Ethernet Switching Feature Guide for EX2300, EX3400, and EX4300 Switches

Modified: 2018-01-26

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

Ethernet Switching Feature Guide for EX2300, EX3400, and EX4300 Switches

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 http://www.juniper.net/support/eula/. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

List of Figures

List of Tables

About the Documentation

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 http://www.juniper.net/techpubs/.

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 http://www.juniper.net/books.

Supported Platforms

For the features described in this document, the following platforms are supported:

- EX Series

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 xv 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 xvi 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*
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, comments, and suggestions so that we can improve the documentation. You can provide feedback by using either of the following methods:

- Online feedback rating system—On any page of the Juniper Networks TechLibrary site at http://www.juniper.net/techpubs/index.html, simply click the stars to rate the content, and use the pop-up form to provide us with information about your experience. Alternately, you can use the online feedback form at http://www.juniper.net/techpubs/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 J-Care or Partner Support Service 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 http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf.
- Product warranties—For product warranty information, visit http://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: http://www.juniper.net/customers/support/
- Search for known bugs: https://prsearch.juniper.net/
- Find product documentation: http://www.juniper.net/documentation/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/
- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: http://kb.juniper.net/InfoCenter/
- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/

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

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at http://www.juniper.net/cm/.
- 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 http://www.juniper.net/support/requesting-support.html.

CHAPTER 1

Configuring Bridging and VLANs

Understanding Bridging and VLANs on EX Series Switches

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 on Juniper Networks EX Series Ethernet Switches:

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 VLAN IDs 1 through 4094 can be assigned to VLANs, while VLAN IDs 0 and 4095 are reserved by Junos OS and cannot be assigned.

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.

In addition to the TPID EtherType value of 0x8100, EX Series switches that run Junos OS that does 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.

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 an EX Series 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 an EX Series switch that runs Junos OS 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 EX Series 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 23.

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 an EX Series 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 an EX Series 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.

In the rare case where you want untagged packets to be recognized by a trunk port on EX Series 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 23.

Trunk Mode and Native VLAN

On an EX Series 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 an EX Series 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 EX Series 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 EX Series switches that run Junos OS that does not use the ELS configuration style support tagged-access mode.

Tagged-access mode accommodates cloud computing scenarios, specifically deployments including servers that adhere to the edge virtual bridging (EVB) standard (IEEE 803.1Qbg).

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.

Additional Advantages 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 an EX Series 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.

Maximum VLANs and VLAN Members Per Switch

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 an EX Series 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 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)

If the configuration of the switch exceeds the recommended VLAN member maximum, a warning message appears in the system log (syslog).

A Default VLAN Is Configured on Most Switches

NOTE: **EX Series switches that run Junos OS with the ELS configuration style do not support a default VLAN.**

Some EX Series switches that run Junos OS that does 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.

The following EX Series switches that run Junos OS that does not support the ELS 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, or EX3300 Ethernet 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.

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.

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 an EX Series switch that supports ELS, see "Adding a Static MAC Address Entry to the Ethernet Switching Table on a Switch with ELS Support (CLI Procedure)" on page 57 To configure a static MAC-based VLAN on an EX Series switch that does not support ELS, see *Adding a Static MAC Address Entry to the Ethernet Switching Table (CLI Procedure)*.

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 and Multiple VLAN Registration Protocol (MVRP).

To pass traffic between two VLANs, the switch uses standard Layer 3 routing protocols, such as static routing, OSPF, and RIP. On EX Series switches, 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 22.

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.

EX Series 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 EX Series 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.

Related Documentation

- Understanding Multiple VLAN Registration Protocol (MVRP) on EX Series Switches on page 79
- Understanding Integrated Routing and Bridging Interfaces and Routed VLAN Interfaces on EX Series Switches on page 64

Configuring VLANs for EX Series Switches (J-Web Procedure)

NOTE: This topic applies only to the J-Web Application package.

You can use the VLAN Configuration page to add a new VLAN or to edit or delete an existing VLAN on an EX Series switch.

To access the VLAN Configuration page:

1. Select **Configure** > **Switching** > **VLAN**.

   The VLAN Configuration page displays a list of existing VLANs. If you select a specific VLAN, the specific VLAN details are displayed in the Details section.

   NOTE: After you make changes to the configuration on this page, you must commit the changes immediately for them to take effect. To commit all changes to the active configuration, select Commit Options > Commit. See Using the Commit Options to Commit Configuration Changes for details about all commit options.

2. Click one of the following options:

- **Add**—Creates a VLAN.
- **Edit**—Edits an existing VLAN configuration.
- **Delete**—Deletes an existing VLAN.

NOTE: If you delete a VLAN, the VLAN configuration for all the associated interfaces is also deleted.

When you are adding or editing a VLAN, enter information as described in Table 3 on page 28.

Table 3: VLAN Configuration Details

Field	Function	Your Action
General tab		
VLAN Name	Specifies a unique name for the VLAN.	Enter a name.
VLAN ID/Range/VLAN ID/List NOTE: EX4300 switches support only VLAN ID/List and not VLAN Range.	Specifies the identifier or range for the VLAN.	Select one of the following options: • VLAN ID—Type a unique identification number from **1** through **4094**. If no value is specified, the ID defaults to 0. • VLAN Range/List—Type a number range to create VLANs with IDs corresponding to the numbers in the range. For example, the range 2–3 creates two VLANs with the IDs 2 and 3.
Description	Describes the VLAN.	Enter a brief description for the VLAN.
MAC-Table-Aging-Time NOTE: This option is not supported on EX4300 switches.	Specifies the maximum time that an entry can remain in the forwarding table before it *ages out*.	Type the number of seconds from **60** through **1000000**.
Input filter	Specifies the VLAN firewall filter that is applied to incoming packets.	To apply an input firewall filter, select the firewall filter from the list.
Output filter	Specifies the VLAN firewall filter that is applied to outgoing packets.	To apply an output firewall filter, select the firewall filter from the list.
Ports tab		
Ports NOTE: This option is not supported on EX4300 switches.	Specifies the ports (interfaces) to be associated with this VLAN for data traffic. You can also remove the port association.	Click one of the following options: • **Add**—Select the ports from the available list. For an EX8200 Virtual Chassis configuration, select the member, FPC, and the interface from the list. • **Remove**—Select the port that you do not want associated with the VLAN.

Table 3: VLAN Configuration Details *(continued)*

Field	Function	Your Action
IP address tab		
IPv4 address	Specifies IPv4 address options for the VLAN.	Select **IPv4 address** to enable the IPv4 address options. To configure IPv4: 1. Enter the IP address. 2. Enter the subnet mask—for example, **255.255.255.0**. You can also specify the address prefix. 3. To apply an input firewall filter to an interface, select the firewall filter from the list. 4. To apply an output firewall filter to an interface, select the firewall filter from the list. 5. Click the **ARP/MAC Details** button. Enter the static IP address and MAC address in the window that is displayed. NOTE: In EX4300 switches, you also need to select **L2 Interface** in the window that is displayed.
IPv6 address	Specifies IPv6 address options for the VLAN.	Select **IPv6 address** to enable the IPv6 address options. To configure IPv6: 1. Enter the IP address—for example: **2001:ab8:85a3::8a2e:370:7334**. 2. Specify the subnet mask.
Voip tab		
Ports	Specifies the ports to be associated with this VLAN for voice traffic. You can also remove the port association.	Click one of the following options: • **Add**—Select the ports from the list of available ports. For an EX8200 Virtual Chassis configuration, select the member, FPC, and the interface from the list. • **Remove**—Select the port that you do not want associated with the VLAN.

Related Documentation

- Configuring VLANs for EX Series Switches (CLI Procedure) on page 30
- *Configuring VLANs for EX Series Switches (CLI Procedure)*
- Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch on page 34
- *Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch*

- Understanding Bridging and VLANs on EX Series Switches on page 19
- Configuring Integrated Routing and Bridging Interfaces (CLI Procedure) on page 68
- *Configuring Routed VLAN Interfaces (CLI Procedure)*

Configuring VLANs for 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 *Configuring VLANs for EX Series Switches (CLI Procedure)*. For ELS details, see *Getting Started with Enhanced Layer 2 Software*.

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? on page 30
- Creating a VLAN Using the Minimum Procedure on page 30
- Creating a VLAN Using All of the Options on page 31
- Configuration Guidelines for VLANs on page 32

Why Create a VLAN?

For switching to begin, you must explicitly configure at least one VLAN, even if your network is simple and you want only one broadcast domain to exist.

Some reasons to create more than one VLAN 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.

Creating a VLAN Using the Minimum Procedure

These steps are required to create a VLAN:

- Uniquely identify the VLAN. You do this by assigning a name and an ID to the VLAN.
- Assign at least one switch port interface to the VLAN for communication. After assigning one or more interfaces to the VLAN, the interfaces function in access mode. 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 a few required steps. The VLAN is created with the name **employee-vlan** and the VLAN ID of **100**. Then, three interfaces are assigned to that VLAN, and these interfaces, which function in access mode, transmit traffic among themselves.

```
[edit] set vlans employee-vlan
[edit] set vlans employee-vlan vlan-id 100
[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 an integrated routing and bridging (IRB) interface. See "Configuring Integrated Routing and Bridging Interfaces (CLI Procedure)" on page 68.

Creating a VLAN Using All of the Options

To configure a VLAN, follow these steps:

1. Create the VLAN by setting the unique VLAN name:

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

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

   ```
   [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-]
   ```

3. Assign at least one interface to the VLAN:

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

NOTE: You can also specify that a 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.

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/destination-prefix
   ```

5. (Optional) Specify the description of the VLAN:

   ```
   [edit vlans]
   user@switch# set vlan-name description text-description
   ```

6. (Optional) For security purposes, 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
   ```

Configuration Guidelines for VLANs

To create a VLAN, you must uniquely identify the VLAN and assign at least one switch port interface to the VLAN for communication. After you assign one or more interfaces to the VLAN, the interfaces function in access mode.

After creating a VLAN, all users connected to interfaces that are 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 an IRB interface. For information about creating an IRB interface, see "Configuring Integrated Routing and Bridging Interfaces (CLI Procedure)" on page 68..

The number of VLANs supported per switch varies. Use the 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 each VLAN is assigned an ID number when it is created.

Related Documentation

- Understanding Bridging and VLANs on EX Series Switches on page 19
- Understanding Integrated Routing and Bridging Interfaces and Routed VLAN Interfaces on EX Series Switches on page 64
- Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch on page 34

Configuring the Native VLAN Identifier on Switches With ELS Support (CLI Procedure)

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 (CLI Procedure)*. For ELS details, see *Getting Started with Enhanced Layer 2 Software*.

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
   ```

Related Documentation

- Understanding Bridging and VLANs on EX Series Switches on page 19
- Example: Connecting Access Switches to a Distribution Switch on page 44
- Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch on page 34
- *Example: Setting Up Basic Bridging and a VLAN on the QFX Series*

Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch with ELS Support

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*. For ELS details, see *Getting Started with Enhanced Layer 2 Software*.

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 4 on page 35 details the topology used in this configuration example.

Table 4: 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

To verify that switching is operational and that **employee-vlan** has been created, perform these tasks:

- Verifying That the VLAN Has Been Created on page 40
- Verifying That Interfaces Are Associated with the Proper VLANs on page 41

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.

Related Documentation

- Configuring VLANs for EX Series Switches (CLI Procedure) on page 30
- Example: Connecting Access Switches to a Distribution Switch on page 44
- Understanding Bridging and VLANs on EX Series Switches on page 19

Example: Connecting Access Switches with ELS Support to a Distribution Switch with ELS Support

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 *Getting Started with Enhanced Layer 2 Software*.

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 on page 45
- Overview and Topology on page 45

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 46 shows an EX9200 distribution switch that is connected to three EX4300 access switches.

Figure 1: Sample Access Switch-Distribution Switch Topology

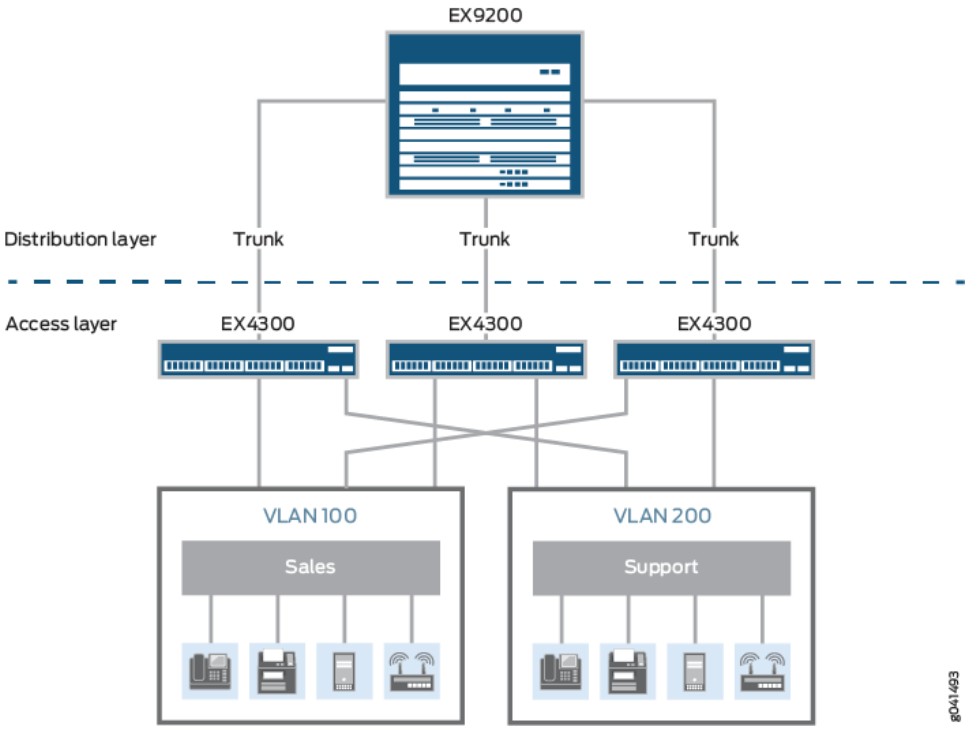


Table 5 on page 46 describes the components of the example topology. The example shows how to configure one of the three access switches. The other access switches could be configured in the same manner.

Table 5: 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)
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

Table 5: Components of the Topology for Connecting an Access Switch to a Distribution Switch *(continued)*

Property	Settings
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

To confirm that the configuration is working properly, perform these tasks:

- Verifying the VLAN Members and Interfaces on the Access Switch on page 55
- Verifying the VLAN Members and Interfaces on the Distribution Switch on page 55

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.

Related Documentation

- Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch on page 34
- Configuring VLANs for EX Series Switches (CLI Procedure) on page 30
- Understanding Bridging and VLANs on EX Series Switches on page 19

CHAPTER 2

Managing MAC Addresses

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

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 (CLI Procedure)*. For ELS details, see *Getting Started with Enhanced Layer 2 Software*.

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 (CLI Procedure)" on page 30.

