



---

# Virtual Chassis Fabric Feature Guide

Release

15.1



---

Modified: 2016-10-26

Juniper Networks, Inc.  
1133 Innovation Way  
Sunnyvale, California 94089  
USA  
408-745-2000  
[www.juniper.net](http://www.juniper.net)

Juniper Networks, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

*Virtual Chassis Fabric Feature Guide*

15.1

Copyright © 2016, Juniper Networks, Inc.

All rights reserved.

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.html>. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

# Table of Contents

	About the Documentation . . . . .	xi
	Documentation and Release Notes . . . . .	xi
	Supported Platforms . . . . .	xi
	Using the Examples in This Manual . . . . .	xi
	Merging a Full Example . . . . .	xii
	Merging a Snippet . . . . .	xii
	Documentation Conventions . . . . .	xiii
	Documentation Feedback . . . . .	xv
	Requesting Technical Support . . . . .	xv
	Self-Help Online Tools and Resources . . . . .	xv
	Opening a Case with JTAC . . . . .	xvi
<b>Part 1</b>	<b>Virtual Chassis Fabric</b>	
<b>Chapter 1</b>	<b>Configuring Virtual Chassis Fabric . . . . .</b>	<b>3</b>
	Virtual Chassis Fabric Overview . . . . .	3
	Understanding Virtual Chassis Fabric Components . . . . .	5
	Spine-and-Leaf Topology . . . . .	5
	Spine Devices . . . . .	6
	Leaf Devices . . . . .	7
	Routing Engine Role . . . . .	7
	Linecard Role . . . . .	7
	Master Routing Engine Election Process . . . . .	8
	Virtual Chassis Ports (VCPs) . . . . .	8
	Automatic Virtual Chassis Port (VCP) Conversion . . . . .	9
	VCF Configuration Options . . . . .	9
	Fabric Mode . . . . .	10
	Mixed Mode . . . . .	10
	Virtual Management Ethernet Interface . . . . .	10
	Virtual Chassis Fabric Port Link Aggregation Group Bundles . . . . .	11
	Virtual Chassis Fabric License Requirements . . . . .	11
	Hardware Requirements for a Virtual Chassis Fabric . . . . .	12
	Software Requirements in a Virtual Chassis Fabric . . . . .	12
	Understanding Virtual Chassis Fabric Configuration . . . . .	13
	Virtual Chassis Fabric Setup . . . . .	13
	Autoprovisioned Virtual Chassis Fabric Configuration . . . . .	13
	Preprovisioned Virtual Chassis Fabric Configuration . . . . .	14
	Nonprovisioned Virtual Chassis Fabric Configuration . . . . .	14
	Configuration File Management in a VCF . . . . .	14
	Logging into a Virtual Chassis Fabric . . . . .	15
	Understanding Interface Numbering . . . . .	15

Understanding Mixed EX Series and QFX Series Virtual Chassis or Virtual Chassis Fabric	15
Virtual Chassis Fabric Summary	16
Understanding Mixed Virtual Chassis Fabric	17
Virtual Chassis Summary for QFX5100, QFX3600, QFX3500, EX4600, and EX4300 Switches	17
Understanding the Routing Engine Role in a Mixed Virtual Chassis Using EX4300, EX4600, QFX3500, QFX3600, or QFX5100 Member Switches	18
Understanding EX4300, QFX3500, QFX3600, and QFX5100 Switches in a Virtual Chassis	19
Understanding Mixed EX4300 and EX4600 Virtual Chassis	19
Understanding EX4200, EX4500, and EX4550 Switches in a Mixed Virtual Chassis	19
Understanding Traffic Flow Through a Virtual Chassis Fabric	20
Smart Trunking Algorithm for Unicast Traffic Forwarding	20
Multicast Distribution Trees for Broadcast, Unknown Unicast, and Multicast Traffic	20
Adaptive Load Balancing	21
Understanding Software Upgrades in a Virtual Chassis Fabric	22
Virtual Chassis Fabric Software Basics	22
Nonstop Software Upgrade (NSSU)	22
Automatic Software Update	22
Traditional Software Upgrade	23
Autoprovisioning a Virtual Chassis Fabric	23
Preprovisioning a Virtual Chassis Fabric	27
Configuring a Nonprovisioned Virtual Chassis Fabric	31
Adding a Device to a Virtual Chassis Fabric	34
Adding a Leaf Device to an Autoprovisioned Virtual Chassis Fabric	35
Adding a Spine Device to an Autoprovisioned Virtual Chassis Fabric	36
Adding a Spine or Leaf Device to a Preprovisioned Virtual Chassis Fabric	38
Adding a Spine or Leaf Device to a Nonprovisioned Virtual Chassis Fabric	40
Removing a Device From a Virtual Chassis Fabric	42
Upgrading Software for a Virtual Chassis Fabric	43
NSSU	43
Automatic Software Update	43
Standard Upgrade	44
Verifying the Member ID, Role, Status, and Neighbor Member Connections of a Virtual Chassis Fabric Member Device	44
Verifying Virtual Chassis Port Connections in a Virtual Chassis Fabric	46
Verifying the Virtual Chassis Fabric Mode Settings	46
Troubleshooting Virtual Chassis Fabric	47
Large-Scale Virtual Chassis Fabric Becomes Unstable When Logging is Enabled	47
Virtual Chassis Port Link Does Not Form	48
QFX5100 Leaf Device Assumes Routing Engine Role	48

