN tag for this secondary VLAN.	**extensive**
Internal Index	VLAN index internal to Junos OS software.	**extensive**
Origin	Manner in which the VLAN was created: **static** or **learn**.	**extensive**

Table 181: show vlans Output Fields *(continued)*

Field Name	Field Description	Level of Output
Protocol	Port-based VLAN or MAC-based VLAN. MAC-based protocol is displayed when VLAN assignment is done either statically or dynamically through 802.1X,	**extensive**
IP addresses	IP address associated with a VLAN.	**extensive**
Number of MAC entries	For MAC-based VLANs created either statically or dynamically, the MAC addresses associated with an interface.	**extensive**
Number of mapping rules	Number of mapping rules for Q-in-Q tunneling (**Push**) and VLAN translation (**Swap**).	
Secondary VLANs	Secondary VLANs associated with a primary VLAN.	**extensive**
Isolated VLANs	Isolated VLANs associated with a primary VLAN.	**extensive**
Community VLANs	Community VLANs associated with a primary VLAN.	**extensive**
VLANs summary	VLAN counts: • **Total**—Total number of VLANs on the switch. • **Configured VLANs**—Number of VLANs that are based on user-configured settings. • **Internal VLANs**—Number of VLANs created by the system with no explicit configuration or protocol—for example, the **default** VLAN and the VLAN created when a trunk interface is not configured with native VLAN membership. • **Temporary VLANs**—Number of VLANs from the previous configuration that the system retains for a limited time after restart. Temporary VLANs are converted into one of the other types of VLAN, or are removed from the system if the current configuration does not require them.	All levels

Table 181: show vlans Output Fields *(continued)*

Field Name	Field Description	Level of Output
Dot1q VLANs summary	802.1Q VLAN counts: • **Total**—Total number of 802.1Q-tagged and untagged VLANs on the switch. • **Tagged VLANs**—Number of 802.1Q-tagged VLANs. • **Untagged VLANs**—Number of untagged 802.1Q VLANs. • **Private VLAN**—Counts of the following kinds of 802.1Q private VLANs (PVLANs): ◦ **Primary VLANs**—Number of primary forwarding private VLANs. ◦ **Community VLANs**—Number of community transporting and forwarding private VLANs. ◦ **Isolated VLANs**—Number of isolated receiving and forwarding private VLANs. ◦ **Inter-switch-isolated VLANs**—Number of inter-switch isolated receiving and forwarding private VLANs.	All levels
Dot1q Tunneled VLANs summary	Q-in-Q-tunneled VLAN counts: • **Total**—Total number of Q-in-Q-tunneled VLANs on the switch. • **Private VLAN**—Counts of primary, community, and isolated Q-in-Q-tunneled private VLANs (PVLANs).	All levels

Sample Output (EX Series and QFX Series Switches)

show vlans (EX Series and QFX Series)

```
user@switch> show vlans
```

```
Name            Tag     Interfaces
default         None
                        xe-0/0/34.0, xe-0/0/33.0, xe-0/0/32.0, xe-0/0/31.0,
                        xe-0/0/30.0, xe-0/0/29.0, xe-0/0/28.0, xe-0/0/27.0,
                        xe-0/0/26.0, xe-0/0/25.0, xe-0/0/19.0, xe-0/0/18.0,
                        xe-0/0/17.0, xe-0/0/16.0, xe-0/0/15.0, xe-0/0/14.0,
                        xe-0/0/13.0, xe-0/0/11.0, xe-0/0/9.0, xe-0/0/8.0,
                        xe-0/0/3.0, xe-0/0/2.0, xe-0/0/1.0
v0001           1
```

```
                        xe-0/0/24.0, xe-0/0/23.0, xe-0/0/22.0, xe-0/0/21.0
v0002           2
                        None
v0003           3
                        None
v0004           4
                        None
v0005           5
                        None
```

show vlans (Private VLANs on EX and QFX Series)

`user@switch>` **show vlans**

```
Name            Tag     Interfaces
__pvlan_pvlan_xe-0/0/46.0__
                        xe-0/0/44.0*, xe-0/0/46.0*
c1
                        xe-0/0/4.0*, xe-0/0/44.0*
c2
                        xe-0/0/28.0*, xe-0/0/44.0*
default
                        None
pvlan           500
                        xe-0/0/4.0*, xe-0/0/28.0*, xe-0/0/44.0*, xe-0/0/46.0*
```

show vlans brief (EX and QFX Series)