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
```

Related Documentation

- Understanding Bridging and VLANs on EX Series Switches on page 19

Configuring MAC Limiting (CLI Procedure)

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 MAC Limiting (CLI Procedure)*. For ELS details, see *Getting Started with Enhanced Layer 2 Software*.

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

For information on configuring an interface to automatically recover from a shutdown caused by MAC limiting, see *Configuring Autorecovery from the Disabled State on Secure or Storm Control Interfaces (CLI Procedure)*. 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 on page 59
- Limiting the Number of MAC Addresses Learned by a VLAN on page 59

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.

Related Documentation

- *Configuring Autorecovery from the Disabled State on Secure or Storm Control Interfaces (CLI Procedure)*
- *Configuring Persistent MAC Learning (CLI Procedure)*

Understanding MAC Address 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.

Related Documentation

- *Configuring MAC Table Aging (CLI Procedure)*
- Configuring MAC Table Aging (CLI Procedure) on page 62
- *Controlling Authentication Session Timeouts (CLI Procedure)*

Configuring MAC Table Aging on 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. For ELS details, see *Getting Started with Enhanced Layer 2 Software*.

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.

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

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

Related Documentation

- Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch on page 34
- Example: Connecting Access Switches to a Distribution Switch on page 44
- Understanding Bridging and VLANs on EX Series Switches on page 19

CHAPTER 3

Configuring Integrated Routing and Bridging Interfaces

Understanding Integrated Routing and Bridging Interfaces and Routed VLAN Interfaces on EX Series Switches

Virtual LANs (VLANs), by definition, 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. 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 traditionally needed a router that connected the VLANs. However, you can also accomplish this forwarding with a switch by configuring one of the following features:

- On Juniper Networks EX Series Ethernet Switches that run Juniper Networks Junos operating system (Junos OS) that supports the Enhanced Layer 2 Software (ELS) configuration style, configure an integrated routing and bridging (IRB) interface.
- On EX Series switches that run Junos OS that does not support ELS, configure a routed VLAN interface (RVI).

NOTE: IRB interfaces and RVIs 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.

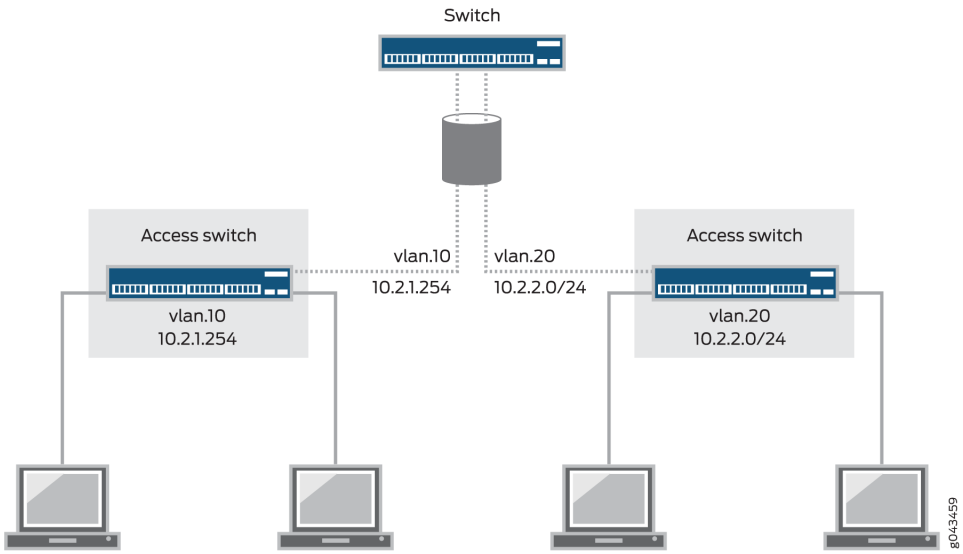
Configuring a switch to route traffic between VLANs reduces complexity and eliminates the costs associated with purchasing, installing, managing, powering, and cooling a router.

These interfaces route only VLAN traffic and work by logically dividing a switch into multiple virtual routing instances, thereby isolating VLAN traffic traveling across the network into virtual segments. These interfaces allow switches to recognize which packets are being sent to another VLAN's MAC addresses—then, packets are bridged (switched) whenever the destination is within the same VLAN and are routed through these interfaces only when necessary. Whenever packets can be switched instead of routed, several layers of processing are eliminated. The switches rely on their Layer 3 capabilities to provide this basic routing between VLANs:

- Two VLANs on the same switch
- Two VLANs on different switches (routing is provided by an intermediary third switch.)

Figure 2 on page 65 illustrates a switch routing VLAN traffic between two access layer switches using one of these interfaces.

Figure 2: An IRB Interface or RVI on a Switch Providing Routing Between Two Access Switches



This topic describes:

- When Should I Use an IRB Interface or RVI? on page 65
- How Does an IRB Interface or RVI Work? on page 65
- Creating an IRB Interface or RVI on page 66
- Viewing IRB Interface and RVI Statistics on page 67
- IRB Interfaces and RVI Functions and Other Technologies on page 67

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

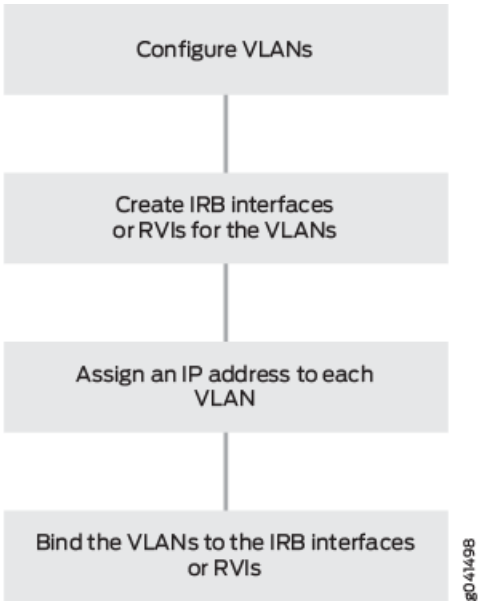
VLANs (for example, VLAN **red** and VLAN **blue**) with corresponding subnets, your interfaces must have a logical unit with an address in the subnet for **red** and a logical unit with an address in the subnet for **blue**. The switch automatically creates direct routes to these subnets and uses these 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

There are four basic steps in creating an IRB interface or RVI as shown in Figure 3 on page 66.

Figure 3: Creating an IRB Interface or RVI



The following explanations correspond to the four steps for creating a VLAN, as depicted in Figure 3 on page 66.

- 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 (CLI Procedure)" on page 68, and for an RVI, see *Configuring Routed VLAN Interfaces (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 6 on page 67 illustrates the IRB interface- and RVI-tracking capability on various switches.

Table 6: 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 71.
- 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 on EX Series Switches*.

Related Documentation

- Understanding Bridging and VLANs on EX Series Switches on page 19

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.

Related Documentation

- Verifying Integrated Routing and Bridging Interface Status and Statistics on page 69
- Understanding Integrated Routing and Bridging Interfaces and Routed VLAN Interfaces on EX Series Switches on page 64

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
                                                           ge-0/0/18.0
default-switch          support                111

...
```

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**.

Related Documentation

- Configuring Integrated Routing and Bridging Interfaces (CLI Procedure) on page 68

CHAPTER 4

Configuring Virtual Routing Interfaces

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.

Related Documentation

- *Understanding Layer 3 Subinterfaces*
- Example: Using Virtual Routing Instances to Route Among VLANs on EX Series Switches on page 73
- Configuring Virtual Routing Instances (CLI Procedure) on page 72

Configuring Virtual Routing Instances on EX Series Switches (CLI Procedure)

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 (CLI Procedure)*, "Configuring VLANs for EX Series Switches (CLI Procedure)" on page 30, or "Configuring VLANs for EX Series Switches (J-Web Procedure)" on page 27.

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**

Related Documentation

- Example: Using Virtual Routing Instances to Route Among VLANs on EX Series Switches on page 73

Example: Using Virtual Routing Instances to Route Among VLANs on EX Series Switches

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 (CLI Procedure)*, "Configuring VLANs for EX Series Switches (CLI Procedure)" on page 30, or "Configuring VLANs for EX Series Switches (J-Web Procedure)" on page 27.

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

To confirm that the configuration is working properly, perform these tasks:

- Verifying That the Routing Instances Were Created on page 76

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.

Related Documentation

- Configuring Virtual Routing Instances (CLI Procedure) on page 72

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.

Related Documentation

- Configuring Virtual Routing Instances (CLI Procedure) on page 72
- Example: Using Virtual Routing Instances to Route Among VLANs on EX Series Switches on page 73

CHAPTER 5

Configuring the Multiple VLAN Registration Protocol

Understanding Multiple VLAN Registration Protocol (MVRP) on EX Series Switches

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 EX Series Ethernet 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.

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.

This topic describes:

How MVRP Updates, Creates, and Deletes VLANs on the 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 the 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 7 on page 82 outlines the MVRP versions and whether or not each version includes the extra byte in the PDU. Table 7 on page 82 also labels each MVRP version with a scenario number, which is used throughout the remainder of this discussion for brevity.

Table 7: 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 8 on page 82 provides a summary of environments that include the various MVRP scenarios and whether or not a particular environment requires you to take action.

Table 8: 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	—
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) (CLI Procedure)*.
Includes MVRP versions in scenarios 1 and 3. MVRP version in scenario 3 is in its default state.	No	—

Table 8: MVRP Environments and Description of Required Actions *(continued)*

Environment	Action Required?	Action Description
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) (CLI Procedure)*. • 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) (CLI Procedure)" on page 83.

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) (CLI Procedure)* or "Configuring Multiple VLAN Registration Protocol (MVRP) (CLI Procedure)" on page 83.

Related Documentation

- Understanding Bridging and VLANs on EX Series Switches on page 19
- *Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches*

Configuring Multiple VLAN Registration Protocol (MVRP) on 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 Junos OS that does not support ELS, see *Configuring Multiple VLAN Registration Protocol (MVRP) (CLI Procedure)*. For ELS details, see *Getting Started with Enhanced Layer 2 Software*.

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.

To enable MVRP or set MVRP options, follow these instructions:

- Enabling MVRP on page 84
- Disabling MVRP on page 84
- Disabling Dynamic VLANs on page 84
- Configuring Timer Values on page 84
- Configuring MVRP Registration Mode on page 86
- Using MVRP in a Mixed-Release Network on page 86

Enabling MVRP

MVRP can only be enabled on trunk interfaces.

To enable MVRP on a trunk interface:

```
[edit protocols mvrp]
user@switch# set interface interface-name
```

See Also

- interface on page 201

Disabling MVRP

MVRP is disabled by default. You only need to perform this procedure if you have previously enabled MVRP.

You can disable MVRP globally only. . To disable MVRP on all trunk interfaces on a switch, use one of the following commands:

```
user@switch# deactivate protocols mvrp
user@switch# delete protocols mvrp
```

See Also

- no-attribute-length-in-pdu on page 225

Disabling Dynamic VLANs

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 on page 226
- no-attribute-length-in-pdu on page 225

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
```

To set the join timer for a specific interface:

```
[edit protocols mvrp]
user@switch# set interface interface-name join-timer milliseconds
```

To set the leave timer for all interfaces on the switch:

```
[edit protocols mvrp]
user@switch# set leave-timer milliseconds
```

To set the leave timer for a specific interface:

```
[edit protocols mvrp]
user@switch# set interface interface-name leave-timer milliseconds
```

To set the leaveall timer for all interfaces on the switch:

```
[edit protocols mvrp]
user@switch# set leaveall-timer seconds
```

To set the leaveall timer for a specific interface:

```
[edit protocols mvrp]
user@switch# set interface interface-name leaveall-timer seconds
```

See Also

- join-timer on page 208
- leave-timer on page 211
- leaveall-timer on page 210

Configuring MVRP Registration Mode

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
```

To set an interface to normal registration mode:

```
[edit protocols mvrp]
user@switch# set interface xe-0/0/1.0 registration normal
```

See Also

- registration on page 241

Using MVRP in a Mixed-Release 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 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.

For more information about this issue, see "Understanding Multiple VLAN Registration Protocol (MVRP) on EX Series Switches" on page 79.

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.

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.

Related Documentation

- Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches on page 87
- Verifying That MVRP Is Working Correctly on page 100
- Understanding Multiple VLAN Registration Protocol (MVRP) on EX Series Switches on page 79

Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches with ELS Support

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*. For ELS details, see *Getting Started with Enhanced Layer 2 Software*.

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 on page 88
- Overview and Topology on page 88
- Configuring VLANs and MVRP on Access Switch A on page 90
- Configuring VLANs and MVRP on Access Switch B on page 92
- Configuring VLANS and MVRP on Distribution Switch C on page 95
- Verification on page 96

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 for EX Series and QFX Series Switches*.
- Multiple Spanning-Tree Protocol (MSTP). For more information about MSTP, see *Understanding MSTP for EX Series and QFX 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. 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) (CLI Procedure)* for details.

Figure 4 on page 89 shows MVRP configured on two access switches and one distribution switch.

Figure 4: MVRP Configured on Two Access Switches and One Distribution Switch for Automatic VLAN Administration

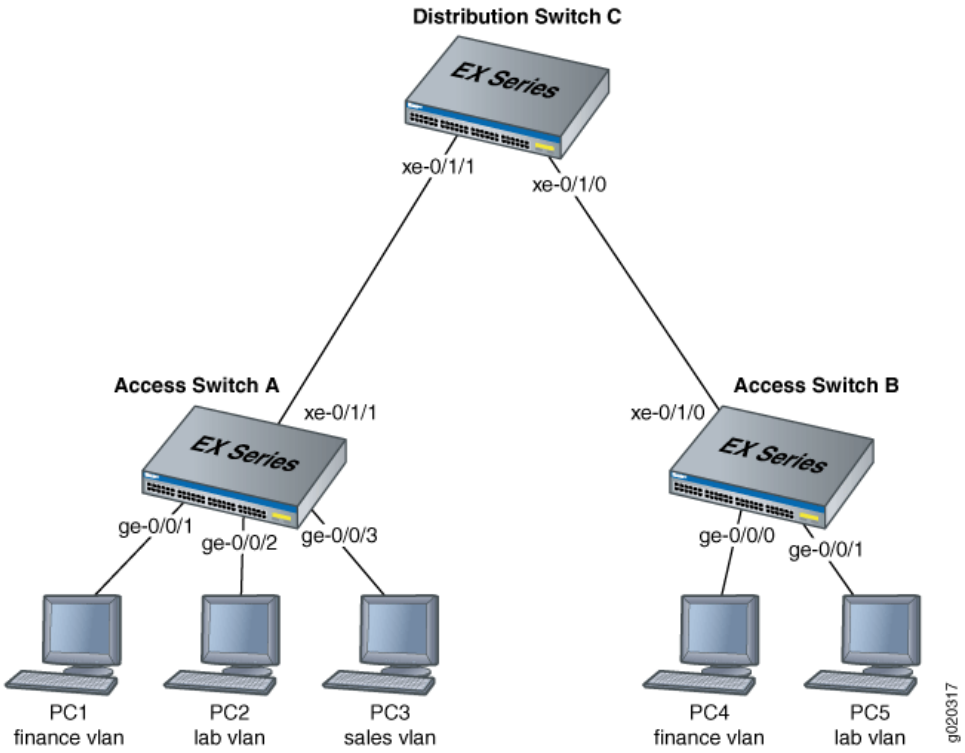


Table 9 on page 89 explains the components of the example topology.

Table 9: 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**

Table 9: Components of the Network Topology *(continued)*

Settings	Settings
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

To confirm that the configuration is updating VLAN membership, perform these tasks:

- Verifying That MVRP Is Enabled on Access Switch A on page 96
- Verifying That MVRP Is Updating VLAN Membership on Access Switch A on page 97
- Verifying That MVRP Is Enabled on Access Switch B on page 97
- Verifying That MVRP Is Updating VLAN Membership on Access Switch B on page 98
- Verifying That MVRP Is Enabled on Distribution Switch C on page 99
- Verifying That MVRP Is Updating VLAN Membership on Distribution Switch C on page 99

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**.

Related Documentation

- Configuring Multiple VLAN Registration Protocol (MVRP) (CLI Procedure) on page 83
- Understanding Multiple VLAN Registration Protocol (MVRP) on EX Series Switches on page 79

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*. For ELS details, see *Getting Started with Enhanced Layer 2 Software*.

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) (CLI Procedure)" on page 83.

Related Documentation

- Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches on page 87
- Understanding Multiple VLAN Registration Protocol (MVRP) on EX Series Switches on page 79

CHAPTER 6

Configuring Q-in-Q Tunneling

Understanding Q-in-Q Tunneling on EX Series Switches with ELS Support

NOTE: This topic applies to Junos OS for EX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style. For ELS details, see *Getting Started with Enhanced Layer 2 Software*.

Q-in-Q tunneling enables service providers on Ethernet access networks to extend a Layer 2 Ethernet connection between two customer sites. Using Q-in-Q tunneling, providers can also 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 VLAN tags are prepended by the service-provider 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 an S-VLAN, a service-provider-specific 802.1Q tag is added to the packet. This additional tag is used to segregate traffic into S-VLANs-. The original customer 802.1Q tag of the packet is

retained and is transmitted transparently, passing through the service provider's network. As the packet leaves the S-VLAN in the downstream direction, the additional 802.1Q tag is removed.

When Q-in-Q tunneling is configured 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 part of the customer network. Therefore, this topic also refers to trunk interfaces as S-VLAN interfaces (network-to-network interfaces [NNI]), and to access interfaces as C-VLAN interfaces (user-network interfaces [UNI]). An access interface can receive both tagged and untagged frames in this case.

An interface can be a member of multiple S-VLANs. You can map one C-VLAN to one S-VLAN (1:1) or many C-VLANs to many S-VLANs (*N:N*). Customer packets that traverse an S-VLAN are double-tagged for an additional layer of segregating or bundling of C-VLANs. C-VLAN and S-VLAN tags are unique—for instance, you can have both a C-VLAN tag of 101 and an S-VLAN tag of 101. 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.

C-VLAN and S-VLAN interfaces accept priority-tagged packets without any configuration.

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.

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 at most 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.

Sending and Receiving Untagged Packets

To enable a C-VLAN or S-VLAN interface to send and receive untagged packets, you must configure a native VLAN for the interface, then specify a VLAN ID for the native VLAN. After performing this configuration, when a C-VLAN or S-VLAN interface receives an untagged packet, it adds the VLAN ID of the native VLAN to the packet and sends the newly tagged packet to the mapped interface.