<b>Chapter 2</b>	<b>Configuration Statements for Virtual Chassis Fabric</b>	<b>49</b>
	[edit virtual-chassis] Configuration Statement Hierarchy	50
	Supported Statements in the [edit virtual-chassis] Hierarchy Level	50
	Unsupported Statements in the [edit virtual-chassis] Hierarchy Level	51
	aliases (Virtual Chassis)	52
	alias-name (Virtual Chassis aliases)	53
	auto-provisioned	54
	auto-sw-update	55
	enhanced-hash-key	57
	fabric-load-balance	60
	fabric-tree-root	61
	id	62
	inactivity-interval (Fabric Load Balance)	63
	location (Virtual Chassis)	64
	mac-persistence-timer	65
	mastership-priority	66
	member	68
	no-management-vlan	69
	no-split-detection	70
	package-name	71
	preprovisioned	72
	role	73
	serial-number	76
	serial-number (Virtual Chassis aliases)	77
	traceoptions (Virtual Chassis)	78
	vcp-snmp-statistics	81
	virtual-chassis	82
<b>Chapter 3</b>	<b>Operational Commands for Virtual Chassis Fabric</b>	<b>85</b>
	clear virtual-chassis vc-port statistics	86
	request session member	88
	request virtual-chassis mode	89
	request virtual-chassis reactivate	92
	request virtual-chassis vc-port	93
	request virtual-chassis vc-port diagnostics optics	95
	show forwarding-options enhanced-hash-key	96
	show snmp mib	101
	show virtual-chassis active-topology	104
	show virtual-chassis device-topology	109
	show virtual-chassis login	115
	show virtual-chassis mode	116
	show virtual-chassis protocol adjacency	119
	show virtual-chassis protocol database	123
	show virtual-chassis protocol interface	127
	show virtual-chassis protocol route	130
	show virtual-chassis protocol statistics	133
	show virtual-chassis	136
	show virtual-chassis vc-path	141
	show virtual-chassis vc-port	143

show virtual-chassis vc-port diagnostics optics . . . . . 147

show virtual-chassis vc-port statistics . . . . . 161

Part 2

Index . . . . . 171

# List of Figures

Part 1	Virtual Chassis Fabric	
Chapter 1	Configuring Virtual Chassis Fabric .....	3
	Figure 1: VCF Spine-and-Leaf Architecture .....	4
	Figure 2: VCF Spine-and-Leaf Architecture .....	6





# List of Tables

	<b>About the Documentation</b> . . . . .	<b>xi</b>
	Table 1: Notice Icons . . . . .	xiii
	Table 2: Text and Syntax Conventions . . . . .	xiii
<b>Part 1</b>	<b>Virtual Chassis Fabric</b>	
<b>Chapter 1</b>	<b>Configuring Virtual Chassis Fabric</b> . . . . .	<b>3</b>
	Table 3: Virtual Chassis Fabric Summary . . . . .	17
	Table 4: Virtual Chassis Summary . . . . .	18
<b>Chapter 3</b>	<b>Operational Commands for Virtual Chassis Fabric</b> . . . . .	<b>85</b>
	Table 5: show forwarding-options enhanced-hash-key Output Fields . . . . .	96
	Table 6: show snmp mib Output Fields . . . . .	102
	Table 7: show virtual-chassis active-topology Output Fields . . . . .	104
	Table 8: show virtual-chassis device-topology Output Fields . . . . .	109
	Table 9: show virtual-chassis mode Output Fields . . . . .	116
	Table 10: show virtual-chassis protocol adjacency Output Fields . . . . .	120
	Table 11: show virtual-chassis protocol database Output Fields . . . . .	123
	Table 12: show virtual-chassis protocol interface Output Fields . . . . .	128
	Table 13: show virtual-chassis protocol route Output Fields . . . . .	130
	Table 14: show virtual-chassis protocol statistics Output Fields . . . . .	133
	Table 15: show virtual-chassis Output Fields . . . . .	136
	Table 16: show virtual-chassis vc-path Output Fields . . . . .	141
	Table 17: show virtual-chassis vc-port Output Fields . . . . .	143
	Table 18: show virtual-chassis vc-port diagnostics optics Output Fields . . . . .	148
	Table 19: show virtual-chassis vc-port statistics Output Fields . . . . .	162



# About the Documentation

- Documentation and Release Notes on page xi
- Supported Platforms on page xi
- Using the Examples in This Manual on page xi
- Documentation Conventions on page xiii
- Documentation Feedback on page xv
- Requesting Technical Support on page xv