`user@switch>` **show vlans brief**

```
                                        Ports
Name            Tag     Address         Active/Total
default         None                    0/23
v0001           1                       0/4
v0002           2                       0/0
v0003           3                       0/0
v0004           4                       0/0
v0005           5                       0/0
v0006           6                       0/0
v0007           7                       0/0
v0008           8                       0/0
v0009           9                       0/0
```

```
v0010          10                          0/2
v0011          11                          0/0
v0012          12                          0/0
v0013          13                          0/0
v0014          14                          0/0
v0015          15                          0/0
v0016          16                          0/0
```

show vlans detail (EX Series and QFX Series)

user@switch> **show vlans detail**

```
VLAN: default, Tag: Untagged, Admin state: Enabled
  Description: None
  Primary IP: None, Number of interfaces: 23 (Active = 0)
  STP: None, RTG: None
  Untagged interfaces: xe-0/0/34.0, xe-0/0/33.0, xe-0/0/32.0, xe-0/0/31.0,
  xe-0/0/30.0, xe-0/0/29.0, xe-0/0/28.0, xe-0/0/27.0, xe-0/0/26.0,
  xe-0/0/25.0, xe-0/0/19.0, xe-0/0/18.0, xe-0/0/17.0, xe-0/0/16.0,
  xe-0/0/15.0, xe-0/0/14.0, xe-0/0/13.0, xe-0/0/11.0, xe-0/0/9.0, xe-0/0/8.0,
  xe-0/0/3.0, xe-0/0/2.0, xe-0/0/1.0,
  Tagged interfaces: None

VLAN: v0001, Tag: 802.1Q Tag 1, Admin state: Enabled
  Description: None
  Primary IP: None, Number of interfaces: 4 (Active = 0)
  Dot1q Tunneling Status: Enabled
  STP: None, RTG: None
  Untagged interfaces: None
  Tagged interfaces: xe-0/0/24.0, xe-0/0/23.0, xe-0/0/22.0, xe-0/0/21.0,

VLAN: v0002, Tag: 802.1Q Tag 2, Admin state: Enabled
  Description: None
  Primary IP: None, Number of interfaces: 0 (Active = 0)
  STP: None, RTG: None
  Untagged interfaces: None
  Tagged interfaces: None

VLAN: v0003, Tag: 802.1Q Tag 3, Admin state: Enabled
  Description: None
  Primary IP: None, Number of interfaces: 0 (Active = 0)
  STP: None, RTG: None
  Untagged interfaces: None
  Tagged interfaces: None
```

```
VLAN: vlan4000, 802.1Q Tag: Untagged, Admin State: Enabled
MAC learning Status: Disabled
Number of interfaces: 0 (Active = 0)
```

show vlans extensive (for a PVLAN spanning multiple switches)

user@switch> **show vlans extensive**

```
VLAN: COM1, Created at: Tue May 11 18:16:05 2010
802.1Q Tag: 100, Internal index: 3, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 3 (Active = 3), Untagged  1 (Active = 1)
      ge-0/0/20.0*, tagged, trunk
      ge-0/0/22.0*, tagged, trunk, pvlan-trunk
      ge-0/0/23.0*, tagged, trunk, pvlan-trunk
      ge-0/0/7.0*, untagged, access

VLAN: __pvlan_primary_ge-0/0/0.0__, Created at: Tue May 11 18:16:05 2010
Internal index: 5, Admin State: Enabled, Origin: Static
Private VLAN Mode: Isolated, Primary VLAN: primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 3 (Active = 3), Untagged  1 (Active = 1)
      ge-0/0/20.0*, tagged, trunk
      ge-0/0/22.0*, tagged, trunk, pvlan-trunk
      ge-0/0/23.0*, tagged, trunk, pvlan-trunk
      ge-0/0/0.0*, untagged, access


VLAN: __pvlan_primary_ge-0/0/2.0__, Created at: Tue May 11 18:16:05 2010
Internal index: 6, Admin State: Enabled, Origin: Static
Private VLAN Mode: Isolated, Primary VLAN: primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 3 (Active = 3), Untagged  1 (Active = 0)
      ge-0/0/20.0*, tagged, trunk
      ge-0/0/22.0*, tagged, trunk, pvlan-trunk
      ge-0/0/23.0*, tagged, trunk, pvlan-trunk
      ge-0/0/2.0, untagged, access


VLAN: __pvlan_primary_isiv__, Created at: Tue May 11 18:16:05 2010
802.1Q Tag: 50, Internal index: 7, Admin State: Enabled, Origin: Static
Private VLAN Mode: Inter-switch-isolated, Primary VLAN: primary
```

```
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 3 (Active = 3), Untagged  0 (Active = 0)
      ge-0/0/20.0*, tagged, trunk
      ge-0/0/22.0*, tagged, trunk, pvlan-trunk
      ge-0/0/23.0*, tagged, trunk, pvlan-trunk


VLAN: community2, Created at: Tue May 11 18:16:05 2010
802.1Q Tag: 20, Internal index: 8, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 3 (Active = 3), Untagged  2 (Active = 2)
      ge-0/0/20.0*, tagged, trunk
      ge-0/0/22.0*, tagged, trunk, pvlan-trunk
      ge-0/0/23.0*, tagged, trunk, pvlan-trunk
      ge-0/0/1.0*, untagged, access
      ge-1/0/6.0*, untagged, access


VLAN: primary, Created at: Tue May 11 18:16:05 2010
802.1Q Tag: 10, Internal index: 2, Admin State: Enabled, Origin: Static
Private VLAN Mode: Primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 3 (Active = 3), Untagged  5 (Active = 4)
      ge-0/0/20.0*, tagged, trunk
      ge-0/0/22.0*, tagged, trunk, pvlan-trunk
      ge-0/0/23.0*, tagged, trunk, pvlan-trunk
      ge-0/0/0.0*, untagged, access
      ge-0/0/1.0*, untagged, access
      ge-0/0/2.0, untagged, access
      ge-0/0/7.0*, untagged, access
      ge-1/0/6.0*, untagged, access

Secondary VLANs: Isolated 2, Community  2, Inter-switch-isolated  1
  Isolated VLANs :
      __pvlan_primary_ge-0/0/0.0__
      __pvlan_primary_ge-0/0/2.0__
  Community VLANs :
      COM1
      community2
  Inter-switch-isolated VLAN :
      __pvlan_primary_isiv__
```