To specify a native VLAN ID, use the **native-vlan-id** statement at the **[edit interfaces *interface-name*]** hierarchy level. When specifying a native VLAN ID on a C-VLAN or S-VLAN physical interface, the value must match the VLAN ID or be included in the VLAN ID list specified on the C-VLAN or S-VLAN logical interface.

For example, on a logical interface for a C-VLAN interface, you specify a C-VLAN ID list of 100-200. Then, on the C-VLAN physical interface, you specify a native VLAN ID of 150. This configuration will 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. If you do not configure a native VLAN on an interface, untagged packets received by the interface are discarded. 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 the global, interface, and VLAN levels:

At the global level, you disable MAC address learning for the switch.

At the interface level, you disable MAC address learning for all VLANs of which the specified interface is a member.

At the VLAN level, you disable MAC address learning for a specified VLAN. MAC addresses that have already been learned for the VLAN are flushed.

Mapping C-VLANs to S-VLANs

There are three ways to map C-VLANs to S-VLANs:

- All-in-One Bundling on page 106
- Many-to-Many Bundling on page 106
- Mapping a Specific Interface on page 106

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

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-Many Bundling

Many-to-many bundling is used to specify which C-VLANs are mapped to which S-VLANs.

Many-to-many bundling is used 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 as in the cases of the all-in-one bundling and many-to-many bundling approaches.

Specific interface mapping has two suboptions for treatment of traffic: push and swap. When traffic that is mapped to a specific interface is pushed, the packet retains its tag as it moves from the C-VLAN to the S-VLAN, then 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, which is also referred to as VLAN rewrite.

It might be useful to have S-VLANs that provide service to multiple customers. Each customer typically has its own S-VLAN plus access to one or more S-VLANs that are used by multiple customers. A specific tag on the customer side is mapped to an S-VLAN. Typically, this functionality is used to keep data from different customers separate or to provide individualized treatment of the packets on a certain interface.

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.

Limitations for Q-in-Q Tunneling

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.

Q-in-Q tunneling supports only two VLAN tags.

Be aware of the following constraints when configuring Q-in-Q tunneling and VLAN translation:

- Most access port security features are not supported with Q-in-Q tunneling and VLAN translation.
- 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
  - IP Source Guard
- The following features are not supported with VLAN translation:
  - Firewall filter applied to a port or VLAN in the output direction
  - VLAN Spanning Tree Protocol
  - Reflective relay

Related Documentation

- Understanding Bridging and VLANs on EX Series Switches on page 19
- Configuring Q-in-Q Tunneling (CLI Procedure) on page 109

Configuring Q-in-Q Tunneling on 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. For ELS details, see *Getting Started with Enhanced Layer 2 Software*.

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 (CLI Procedure)" on page 30 or "Configuring VLANs for EX Series Switches (J-Web Procedure)" on page 27.

Configure Q-in-Q tunneling by using one of the following methods to map C-VLANs to S-VLANs:

- Configuring All-in-One Bundling on page 109
- Configuring Many-to-Many Bundling on page 111
- Configuring a Specific Interface Mapping with VLAN Rewrite Option on page 113

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***

   CAUTION: **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. 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 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**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
```

Related Documentation

- Understanding Q-in-Q Tunneling on EX Series Switches on page 103

CHAPTER 7

Configuring Layer 2 Protocol Tunneling

Understanding Layer 2 Protocol Tunneling on EX Series Switches That Support Enhanced Layer 2 Software (ELS)

NOTE: This topic describes Layer 2 protocol tunneling (L2PT) on 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 *Understanding Layer 2 Protocol Tunneling on EX Series Switches*. For ELS details, see *Getting Started with Enhanced Layer 2 Software*.

L2PT enables service providers to send Layer 2 protocol data units (PDUs) across the provider's cloud and deliver them to Juniper Networks EX Series Ethernet 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.

See Feature Explorer for the list of EX Series switches that support L2PT.

This topic includes:

Layer 2 Protocols Supported by L2PT on EX Series Switches

L2PT supports tunneling the following Layer 2 protocols on EX Series switches (unless otherwise noted). See protocol for details on which protocols can be tunneled on each type of EX Series switch and the available configuration options to enable tunneling the supported protocols.

- Cisco Discovery Protocol (CDP)
- Ethernet Local Management Interface (E-LMI)—EX4300, EX4600, and EX9200 switches only
- Generic Attribute Registration Protocol (GARP) VLAN Registration Protocol (GVRP)
- IEEE 802.1X authentication—EX4300, EX4600, and EX9200 switches only
- 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.

- Link Aggregation Control Protocol (LACP)

  NOTE: If you enable L2PT for untagged LACP packets, do not configure LACP on the corresponding access interface.

- Link Layer Discovery Protocol (LLDP)
- Multiple MAC Registration Protocol (MMRP)—EX4300, EX4600, and EX9200 switches only
- Multiple VLAN Registration Protocol (MVRP)
- Per-VLAN Spanning Tree and Per-VLAN Spanning Tree Plus (PVST+) Protocols—EX9200 switches only
- Spanning Tree Protocol (STP), Rapid Spanning Tree Protocol (RSTP), and Multiple Spanning Tree Protocol (MSTP)
- Unidirectional Link Detection (UDLD)—EX4300, EX4600, and EX9200 switches only
- VLAN Spanning Tree Protocol (VSTP)

  NOTE: EX9200 switches do not have a separate option to enable VSTP. The L2PT protocol configuration statement option that enables tunneling PVST and PVST+ (**pvstp**) also enables tunneling VSTP.

- VLAN Trunking Protocol (VTP)

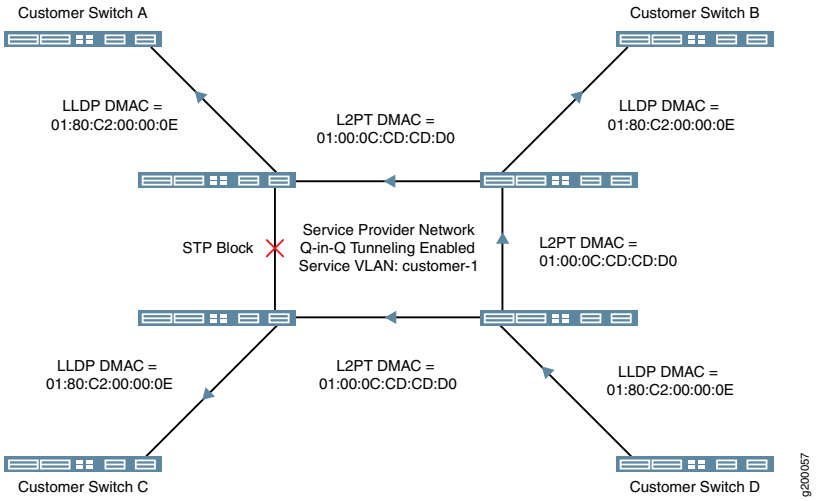
NOTE: CDP, UDLD, and VTP cannot be configured on EX Series switches. L2PT does, however, tunnel CDP, UDLD, and VTP PDUs.

How L2PT Works

L2PT works by encapsulating Layer 2 PDUs, tunneling them across a service provider network, and decapsulating them for delivery to their destination switches. L2PT encapsulates Layer 2 PDUs by enabling the ingress provider edge (PE) device to rewrite the PDUs' destination media access control (MAC) addresses before forwarding them to the service provider network. The devices in the service provider network treat these encapsulated PDUs as multicast Ethernet packets. On receiving these PDUs, the egress PE devices decapsulate them by replacing the destination MAC (DMAC) addresses with the address of the Layer 2 protocol that is being tunneled before forwarding the PDUs to their destination switches.

This process is illustrated in the following example for tunneling LLDP packets in Figure 5 on page 119:

Figure 5: L2PT 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 adds the L2PT DMAC 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 restore the LLDP DMAC and send it to Customer Switches A, B, and C.

The destination switches identify the tunneled Layer 2 control protocol by the encapsulated MAC address. The destination MAC addresses used by different protocols are listed in Table 10 on page 120:

Table 10: Protocol Destination MAC Addresses

Protocol	Ethernet Encapsulation	MAC Address
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
IEEE 802.1X	Ether-II	01:80:C2:00:00:03
IEEE 802.3AH	Ether-II	01:80:C2:00:00:02
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, and 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

When a PE device receives a Layer 2 control PDU from any of the customer PE devices, it changes the destination MAC address to 01:00:0C:CD:CD:D0. The modified packet is then sent to the provider network. 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 same MAC address (01:00:0C:CD:CD:D0). Then they identify the packet type by doing deeper packet inspection and replace the destination MAC address 01:00:0C:CD:CD:D0 with the appropriate destination address. The modified PDUs are sent out to the customer PE devices, thus ensuring the Layer 2 control PDUs are delivered, in their original state, across the provider network. The L2PT protocol is valid for all types of packets (untagged, tagged, and Q-in-Q tagged).

L2PT and Q-in-Q Tunneling on EX Series Switches

You must 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* and related topics, or for other EX Series switches, see "Understanding Q-in-Q Tunneling on EX Series Switches" on page 103.

Related Documentation

- Understanding Q-in-Q Tunneling on EX Series Switches on page 103
- Configuring Layer 2 Protocol Tunneling on EX Series Switches with ELS Support (CLI Procedure) on page 122
- Configuring Q-in-Q Tunneling (CLI Procedure) on page 109
- *Configuring VLAN Encapsulation*

Configuring Layer 2 Protocol Tunneling on EX Series Switches with ELS Support (CLI Procedure)

NOTE: This topic applies to 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 Layer 2 Protocol Tunneling on EX Series Switches (CLI Procedure)*. For ELS details, see *Getting Started with Enhanced Layer 2 Software*.

Layer 2 protocol tunneling (L2PT) enables you 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.

To configure L2PT on an EX Series switch, you must first configure a Q-in-Q interface or group of interfaces. For information about configuring Q-in-Q tunneling on EX9200 switches, see *Configuring VLAN Encapsulation*, *Configuring Inner and Outer TPIDs and VLAN IDs*, and *Stacking a VLAN Tag*. For information about configuring Q-in-Q tunneling on other EX Series switches, see "Configuring Q-in-Q Tunneling (CLI Procedure)" on page 109.

NOTE: When you enable L2PT tunneling for a protocol on one user-to-network interface (UNI) in a bridge domain or VLAN, all UNIs in the bridge domain or VLAN should also be configured 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).

- To configure L2PT on a specified Q-in-Q interface, enable MAC address rewriting for Layer 2 protocol tunneling and select the Layer 2 protocol to be tunneled from the list of available options for the type of switch being configured (see **protocol**):

  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 multiple times to select the desired protocols.

  ```
  [edit protocols]
  user@switch# set layer2-control mac-rewrite interface interface-name protocol protocol-name
  ```

  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 0 encapsulation vlan-bridge
set interfaces xe-1/1/3 unit 0 vlan-id 10
set interfaces xe-1/1/3 native-vlan-id 10
set interfaces xe-1/1/3 unit 0 input-vlan-map push
set interfaces xe-1/1/3 unit 0 input-vlan-map vlan-id 100
set interfaces xe-1/1/3 unit 0 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 EX2300, EX3400, EX4300, or EX4600 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
```

- To check the protocols that L2PT is configured to tunnel on an interface, enter the **show mac-rewrite interface** command in operational mode and specify the interface name, as follows:

```
user@switch> show mac-rewrite interface ge-0/0/0
Interface       Protocols

  ge-0/0/0       LLDP STP
```

- (EX9200 switches only) For information on how to detect and clear an interface configured for 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*.

Related Documentation

- Understanding Layer 2 Protocol Tunneling on EX Series Switches That Support Enhanced Layer 2 Software (ELS) on page 117
- Understanding Q-in-Q Tunneling on EX Series Switches on page 103
- Configuring Q-in-Q Tunneling (CLI Procedure) on page 109
- *Configuring VLAN Encapsulation*
- *Stacking a VLAN Tag*

CHAPTER 8

Configuring Redundant Trunk Groups

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" on page 137.

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 6 on page 127, 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 6 on page 127 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 6: Redundant Trunk Group, Link 1 Active

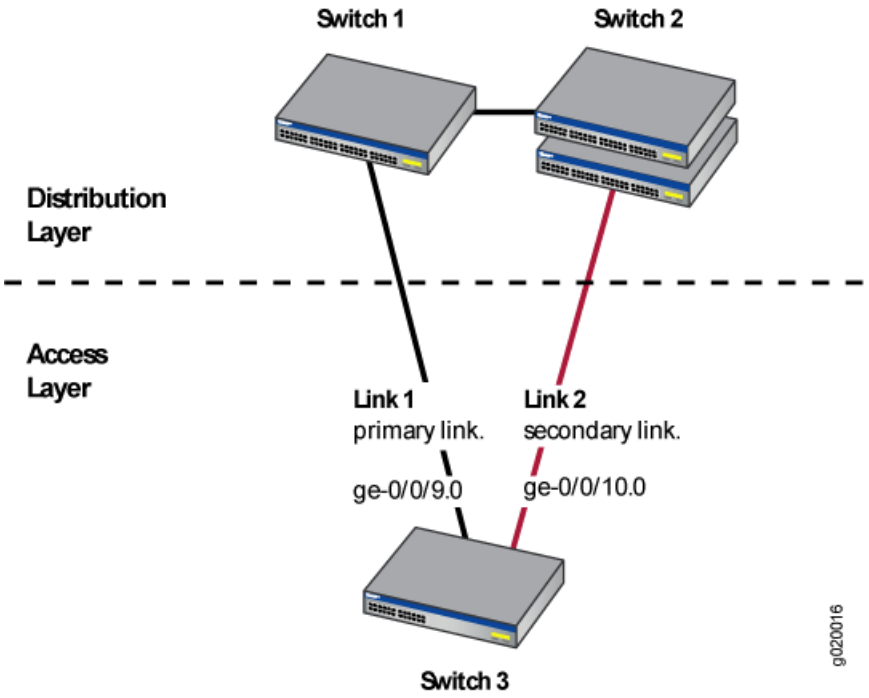
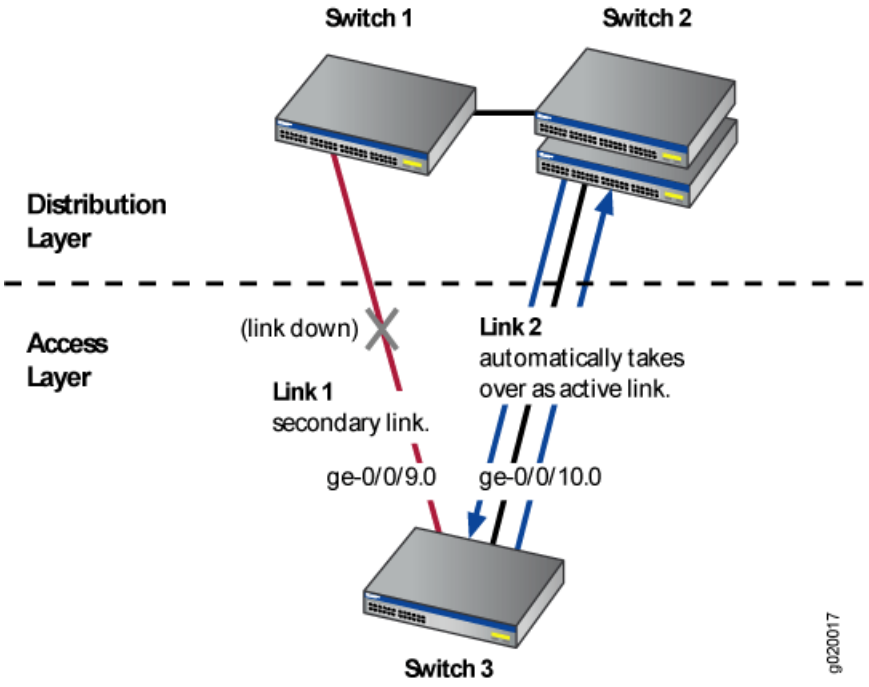


Figure 7 on page 127 illustrates how the redundant trunk link topology works when the primary link goes down.

Figure 7: 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 1 and Switch 2.

Related Documentation

- *Example: Configuring Redundant Trunk Links for Faster Recovery*
- Example: Configuring Redundant Trunk Links for Faster Recovery on page 129

Configuring Redundant Trunk Groups on EX Series Switches (J-Web Procedure)

NOTE: This topic applies only to the J-Web Application package.