## 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:

- QFX Series
- 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.xml;
    }
  }
}
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.xml; }
```

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 xiii](#) 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 xiii](#) defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
<b>Bold text like this</b>	Represents text that you type.	To enter configuration mode, type the <b>configure</b> command:  user@host> <b>configure</b>

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

Convention	Description	Examples
Fixed-width text like this	Represents output that appears on the terminal screen.	user@host> <b>show chassis alarms</b>  No alarms currently active
<i>Italic text like this</i>	<ul style="list-style-type: none"><li>Introduces or emphasizes important new terms.</li><li>Identifies guide names.</li><li>Identifies RFC and Internet draft titles.</li></ul>	<ul style="list-style-type: none"><li>A policy <i>term</i> is a named structure that defines match conditions and actions.</li><li><i>Junos OS CLI User Guide</i></li><li>RFC 1997, <i>BGP Communities Attribute</i></li></ul>
<i>Italic text like this</i>	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name:  [edit] root@# <b>set system domain-name</b> <i>domain-name</i>
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	<ul style="list-style-type: none"><li>To configure a stub area, include the <b>stub</b> statement at the [edit protocols ospf area area-id] hierarchy level.</li><li>The console port is labeled <b>CONSOLE</b>.</li></ul>
< > (angle brackets)	Encloses optional keywords or variables.	<b>stub &lt;default-metric <i>metric</i>&gt;;</b>
(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.	<b>broadcast   multicast</b>  <b>(<i>string1</i>   <i>string2</i>   <i>string3</i>)</b>
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	<b>rsvp { # Required for dynamic MPLS only</b>
[ ] (square brackets)	Encloses a variable for which you can substitute one or more values.	<b>community name members [</b> <i>community-ids</i> <b>]</b>
Indentation and braces ( { } )	Identifies a level in the configuration hierarchy.	[edit] routing-options { static { route default { nexthop <i>address</i> ; retain; } } }
;(semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
GUI Conventions		
Bold text like this	Represents graphical user interface (GUI) items you click or select.	<ul style="list-style-type: none"><li>In the Logical Interfaces box, select <b>All Interfaces</b>.</li><li>To cancel the configuration, click <b>Cancel</b>.</li></ul>

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

Convention	Description	Examples
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select <b>Protocols&gt;Ospf</b> .

## 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](mailto: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: <http://www2.juniper.net/kb/>
- Find product documentation: <http://www.juniper.net/techpubs/>
- 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://tools.juniper.net/SerialNumberEntitlementSearch/>

## 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>.



## PART 1

# Virtual Chassis Fabric

- [Configuring Virtual Chassis Fabric on page 3](#)
- [Configuration Statements for Virtual Chassis Fabric on page 49](#)
- [Operational Commands for Virtual Chassis Fabric on page 85](#)



## CHAPTER 1

# Configuring Virtual Chassis Fabric

- [Virtual Chassis Fabric Overview on page 3](#)
- [Understanding Virtual Chassis Fabric Components on page 5](#)
- [Understanding Virtual Chassis Fabric Configuration on page 13](#)
- [Understanding Mixed EX Series and QFX Series Virtual Chassis or Virtual Chassis Fabric on page 15](#)
- [Understanding Traffic Flow Through a Virtual Chassis Fabric on page 20](#)
- [Understanding Software Upgrades in a Virtual Chassis Fabric on page 22](#)
- [Autoprovisioning a Virtual Chassis Fabric on page 23](#)
- [Preprovisioning a Virtual Chassis Fabric on page 27](#)
- [Configuring a Nonprovisioned Virtual Chassis Fabric on page 31](#)
- [Adding a Device to a Virtual Chassis Fabric on page 34](#)
- [Removing a Device From a Virtual Chassis Fabric on page 42](#)
- [Upgrading Software for a Virtual Chassis Fabric on page 43](#)
- [Verifying the Member ID, Role, Status, and Neighbor Member Connections of a Virtual Chassis Fabric Member Device on page 44](#)
- [Verifying Virtual Chassis Port Connections in a Virtual Chassis Fabric on page 46](#)
- [Verifying the Virtual Chassis Fabric Mode Settings on page 46](#)
- [Troubleshooting Virtual Chassis Fabric on page 47](#)

## Virtual Chassis Fabric Overview

---

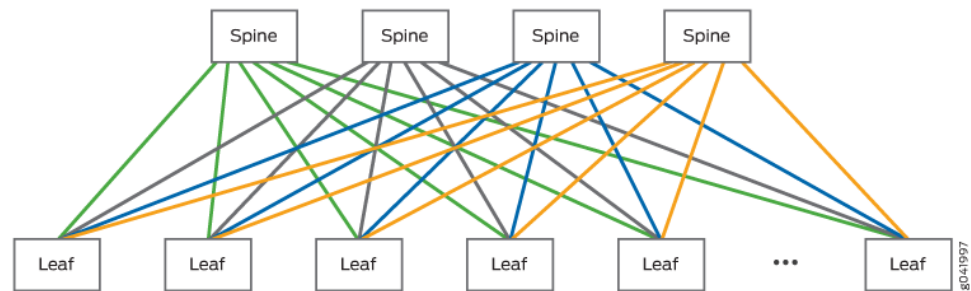
The Juniper Networks Virtual Chassis Fabric (VCF) provides a low-latency, high-performance fabric architecture that can be managed as a single device. VCF is an evolution of the Virtual Chassis feature, which enables you to interconnect multiple devices into a single logical device, inside of a fabric architecture. The VCF architecture is optimized to support small and medium-sized data centers that contain a mix of 1-Gbps, 10-Gbps, and 40-Gbps Ethernet interfaces.



Video: [What is Virtual Chassis Fabric?](#)

A VCF is constructed using a spine-and-leaf architecture. In the spine-and-leaf architecture, each spine device is interconnected to each leaf device. A VCF supports up to twenty total devices, and up to four devices can be configured as spine devices. See [Figure 1 on page 4](#) for an illustration of the VCF spine-and-leaf architecture.

**Figure 1: VCF Spine-and-Leaf Architecture**



Each spine device must be a QFX5100 device. In an optimal VCF configuration, the leaf devices are also QFX5100 devices. You can, however, also create a mixed VCF by configuring QFX3600, QFX3500, and EX4300 switches as leaf devices. See [“Understanding Virtual Chassis Fabric Components” on page 5](#) for more information about the spine-and-leaf architecture.

A VCF provides the following benefits:

- **Latency**—VCF provides predictable low latency because it uses a fabric architecture that ensures each device is one or two hops away from every other device in the fabric. The weighted algorithm that makes traffic-forwarding decisions in a VCF is designed to avoid congestion and ensures low latency by intelligently forwarding traffic over all paths within the VCF to any destination device., ensuring predictable low latency for all traffic traversing the VCF.
- **Resiliency**—The VCF architecture provides a resilient framework because traffic has multiple paths across the fabric. Traffic is, therefore, easily diverted within the fabric when a device or link fails.
- **Flexibility**—You can easily expand the size of your VCF by adding devices to the fabric as your networking needs grow.
- **Investment protection**—In environments that need to expand because the capabilities of a traditional QFX5100, QFX3600, QFX3500, or EX4300 Virtual Chassis are maximized, a VCF is often a logical upgrade option because it enables the system to evolve without having to remove the existing, previously purchased devices from the network.