show vlans extensive (Port-Based on EX Series and QFX Series)

user@switch> **show vlans extensive**

```
VLAN: default, created at Mon Feb  4 12:13:47 2008
  Tag: None, Internal index: 0, Admin state: Enabled, Origin: static
  Description: None
  Customer VLAN ranges:
                    1-4100
    Protocol: Port based
  IP addresses: None
  STP: None, RTG: None.
  Number of interfaces: Tagged 0 (Active = 0), Untagged 23 (Active = 0)
    xe-0/0/34.0 (untagged, access)
    xe-0/0/33.0 (untagged, access)
    xe-0/0/32.0 (untagged, access)
    xe-0/0/31.0 (untagged, access)
    xe-0/0/30.0 (untagged, access)
    xe-0/0/29.0 (untagged, access)
    xe-0/0/28.0 (untagged, access)
    xe-0/0/27.0 (untagged, access)
    xe-0/0/26.0 (untagged, access)
    xe-0/0/25.0 (untagged, access)
    xe-0/0/19.0 (untagged, access)
    xe-0/0/18.0 (untagged, access)
    xe-0/0/17.0 (untagged, access)
    xe-0/0/16.0 (untagged, access)
    xe-0/0/15.0 (untagged, access)
    xe-0/0/14.0 (untagged, access)
    xe-0/0/13.0 (untagged, access)
    xe-0/0/11.0 (untagged, access)
    xe-0/0/9.0 (untagged, access)
    xe-0/0/8.0 (untagged, access)
    xe-0/0/3.0 (untagged, access)
    xe-0/0/2.0 (untagged, access)
    xe-0/0/1.0 (untagged, access)

Secondary VLANs: Isolated 1, Community  1
  Isolated VLANs :
      __pvlan_pvlan_xe-0/0/3.0__
  Community VLANs :
      comm1

VLAN: v0001, created at Mon Feb  4 12:13:47 2008
  Tag: 1, Internal index: 1, Admin state: Enabled, Origin: static
```

```
  Description: None
  Protocol: Port based, Layer 3 interface: None
  IP addresses: None
  STP: None, RTG: None.
  Number of interfaces: Tagged 4 (Active = 0), Untagged 0 (Active = 0)
    xe-0/0/24.0 (tagged, trunk)
    xe-0/0/23.0 (tagged, trunk)
    xe-0/0/22.0 (tagged, trunk)
    xe-0/0/21.0 (tagged, trunk)

VLAN: v0002, created at Mon Feb  4 12:13:47 2008
  Tag: 2, Internal index: 2, Admin state: Enabled, Origin: static
  Description: None
  Protocol: Port based, Layer 3 interface: None
  IP addresses: None
  STP: None, RTG: None.
  Number of interfaces: Tagged 0 (Active = 0), Untagged 0 (Active = 0)
    None

VLAN: v0003, created at Mon Feb  4 12:13:47 2008
  Tag: 3, Internal index: 3, Admin state: Enabled, Origin: static
  Description: None
  Protocol: Port based, Layer 3 interface: None
  IP addresses: None
  STP: None, RTG: None.
  Number of interfaces: Tagged 0 (Active = 0), Untagged 0 (Active = 0)
    None
```

show vlans extensive (MAC-based)