A redundant trunk link provides a simple solution for network recovery when a trunk interface goes down. Traffic is routed to another trunk interface, keeping network convergence time to a minimum. You can configure redundant trunk groups (RTGs) with a primary link and a secondary link on trunk interfaces, or configure dynamic selection of the active interface. If the primary link fails, the secondary link automatically takes over without waiting for normal Spanning Tree Protocol (STP) convergence. An RTG can be created only if the following conditions are satisfied:

- A minimum of two trunk interfaces that are not part of any RTG are available.
- All the selected trunk interfaces to be added to the RTG have the same VLAN configuration.
- The selected trunk interfaces are not part of a spanning-tree configuration.

To configure an RTG by using the J-Web interface:

1. Select **Configure** > **Switching** > **RTG**.

   The RTG Configuration page displays a list of existing RTGs. If you select a specific RTG, the details of the selected RTG are displayed in the Details of group section.

   NOTE: After you make changes to the configuration on this page, you must commit the changes for them to take effect. To commit all changes to the active configuration, select **Commit Options** > **Commit**. See Using the Commit Options to Commit Configuration Changes for details about all commit options.

2. Click one of the following options:
   - **Add**—Creates an RTG.
   - **Edit**—Modifies an RTG.
   - **Delete**—Deletes an RTG.

When you are adding or editing an RTG, enter information as described in Table 11 on page 129.

3. Click **OK** to apply changes to the configuration or click **Cancel** to cancel without saving changes.

Table 11: RTG Configuration Fields

Field	Function	Your Action
Group Name	Specifies a unique name for the RTG.	Enter a name. NOTE: Only on EX4300 switches, you can select the name from a list.
Member Interface 1	Specifies a logical interface containing multiple trunk interfaces.	Select a trunk interface from the list.
Member Interface 2	Specifies a trunk interface containing multiple VLANs.	Select a trunk interface from the list.
Select Primary Interface	Enables you to specify one of the interfaces in the RTG as the primary link. The interface without this option is the secondary link in the RTG.	1. Select the option button. 2. Select the primary interface.
Dynamically select my active interface	Specifies that the system dynamically select the active interface.	Select the option button.

Related Documentation

- Example: Configuring Redundant Trunk Links for Faster Recovery on page 129
- *Example: Configuring Redundant Trunk Links for Faster Recovery*
- Understanding Redundant Trunk Links on page 126

Example: Configuring Redundant Trunk Links for Faster Recovery on Devices with ELS Support

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 *Getting Started with Enhanced Layer 2 Software*.

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 8 on page 132).

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 8 on page 132 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 12 on page 132 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 8: Topology for Configuring the Redundant Trunk Links

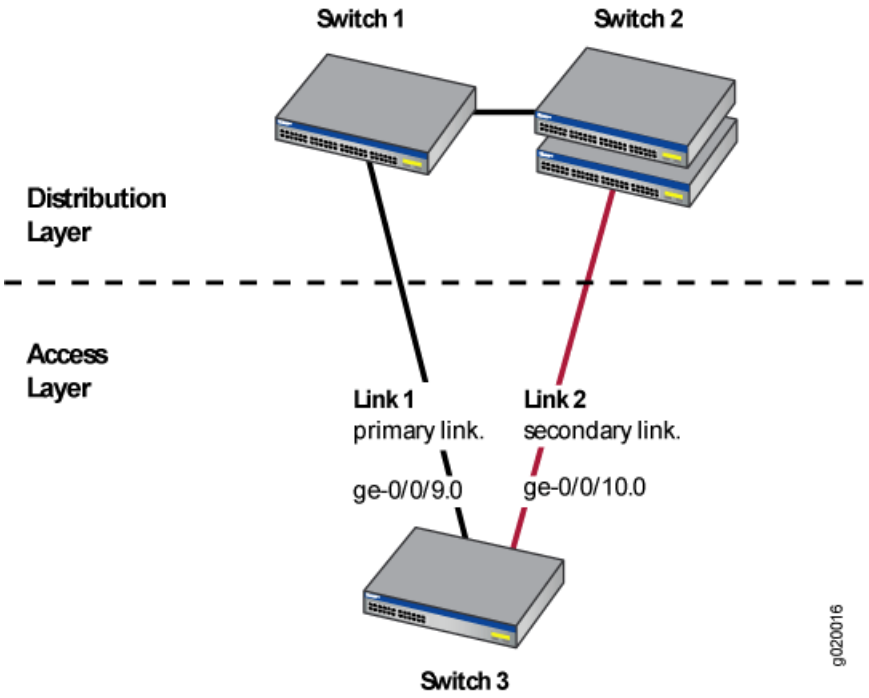


Table 12: Components of the Redundant Trunk Link Topology

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 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

To confirm that the configuration is set up correctly, perform this task:

- Verifying That a Redundant Trunk Group Was Created on page 134

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.

Related Documentation

- Understanding Redundant Trunk Links on page 126

CHAPTER 9

Configuring Q-in-Q Support on Redundant Trunk Links Using LAGs with Link Protection

Q-in-Q Support on Redundant Trunk Links Using LAGs with Link Protection

Understanding Q-in-Q Support on RTGs Using LAGs with Link Protection

Redundant trunk links provide 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 about using redundant trunk links in a legacy redundant trunk groups (RTG) setup—that is, an RTG configuration that does not support Q-in-Q or service-provider configurations—see "Understanding Redundant Trunk Links" on page 126.

You can use this feature of redundant trunk links (or RTG) with Q-in-Q support using LAGs with link protection in both service provider and enterprise configurations.

This feature of RTG with Q-in-Q support includes support for the following items that are *not* supported in legacy RTG configurations:

- Configuration of flexible VLAN tagging on the same LAG that supports the redundant links configurations
- Multiple redundant-link configurations on one physical interface
- Multicast convergence

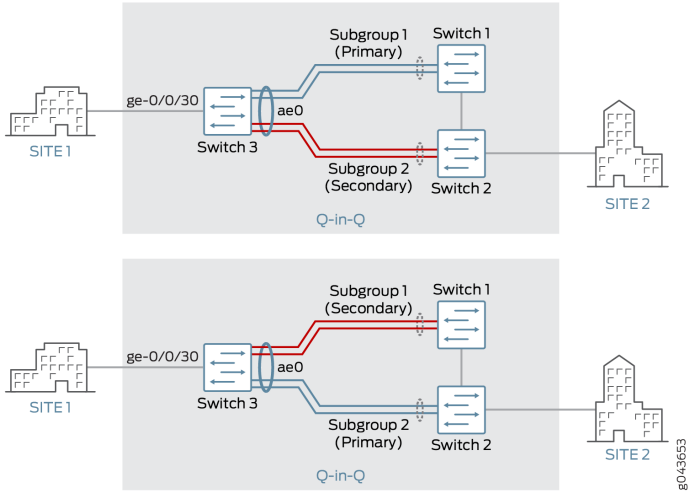
The redundant trunk link configuration (also known as a "redundant trunk group (RTG) configuration") contains two links: a *primary, or active,* link and a *secondary* link. If the primary link fails, the secondary link automatically starts forwarding data traffic without waiting for normal spanning-tree protocol convergence.

Data traffic is forwarded only on the primary link. Data traffic received on the secondary link is dropped.

While data traffic is blocked on the secondary link, Layer 2 control traffic is still permitted. For example, you can run an LLDP session 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 cannot be in both a redundant trunk link and in a spanning-tree protocol topology at the same time. You must disable RSTP on an interface if a redundant trunk link is configured on that interface. Spanning-tree protocols can, however, continue operating on other interfaces on those switches.

Figure 9: Q-in-Q with Redundant Trunk Links Using LAGs with Link Protection



The top of Figure 1 shows three switches in a topology for redundant trunk links on a LAG with flexible VLAN tagging. This particular configuration also includes subgroups that contain multiple links—there can be just two subgroups on the LAG, and both subgroups must have the same number of links. (For more about subgroups, see Configuring Redundant Trunk Links on an LACP LAG (N:N Link Protection with Subgroups), later in this document.) Switch 3 is connected to Switch 1 through Subgroup 1 and to Switch 2 through Subgroup 2. Subgroups 1 and 2 are in an aggregated Ethernet bundle, or link aggregation group (LAG), with interface name ae0. Subgroup 1 is designated as the primary link, and Subgroup 2 is designated as the secondary link. Traffic flows between Switch 3 and Switch 1 through Subgroup 1. While Subgroup 1 is active, Subgroup 2 blocks data traffic.

The bottom of Figure 1 illustrates how the redundant trunk link topology works when the primary link goes down.

When Subgroup 1 between Switch 1 and Switch 3 goes down, Subgroup 2 takes over as the primary (active) link. Traffic flows between Switch 3 and Switch 2 through Subgroup 2.

Configuring Redundant Trunk Links on a LAG with Link Protection and Flexible VLAN Tagging

There are several variations on the configuration of redundant trunk links on a LAG with link protection and with flexible VLAN tagging.

NOTE: **For illustration purposes only, the following configuration tasks show absolute values, such as ge-0/0/30, rather than variables such as *interface-name*.**

1. Configuring Redundant Trunk Links on an LACP LAG (N:N Link Protection with Subgroups) on page 140
2. Configuring Redundant Trunk Links on a Static LAG (1:1 Link Protection) on page 140

3. Configuring Redundant Trunk Links on a LAG with Multiple Logical Interfaces (1:1 Link Protection) on page 141
4. Verifying That Redundant Trunk Links Are Available on the LAG and Viewing Active Links on page 142

Configuring Redundant Trunk Links on an LACP LAG (N:N Link Protection with Subgroups)

1. Configure the ingress interface:

```
set interfaces ge-0/0/30 enable
set interfaces ge-0/0/30 flexible-vlan-tagging
set interfaces ge-0/0/30 encapsulation extended-vlan-bridge
set interfaces ge-0/0/30 unit 30 vlan-id 30
set interfaces ge-0/0/30 unit 30 input-vlan-map push
set interfaces ge-0/0/30 unit 30 output-vlan-map pop
set vlans qinqvlan interface ge-0/0/30.30
```

2. Assign interfaces to the LAG, and assign those interfaces to two link-protection subgroups:

```
set interfaces ge-0/0/14 ether-options 802.3ad ae0
set interfaces ge-0/0/15 ether-options 802.3ad ae0
set interfaces ge-0/0/16 ether-options 802.3ad ae0
set interfaces ge-0/0/17 ether-options 802.3ad ae0
set interfaces ge-0/0/14 ether-options 802.3ad link-protection-sub-group subg1
set interfaces ge-0/0/15 ether-options 802.3ad link-protection-sub-group subg1
set interfaces ge-0/0/16 ether-options 802.3ad link-protection-sub-group subg2
set interfaces ge-0/0/17 ether-options 802.3ad link-protection-sub-group subg2
```

3. Assign LACP values, configure redundant trunk links (using the **rtg-config** statement) on the LAG, set one subgroup as primary and one as backup, and configure Q-in-Q on the LAG:

```
set interfaces ae0 flexible-vlan-tagging
set interfaces ae0 encapsulation extended-vlan-bridge
set interfaces ae0 aggregated-ether-options lacp active
set interfaces ae0 aggregated-ether-options lacp periodic fast
set interfaces ae0 aggregated-ether-options lacp link-protection rtg-config
set interfaces ae0 aggregated-ether-options lacp system-id 00:00:00:01:00:02
set interfaces ae0 aggregated-ether-options link-protection-sub-group subg1 primary
set interfaces ae0 aggregated-ether-options link-protection-sub-group subg2 backup
set interfaces ae0 unit 300 vlan-id 40
set vlans qinqvlan interface ae0.300
```

Configuring Redundant Trunk Links on a Static LAG (1:1 Link Protection)

1. Configure the ingress interface:

```
set interfaces ge-1/0/0 flexible-vlan-tagging
set interfaces ge-1/0/0 encapsulation extended-vlan-bridge
set interfaces ge-1/0/0 unit 4001 vlan-id-list 1-100
set interfaces ge-1/0/0 unit 4001 input-vlan-map push
set interfaces ge-1/0/0 unit 4001 output-vlan-map pop
```

2. Assign interfaces to the LAG, configure redundant trunk links (using the **rtg-config** statement) on the LAG, and configure Q-in-Q on the LAG:

```
set interfaces ae0 enable
set interfaces ae0 flexible-vlan-tagging
set interfaces ae0 encapsulation extended-vlan-bridge
set interfaces ae0 aggregated-ether-options link-protection rtg-config
set interfaces ae0 aggregated-ether-options minimum-links 1
set interfaces ae0 unit 300 vlan-id 40
set vlans qinqvlan interface ae0.300
set vlans qinqvlan interface ge-1/0/0.4001
```

3. Assign the member interfaces on the LAG and assign the two redundant trunk links to the interfaces; set one of the links as primary and one as backup:

```
set interfaces ge-1/0/22 enable
set interfaces ge-1/0/22 ether-options 802.3ad ae0
set interfaces ge-1/0/22 ether-options 802.3ad primary
set interfaces ge-1/0/23 enable
set interfaces ge-1/0/23 ether-options 802.3ad ae0
set interfaces ge-1/0/23 ether-options 802.3ad backup
```

Configuring Redundant Trunk Links on a LAG with Multiple Logical Interfaces (1:1 Link Protection)

1. Configure the ingress interface:

```
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 300 vlan-id 300
set interfaces ge-0/0/0 unit 300 input-vlan-map push
set interfaces ge-0/0/0 unit 300 output-vlan-map pop
set vlans qinqvlan interface ge-0/0/0.300
set interfaces ge-0/0/0 unit 400 vlan-id 400
set interfaces ge-0/0/0 unit 400 input-vlan-map push
set interfaces ge-0/0/0 unit 400 output-vlan-map pop
set vlans qinqvlan interface ge-0/0/0.400
```

2. Assign interfaces to the LAG, configure redundant trunk links (using the **rtg-config** statement) on the LAG, and configure Q-in-Q on the LAG:

```
set interfaces ae0 enable
set interfaces ae0 flexible-vlan-tagging
set interfaces ae0 encapsulation extended-vlan-bridge
set interfaces ae0 aggregated-ether-options link-protection rtg-config
set interfaces ae0 aggregated-ether-options minimum-links 1
set interfaces ae0 unit 300 vlan-id 300
set vlans qinqvlan interface ae0.300
set vlans qinqvlan interface ge-0/0/0.300
set interfaces ae0 unit 400 vlan-id 400
set vlans qinqvlan interface ae0.400
set vlans qinqvlan interface ge-0/0/0.400
```

3. Assign the member interfaces on the LAG and assign the two redundant trunk links to the *logical* interfaces; set one of the links as primary and one as backup:

```
set interfaces ge-0/0/0 enable
set interfaces ge-0/0/0 ether-options 802.3ad ae0
set interfaces ge-0/0/0.300 ether-options 802.3ad primary
set interfaces ge-0/0/0.400 ether-options 802.3ad backup
```

See Also

- show mac-refresh on page 271
- *show lacp interfaces*
- *show interfaces ge-*

Verifying That Redundant Trunk Links Are Available on the LAG and Viewing Active Links

Purpose Verify that the redundant trunk links are available on the LAG, and see which interfaces are configured as the primary (active) links.

Action Use the following **show** commands:

- show mac-refresh***interface-name***—Display whether redundant trunk links on a LAG with link protection are enabled on the specified interface.
- **show interfaces ge-** ***interface-name*** **extensive** or **show interfaces xe-** ***interface-name*** **extensive**—On a static LAG, display which interface is set as the primary member.
- **show lacp interfaces**—On an LACP LAG, display which member interfaces are active and which are down.

Related Documentation

- Understanding Redundant Trunk Links on page 126

CHAPTER 10

Configuring Proxy ARP

Understanding Proxy ARP on EX Series Switches

You can configure proxy Address Resolution Protocol (ARP) on your Juniper Networks EX Series Ethernet Switch 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.

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.)

EX Series switches support two modes of proxy ARP, 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 on EX Series Switches

We recommend these best practices for configuring proxy ARP on the switches:

- Set proxy ARP on the interfaces that you want, including IRB interfaces or RVIs, to restricted mode.
- If you set proxy ARP to unrestricted, disable gratuitous ARP requests on each interface enabled for proxy ARP.

Related Documentation

- Example: Configuring Proxy ARP on an EX Series Switch on page 145
- Configuring Proxy ARP on Devices with ELS Support (CLI Procedure) on page 145

Configuring Proxy ARP on Devices with ELS Support (CLI Procedure)

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 (CLI Procedure)* or *Configuring Proxy ARP on Switches*. For ELS details, see *Getting Started with Enhanced Layer 2 Software*.