- **Manageability**—VCF provides multiple features that simplify configuration and management. VCF, for instance, has an autoprovisioning feature that enables you to plug and play devices into the fabric after minimal initial configuration. VCF leverages many of the existing configuration procedures from a Virtual Chassis, so that you can configure and maintain a VCF easily if you are already familiar with the procedures for configuring and maintaining a Virtual Chassis.

- Related Documentation**
- [Network Configuration Example: MetaFabric™ Architecture 1.1: Configuring Virtual Chassis Fabric and Network Director 1.6](#)
  - [Understanding Virtual Chassis Fabric Components on page 5](#)
  - [Understanding Virtual Chassis Fabric Configuration on page 13](#)
  - [Autoprovisioning a Virtual Chassis Fabric on page 23](#)
  - [Preprovisioning a Virtual Chassis Fabric on page 27](#)

---

## Understanding Virtual Chassis Fabric Components

This topic describes the components of a Virtual Chassis Fabric (VCF).

This topic covers:

- [Spine-and-Leaf Topology on page 5](#)
- [Spine Devices on page 6](#)
- [Leaf Devices on page 7](#)
- [Routing Engine Role on page 7](#)
- [Linecard Role on page 7](#)
- [Master Routing Engine Election Process on page 8](#)
- [Virtual Chassis Ports \(VCPs\) on page 8](#)
- [Automatic Virtual Chassis Port \(VCP\) Conversion on page 9](#)
- [VCF Configuration Options on page 9](#)
- [Fabric Mode on page 10](#)
- [Mixed Mode on page 10](#)
- [Virtual Management Ethernet Interface on page 10](#)
- [Virtual Chassis Fabric Port Link Aggregation Group Bundles on page 11](#)
- [Virtual Chassis Fabric License Requirements on page 11](#)
- [Hardware Requirements for a Virtual Chassis Fabric on page 12](#)
- [Software Requirements in a Virtual Chassis Fabric on page 12](#)

### Spine-and-Leaf Topology

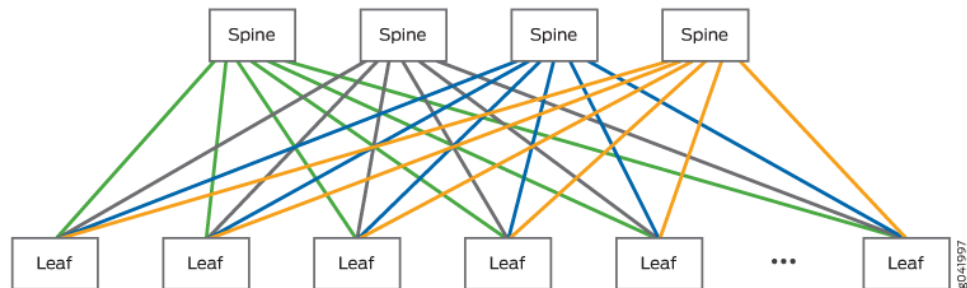
The VCF uses a spine-and-leaf architecture where each device in the fabric is either a spine device or a leaf device.

A VCF can have up to four spine devices, and up to twenty total devices. Each spine device has at least one direct Virtual Chassis port (VCP) connection to each leaf device in the VCF.

All traffic entering a leaf device can, therefore, be forwarded to any directly connected spine device and is always two hops away from any other leaf device—leaf device to leaf device traffic travels from the source leaf device to a spine device to the destination leaf device—within the VCF.

See [Figure 2 on page 6](#) for an illustration of the VCF spine-and-leaf architecture:

**Figure 2: VCF Spine-and-Leaf Architecture**



Traffic is forwarded through a VCF using a weighted algorithm designed to avoid congestion. Traffic travelling across the VCF from one leaf device to another leaf device is forwarded using the best path available at the time, so any connection to a spine device can be used to transport traffic from one leaf device to another leaf device.

## Spine Devices

A spine device:

- Must be a QFX5100 device.
- Can be configured into the Routing Engine or linecard role. In a VCF, two spine devices must be configured into the Routing Engine role. The remaining spine devices must be configured into the linecard role.
- Has a direct connection to each leaf device.
- Typically connects a router, firewall, or other data center networking device to the VCF.

A VCF should always have at least two active spine devices. A VCF supports up to four spine devices.



**BEST PRACTICE:** We recommend using the following QFX5100 switches as spine devices:

- QFX5100-24Q switches, in deployments where devices are connecting to the VCF using the 10-Gbps Ethernet interfaces on the leaf devices, or using a mix of 10-Gbps and 1-Gbps Ethernet interfaces on the leaf devices.
- QFX5100-96S or QFX5100-48S, in deployments where devices are connecting to the VCF using 1-Gbps Ethernet interfaces only on the leaf devices.

QFX5100-48T switches are not supported as spine devices.

## Leaf Devices

A leaf device:

- Is optimally a QFX5100 device, but can also be a QFX3500, QFX3600, or EX4300 device.
- Has a direct connection to each spine device.
- Always operates in the linecard role.
- Typically connects an endpoint device—for instance, a server or other storage device in a data center—to the VCF.

A VCF can have up to twenty total devices and up to four devices can be configured as spine devices. The devices that are not spine devices in a VCF operate as leaf devices.

## Routing Engine Role

A VCF has two devices operating in the Routing Engine role—a master Routing Engine and a backup Routing Engine.

The device that functions as the master Routing Engine:

- Is a spine device.
- Manages the member devices.
- Runs the chassis management processes and control protocols.
- Represents all the member devices interconnected within the VCF configuration. (The hostname and other parameters that you assign to this device during setup apply to all members of the VCF.)

The device that functions as the backup Routing Engine:

- Is a spine device.
- Maintains a state of readiness to take over the master role if the master fails.
- Synchronizes with the master in terms of protocol states, forwarding tables, and so forth, so that it preserves routing information and maintains network connectivity without disruption when the master is unavailable.

In a VCF, two spine devices are configured into the Routing Engine role. The remaining spine devices are configured into the linecard role.

A spine device operating in the linecard role can complete all spine-related functions with no limitations within a VCF.

## Linecard Role

A spine or a leaf device can be configured into the linecard role in a VCF.

In a VCF, two spine devices are configured into the Routing Engine role. The remaining spine devices are configured into the linecard role. A spine device operating in the linecard

role can complete all spine-related functions with no limitations within a VCF. A spine device operating in the linecard role does not become a Routing Engine when the master or backup Routing Engine fails.

All leaf devices in a VCF operate in the linecard role. In autoprovisioned configurations, leaf devices are assigned the linecard role when they are cabled into the VCF. In preprovisioned configurations, leaf devices are manually configured into the linecard role. In nonprovisioned configurations, leaf devices are assigned the linecard role according to the master election algorithm, which uses the mastership priority values to set the roles of each device in the VCF.

A member that functions in the linecard role in a VCF:

- Runs only a subset of Junos OS.
- Detects certain error conditions (such as an unplugged cable) on any interfaces that have been configured on it through the device functioning as the master Routing Engine.

## Master Routing Engine Election Process

The device in the master Routing Engine role in a VCF is always a spine device.

In a preprovisioned or autoprovisioned VCF, two spine devices are assigned the Routing Engine role during the configuration process. The spine device that has been powered on the longest assumes the master Routing Engine role; the spine device that has been powered on the second longest assumes the backup Routing Engine role.