`user@switch>` **show vlans extensive**

```
VLAN: default, Created at: Thu May 15 13:43:09 2008
Internal index: 3, Admin State: Enabled, Origin: Static
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 0 (Active = 0), Untagged  2 (Active = 2)
      ge-0/0/0.0*, untagged, access
      ge-0/0/14.0*, untagged, access

VLAN: vlan_dyn, Created at: Thu May 15 13:43:09 2008
Internal index: 4, Admin State: Enabled, Origin: Static
Protocol: Port Mode
Number of interfaces: Tagged 0 (Active = 0), Untagged  0 (Active = 0)
Protocol: MAC Based
```

```
Number of MAC entries: 6
      ge-0/0/0.0*
                  00:00:00:00:00:02 (untagged)
                  00:00:00:00:00:03 (untagged)
                  00:00:00:00:00:04 (untagged)
                  00:00:00:00:00:05 (untagged)
                  00:00:00:00:00:06 (untagged)
                  00:00:00:00:00:07 (untagged)
```

show vlans (Q-in-Q Tunneling on EX Series and QFX Series)

`user@switch>` **show vlans dot1q-tunneling**

```
Name           Tag     Interfaces
sv100          100     xe-0/0/4.0*, xe-0/0/15.0*
```

show vlans extensive (Q-in-Q Tunneling on EX Series and QFX Series)

`user@switch>` **show vlans sv100 extensive**

```
VLAN: sv100, Created at: Sat Sep 10 12:53:52 2011
802.1Q Tag: 100, Internal index: 2, Admin State: Enabled, Origin: Static
Dot1q Tunneling Status: Enabled
Customer VLAN ranges:
                    10-20
                    40-50
Protocol: Port Mode
Number of interfaces: Tagged 1 (Active = 1), Untagged  0 (Active = 0)
      ge-0/0/0.0, tagged, trunk

Number of mapping rules:
      Push 1 (Active = 0), Policy 0 (Active = 0), Swap 0 (Active = 0)

      xe-0/0/3.0*, 300, push
```

show vlans extensive (Q-in-Q Tunneling and L2TP on EX Series and QFX Series)

`user@switch>` **show vlans v1 extensive**

```
VLAN: v1, Created at: Fri Mar 2 05:07:38 2012
802.1Q Tag: 100, Internal index: 4, Admin State: Enabled, Origin: Static
```

```
Dot1q Tunneling status: Enabled
Layer2 Protocol Tunneling status: Enabled
```

show vlans sort-by tag (EX Series and QFX Series)

`user@switch>` **show vlans sort-by tag**

```
Name            Tag     Interfaces
default                        None
__vlan-x_1__    1
                        None
__vlan-x_2__    2
                        None
__vlan-x_3__    3
                        None
__vlan-x_4__    4
                        None
__vlan-x_5__    5
                        None
__vlan-x_6__    6
                        None
__vlan-x_7__    7
                        None
__vlan-x_8__    8
                        None
__vlan-x_9__    9
                        None
__vlan-x_10__   10
                        None
__vlan-x_11__   11
                        None
__vlan-x_12__   12
                        None
__vlan-x_13__   13
                        None
__vlan-x_14__   14
                        None
__vlan-x_15__   15
                        None
__vlan-x_16__   16
                        None
__vlan-x_17__   17
                        None
__vlan-x_18__   18
```

```
                        None
__vlan-x_19__   19
                        None
__vlan-x_20__   20
                        None
```

show vlans sort-by name (EX Series and QFX Series)

`user@switch>` **show vlans sort-by employee**

```
Name              Tag   Interfaces

__employee_120__ 120
                        xe-0/0/22.0*
__employee_121__ 121
                        xe-0/0/22.0*
__employee_122__ 122
                        xe-0/0/22.0*
__employee_123__ 123
                        xe-0/0/22.0*
__employee_124__ 124
                        xe-0/0/22.0*
__employee_125__ 125
                        xe-0/0/22.0*
__employee_126__ 126
                        xe-0/0/22.0*
__employee_127__ 127
                        xe-0/0/22.0*
__employee_128__ 128
                        xe-0/0/22.0*
__employee_129__ 129
                        xe-0/0/22.0*
__employee_130__ 130
                        xe-0/0/22.0*
```

show vlans tag (EX Series and QFX Series)

`user@switch>` **show vlans employee**

```
Name              Tag   Interfaces
```

```
__employee_120__ 120
                        xe-0/0/22.0*
__employee_121__ 121
                        xe-0/0/22.0*
__employee_122__ 122
                        xe-0/0/22.0*
__employee_123__ 123
                        xe-0/0/22.0*
__employee_124__ 124
                        xe-0/0/22.0*
__employee_125__ 125
                        xe-0/0/22.0*
__employee_126__ 126
                        xe-0/0/22.0*
__employee_127__ 127
                        xe-0/0/22.0*
__employee_128__ 128
                        xe-0/0/22.0*
__employee_129__ 129
                        xe-0/0/22.0*
__employee_130__ 130
                        xe-0/0/22.0*
```

show vlans sort-by tag (EX Series)