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
```

Related Documentation

- Example: Configuring Proxy ARP on an EX Series Switch on page 145
- Verifying That Proxy ARP Is Working Correctly on page 148
- Configuring Integrated Routing and Bridging Interfaces (CLI Procedure) on page 68

Example: Configuring Proxy ARP on an EX Series Switch

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 MAC address. With proxy ARP enabled, the switch captures and routes traffic to the intended destination.

This example shows how to configure proxy ARP on an access switch:

Requirements

This example uses the following hardware and software components:

- One EX Series switch
- Junos OS Release 10.0 or later for EX Series switches

Overview and Topology

This example shows the configuration of proxy ARP on an interface of an EX Series switch using restricted mode. In restricted mode, the switch does not act as a proxy for hosts on the same subnet.

The topology for this example consists of one EX Series switch. When a host wants to communicate with a host that is not already in its ARP table, it broadcasts an ARP request for the MAC address of the destination host:

- When proxy ARP is not enabled, a host that shares the same IP address replies directly to the ARP request, providing its MAC address, and future transmissions are sent directly to the destination host MAC address.
- When proxy ARP is enabled, the switch responds to ARP requests, providing the switch's MAC address—even when the destination IP address is the same as the source IP address. Thus, communications must be sent through the switch and then routed through the switch to the appropriate destination.

Configuration

To configure proxy ARP, perform the following tasks:

CLI Quick Configuration

To quickly configure proxy ARP on an interface, copy the following command and paste it into the switch terminal window:

```
[edit]
set interfaces ge-0/0/3 unit 0 proxy-arp restricted
```

Step-by-Step Procedure

You configure proxy ARP on individual interfaces.

1. To configure proxy ARP on an 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 does not act as 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 a situation wherein the switch's response to a gratuitous ARP request appears to the host to be an indication of an IP conflict.

```
[edit interfaces]
user@switch# set ge-0/0/3 no-gratuitous-arp-request
```

Results

Display the results of the configuration:

```
user@switch> show configuration
  interfaces {
    ge-0/0/3 {
      unit 0 {
        proxy-arp restricted;
        family ethernet-switching;
      }
    }
```

Verification

To verify that the switch is sending proxy ARP messages, perform these tasks:

- Verifying That the Switch Is Sending Proxy ARP Messages on page 147

Verifying That the Switch Is Sending Proxy ARP Messages

Purpose

Verify that the switch is sending proxy ARP messages.

Action

List the system statistics for ARP messages:

```
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.

Related Documentation

- Configuring Proxy ARP on Devices with ELS Support (CLI Procedure) on page 145
- Understanding Proxy ARP on EX Series Switches on page 143

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.

Related Documentation

- *Configuring Proxy ARP on Switches*
- Configuring Proxy ARP on Devices with ELS Support (CLI Procedure) on page 145

CHAPTER 11

Configuring Private VLANs

Understanding Private VLANs on EX Series Switches

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 the same VLAN, splitting a broadcast domain into multiple isolated broadcast subdomains and essentially putting secondary VLANs inside a primary VLAN.

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:

Typical Structure and Primary Application of PVLANs

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.

Following are the types of domains, interfaces, and ports that you configure within a PVLAN:

- 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—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.
- Community 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.
- Interswitch isolated VLAN—An interswitch isolated VLAN is a secondary VLAN nested within the primary VLAN. This VLAN is used to forward isolated VLAN traffic from one switch to another through a PVLAN trunk port.
- Promiscuous 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.

- RVI—On some EX switches, you can configure one RVI for the primary VLAN. When configured, this RVI routes Layer 3 packets received by isolated and community VLAN interfaces.
- 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.

Figure 10 on page 153 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 10: Private VLAN on a Single EX Switch

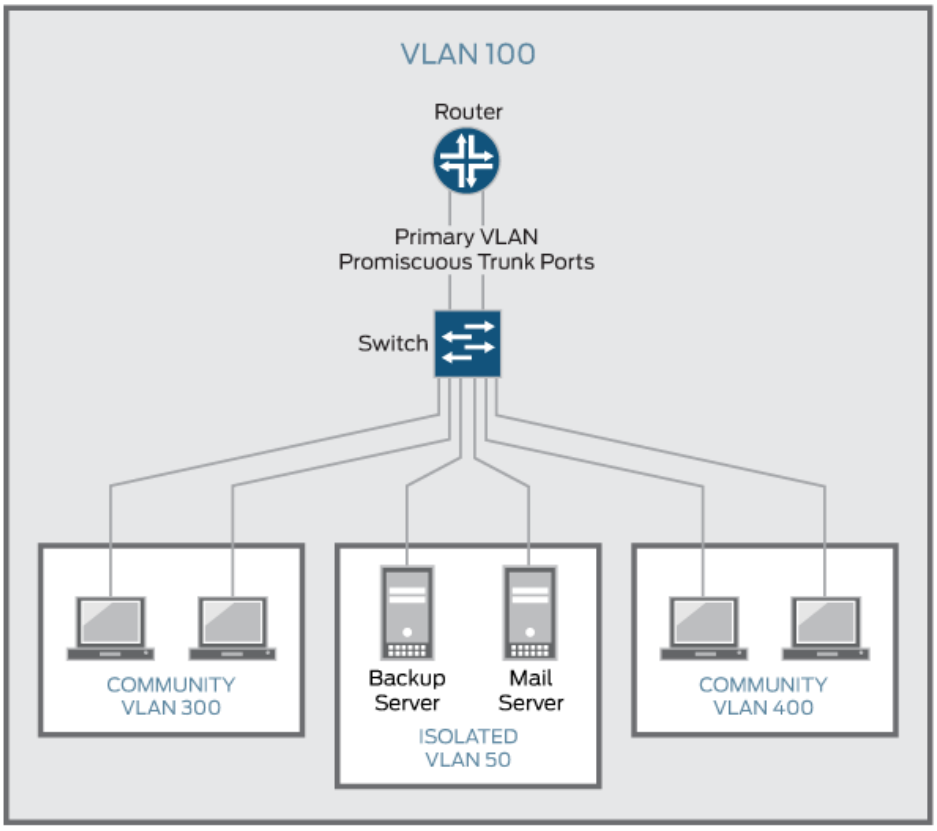
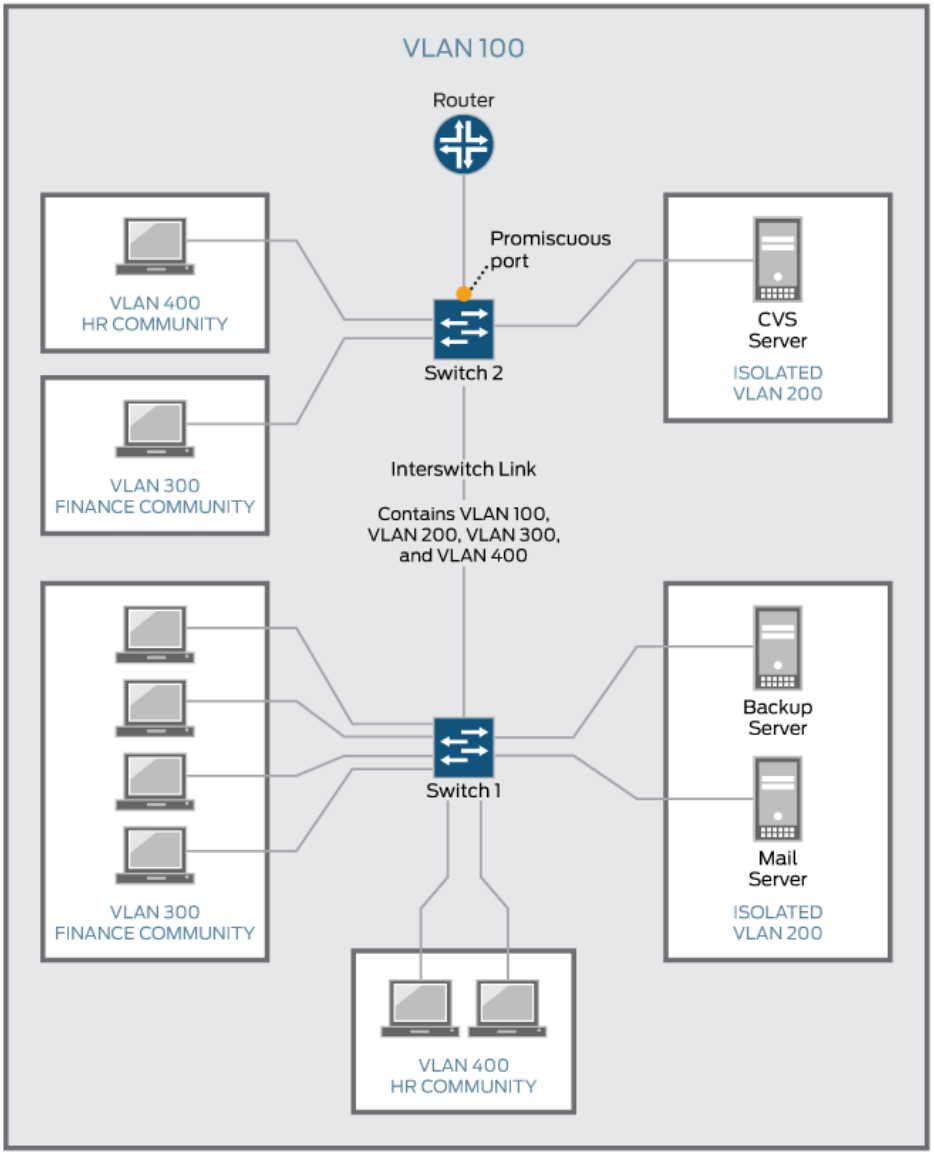


Figure 11 on page 154 shows a PVLAN spanning multiple switches, where the primary VLAN (VLAN **100**) contains two community VLANs (VLAN **300** and VLAN **400**) and one isolated VLAN (VLAN 200). It also shows that Switches 1 and 2 are connected through an interswitch link (PVLAN trunk link).

Figure 11: PVLAN Spanning Multiple EX Series Switches



Also, the PVLANs shown in Figure 10 on page 153 and Figure 11 on page 154 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 10 on page 153 or one of the switches shown in Figure 11 on page 154 (on some EX switches).

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)* and *Creating a Private VLAN Spanning Multiple EX Series Switches (CLI Procedure)*, respectively. For information about configuring an RVI, see *Configuring a Routed VLAN Interface in a Private VLAN (CLI Procedure)*.

Routing Between Isolated and Community VLANs

To route Layer 3 traffic between isolated and community VLANs, you must either connect a router to a promiscuous port, as shown in Figure 10 on page 153 and Figure 11 on page 154, 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.

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 13 on page 155 indicates when an 802.1Q tag is needed on the primary VLAN or on secondary VLANs.

Table 13: 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 802.1Q tag needed.	Specify an 802.1Q tag for each community VLAN by setting a VLAN ID. Specify an 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.

PVLANs Use Four Different Ethernet Switch Port Types

Table 14 on page 156 summarizes whether or not Layer 2 connectivity exists between the different types of ports within a PVLAN.

Table 14: PVLAN Ports and Layer 2 Connectivity

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 14 on page 156, 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.

Related Documentation

- Understanding Bridging and VLANs on EX Series Switches on page 19
- *Example: Configuring a Private VLAN on a Single EX Series Switch*
- *Example: Configuring a Private VLAN Spanning Multiple EX Series Switches*

Understanding PVLAN Traffic Flows Across Multiple Switches

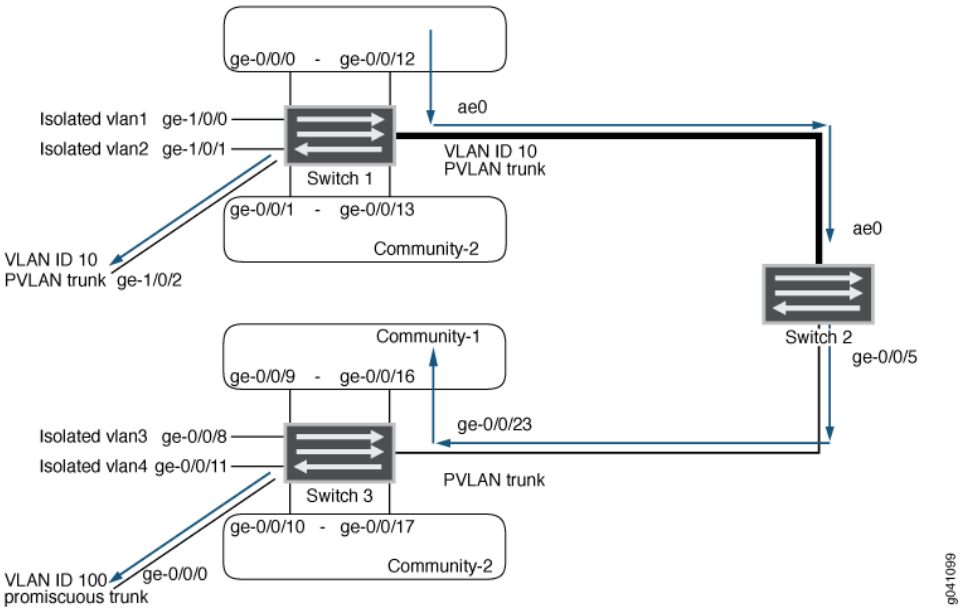
This topic illustrates and explains three different traffic flows on a sample multiswitch network configured with a private VLAN (PVLAN). PVLANs restrict traffic flows through their member switch ports (which are called “private ports”) so that they communicate only with a specific uplink trunk port 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 12 on page 157 represent this traffic flow.

Figure 12: 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

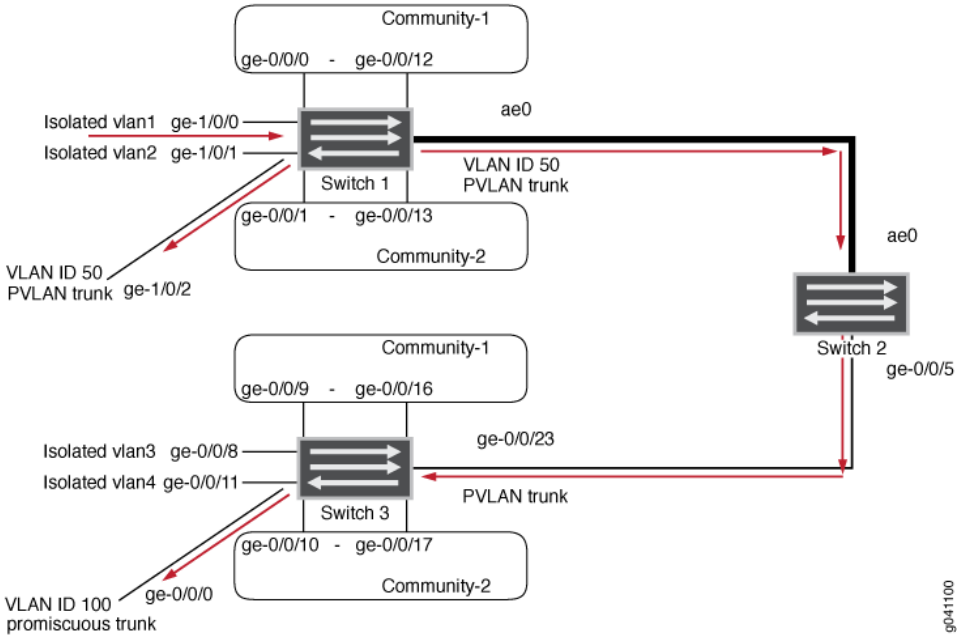
In this scenario, this activity takes place on Switch 3:

- 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 13 on page 158 represent this traffic flow.

Figure 13: 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

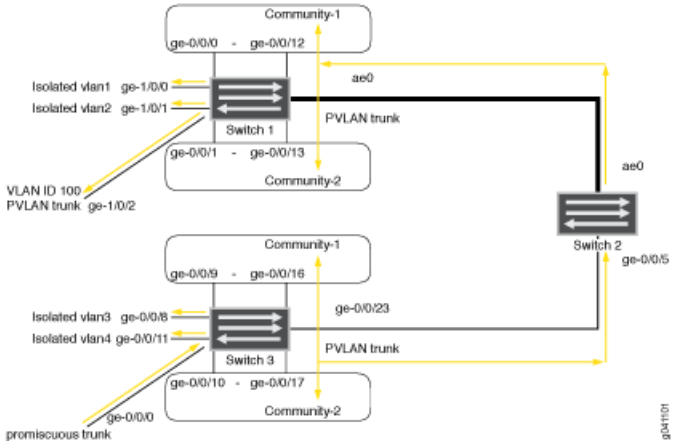
In this scenario, this activity takes place on Switch 3:

- 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 14 on page 159 represent this traffic flow.

Figure 14: 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

Related Documentation

- Understanding Private VLANs on EX Series Switches on page 151
- *Understanding Private VLANs*

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)*. For ELS details, see *Getting Started with Enhanced Layer 2 Software*.

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*.

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.