In a nonprovisioned VCF, the master and backup Routing Engines are selected using the following algorithm:

1. Choose the QFX5100 device with the highest user-configured mastership priority (255 is the highest possible value) as the master Routing Engine, and the QFX5100 switch with the second highest mastership priority value as the backup Routing Engine.

A QFX5100 switch with a mastership priority of 0 will always stay in the linecard role.

2. Choose the QFX5100 device that was master the last time the VCF booted.
3. Choose the QFX5100 device that has been included in the VCF configuration for the longest period of time.
4. Choose the QFX5100 device with the lowest MAC address.

QFX3500, QFX3600, and EX4300 devices never assume the master or backup Routing Engine role in a VCF.

We strongly recommend that you configure the mastership priority of the QFX5100 devices in your VCF to ensure that the correct devices assume their intended roles when you configure your VCF using a nonprovisioned configuration.

## Virtual Chassis Ports (VCPs)

Virtual Chassis ports (VCPs) are used in a VCF to interconnect leaf devices to spine devices. All control and data traffic in a VCF is transported over VCPs.



VCPs in a VCF are either SFP+ connections that support 10-Gbps or QSFP+ connections that support 40-Gbps.

10-Gbps SFP+ and 40-Gbps QSFP+ links are automatically converted into VCPs in most scenarios when a device is added to an autoprovisioned or preprovisioned VCF. Automatic VCP conversion is discussed in more detail in the following section.

You can manually configure a 10-Gbps SFP+ and 40-Gbps QSFP+ link into a VCP.

Channelized interfaces cannot be configured into VCPs.

## Automatic Virtual Chassis Port (VCP) Conversion

10-Gbps SFP+ and 40-Gbps QSFP+ links are not configured into VCPs, by default.

10-Gbps SFP+ and 40-Gbps QSFP+ links are automatically converted into VCPs when:

- Link Layer Discovery Protocol (LLDP) is enabled on the interfaces on both ends of the link. LLDP is enabled by default.
- the device being added to the VCF is configured into fabric mode.
- The interfaces on both ends of the link are not configured as VCPs. The following interfaces are configured as VCPs:
  - The 40-Gbps QSFP+ port on an EX4300 switch, by default.
  - Any interface in the VCF that has been a VCP. If a device is removed from a VCF, the interface that was interconnected to the removed device remains configured as a VCP until it is configured into a network port using the **request virtual-chassis vc-port delete** command.
  - Any interface that has been configured into a VCP using the **request virtual-chassis vc-port set** command.

To change any of the above interfaces into a network interface so that the interface can become eligible for automatic VCP conversion, use the **request virtual-chassis vc-port delete** command.

- one of the devices is already part of a VCF that was autoprovisioned or preprovisioned.

Automatic VCP conversion does not work in nonprovisioned VCFs.

Automatic VCP conversion does not convert a VCP interface into a network interface when a device is removed from a VCF. If automatic VCP conversion has converted an interface into a VCP and you want the interface to function as a network interface, you must manually disable the VCP interface.

## VCF Configuration Options

You can configure a VCF using autoprovisioned, preprovisioned, or nonprovisioned configuration.

Autoprovisioned configuration allows you to *plug and play* leaf devices into a VCF after completing a minimal initial configuration procedure.

In a preprovisioned configuration, you deterministically control the devices in your VCF by associating each device's serial number to a member ID and role.

Nonprovisioned configuration is possible, but not recommended for most VCF installations. Nonprovisioned configuration is a highly manual procedure that should only be performed by expert users.

See [“Understanding Virtual Chassis Fabric Configuration” on page 13](#) for additional information on the VCF configuration options.

## Fabric Mode

A device must be configured into fabric mode in order for it to join a VCF. You should always configure a device into fabric mode before interconnecting it into a VCF.

In preprovisioned and nonprovisioned configurations, a device is not participating as a VCF member until it is configured into fabric mode.

In autoprovisioned configurations, a spine device is not participating as a VCF member until it is configured into fabric mode. A spine device that is not configured into fabric mode is configured into fabric mode when it is interconnected into the VCF. The final step of the process of configuring the device into fabric mode is a device reboot. We strongly recommend configuring the spine device into fabric mode before interconnecting it into the VCF to eliminate this reboot.

A leaf device in an autoprovisioned configuration is also rebooted to complete the fabric mode configuration when it is interconnected into a VCF without being set into fabric mode. You can avoid the downtime that accompanies the reboot by setting the device into fabric mode before interconnecting it into the VCF.

A standalone device that is not part of a VCF should never be configured into fabric mode. A device is not in fabric mode, by default.

## Mixed Mode

The optimal method of configuring a VCF is to use QFX5100 devices only. A VCF composed entirely of QFX5100 devices supports the largest breadth of features at the highest scalability while also supporting the highest number of high-speed interfaces.

You can, however, configure other devices as leaf devices in your VCF. QFX5100, QFX3600, QFX3500, or EX4300 devices can be used as leaf devices in a VCF.

If you use QFX3600, QFX3500, or EX4300 devices as leaf devices in your VCF, you must configure all devices in your VCF into mixed mode.

A device that is not part of a Virtual Chassis or a VCF with other devices should never be configured into mixed mode. A device is not configured into mixed mode, by default.

## Virtual Management Ethernet Interface

VCF configuration can be managed remotely using a global management interface called the virtual management Ethernet (VME) interface. The VME interface is a logical interface representing all of the out-of-band management ports on the member devices. When

you connect to the VCF using the VME interface's IP address, the connection is always redirected to the device acting in the master Routing Engine role.

A VME interface should always be used to configure a VCF. The VME interface is not tied to a device, so it can always be used to log in to the VCF even after the master Routing Engine changes.

We strongly recommend cabling the management port on all devices acting as Routing Engines to the network to ensure that you always have a direct connection to the master Routing Engine through the VME interface, regardless of which device assumes the master Routing Engine role. The management ports on leaf devices can also be used by the VME interface to access the VCF, so you can also cable leaf device management ports to the network, if desired.