`user@switch>` **show vlans sort-by tag**

```
Name            Tag     Interfaces
default                 None
__vlan-x_1__    1
                        None
__vlan-x_2__    2
                        None
__vlan-x_3__    3
                        None
__vlan-x_4__    4
                        None
__vlan-x_5__    5
                        None
__vlan-x_6__    6
                        None
__vlan-x_7__    7
                        None
__vlan-x_8__    8
```

```
                        None
__vlan-x_9__    9
                        None
__vlan-x_10__   10
                        None
__vlan-x_11__   11
                        None
__vlan-x_12__   12
                        None
__vlan-x_13__   13
                        None
__vlan-x_14__   14
                        None
__vlan-x_15__   15
                        None
__vlan-x_16__   16
                        None
__vlan-x_17__   17
                        None
__vlan-x_18__   18
                        None
__vlan-x_19__   19
                        None
__vlan-x_20__   20
                        None
```

show vlans employee (vlan-range-name)

`user@switch>` **show vlans employee**

```
Name             Tag   Interfaces

__employee_120__ 120
                       ge-0/0/22.0*
__employee_121__ 121
                       ge-0/0/22.0*
__employee_122__ 122
                       ge-0/0/22.0*
__employee_123__ 123
                       ge-0/0/22.0*
__employee_124__ 124
                       ge-0/0/22.0*
__employee_125__ 125
```

```
                      ge-0/0/22.0*
__employee_126__ 126
                      ge-0/0/22.0*
__employee_127__ 127
                      ge-0/0/22.0*
__employee_128__ 128
                      ge-0/0/22.0*
__employee_129__ 129
                      ge-0/0/22.0*
__employee_130__ 130
                      ge-0/0/22.0*
```

show vlans summary (EX Series)

`user@switch>` **show vlans summary**

```
VLANs summary:
         Total: 8,   Configured VLANs: 5
         Internal VLANs: 1,  Temporary VLANs: 0

     Dot1q VLANs summary:
         Total: 8,  Tagged VLANs: 2, Untagged VLANs: 6
         Private VLAN:
             Primary VLANs: 2,  Community VLANs: 2, Isolated VLANs: 3

     Dot1q Tunneled VLANs summary:
         Total: 0
         Private VLAN:
             Primary VLANs: 0,  Community VLANs: 0, Isolated VLANs: 0

     Dynamic VLANs:
        Total: 2,  Dot1x: 2, MVRP: 0
```

Sample Output: EX Series with ELS Switches and MX Routers

show vlans brief (EX Series Switch)

`user@switch>` **show vlans brief**

```
Routing instance        VLAN name             Tag          Interfaces
default-switch          c1                    20
```

```
                                                    ge-0/0/0.0*
                                                    ge-1/0/0.0*
                                                    ge-2/0/0.0*
default-switch          c2                  30
                                                    ge-0/0/0.0*
                                                    ge-2/0/0.0*
default-switch          default             1

default-switch          iso                 10
                                                    ge-0/0/1.0*
default-switch          iso1                50
                                                    ge-0/0/0.0*
                                                    ge-2/0/0.0*
default-switch          pri                 100
                                                    ge-0/0/0.0*
                                                    ge-1/0/0.0*
                                                    ge-2/0/0.0*
```

show vlans brief (MX Routers)

`user@host>` **show vlans brief**

```
Routing instance        VLAN name           Tag         Interfaces
VPLS-1                  __VPLS-1__             all
                                                       ae1.0
VPLS-2                  __VPLS-2__             all
                                                       ae3.0
                                                       ge-3/1/2.0
                                                       vt-3/3/10.1048576
default-switch          VLAN1000               1000
                                                       ae26.0
default-switch          VLAN101                101
                                                       ae20.0
default-switch          VLAN102                102
                                                       ae20.0
default-switch          VLAN103                103
                                                       ae20.0
default-switch          VLAN104                104
                                                       ae20.0
default-switch          VLAN105                105
                                                       ae20.0
default-switch          VLAN106                106
                                                       ae20.0
```

```
default-switch          VLAN107                 107
                                                            ae20.0
default-switch          VLAN108                 108
                                                            ae20.0
[...output truncated...]
```

show vlans detail (EX Series Switch)