Related Documentation

- *Understanding Private VLANs*
- Creating a Private VLAN Spanning Multiple EX Series Switches (CLI Procedure) on page 162

Creating a Private VLAN Spanning Multiple EX Series Switches (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)*. For ELS details, see *Getting Started with Enhanced Layer 2 Software*.

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*.

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-private-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.

Related Documentation

- *Understanding Private VLANs*
- Example: Configuring a Private VLAN on a Single Switch with ELS Support on page 164
- Creating a Private VLAN on a Single Switch with ELS Support (CLI Procedure) on page 160

Example: Configuring a Private VLAN on a Single Switch with ELS Support

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*. For ELS details, see *Getting Started with Enhanced Layer 2 Software*.

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 on page 165
- Overview and Topology on page 165
- Configuration on page 166
- Verification on page 168

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 15 on page 165 lists the interfaces of the topology used in the example.

Table 15: 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

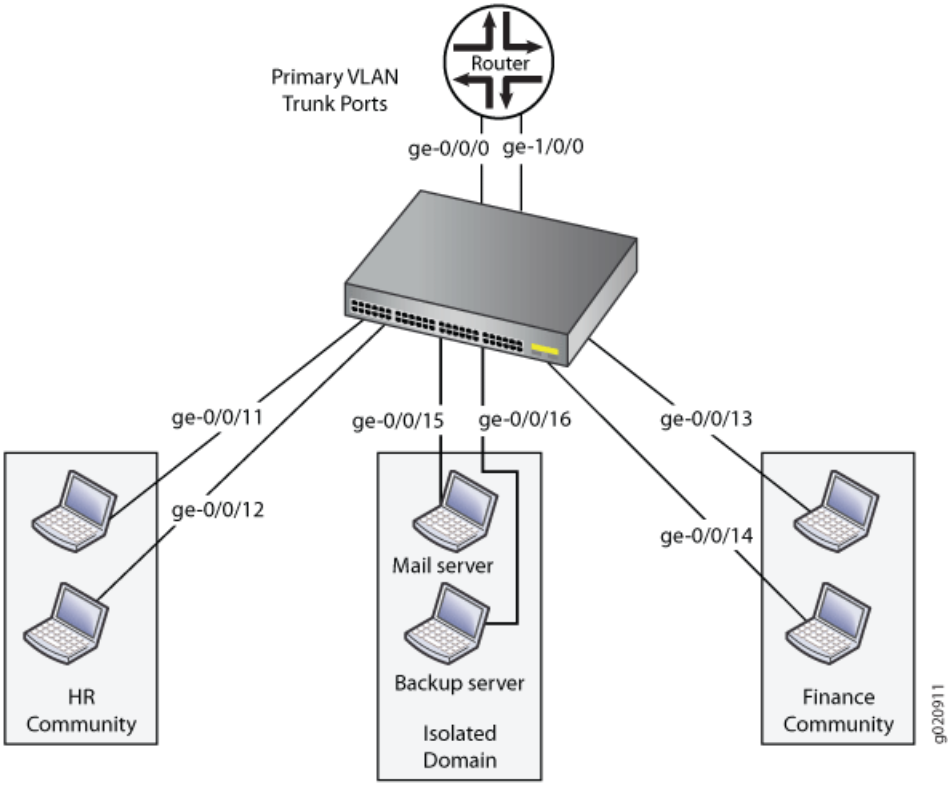
Table 16 on page 165 lists the VLAN IDs of the topology used in the example.

Table 16: 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 15 on page 166 shows the topology for this example.

Figure 15: 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
```

Results

Check the results of the configuration:

```
user@switch> show configuration
```

Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying That the Private VLAN and Secondary VLANs Were Created on page 168

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 extensive
```

Meaning The output shows that the primary VLAN was created and identifies the interfaces and secondary VLANs associated with it.

Related Documentation

- *Understanding Private VLANs*
- Creating a Private VLAN on a Single Switch with ELS Support (CLI Procedure) on page 160
- Creating a Private VLAN Spanning Multiple EX Series Switches (CLI Procedure) on page 162

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**.

Related Documentation

- *Creating a Private VLAN on a Single EX Series Switch (CLI Procedure)*
- Creating a Private VLAN Spanning Multiple EX Series Switches (CLI Procedure) on page 162
- Creating a Private VLAN on a Single Switch with ELS Support (CLI Procedure) on page 160
- Creating a Private VLAN Spanning Multiple EX Series Switches (CLI Procedure) on page 162

CHAPTER 12

Configuring MAC Notification

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.

Related Documentation

- Configuring MAC Notification (CLI Procedure) on page 176
- *Configuring SNMP (J-Web Procedure)*

Configuring MAC Notification (CLI Procedure)

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 (CLI Procedure)*. For ELS details, see *Getting Started with Enhanced Layer 2 Software*.

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 on page 176
- Disabling MAC Notification on page 176
- Setting the MAC Notification Interval on page 177

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
```

Related Documentation

- Verifying That MAC Notification Is Working Properly on page 177

Verifying That MAC Notification Is Working Properly on an EX Series Switch

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

Action Verify that MAC notification is enabled while also verifying the MAC notification interval setting.

```
user@switch> show ethernet-switching mac-notification
Notification Status: Enabled
Notification Interval: 30
```

Meaning 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.

Related Documentation

- Configuring MAC Notification (CLI Procedure) on page 176
- *Configuring MAC Notification on Switches with ELS Support (CLI Procedure)*

CHAPTER 13

Configuration Statements

address

List of Syntax

MX Series and EX Series (dynamic-profiles) on page 181
QFX Series and QFabric (interfaces) on page 181

MX Series and EX Series (dynamic-profiles)

```
address (ip-address | ipv6-address);
```

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*

aggregated-ether-options

Syntax

```
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
}
```

Hierarchy Level [edit interfaces ae*x*]

Release Information Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.3R2.

Description Configure the aggregated Ethernet properties of a specific aggregated Ethernet 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

- *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 (CLI Procedure)*
- Configuring Q-in-Q Tunneling (CLI Procedure) on page 109
- *Junos OS Ethernet Interfaces Configuration Guide*

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 Access 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*
- *Configuring Gigabit Ethernet Interfaces (CLI Procedure)*
- *Configuring Gigabit and 10-Gigabit Ethernet Interfaces*
- *Using DHCP Relay Agent Option 82 Information*
- *Junos OS Network Interfaces Library for Routing Devices*
- Example: Connecting Access Switches to a Distribution Switch on page 44

description (VLANs)

Syntax description *text-description*;

Hierarchy Level [edit vlans *vlan-name*]

Release Information Statement introduced in Junos OS Release 9.0 for EX Series switches.
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 of 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 contain 256 characters. If the text includes spaces, enclose the entire text in quotation marks.

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation

- *show vlans*
- *Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch*
- Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch on page 34
- Understanding Bridging and VLANs on EX Series Switches on page 19

encapsulation (Physical Interface)

Syntax 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);

Hierarchy Level [edit interfaces *interface-name*],
[edit interfaces rlsq *number:number*]

Release Information Statement introduced before Junos OS Release 7.4.
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 Specify the physical link-layer encapsulation type.

NOTE: Not all encapsulation types are supported on the switches. See the switch CLI.

Default **ppp**—Use serial PPP encapsulation.

Options

NOTE: Frame Relay, ATM, PPP, SONET, and SATSOP options are not supported on the 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-ccc—Use Ethernet CCC encapsulation on Ethernet interfaces that must accept packets carrying standard Tag Protocol ID (TPID) values. For 8-port, 12-port, and 48-port Fast Ethernet PICs, CCC is not supported.

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.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *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*
- *Configuring VLAN and Extended VLAN Encapsulation*
- *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

Syntax

```
ether-options {
  802.3ad {
    aex;
    (backup | primary);
    lacp {
      force-up;
      port-priority
    }
  }
  (auto-negotiation | no-auto-negotiation);
  ethernet-switch-profile {
    tag-protocol-id;
  }
  (flow-control | no-flow-control);
  ieee-802-3az-eee;
  link-mode mode;
  (loopback | no-loopback);
  speed (speed | auto-negotiation);
}
```

Hierarchy Level

[edit interfaces *interface-name*],
[edit interfaces interface-range *range*]

Release Information

Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.3R2.

Description

Configure Ethernet properties for a Gigabit Ethernet interface or a 10-Gigabit Ethernet 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

- *Configuring Gigabit Ethernet Interfaces (CLI Procedure)*
- *Configuring Gigabit Ethernet Interfaces (CLI Procedure)*
- *Configuring Gigabit Ethernet Interfaces (J-Web Procedure)*
- *Configuring LACP Link Protection of Aggregated Ethernet Interfaces (CLI Procedure)*
- Configuring Q-in-Q Tunneling (CLI Procedure) on page 109
- *Junos OS Ethernet Interfaces Configuration Guide*

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 Access 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 MAC Address Filtering*
- *Stacking and Rewriting Gigabit Ethernet VLAN Tags Overview*
- Configuring Q-in-Q Tunneling (CLI Procedure) on page 109

filter (VLANs)

Syntax filter (input | output) *filter-name*;

Hierarchy Level [edit vlans *vlan-name*]

Release Information Statement introduced in Junos OS Release 9.0 for EX Series switches.

Description Apply a firewall filter to traffic coming into or exiting from the 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.

Related Documentation

- *Example: Configuring Firewall Filters for Port, VLAN, and Router Traffic on EX Series Switches*
- *Configuring Firewall Filters (CLI Procedure)*
- *Configuring Firewall Filters (J-Web Procedure)*
- *Firewall Filters for EX Series Switches Overview*
- Configuring VLANs for EX Series Switches (CLI Procedure) on page 30

flexible-vlan-tagging

Syntax flexible-vlan-tagging;

Hierarchy Level [edit interfaces ae*x*],
[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

- *Configuring Mixed Tagging*
- *Configuring Flexible VLAN Tagging on PTX Series Packet Transport Routers*
- *Configuring Double-Tagged VLANs on Layer 3 Logical Interfaces*

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.
Support for logical systems added in Junos OS Release 9.6.

Description Configure the timeout interval for entries in the MAC table.

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

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*
- Configuring MAC Table Aging (CLI Procedure) on page 62
- *Configuring MAC Table Aging on Switches with ELS Support*

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.
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*
- Understanding Q-in-Q Tunneling on EX Series Switches on page 103

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 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 (Gigabit Ethernet IQ and 10-Gigabit Ethernet with SFPP). 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*
- output-vlan-map (Gigabit Ethernet IQ and 10-Gigabit Ethernet with SFPP) on page 230
- Configuring Q-in-Q Tunneling (CLI Procedure) on page 109

interface (MVRP)

Syntax

```
interface (all | interface-name) {
  disable;
  join-timer milliseconds;
  leave-timer milliseconds;
  leaveall-timer milliseconds;
  registration (forbidden | normal);
}
```

Hierarchy Level [edit protocols mvrp]

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 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*
- *Configuring Multiple VLAN Registration Protocol (MVRP) (CLI Procedure)*

interface (VLANs)

Syntax

```
interface interface-name {
  egress;
  ingress;
  mapping (native (push | swap) | policy | tag (push | swap));
  pvlan-trunk;
}
```

Hierarchy Level

[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.

Description

For a specific VLAN, configure an interface.

Options

interface-name—Name of a Gigabit Ethernet interface.

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

- *Configuring VLANs for EX Series Switches (CLI Procedure)*
- *Configuring Q-in-Q Tunneling (CLI Procedure)*
- Configuring Q-in-Q Tunneling (CLI Procedure) on page 109

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 for EX4300 switches.
Statement introduced in Junos OS Release 15.1X53-D55 for EX2300 and EX3400 switches.
Statement introduced in Junos OS Release 17.4R1 for EX4600 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 Through a Network Overview*
- Understanding Layer 2 Protocol Tunneling on EX Series Switches That Support Enhanced Layer 2 Software (ELS) on page 117

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

- *Layer 2 Learning and Forwarding for Bridge Domains Overview*
- *Layer 2 Learning and Forwarding for VLANs Overview*
- *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*

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 *Getting Started with Enhanced Layer 2 Software*.

(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*.

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*
- *Configuring a Logical Interface for Trunk Mode*
- Example: Connecting Access Switches to a Distribution Switch on page 44
- *Tunnel Services Overview*
- *Tunnel Interface Configuration on MX Series Routers Overview*

join-timer (MVRP)

Syntax join-timer *milliseconds*;

Hierarchy Level [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 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***—Number of milliseconds that the interface must wait before sending MVRP PDUs.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- leave-timer on page 211
- leaveall-timer on page 210
- *Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches*
- *Configuring Multiple VLAN Registration Protocol (MVRP) (CLI Procedure)*

l3-interface (VLANs)

Syntax

```
l3-interface l3-interface-name.logical-interface-number {
    l3-interface-ingress-counting;
}
```

Hierarchy Level [edit vlans *vlan-name*]

Release Information Statement introduced in Junos OS Release 9.0 for EX Series switches.

Description Associate a Layer 3 interface with the VLAN. Configure Layer 3 interfaces on trunk ports to allow the interface to transfer traffic between multiple VLANs. Within a VLAN, traffic is bridged, while across VLANs, traffic is routed.

Default No Layer 3 (routing) interface is associated with the VLAN.

Options *l3-interface-name.logical-interface-number*—Name of the Layer 3 interface and number of the logical interface defined by using the **set interfaces** vlan **unit** command. The name of the Layer 3 interface is irb for an integrated routing and bridging (IRB) interface, and vlan for a routed VLAN interface (RVI). The number of the logical interface is the same number that you configure in the **unit** statement.

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

- *show ethernet-switching interfaces*
- show ethernet-switching interface on page 254
- *show vlans*
- show vlans on page 284
- *Configuring Routed VLAN Interfaces (CLI Procedure)*
- Configuring Integrated Routing and Bridging Interfaces (CLI Procedure) on page 68

leaveall-timer (MVRP)

Syntax leaveall-timer *interval*;

Hierarchy Level

- For platforms with ELS:

  [edit protocols mvrp],
  [edit protocols mvrp interface *interface-name*]

- For platforms without ELS:

  [edit protocols mvrp interface (all | *interface-name*)]

Release Information Statement introduced in Junos OS Release 10.0 for EX Series switches.
Hierarchy level **[edit protocols mvrp]** introduced in Junos OS Release 13.2X50-D10 (ELS). (See *Getting Started with Enhanced Layer 2 Software* for information about ELS.)
Statement introduced in Junos OS Release 13.1 for the QFX Series.

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 *interval*—Number of seconds or milliseconds between the sending of Leave All messages.

Default: 10 seconds, or 10,000 milliseconds

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- join-timer on page 208
- leave-timer on page 211
- *Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches*
- Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches on page 87
- *Configuring Multiple VLAN Registration Protocol (MVRP) (CLI Procedure)*
- Configuring Multiple VLAN Registration Protocol (MVRP) (CLI Procedure) on page 83

leave-timer (MVRP)

Syntax leave-timer *milliseconds*;

Hierarchy Level [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 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***—Number of milliseconds 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.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- join-timer on page 208
- leaveall-timer on page 210
- *Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches*
- *Configuring Multiple VLAN Registration Protocol (MVRP) (CLI Procedure)*

link-protection-sub-group (aggregated-ether-options)

Syntax