## Virtual Chassis Fabric Port Link Aggregation Group Bundles

You can increase the bandwidth on links configured as VCPs within a VCF between two devices by configuring multiple same-speed links between two devices into VCPs. If, for instance, you configure two 40-Gbps QSFP+ links that are connecting the same devices in a VCF into VCPs, the two VCP links form one LAG bundle with two member links and 80-Gbps of total available bandwidth.

A VCP LAG bundle provides more bandwidth than a single VCP link can provide. A VCP LAG bundle also improves performance by load-sharing traffic across links within the bundle, and provides redundancy because traffic can be forwarded across another member link in the VCP LAG bundle when one VCP link fails.

VCP LAG bundling occurs automatically when same-speed VCP links are configured between two devices. No user configuration is required. VCP LAG bundling works only on same-speed VCP links; 10-Gbps and 40-Gbps links cannot be in the same VCP LAG bundle.

## Virtual Chassis Fabric License Requirements

A feature license is required to configure a VCF. The VCF feature license is an independent feature license; the enhanced feature licenses (EFLs) or advanced feature licenses (AFLs) that must be purchased to enable some features on some Juniper switches cannot be purchased to enable VCF.

For a VCF deployment, two license keys are recommended for redundancy—one for the device in the master Routing Engine role and the other for the device in the backup Routing Engine role.

Feature licenses are also required to configure advanced features on a Virtual Chassis Fabric. For a Virtual Chassis Fabric deployment, two license keys are recommended for redundancy—one for the device in the master Routing Engine role and the other for the device in the backup Routing Engine role. See *Software Features That Require Licenses on the QFX Series*.

To purchase feature licenses for VCF, contact your Juniper Networks sales representative (<http://www.juniper.net/us/en/contact-us/sales-offices>). The Juniper sales representative will provide you with the feature license files and license keys. You will be asked to supply

the chassis serial number of your switch; you can obtain the serial number by running the **show virtual-chassis** command.

## Hardware Requirements for a Virtual Chassis Fabric

A VCF can contain up to four devices configured as spines and up to twenty total devices.

All spine devices must be QFX5100 devices. We recommend optimizing the performance of your VCF by also configuring QFX5100 devices as your leaf devices. A non-mixed VCF has the highest port density and feature support for a VCF in addition to supporting more spine devices. Nevertheless, you can configure any combination of QFX5100, QFX3600, QFX3500, or EX4300 devices into leaf devices within your VCF.



**BEST PRACTICE:** We recommend using the following QFX5100 switches as spine devices:

- QFX5100-24Q switches, in deployments where devices are connecting to the VCF using the 10-Gbps Ethernet interfaces on the leaf devices, or using a mix of 10-Gbps and 1-Gbps Ethernet interfaces on the leaf devices.
- QFX5100-96S or QFX5100-48S, in deployments where devices are connecting to the VCF using 1-Gbps Ethernet interfaces only on the leaf devices.

QFX5100-48T switches are not supported as spine devices.

## Software Requirements in a Virtual Chassis Fabric

All devices in a VCF must be running the same version of Junos OS software that supports VCF.

The devices in the VCF must be using the version of software for standalone switches.

The flex software bundle is supported on non-mixed VCFs using QFX5100 member switches only. You cannot use the flex software bundle in mixed VCFs. The flex software bundle is the software that includes “jinstall-qfx-5-flex” text in the filename when it is downloaded from the Software Center.

We recommend configuring a device to the Junos OS release running on the VCF before interconnecting it into the VCF. For additional information on VCF software upgrades, see [“Understanding Software Upgrades in a Virtual Chassis Fabric” on page 22](#).

For information on software upgrade options for an operational VCF, see [“Understanding Software Upgrades in a Virtual Chassis Fabric” on page 22](#).

### Related Documentation

- [Network Configuration Example: MetaFabric™ Architecture 1.1: Configuring Virtual Chassis Fabric and Network Director 1.6](#)
- [Autoprovisioning a Virtual Chassis Fabric on page 23](#)
- [Preprovisioning a Virtual Chassis Fabric on page 27](#)
- [Understanding Virtual Chassis Fabric Configuration on page 13](#)

- [Understanding Software Upgrades in a Virtual Chassis Fabric on page 22](#)
- [Virtual Chassis Fabric Overview on page 3](#)

## Understanding Virtual Chassis Fabric Configuration

---

This topic describes the configuration options available for your Virtual Chassis Fabric (VCF).

This topic covers:

- [Virtual Chassis Fabric Setup on page 13](#)
- [Configuration File Management in a VCF on page 14](#)
- [Logging into a Virtual Chassis Fabric on page 15](#)
- [Understanding Interface Numbering on page 15](#)

### Virtual Chassis Fabric Setup

You must setup your VCF using one of the following options:

- [Autoprovisioned Virtual Chassis Fabric Configuration on page 13](#)
- [Preprovisioned Virtual Chassis Fabric Configuration on page 14](#)
- [Nonprovisioned Virtual Chassis Fabric Configuration on page 14](#)

#### Autoprovisioned Virtual Chassis Fabric Configuration

---

Autoprovisioned configuration allows you to “plug and play” leaf devices into a VCF after minimal initial configuration.

The minimal configuration requirements for autoprovisioning a VCF include setting the configuration mode to autoprovisioned and explicitly identifying the spine devices in your VCF by serial number. After this minimal configuration is complete, all supported devices—supported devices are either devices that have been zeroized or devices in factory default mode that have never been configured into a Virtual Chassis or VCF—are automatically added to the VCF as leaf devices when they are cabled to spine devices using supported 10-Gbps SFP+ ports or 40-Gbps QSFP+ ports. The Virtual Chassis ports (VCPs) are created automatically. Other parameters such as fabric and mixed mode are automatically detected and set.

A spine device in an autoprovisioned configuration should be configured into fabric mode before being interconnected into a VCF. A spine device in an autoprovisioned VCF must also have the same mixed mode setting as other member devices in the VCF. You should configure your spine device into fabric mode and, if necessary, mixed mode before interconnecting it into the VCF.

A leaf device in an autoprovisioned configuration is rebooted to complete the fabric mode configuration when it is interconnected into a VCF without being set into fabric mode. The leaf device is also rebooted if the device needs to be configured into or out of mixed mode to participate in the VCF. You can avoid the downtime that accompanies the reboot