user@switch> **show vlans detail**

```
Routing instance: default-switch
  VLAN Name: c1                             State: Active
Tag: 20
PVLAN type : Community
Internal index: 16, Generation Index: 21, Origin: Static
MAC aging time: 300 seconds
Interfaces:
    ge-0/0/0.0*,tagged,trunk
    ge-1/0/0.0*,tagged,trunk
    ge-2/0/0.0*,tagged,trunk
Number of interfaces: Tagged 3    , Untagged 0
Total MAC count: 0

Routing instance: default-switch
  VLAN Name: c2                             State: Active
Tag: 30
PVLAN type : Community
Internal index: 17, Generation Index: 22, Origin: Static
MAC aging time: 300 seconds
Interfaces:
    ge-0/0/0.0*,tagged,trunk
    ge-2/0/0.0*,tagged,trunk
Number of interfaces: Tagged 2    , Untagged 0
Total MAC count: 0

Routing instance: default-switch
  VLAN Name: default                        State: Active
Tag: 1
Internal index: 5, Generation Index: 5, Origin: Static
MAC aging time: 300 seconds
Number of interfaces: Tagged 0    , Untagged 0
Total MAC count: 0

Routing instance: default-switch
```

```
  VLAN Name: iso                            State: Active
Tag: 10
Internal index: 14, Generation Index: 19, Origin: Static
MAC aging time: 300 seconds
Interfaces:
    ge-0/0/1.0*,untagged,access
Number of interfaces: Tagged 0    , Untagged 1
Total MAC count: 0

Routing instance: default-switch
  VLAN Name: iso1                           State: Active
Tag: 50
PVLAN type : Isolated
Internal index: 15, Generation Index: 20, Origin: Static
MAC aging time: 300 seconds
Interfaces:
    ge-0/0/0.0*,tagged,trunk
    ge-2/0/0.0*,tagged,trunk
Number of interfaces: Tagged 2    , Untagged 0
Total MAC count: 0

Routing instance: default-switch
  VLAN Name: pri                            State: Active
Tag: 100
PVLAN type : Primary
Isolated VLAN :
vlan-id : 50 vlan name : iso1
Community VLAN :
vlan-id : 20 vlan name : c1
vlan-id : 30 vlan name : c2
Internal index: 9, Generation Index: 14, Origin: Static
MAC aging time: 300 seconds
Interfaces:
    ge-0/0/0.0*,tagged,trunk
    ge-1/0/0.0*,tagged,trunk
    ge-2/0/0.0*,tagged,trunk
Number of interfaces: Tagged 3    , Untagged 0
Total MAC count: 0
```

show vlans detail (MX Routers)

`user@host>` **show vlans detail**

```
Routing instance: VPLS-1
  VLAN Name: __VPLS-1__                     State: Active
Tag: all
Internal index: 2, Generation Index:  , Origin: Dynamic
Interfaces:
    ae1.0,tagged
Number of interfaces: Tagged 1    , Untagged 0
Total MAC count: 0

Routing instance: VPLS-2
  VLAN Name: __VPLS-2__                     State: Active
Tag: all
Internal index: 3, Generation Index:  , Origin: Dynamic
Interfaces:
    ae3.0,tagged
    ge-3/1/2.0,tagged
    vt-3/3/10.1048576,tagged
Number of interfaces: Tagged 3    , Untagged 0
Total MAC count: 4

Routing instance: default-switch
  VLAN Name: VLAN1000                       State: Active
Tag: 1000
Internal index: 4, Generation Index: 1, Origin: Static
Layer 3 interface: irb.1000
Interfaces:
    ae26.0,tagged,trunk
Number of interfaces: Tagged 1    , Untagged 0
Total MAC count: 0

Routing instance: default-switch
  VLAN Name: VLAN101                        State: Active
Tag: 101
Internal index: 5, Generation Index: 2, Origin: Static
Layer 3 interface: irb.101
Interfaces:
    ae20.0,tagged,trunk
Number of interfaces: Tagged 1    , Untagged 0
Total MAC count: 1

Routing instance: default-switch
  VLAN Name: VLAN102                        State: Active
Tag: 102
Internal index: 6, Generation Index: 3, Origin: Static
```

```
Layer 3 interface: irb.102
Interfaces:
    ae20.0,tagged,trunk
Number of interfaces: Tagged 1    , Untagged 0
Total MAC count: 1
[...output truncated...]
```

show vlans extensive (EX Series Switch)