```
link-protection-sub-group group-name {
  [primary | backup];
}
```

Hierarchy Level [edit interfaces ae*x* aggregated-ether-options]

Release Information Statement introduced in Junos OS Release 14.1X53-D10 for EX Series switches.

Description Create and name a link protection subgroup.

Link protection subgroups allow you to provide link protection to a collection of Ethernet links within a link aggregation group (LAG). If you need to provide link protection to a single link in a LAG, you do not need to configure link protection subgroups.

A link protection subgroup includes multiple links within the LAG. If one link in the primary link protection subgroup fails, traffic is forwarded over the links in the backup link protection subgroup.

Links within the LAG are added to the link protection subgroup using the link-protection-sub-group statement in the [**edit interfaces *interface-name* ether-options 802.3ad**] hierarchy.

Options

group-name—User-provided name of the link protection subgroup.

primary—Subgroup is the primary subgroup.

backup—Subgroup is the backup subgroup.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Configuring Aggregated Ethernet Links (CLI Procedure)*
- *Configuring LACP Link Protection of Aggregated Ethernet Interfaces (CLI Procedure)*
- Q-in-Q Support on Redundant Trunk Links Using LAGs with Link Protection on page 137

link-protection-sub-group (802.3ad)

Syntax link-protection-sub-group *group-name*;

Hierarchy Level [edit interfaces *interface-name* ether-options 802.3ad]

Release Information Statement introduced in Junos OS Release 14.1X53-D10 for EX Series switches.

Description Add an interface in an aggregated Ethernet bundle into a link-protection subgroup.

A link protection subgroup is created and named using the **link-protection-sub-group** statement in the **[edit interfaces ae*x* aggregated-ether-options]** hierarchy.

Options ***group-name***—Name of the link protection subgroup that will include this interface after this statement is entered. The link protection subgroup is named when it is created using the **link-protection-sub-group** statement in the **[edit interfaces ae*x* aggregated-ether-options]** hierarchy.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Configuring Aggregated Ethernet Links (CLI Procedure)*
- *Configuring LACP Link Protection of Aggregated Ethernet Interfaces (CLI Procedure)*
- Q-in-Q Support on Redundant Trunk Links Using LAGs with Link Protection on page 137

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 (CLI Procedure)*

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

- *Layer 2 Learning and Forwarding for Bridge Domains Overview*
- *Layer 2 Learning and Forwarding for VLANs Overview*
- *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*

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 for EX4300 switches.
Statement introduced in Junos OS Release 15.1X53-D55 for EX2300 and EX3400 switches.
Statement introduced in Junos OS Release 17.4R1 for EX4600 switches.

Description

Enable rewriting of the MAC address for Layer 2 protocol tunneling. When a control packet for a supported protocol is received on a service provider edge port configured for Layer 2 protocol tunneling (L2PT), the multicast destination MAC address is rewritten with the predefined multicast tunneling MAC address of 01:00:0c:cd:cd:d0. The packet is transported across the provider network transparently to the other end of the tunnel, and the original multicast destination MAC address is restored when the packet is transmitted.

Refer to protocol for the list of protocols that can be configured for L2PT on different devices.

To see the protocols for which L2PT tunneling is enabled for an interface, enter the show mac-rewrite interface command.

On MX 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 must subsequently be re-enabled 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 Through a Network Overview*
- Understanding Layer 2 Protocol Tunneling on EX Series Switches That Support Enhanced Layer 2 Software (ELS) on page 117

- Configuring Layer 2 Protocol Tunneling on EX Series Switches with ELS Support (CLI Procedure) on page 122
- **show mac-rewrite interface on page 272**
- *clear error mac-rewrite*

members

Syntax members [(all | *names* | *vlan-ids*)];

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* 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.

Description For trunk interfaces, configure the VLANs that 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 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.

NOTE: Since VLAN members are limited, specifying all could cause the number of VLAN members to exceed the limit at some point.

names—Name of one or more VLANs. VLAN IDs are applied automatically in this case.

NOTE: **all** cannot be a VLAN name.

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

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *show ethernet-switching interfaces*
- show ethernet-switching interface on page 254
- *show vlans*
- *Configuring Gigabit Ethernet Interfaces (CLI Procedure)*
- *Configuring Gigabit Ethernet Interfaces (CLI Procedure)*
- *Configuring Gigabit Ethernet Interfaces (J-Web Procedure)*
- *Configuring VLANs for EX Series Switches (CLI Procedure)*
- Configuring VLANs for EX Series Switches (CLI Procedure) on page 30

mvrp

Syntax

```
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 (Spanning Trees) {
    file filename <files number > <size size> < world-readable | no-world-readable>;
    flag <flag> <disable>;
  }
}
```

Hierarchy Level [edit protocols]

Release Information Statement introduced in Junos OS Release 13.2X50-D10 for EX Series switches.

Description

NOTE: If your switch command-line interface (CLI) displays different options for the mvrp statement from the options shown in this document, see *mvrp*.

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: In 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) (CLI Procedure)" on page 83.

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 EX Series Switches on page 87

native-vlan-id

Syntax native-vlan-id *number*;

Hierarchy Level [edit interfaces ge-*fpc/pic/port*],
[edit interfaces *interface-name*]

Release Information Statement introduced in Junos OS Release 8.3.
Statement introduced in Junos OS Release 12.2 for ACX Series Universal Access 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 mixed tagging support for untagged packets on a port for the following:

- 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

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.

NOTE: **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 interface-mode statement, untagged packets are accepted and forwarded within the bridge domain or VLAN that is configured with the matching VLAN ID.

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 *number*—VLAN ID number.

Range: (ACX Series routers and EX Series switches) 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 Mixed Tagging Support for Untagged Packets*
- *Configuring Access Mode on a Logical Interface*
- Configuring the Native VLAN Identifier on Switches With ELS Support (CLI Procedure) on page 33
- Understanding Bridging and VLANs on EX Series Switches on page 19
- **flexible-vlan-tagging on page 197**
- Understanding Q-in-Q Tunneling on EX Series Switches on page 103
- *no-native-vlan-insert*
- *Sending Untagged Traffic Without VLAN ID to Remote End*

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.

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) (CLI Procedure) on page 83
- Understanding Multiple VLAN Registration Protocol (MVRP) on EX Series Switches on page 79

no-dynamic-vlan

Syntax no-dynamic-vlan;

Hierarchy Level [edit protocols mvrp]
[edit protocols mvrp]

Release Information Statement introduced in Junos OS Release 10.0 for EX Series switches.

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) (CLI Procedure)*
- Configuring Multiple VLAN Registration Protocol (MVRP) (CLI Procedure) on page 83

no-gratuitous-arp-request

Syntax no-gratuitous-arp-request;

Hierarchy Level [edit interfaces *interface-name*]

Release Information Statement introduced in Junos OS Release 9.0 for EX Series switches.

Description Configure the switch not to respond to gratuitous ARP requests. You can disable responses to gratuitous ARP requests on Layer 2 Ethernet switching interfaces, and integrated routing and bridging (IRB) interfaces or routed VLAN interfaces (RVIs). (On EX Series switches that use 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.)

Default Gratuitous ARP responses are enabled on all Ethernet switching interfaces, and IRB interfaces or RVIs.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Proxy ARP on an EX Series Switch on page 145
- *Configuring Proxy ARP (CLI Procedure)*
- Configuring Proxy ARP on Devices with ELS Support (CLI Procedure) on page 145

no-mac-learning

Syntax no-mac-learning;

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* 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.
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 12.3 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 MX Series routers and EX Series switches, disable 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 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*
- *Layer 2 Learning and Forwarding for Bridge Domains Overview*
- *Layer 2 Learning and Forwarding for VLANs Overview*
- *Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports*
- Understanding Bridging and VLANs on EX Series Switches on page 19
- Understanding Q-in-Q Tunneling on EX Series Switches on page 103

output-vlan-map (Gigabit Ethernet IQ and 10-Gigabit Ethernet with SFPP)

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.

Description

For Gigabit Ethernet IQ and 10-Port 10-Gigabit Ethernet SFPP interfaces only, define 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*
- input-vlan-map on page 200

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 3D Universal Edge Routers.
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.

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 (CLI Procedure) on page 58
- *Configuring Persistent MAC Learning (CLI Procedure)*
- *Layer 2 Learning and Forwarding for Bridge Domains Overview*
- *Layer 2 Learning and Forwarding for VLANs Overview*
- *Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports*
- *Layer 2 Learning and Forwarding for VLANs Overview*
- *Layer 2 Learning and Forwarding for VLANs Acting as a Switch for a Layer 2 Trunk Port*

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*
- Configuring Q-in-Q Tunneling (CLI Procedure) on page 109

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-options redundant-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 *Getting Started with Enhanced Layer 2 Software* 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

- *Example: Configuring Redundant Trunk Links for Faster Recovery*
- Example: Configuring Redundant Trunk Links for Faster Recovery on page 129
- Q-in-Q Support on Redundant Trunk Links Using LAGs with Link Protection on page 137

protocol

List of Syntax

Syntax (MX Series Routers) on page 236
Syntax (EX2300 and EX3400 Switches) on page 236
Syntax (EX4300 and EX4600 Switches) on page 236
Syntax (EX9200 Switches) on page 236

Syntax (MX Series Routers)

```
protocol (cdp | pvstp | stp | vtp);
```

Syntax (EX2300 and EX3400 Switches)

```
protocol (cdp | gvrp | ieee8023ah | lacp | lldp | mvrp | stp | vstp | vtp);
```

Syntax (EX4300 and EX4600 Switches)

```
protocol (cdp | elmi | gvrp | ieee8021x | ieee8023ah | lacp | lldp | mmrp | mvrp | stp | udld |
    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 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 for EX4300 switches
Statement introduced in Junos OS Release 15.1X53-D55 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.

Description

Configure the protocol to be tunneled on an interface for Layer 2 protocol tunneling (L2PT). To enable tunneling multiple protocols, include multiple **protocol** statements.

Not all protocols listed in the Options section can be tunneled on all devices. The Syntax and Release Information sections list the available options for the protocols that can be tunneled by different devices as of a particular Junos OS release.

When a control packet for a supported protocol is received on a service provider edge port configured for Layer 2 protocol tunneling (L2PT), the multicast destination MAC address is rewritten with the predefined multicast tunneling MAC address of 01:00:0c:cd:cd:d0. The packet is transported across the provider network transparently to the other end of the tunnel, and the original multicast destination MAC address is restored when the packet is transmitted.

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 Through a Network Overview*
- *Layer 2 Protocol Tunneling Configuration Guidelines*
- *Configuring Layer 2 Protocol Tunneling*
- Understanding Layer 2 Protocol Tunneling on EX Series Switches That Support Enhanced Layer 2 Software (ELS) on page 117
- Configuring Layer 2 Protocol Tunneling on EX Series Switches with ELS Support (CLI Procedure) on page 122

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*
- *Configuring Proxy ARP (CLI Procedure)*
- Example: Configuring Proxy ARP on an EX Series Switch on page 145
- *Configuring Gratuitous ARP*

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*
- Configuring Q-in-Q Tunneling (CLI Procedure) on page 109

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 *Getting Started with Enhanced Layer 2 Software* 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*
- Example: Configuring Redundant Trunk Links for Faster Recovery on page 129
- Understanding Redundant Trunk Links on page 126

registration

Syntax registration (forbidden | normal);

Hierarchy Level [edit protocols mvrp interface (all | *interface-name*)],
[edit protocols mvrp interface *interface-name*]

Release Information Statement introduced in Junos OS Release 10.0 for EX Series switches.

Description Specifies the Multiple VLAN Registration Protocol (MVRP) registration mode for the interface if MVRP is enabled.

Default normal

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.

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) (CLI Procedure)*
- Configuring Multiple VLAN Registration Protocol (MVRP) (CLI Procedure) on page 83

rtg-config

Syntax

```
rtg-config {
  preempt-cutover-timer seconds;
}
```

Hierarchy Level

[edit interfaces *name* aggregated-ether-options lacp link-protection]
[edit interfaces *name* aggregated-ether-options link-protection]

Release Information

Statement introduced in Junos OS Release 17.4R1 for EX Series switches.

Description

Enable a redundant trunk group (RTG), with Q-in-Q support, on a link aggregation group (LAG) with link protection.

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

interface

Related Documentation

- Q-in-Q Support on Redundant Trunk Links Using LAGs with Link Protection on page 137

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*
- Configuring Q-in-Q Tunneling (CLI Procedure) on page 109

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 Access 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*
- Configuring Q-in-Q Tunneling (CLI Procedure) on page 109

vlan (802.1Q Tagging)

Syntax

```
vlan {
  members [(all | names | vlan-ids)];
}
```

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.

Description Bind an 802.1Q VLAN tag ID to a logical interface.

The remaining statement is 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

- *show ethernet-switching interfaces*
- show ethernet-switching interface on page 254
- *Example: Setting Up Bridging with Multiple VLANs for EX Series Switches*
- *Configuring Routed VLAN Interfaces (CLI Procedure)*
- Configuring Integrated Routing and Bridging Interfaces (CLI Procedure) on page 68
- Understanding Bridging and VLANs on EX Series Switches on page 19
- *Junos OS Ethernet Interfaces Configuration Guide*

vlan-id (802.1Q Tagging)

Syntax vlan-id *number*;

Hierarchy Level [edit vlans *vlan-name*]

Release Information Statement introduced in Junos OS Release 9.0 for EX Series switches.

Description 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.

Default If you use the default factory configuration, all traffic originating on the VLAN is untagged and has a VLAN identifier of **1**.

Options *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 system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation

- Example: Connecting Access Switches to a Distribution Switch on page 44
- Example: Configuring a Private VLAN on a Single Switch with ELS Support on page 164
- Creating a Private VLAN on a Single Switch with ELS Support (CLI Procedure) on page 160
- Creating a Private VLAN Spanning Multiple EX Series Switches (CLI Procedure) on page 162

vlan-id (VLAN Tagging and Layer 3 Subinterfaces)

Syntax vlan-id *vlan-id-number*;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number*]

Release Information Statement introduced in Junos OS Release 9.2 for EX Series switches.

Description 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 ***vlan-id-number***—A valid VLAN identifier.
Range: 1 through 4094

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *vlan-tagging*
- *Example: Configuring Layer 3 Subinterfaces for a Distribution Switch and an Access Switch*
- *Configuring Gigabit Ethernet Interfaces (CLI Procedure)*
- *Configuring Gigabit Ethernet Interfaces (CLI Procedure)*
- *Configuring Gigabit Ethernet Interfaces (J-Web Procedure)*
- *Configuring a Layer 3 Subinterface (CLI Procedure)*
- Configuring Q-in-Q Tunneling (CLI Procedure) on page 109
- *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.

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, 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*
- *Configuring VLAN Identifiers for Bridge Domains and VPLS Routing Instances*
- *Configuring VLAN Identifiers for VLANs and VPLS Routing Instances*

vlans

Syntax

```
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;
  }
}
```

Hierarchy Level

[edit],
[edit routing-instances *routing-instance-name*]

Release Information

Statement introduced in Junos OS Release 9.0 for EX Series switches.

Description

Configure VLAN properties on EX Series switches. 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.

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.

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 (CLI Procedure)*
- Configuring VLANs for EX Series Switches (CLI Procedure) on page 30
- Configuring Q-in-Q Tunneling (CLI Procedure) on page 109
- Configuring Integrated Routing and Bridging Interfaces (CLI Procedure) on page 68
- Understanding Bridging and VLANs on EX Series Switches on page 19

CHAPTER 14

Operational Commands

- show ethernet-switching interface
- show ethernet-switching table
- show interfaces irb
- show mac-refresh
- show mac-rewrite interface
- show mvrp
- show mvrp dynamic-vlan-memberships
- show mvrp statistics
- show redundant-trunk-group
- show system statistics arp
- show vlans

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 255
show ethernet-switching interface detail on page 256

Output Fields

Table 17 on page 254 describes the output fields for the **show ethernet-switching interface** command. Output fields are listed in the approximate order in which they appear.

Table 17: show ethernet-switching interface Output Fields

Field Name	Field Description
Logical interface	Name of the logical interface.
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.

Table 17: show ethernet-switching interface Output Fields *(continued)*

Field Name	Field Description
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.
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 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 table

Syntax

```
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)>
```

Release Information

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

(EX Series switches only) Display Layer 2 MAC address information.

Options

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.

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

List of Sample Output

show ethernet-switching table on page 259
show ethernet-switching table brief on page 260
show ethernet-switching table count on page 261
show ethernet-switching table extensive on page 262

Output Fields Table 18 on page 258 describes the output fields for the **show ethernet-switching table** command. Output fields are listed in the approximate order in which they appear.

Table 18: 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.
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.

Table 18: show ethernet-switching table Output fields *(continued)*

Field Name	Field Description
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.

Sample Output

show ethernet-switching table

```
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
```

show interfaces irb

Syntax

show interfaces irb
<brief | detail | extensive | terse>
<descriptions>
<media>
<routing-instance *instance-name*>
<snmp-index *snmp-index*>
<statistics>

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 13.2 for the QFX Series

Description

Display integrated routing and bridging interfaces information.

Options

brief | detail | extensive | terse—(Optional) Display the specified level of output.

descriptions—(Optional) Display interface description strings.

media—(Optional) Display media-specific information about network interfaces.

routing-instance *instance-name*—(Optional) Display information for the interface with the specified SNMP index.

snmp-index *snmp-index*—(Optional) Display information for the interface with the specified SNMP index.

statistics—(Optional) Display static interface statistics.

Additional Information

Integrated routing and bridging (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 VLAN that has a Layer 3 protocol configured.

Required Privilege Level

view

List of Sample Output

show interfaces irb extensive on page 268
show interfaces irb snmp-index on page 269

Output Fields

Table 19 on page 264 lists the output fields for the **show interfaces irb** command. Output fields are listed in the approximate order in which they appear.