of the leaf device by setting the leaf device into fabric mode and into or out of mixed mode before interconnecting it into the VCF.

### Preprovisioned Virtual Chassis Fabric Configuration

---

In a preprovisioned configuration, you deterministically control the devices in your VCF by associating each device's serial number to a member ID and role.

The advantage of configuring a VCF using a preprovisioned configuration is that you can explicitly control which devices are added to your VCF, and in what roles. VCF configuration, notably, occurs automatically when two devices that have been configured into fabric mode (and mixed mode, if applicable) are interconnected by a supported 10-Gbps SFP+ port or a 40-Gbps QSFP+ port after the preprovisioned configuration is defined.

The disadvantage of using a preprovisioned configuration is that the configuration process is more manual than the autoprovisioned configuration process.

### Nonprovisioned Virtual Chassis Fabric Configuration

---



**CAUTION:** We discourage nonprovisioned VCF configuration. You can configure all aspects of a VCF using autoprovisioned or preprovisioned configuration. Nonprovisioned VCF configuration should only be used by VCF experts in specialized scenarios.

---

A nonprovisioned VCF is the default method for creating a VCF; it is the configuration mode used when a VCF has not been configured into autoprovisioned or preprovisioned mode.

In a nonprovisioned VCF, member roles are determined by a mastership election algorithm. The first value checked by the mastership election algorithm is the mastership priority value. The switches with the highest mastership priority values assume the master and backup Routing Engine roles in a VCF.

If two or more devices have the same mastership priority value and are candidates for the Routing Engine role, the mastership election algorithm uses other parameters to determine which device is elected as the Routing Engine. See [“Understanding Virtual Chassis Fabric Components” on page 5](#).

The default mastership priority value for all devices is 128. You should always configure two QFX5100 switches acting as spine devices with the highest mastership priority to ensure the Routing Engine role is assigned to a spine device..

In a nonprovisioned VCF, you must manually configure every VCF.

## Configuration File Management in a VCF

You configure a VCF by logging onto the master Routing Engine and making configuration changes. See the next section for information on logging into a VCF.

The configuration file that is modified when you are on the master Routing Engine is automatically shared with all other devices in the VCF when it is committed. Each device stores its own copy of the configuration file.

## Logging into a Virtual Chassis Fabric

The recommended method of logging into a VCF is through the use of a Virtual Management Ethernet (VME) interface. The VME interface is a logical interface representing all of the out-of-band management ports on the member devices. When you connect to the VCF configuration using the VME interface's IP address, the connection is always redirected to the management port on device in the master Routing Engine role. The VME interface is not tied to a device, so it can always be used to log in to the VCF even after the master Routing Engine changes. We recommend cabling the management ports—an *me* or *em* interface—on each Routing Engine in your VCF to support the VME interface.

If you log in to the console port of any member device in a VCF, your session is automatically redirected to the device acting in the master Routing Engine role.

## Understanding Interface Numbering

Interfaces in Junos OS are specified as follows:

*type-fpc/pic/port*

A VCF applies this convention as follows:

- *type*—The interface type.
- *fpc*—Flexible PIC Concentrator. In a VCF, the *fpc* is the member ID of the switch. For instance, the *fpc* of member 16 in the VCF is 16.
- *pic*—the number of the PIC (Physical Interface Card) on the member device.
- *port*—the port number.

For more detailed information on interface numbering, see *Understanding Interface Naming Conventions*.

### Related Documentation

- [Network Configuration Example: MetaFabric™ Architecture 1.1: Configuring Virtual Chassis Fabric and Network Director 1.6](#)
- [Autoprovisioning a Virtual Chassis Fabric on page 23](#)
- [Preprovisioning a Virtual Chassis Fabric on page 27](#)
- [Configuring a Nonprovisioned Virtual Chassis Fabric on page 31](#)

## Understanding Mixed EX Series and QFX Series Virtual Chassis or Virtual Chassis Fabric

This topic describes the requirements for a mixed Virtual Chassis or a mixed Virtual Chassis Fabric (VCF).

A mixed Virtual Chassis includes two or more types of EX Series switches, two or more types of QFX Series switches, or a mix of EX and QFX Series switches.



**NOTE:** EX2200, EX2300, EX3300, and EX3400 switches cannot be part of any mixed Virtual Chassis.

A mixed VCF is any VCF that includes two or more types of member switches. Because a VCF must use a QFX5100 switch as a spine device, a mixed VCF is any VCF that includes EX4300, QFX3500, or QFX3600 member switches in addition to the required QFX5100 switches.



**NOTE:** The optimal VCF topology is to use QFX5100 devices only. A VCF composed entirely of QFX5100 devices supports the largest breadth of features at the highest scalability while also supporting the highest number of high-speed interfaces.

This topic covers:

- [Virtual Chassis Fabric Summary on page 16](#)
- [Understanding Mixed Virtual Chassis Fabric on page 17](#)
- [Virtual Chassis Summary for QFX5100, QFX3600, QFX3500, EX4600, and EX4300 Switches on page 17](#)
- [Understanding the Routing Engine Role in a Mixed Virtual Chassis Using EX4300, EX4600, QFX3500, QFX3600, or QFX5100 Member Switches on page 18](#)
- [Understanding EX4300, QFX3500, QFX3600, and QFX5100 Switches in a Virtual Chassis on page 19](#)
- [Understanding Mixed EX4300 and EX4600 Virtual Chassis on page 19](#)
- [Understanding EX4200, EX4500, and EX4550 Switches in a Mixed Virtual Chassis on page 19](#)

## Virtual Chassis Fabric Summary

[Table 3 on page 17](#) provides a high-level overview of the permitted hardware allowed in the routing engine and line card roles of a mixed and a non-mixed VCF. The table also includes license requirements and supported configuration methods.



Table 3: Virtual Chassis Fabric Summary

Category	Allowed Routing Engines	Allowed Line Cards	License Requirement	Configuration Methods
Non-mixed	QFX5100	QFX5100	Yes (on two QFX5100 switches operating in master and backup Routing Engine roles)	Autoprovisioning Preprovisioning Nonprovisioning (not recommended)
Mixed	QFX5100	QFX5100 QFX3600 QFX3500 EX4300	Yes (on two QFX5100 switches operating in master and backup Routing Engine roles)	Autoprovisioning Preprovisioning Nonprovisioning (not recommended)

### Understanding Mixed Virtual Chassis Fabric

A VCF must use a QFX5100 switch in the spine role. A mixed VCF is, therefore, any VCF that includes EX4300, QFX3500, or QFX3600 member switches in addition to the required QFX5100 switch.