`user@switch>` **show vlans extensive**

```
Routing instance: default-switch
  VLAN Name: c1                             State: Active
Tag: 20
PVLAN type : Community
Internal index: 16, Generation Index: 21, Origin: Static
MAC aging time: 300 seconds
Interfaces:
    ge-0/0/0.0*,tagged,trunk
    ge-1/0/0.0*,tagged,trunk
    ge-2/0/0.0*,tagged,trunk
Number of interfaces: Tagged 3    , Untagged 0
Total MAC count: 0

Routing instance: default-switch
  VLAN Name: c2                             State: Active
Tag: 30
PVLAN type : Community
Internal index: 17, Generation Index: 22, Origin: Static
MAC aging time: 300 seconds
Interfaces:
    ge-0/0/0.0*,tagged,trunk
    ge-2/0/0.0*,tagged,trunk
Number of interfaces: Tagged 2    , Untagged 0
Total MAC count: 0

Routing instance: default-switch
  VLAN Name: default                        State: Active
Tag: 1
Internal index: 5, Generation Index: 5, Origin: Static
MAC aging time: 300 seconds
Number of interfaces: Tagged 0    , Untagged 0
Total MAC count: 0
```

```
Routing instance: default-switch
  VLAN Name: iso                            State: Active
Tag: 10
Internal index: 14, Generation Index: 19, Origin: Static
MAC aging time: 300 seconds
Interfaces:
    ge-0/0/1.0*,untagged,access
Number of interfaces: Tagged 0    , Untagged 1
Total MAC count: 0

Routing instance: default-switch
  VLAN Name: iso1                           State: Active
Tag: 50
PVLAN type : Isolated
Internal index: 15, Generation Index: 20, Origin: Static
MAC aging time: 300 seconds
Interfaces:
    ge-0/0/0.0*,tagged,trunk
    ge-2/0/0.0*,tagged,trunk
Number of interfaces: Tagged 2    , Untagged 0
Total MAC count: 0

Routing instance: default-switch
  VLAN Name: pri                            State: Active
Tag: 100
PVLAN type : Primary
Isolated VLAN :
vlan-id : 50 vlan name : iso1
Community VLAN :
vlan-id : 20 vlan name : c1
vlan-id : 30 vlan name : c2
Internal index: 9, Generation Index: 14, Origin: Static
MAC aging time: 300 seconds
Interfaces:
    ge-0/0/0.0*,tagged,trunk
    ge-1/0/0.0*,tagged,trunk
    ge-2/0/0.0*,tagged,trunk
Number of interfaces: Tagged 3    , Untagged 0
Total MAC count: 0
```

show vlans extensive (MX Routers)

`user@host>` **show vlans extensive**

```
Routing instance: default-switch
  VLAN Name: VLAN_10                        State: Active
Tag: 10
Internal index: 2, Generation Index: 1, Origin: Static
MAC aging time: 300 seconds
Interfaces:
    ge-1/0/3.0*,tagged,trunk
Number of interfaces: Tagged 1    , Untagged 0
Total MAC count: 0

Routing instance: default-switch
  VLAN Name: VLAN_20                        State: Active
Tag: 20
Internal index: 3, Generation Index: 2, Origin: Static
MAC aging time: 300 seconds
Interfaces:
    ge-1/0/3.0*,tagged,trunk
Number of interfaces: Tagged 1    , Untagged 0
Total MAC count: 0
```

Sample Output (SRX Devices)

show vlans (SRX devices)

`user@host>` **show vlans**

```
Routing instance        VLAN name             Tag          Interfaces
default-switch          vlan-22                 22

default-switch          vlan-333                333
                                                             ge-0/0/3.0*
                                                             ge-0/0/4.0*
default-switch          default               1

default-switch          vlan100               100
                                                             ge-0/0/1.0*
```

show vlans brief (SRX devices)

`user@host>` **show vlans brief**

```
Routing instance        VLAN name             Tag          Interfaces
default-switch          vlan-22                  22
                                                            ge-0/0/3.0*
                                                            ge-0/0/4.0*
default-switch          default               1
default-switch          vlan100               100
                                                            ge-0/0/1.0*
```

show vlans detail (SRX devices)

`user@host>` **show vlans detail**

```
Routing instance: default-switch
  VLAN Name: vlan-22                          State: Active
Tag: 22
Internal index: 2, Generation Index: 1, Origin: Static
MAC aging time: 300 seconds
VXLAN Enabled : No
Number of interfaces: Tagged 0    , Untagged 0
Total MAC count: 0

Routing instance: default-switch
  VLAN Name: vlan-333                         State: Active
Tag: 333
Internal index: 3, Generation Index: 2, Origin: Static
MAC aging time: 300 seconds
VXLAN Enabled : No
Interfaces:
    ge-0/0/3.0*,tagged,trunk
    ge-0/0/4.0*,tagged,trunk
Number of interfaces: Tagged 2    , Untagged 0
Total MAC count: 0
```

traceroute ethernet

Syntax

```
traceroute ethernet
local-mep mep-id
maintenance-association ma-name
maintenance-domain md-name
<ttl value>
<wait seconds>
mac-address | mep-id
<detail>
```

Release Information

Command introduced in Junos OS Release 9.0.
mep-id option introduced in Junos OS Release 9.1.
local-mep option introduced in Junos OS Release 15.1

Description

Triggers the linktrace protocol to trace the route between two maintenance points. The result of the traceroute protocol is stored in the path database. To display the path database, use the **show oam ethernet connectivity-fault-management path-database** command.

Before using the traceroute command, you can verify the remote MEP's MAC address using the **show oam ethernet connectivity-fault-management path-database** command.