Table 19: show interfaces irb Output Fields

Field Name	Field Description	Level of Output
Physical Interface		
Physical interface	Name of the physical interface.	All levels

Table 19: show interfaces irb Output Fields *(continued)*

Field Name	Field Description	Level of Output
Enabled	State of the physical interface. Possible values are described in the "Enabled Field" section under *Common Output Fields Description*.	All levels
Proto	Protocol configured on the interface.	**terse**
Interface index	Physical interface index number, which reflects its initialization sequence.	**detail extensive** none
SNMP ifIndex	SNMP index number for the physical interface.	**detail extensive** none
Type	Physical interface type.	**detail extensive** none
Link-level type	Encapsulation being used on the physical interface.	**detail extensive brief** none
MTU	MTU size on the physical interface.	**detail extensive brief** none
Clocking	Reference clock source: **Internal** or **External**. Always unspecified on IRB interfaces.	**detail extensive brief**
Speed	Speed at which the interface is running. Always unspecified on IRB interfaces.	**detail extensive brief**
Device flags	Information about the physical device. Possible values are described in the "Device Flags" section under *Common Output Fields Description*.	**detail extensive brief** none
Interface flags	Information about the interface. Possible values are described in the "Interface Flags" section under *Common Output Fields Description*.	**detail extensive brief** none
Link type	Physical interface link type: **full duplex** or **half duplex**.	**detail extensive** none
Link flags	Information about the link. Possible values are described in the "Links Flags" section under *Common Output Fields Description*.	**detail extensive** none
Physical Info	Physical interface information.	All levels
Hold-times	Current interface hold-time up and hold-time down, in milliseconds.	**detail extensive**
Current address	Configured MAC address.	**detail extensive** none
Hardware address	MAC address of the hardware.	**detail extensive** none
Alternate link address	Backup address of the link.	**detail extensive**
Last flapped	Date, time, and how long ago the interface went from down to up. The format is **Last flapped: year-month-day hours:minutes:seconds timezone (hours:minutes:seconds ago)**. For example, **Last flapped: 2002-04-26 10:52:40 PDT (04:33:20 ago)**.	**detail extensive** none

Table 19: show interfaces irb Output Fields *(continued)*

Field Name	Field Description	Level of Output
Statistics last cleared	Time when the statistics for the interface were last set to zero.	**detail extensive**
Traffic statistics	Number and rate of bytes and packets received and transmitted on the physical interface. • **Input bytes**—Number of bytes received on the interface. • **Output bytes**—Number of bytes transmitted on the interface. • **Input packets**—Number of packets received on the interface • **Output packets**—Number of packets transmitted on the interface.	**detail extensive**
IPv6 transit statistics	Number of IPv6 transit bytes and packets received and transmitted on the physical interface if IPv6 statistics tracking is enabled. • **Input bytes**—Number of bytes received on the interface. • **Output bytes**—Number of bytes transmitted on the interface. • **Input packets**—Number of packets received on the interface. • **Output packets**—Number of packets transmitted on the interface.	**detail extensive**
Input errors	Input errors on the interface. The following paragraphs explain the counters whose meaning might not be obvious: • **Errors**—Sum of the incoming frame aborts and FCS errors. • **Drops**—Number of packets dropped by the input queue of the I/O Manager ASIC. If the interface is saturated, this number increments once for every packet that is dropped by the ASIC's RED mechanism. • **Framing errors**—Number of packets received with an invalid frame checksum (FCS). • **Runts**—Number of frames received that are smaller than the runt threshold. • **Giants**—Number of frames received that are larger than the giant threshold. • **Policed discards**—Number of frames that the incoming packet match code discarded because they were not recognized or not of interest. Usually, this field reports protocols that the Junos OS does not handle. • **Resource errors**—Sum of transmit drops.	**detail extensive**
Output errors	Output errors on the interface. The following paragraphs explain the counters whose meaning might not be obvious: • **Carrier transitions**—Number of times the interface has gone from **down** to **up**. This number does not normally increment quickly, increasing only when the cable is unplugged, the far-end system is powered down and up, or another problem occurs. If the number of carrier transitions increments quickly (perhaps once every 10 seconds), the cable, the far-end system, or the DPC is malfunctioning. • **Errors**—Sum of the outgoing frame aborts and FCS errors. • **Drops**—Number of packets dropped by the output queue of the I/O Manager ASIC. If the interface is saturated, this number increments once for every packet that is dropped by the ASIC's RED mechanism. • **MTU errors**—Number of packets whose size exceeded the MTU of the interface. • **Resource errors**—Sum of transmit drops.	**detail extensive**

Table 19: show interfaces irb Output Fields *(continued)*

Field Name	Field Description	Level of Output
Logical Interface		
Logical interface	Name of the logical interface.	All levels
Index	Index number of the logical interface (which reflects its initialization sequence).	**detail extensive** none
SNMP ifIndex	SNMP interface index number of the logical interface.	**detail extensive** none
Generation	Unique number for use by Juniper Networks technical support only.	**detail extensive**
Flags	Information about the logical interface. Possible values are described in the "Logical Interface Flags" section under *Common Output Fields Description*.	**detail extensive**
Encapsulation	Encapsulation on the logical interface.	**detail extensive**
Bandwidth	Dummy value that is ignored by an IRB interface. IRB interfaces are pseudo interfaces and do not have physical bandwidth associated with them.	**detail extensive**
Routing Instance	Routing instance IRB is configured under.	**detail extensive**
Bridging Domain	Bridging domain IRB is participating in.	**detail extensive**
Traffic statistics	Number and rate of bytes and packets received and transmitted on the logical interface. • **Input bytes**—Number of bytes received on the interface. • **Output bytes**—Number of bytes transmitted on the interface. • **Input packets**—Number of packets received on the interface • **Output packets**—Number of packets transmitted on the interface.	**detail extensive**
IPv6 transit statistics	Number of IPv6 transit bytes and packets received and transmitted on the logical interface if IPv6 statistics tracking is enabled. • **Input bytes**—Number of bytes received on the interface. • **Output bytes**—Number of bytes transmitted on the interface. • **Input packets**—Number of packets received on the interface. • **Output packets**—Number of packets transmitted on the interface.	**detail extensive**
Local statistics	Statistics for traffic received from and transmitted to the Routing Engine.	**detail extensive**
Transit statistics	Statistics for traffic transiting the router.	**detail extensive**
Protocol	Protocol family configured on the local interface. Possible values are described in the "Protocol Field" section under *Common Output Fields Description*.	**detail extensive**
MTU	Maximum transmission unit size on the logical interface.	**detail extensive**

Table 19: show interfaces irb Output Fields *(continued)*

Field Name	Field Description	Level of Output
Maximum labels	Maximum number of MPLS labels configured for the MPLS protocol family on the logical interface.	**detail extensive** none
Generation	Unique number for use by Juniper Networks technical support only.	**detail extensive**
Route table	Routing table in which the logical interface address is located. For example, **0** refers to the routing table inet.0.	**detail extensive**
Addresses, Flags	Information about address flags. Possible values are described in the "Addresses Flags" section under *Common Output Fields Description*.	**detail extensive**
Policer	The policer that is to be evaluated when packets are received or transmitted on the interface.	**detail extensive**
Flags	Information about the logical interface. Possible values are described in the "Logical Interface Flags" section under *Common Output Fields Description*.	**detail extensive**

Sample Output

show interfaces irb extensive

```
user@host> show interfaces irb extensive
Physical interface: irb, Enabled, Physical link is Up
  Interface index: 129, SNMP ifIndex: 23, Generation: 130
  Type: Ethernet, Link-level type: Ethernet, MTU: 1514, Clocking: Unspecified,
Speed: Unspecified
  Device flags   : Present Running
  Interface flags: SNMP-Traps
  Link type      : Full-Duplex
  Link flags     : None
  Physical info  : Unspecified
  Hold-times     : Up 0 ms, Down 0 ms
  Current address: 02:00:00:00:00:30, Hardware address: 02:00:00:00:00:30
  Alternate link address: Unspecified
  Last flapped   : Never
  Statistics last cleared: Never
  Traffic statistics:
   Input  bytes  :                   0
   Output bytes  :                   0
   Input  packets:                   0
   Output packets:                   0
   IPv6 transit statistics:
    Input  bytes  :                  0
    Output bytes  :                  0
    Input  packets:                  0
    Output packets:                  0
  Input errors:
    Errors: 0, Drops: 0, Framing errors: 0, Runts: 0, Giants: 0, Policed discards:
 0, Resource errors: 0
  Output errors:
    Carrier transitions: 0, Errors: 0, Drops: 0, MTU errors: 0, Resource errors:
 0
```

```
  Logical interface irb.0 (Index 68) (SNMP ifIndex 70) (Generation 143)
    Flags: Hardware-Down SNMP-Traps 0x4000 Encapsulation: ENET2
    Bandwidth: 1000mbps
    Routing Instance: customer_0 Bridging Domain: bd0
    Traffic statistics:
     Input  bytes  :                   0
     Output bytes  :                   0
     Input  packets:                   0
     Output packets:                   0
     IPv6 transit statistics:
      Input  bytes  :                  0
      Output bytes  :                  0
      Input  packets:                  0
      Output packets:                  0
    Local statistics:
     Input  bytes  :                   0
     Output bytes  :                   0
     Input  packets:                   0
     Output packets:                   0
    Transit statistics:
     Input  bytes  :                   0                    0 bps
     Output bytes  :                   0                    0 bps
     Input  packets:                   0                    0 pps
     Output packets:                   0                    0 pps
     IPv6 transit statistics:
      Input  bytes  :                  0
      Output bytes  :                  0
      Input  packets:                  0
      Output packets:                  0
    Protocol inet, MTU: 1500, Generation: 154, Route table: 0
      Addresses, Flags: Dest-route-down Is-Preferred Is-Primary
        Destination: 10.51.1/8, Local: 10.51.1.2, Broadcast: 10.51.1.255,
        Generation: 155
    Protocol multiservice, MTU: 1500, Generation: 155, Route table: 0
      Flags: Is-Primary
      Policer: Input: __default_arp_policer
```

show interfaces irb snmp-index

```
user@host> show interfaces irb snmp-index 25
Physical interface: irb, Enabled, Physical link is Up
  Interface index: 128, SNMP ifIndex: 25
  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: 02:00:00:00:00:30, Hardware address: 02:00:00:00:00:30
  Last flapped   : Never
    Input packets : 0
    Output packets: 0

  Logical interface irb.0 (Index 68) (SNMP ifIndex 70)
    Flags: Hardware-Down SNMP-Traps 0x4000 Encapsulation: ENET2
    Bandwidth: 1000mbps
    Routing Instance: customer_0 Bridging Domain: bd0
    Input packets : 0
    Output packets: 0
    Protocol inet, MTU: 1500
      Addresses, Flags: Dest-route-down Is-Preferred Is-Primary
        Destination: 10.51.1/8, Local: 10.51.1.2, Broadcast: 10.51.1.255
```

```
Protocol multiservice, MTU: 1500
  Flags: Is-Primary
```

show mac-refresh

Syntax show mac-refresh *interface-name*

Release Information Command introduced in Junos OS Release 17.4R1 for EX Series switches.

Description Discover whether redundant trunk links on a LAG with link protection are enabled on the specified interface.

Options ***interface-name***—Name of the interface.

Required Privilege Level view

Related Documentation
- Q-in-Q Support on Redundant Trunk Links Using LAGs with Link Protection on page 137

List of Sample Output show mac-refresh interface ge-0/0/0 on page 271

Output Fields Table 20 on page 271 describes the output fields for the **show mac-refresh** ***interface-name*** command. Output fields are listed in the approximate order in which they appear.

Table 20: show mac-refresh Output Fields

Field Name	Field Description
Interface Name	Name of the interface.
RTG-Config	State of the interface: **Enabled** or **Not enabled**.
Preemptive Timer ***seconds***	Value set through the preemptive-timer statement.

Sample Output

show mac-refresh interface ge-0/0/0

```
user@switch> show mac-refresh interface ge-0/0/0
Interface Name : ge-0/0/0
RTG-Config : Enabled
Preemptive-Timer: 5
```

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 for EX4300 switches.
Command introduced in Junos OS Release 15.1X53-D55 for EX2300 and EX3400 switches.
Command introduced in Junos OS Release 17.4R1 for EX4600 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*
- mac-rewrite on page 217
- protocol on page 236
- *Understanding Layer 2 Protocol Tunneling Through a Network Overview*
- *Layer 2 Protocol Tunneling Configuration Guidelines*
- *Configuring Layer 2 Protocol Tunneling*
- Understanding Layer 2 Protocol Tunneling on EX Series Switches That Support Enhanced Layer 2 Software (ELS) on page 117
- Configuring Layer 2 Protocol Tunneling on EX Series Switches with ELS Support (CLI Procedure) on page 122

List of Sample Output

show mac-rewrite interface on page 273
show mac-rewrite interface (EX Series Switches) on page 273

Output Fields

Table 21 on page 272 lists the output fields for the **show mac-rewrite interface** command. Output fields are listed in the approximate order in which they appear.

Table 21: 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**

Table 21: show mac-rewrite interface Output Fields *(continued)*

Field Name	Field Description	Level of Output
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 10.1.
Command introduced in Junos OS Release 13.2X50-D10 for EX Series switches.

Description Display Multiple VLAN Registration Protocol (MVRP) configuration information.

Required Privilege Level view

Related Documentation

- *show mvrp applicant-state*
- show mvrp dynamic-vlan-memberships on page 276
- *show mvrp interface*
- *show mvrp registration-state*
- *show mvrp statistics*

List of Sample Output show mvrp on page 275

Output Fields Table 22 on page 274 lists the output fields for the **show mvrp** command. Output fields are listed in the approximate order in which they appear.

Table 22: show mvrp Output Fields

Field Name	Field Description
MVRP dynamic VLAN creation	Displays whether global MVRP dynamic VLAN creation is **Enabled** 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.

Sample Output

show mvrp

```
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
```

show mvrp dynamic-vlan-memberships

Syntax show mvrp dynamic-vlan-memberships

Release Information Command introduced in Junos OS Release 10.1.
Command introduced in Junos OS Release 13.2X50-D10 for EX Series switches.

Description Display all VLANs that have been created dynamically using Multiple VLAN Registration Protocol (MVRP) on the router or switch.

Required Privilege Level clear

Related Documentation

- show mvrp on page 274
- *show mvrp applicant-state*
- *show mvrp interface*
- *show mvrp registration-state*
- *show mvrp statistics*

List of Sample Output show mvrp dynamic-vlan-memberships on page 276

Output Fields Table 23 on page 276 lists the output fields for the **show mvrp dynamic-vlan-memberships** command. Output fields are listed in the approximate order in which they appear.

Table 23: 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

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
    100 (s)       ge-11/3/0
    200 (s)       ge-11/3/0
    300 (s)
```

show mvrp statistics

Syntax
show mvrp statistics
<interface *interface-name*>
<routing-instance *routing-instance-name*>

Release Information
Command introduced in Junos OS Release 13.2X50-D10 (ELS).

Description
Display Multiple VLAN Registration Protocol (MVRP) statistics in the form of Multiple Registration Protocol data unit (MRPDU) messages.

Required Privilege Level
view

Related Documentation
- show mvrp on page 274
- Verifying That MVRP Is Working Correctly on page 100

List of Sample Output
show mvrp statistics on page 278

Output Fields
Table 24 on page 277 lists the output fields for the **show mvrp statistics** command. Output fields are listed in the approximate order in which they appear.

Table 24: 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.
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) (CLI Procedure)" on page 83.

Table 24: show mvrp statistics Output Fields *(continued)*

Field Name	Field Description
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) (CLI Procedure)" on page 83.
Received New	Number of New messages received on the switch.

Sample Output

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 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*
- Example: Configuring Redundant Trunk Links for Faster Recovery on page 129
- Understanding Redundant Trunk Links on page 126

List of Sample Output show redundant-trunk-group group-name Group1 on page 281

Output Fields Table 25 on page 280 lists the output fields for the **show redundant-trunk-group** command. Output fields are listed in the approximate order in which they appear.

Table 25: 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.
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

Syntax show system statistics arp

Release Information Command introduced in Junos OS Release 9.6 for EX Series switches.

Description Display system-wide Address Resolution Protocol (ARP) statistics.

Required Privilege Level view

Related Documentation
- Example: Configuring Proxy ARP on an EX Series Switch on page 145
- Verifying That Proxy ARP Is Working Correctly on page 148

show system statistics arp

```
user@switch> show system statistics arp
arp:
        90060 datagrams received
        34 ARP requests received
        610 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
        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
```

show vlans

Syntax

show vlans
<brief | detail | extensive>
<instance *instance-name*>
<logical-system *logical-system-name*>
<operational>
<*vlan-name*)>
<interface *interface-name*>

Release Information

Command introduced in Junos OS Release 12.3R2.
Command introduced in Junos OS Release 12.3R2 for EX Series switches.
Option **interface** introduced in Junos OS Release 13.2X50-D10 (ELS).

Description

(MX Series routers and EX Series switches only) Display VLAN information.

Options

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.

Required Privilege Level

view

List of Sample Output

Sample Output

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

```
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

```
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

```
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
```