Options

local-mep *mep-id*—(Required when multiple MEPs are configured) Identifier for the local maintenance endpoint.

detail—(Optional) Provide detailed information of the responder hostname, ingress port name, egress port name, TTL, and relay action.

mac-address—Destination unicast MAC address of the remote maintenance point.

mep-id—MEP identifier of the remote maintenance point. The range of values is 1 through 8191.

maintenance-association *ma-name*—Specifies an existing maintenance association from the set of configured maintenance associations.

maintenance-domain *md-name*—Specifies an existing maintenance domain from the set of configured maintenance domains.

ttl *value*—Number of hops to use in the linktrace request. The range is 1 to 255 hops. The default is 4.

wait *seconds*—(Optional) Maximum time to wait for a response to the traceroute request. The range is 1 to 255 seconds. The default is 5.

Required Privilege Level
network

List of Sample Output
traceroute ethernet on page 1677
traceroute ethernet detail on page 1678

Output Fields
Table 182 on page 1676 lists the output fields for the **traceroute ethernet** command. Output fields are listed in the approximate order in which they appear.

Table 182: traceroute ethernet Output Fields

Field Name	Field Description
Linktrace to	MAC address of the destination maintenance point.
Interface	Local interface used to send the linktrace message (LTM).
Maintenance Domain	Maintenance domain specified in the traceroute command.
Level	Maintenance domain level configured.
Maintenance Association	Maintenance association specified in the traceroute command.
Local Mep	The local maintenance end point identifier.
Transaction Identifier	4-byte identifier maintained by the MEP. Each LTM uses a transaction identifier. The transaction identifier is maintained globally across all Maintenance Domains. Use the transaction identifier to match an incoming linktrace response (LTR), with a previously sent LTM.
Hop	Sequential hop count of the linktrace path.
TTL	Number of hops remaining in the linktrace message. The time to live (TTL) is decremented at each hop.
Source MAC address	MAC address of the 802.1ag node responding to the LTM or the source MAC address of the LTR.

Table 182: traceroute ethernet Output Fields *(continued)*

Field Name	Field Description
Next-hop MAC address	MAC address of the egress interface of the node to which the LTM is forwarded or the next-hop MAC address derived from the next egress identifier in the Egress-ID TLV of the LTR PDU.
Responder Hostname	The hostname of the responding router. A valid hostname is received only when the responding system is a Juniper Networks router.
Ingress port name	The port name for ingress connections.
Egress port name	The port name for egress connections.
Flags	The configurable flags can include: • H— Hardware only, incoming LT frame has hardware bit set. • T— Terminal MEP, responder is a terminating MEP. • F— FWD yes, LTM frame is relayed further.
Relay Action	The associated relay action. Relay action can be one of the following: • RlyHit— Relay hit; target MAC address matches the MP mac address. • RlyFDB— Relay FDB; output port decided by consulting forwarding database. • RlyMPDB— Relay MIP; output port decided by consulting MIP database.

Sample Output

traceroute ethernet

`user@host>` **traceroute ethernet maintenance-domain md1 maintenance-association ma1 00:01:02:03:04:05**

```
Linktrace to 00:01:02:03:04:05, Interface : ge-5/0/0.0
      Maintenance Domain: MD1, Level: 7
      Maintenance Association: MA1, Local Mep: 1
```

```
Hop     TTL      Source MAC address        Next hop MAC address
Transaction Identifier:100001
1       63       00:00:aa:aa:aa:aa         00:00:ab:ab:ab:ab
2       62       00:00:bb:bb:bb:bb         00:00:bc:bc:bc:bc
3       61       00:00:cc:cc:cc:cc         00:00:cd:cd:cd:cd
4       60       00:01:02:03:04:05         00:00:00:00:00:00
```

traceroute ethernet detail

user@host> **run traceroute ethernet maintenance-domain md6 maintenance-association ma6 mep 101 detail**

```
Linktrace to 00:00:5E:00:53:CC, Interface : ge-1/0/0.1
Maintenance Domain: md6, Level: 6
Maintenance Association: ma6, Local Mep: 201
Transaction Identifier: 2077547465

Legend for RelayAction:
RlyHit  -- Relay hit, Target MAC address matches the MP mac address
RlyFDB  -- Relay FDB, output port decided by consulting FDB database
RlyMPDB -- Relay MIP, output port decided by consulting MIP database

Legend for Flags:
H -- Hardware only,incoming LT frame has hardware bit set
T -- Terminal MEP, responder is a terminating MEP
F -- FWD yes, LTM frame is relayed further

TTL   Responder Hostname    Ingress port name    Egress port name      RelayAction
Responder Service     Ingress MAC address   Egress MAC address     Flags
62    host1                  ge-1/0/0.1            ge-2/3/0.1             RlyFDB
br1                   00:00:5E:00:53:00     00:00:5E:00:53:A0      HF-
63    host2                 ge-2/3/0.1            ge-1/0/0.1             RlyFDB
br1                   00:00:5E:00:53:AA     00:00:5E:00:53:A2      HF-
61    host3                 ge-1/0/0.1            --:--                  RlyHit
br1                   00:00:5E:00:53:B0           --:--            H-T
```