The optimal method of configuring a VCF is to use QFX5100 devices only. A non-mixed VCF composed entirely of QFX5100 devices supports the largest breadth of features at the highest scalability while also supporting the highest number of high-speed interfaces. You can, however, also configure a mixed VCF.

If you use QFX3600, QFX3500, or EX4300 devices as leaf devices in your VCF, you must configure all devices in your VCF into mixed mode. If you are turning a non-mixed VCF into a mixed VCF, you have to reboot the VCF to change the mixed mode setting.

### Virtual Chassis Summary for QFX5100, QFX3600, QFX3500, EX4600, and EX4300 Switches

Table 4 on page 18 provides a high-level overview of the permitted hardware allowed in the routing engine and line card roles of a mixed and a non-mixed Virtual Chassis for QFX5100, QFX3600, QFX3500, EX4600, and EX4300 switches. The table also includes license requirements and supported configuration methods.

Table 4: Virtual Chassis Summary

Category	Allowed Routing Engines	Allowed Line Cards	License Requirement	Configuration Methods
Non-mixed	QFX5100	QFX5100	No	Nonprovisioning Preprovisioning
	QFX3600 QFX3500	QFX3600 QFX3500	No	Nonprovisioning Preprovisioning
	EX4600	EX4600	No	Nonprovisioning Preprovisioning
	EX4300	EX4300	No	Nonprovisioning Preprovisioning
Mixed	QFX5100	QFX5100 QFX3600 QFX3500 EX4300	No	Nonprovisioning Preprovisioning
	QFX3600 QFX3500	QFX3600 QFX3500 EX4300	No	Nonprovisioning Preprovisioning
	EX4600	EX4600 EX4300	No	Nonprovisioning Preprovisioning

### Understanding the Routing Engine Role in a Mixed Virtual Chassis Using EX4300, EX4600, QFX3500, QFX3600, or QFX5100 Member Switches

In a mixed Virtual Chassis, the switch in the master Routing Engine role determines which switches are supported in the line card role of the mixed Virtual Chassis.

When a mixed Virtual Chassis is using a QFX5100 switch in the master Routing Engine role, you can use QFX5100, QFX3600, QFX3500, or EX4300 switches in the line card role.

When a mixed Virtual Chassis is using a QFX3600 or QFX3500 switch in the master Routing Engine role, you can use QFX3600, QFX3500, or EX4300 switches in the line card role.

In a mixed EX4300 and EX4600 Virtual Chassis, an EX4600 switch automatically assumes the Routing Engine role.

EX4600 switches can only be in a mixed Virtual Chassis with EX4300 switches. EX4600 switches cannot be in a mixed Virtual Chassis with QFX5100, QFX3600, or QFX3500 switches.

We recommend always configuring the same type of switch into the master and backup Routing Engine role, to ensure that the switch operating in the master role remains the same type of switch in the event of a switchover.

In most mixed Virtual Chassis, you must configure your Virtual Chassis to ensure a switch that supports the master Routing Engine assumes the master Routing Engine role. Without user configuration, any switch—with the exception of the EX4300 switch, which can never assume the master or backup Routing Engine role in a mixed Virtual Chassis or VCF—can assume the master or backup Routing Engine role.

### Understanding EX4300, QFX3500, QFX3600, and QFX5100 Switches in a Virtual Chassis

Up to ten EX4300 switches, QFX3500 switches, QFX3600 switches, and QFX5100 switches can be interconnected using Virtual Chassis ports (VCPs) to form a mixed or non-mixed Virtual Chassis. The mixed Virtual Chassis supports up to ten member switches regardless of the switches that compose the mixed Virtual Chassis.

EX4300 switches can also be interconnected into a mixed Virtual Chassis with EX4600 switches. See the following section for information on mixed EX4300 and EX4600 Virtual Chassis.

### Understanding Mixed EX4300 and EX4600 Virtual Chassis

EX4300 switches and EX4600 switches can be interconnected into the same Virtual Chassis. An EX4600 switch automatically assumes the master Routing Engine role in a mixed EX4300 and EX4600 Virtual Chassis, since EX4300 switches cannot assume the Routing Engine role in a mixed Virtual Chassis. EX4600 switches cannot be in a mixed Virtual Chassis with any other type of switch.

The mixed Virtual Chassis supports up to ten member switches.

### Understanding EX4200, EX4500, and EX4550 Switches in a Mixed Virtual Chassis

EX4200 switches, EX4500 switches, and EX4550 switches can be interconnected into the same Virtual Chassis to form a mixed EX4200 and EX4500 Virtual Chassis, mixed EX4200 and EX4550 Virtual Chassis, mixed EX4500 and EX4550 Virtual Chassis, or mixed EX4200, EX4500, and EX4550 Virtual Chassis. The mixed Virtual Chassis supports up to 10 member switches regardless of whether the switches are EX4200 switches, EX4500 switches, or EX4550 switches. Any model of EX4200, EX4500, or EX4550 switch can be interconnected into the same mixed Virtual Chassis. The master election process that decides member switch roles in a mixed Virtual Chassis is identical to the master election process in a non-mixed Virtual Chassis, so any member switch in a mixed Virtual Chassis can assume the master, backup, or linecard role.

EX4200 switches, EX4500 switches, and EX4550 switches cannot be interconnected into a Virtual Chassis with any other switches.

#### Related Documentation

- [Virtual Chassis Fabric Overview on page 3](#)
- [Understanding QFX Series Virtual Chassis](#)
- [EX Series Virtual Chassis Overview](#)
- [Understanding Virtual Chassis Fabric Components on page 5](#)

- [Understanding QFX Series Virtual Chassis Components](#)
- [Understanding EX Series Virtual Chassis Components](#)

## Understanding Traffic Flow Through a Virtual Chassis Fabric

---

This topic describes the ways traffic is managed within the members of a Virtual Chassis Fabric (VCF).

It contains the following sections:

- [Smart Trunking Algorithm for Unicast Traffic Forwarding on page 20](#)
- [Multicast Distribution Trees for Broadcast, Unknown Unicast, and Multicast Traffic on page 20](#)
- [Adaptive Load Balancing on page 21](#)

### Smart Trunking Algorithm for Unicast Traffic Forwarding

A Virtual Chassis Fabric (VCF) forwards unicast traffic using a smart trunking algorithm that sends all traffic across multiple paths based on end-to-end available bandwidth. The smart trunking algorithm avoids unnecessary congestion due to improper traffic allocation while optimizing fabric bandwidth utilization because traffic is forwarded through the VCF relative to available bandwidth.

The smart trunking algorithm works by considering the overall available path bandwidth of each path in the VCF when making traffic-forwarding decisions, and then forwarding traffic across the paths relative to available path bandwidth. If a VCF with two spine devices, for instance, has one path from leaf device 1 to leaf device 4 that contains two 40-Gbps QSFP+ links and a second path from leaf device 1 to leaf device 4 that contains two 10-Gbps SFP+ links, the algorithm tries to balance traffic sent on the paths so that four times more packets are sent on the first path with 40 Gbps of available bandwidth across the entire path than are sent on the second path with 10 Gbps of total bandwidth.

You can optimize how traffic is forwarded through the VCF by adding spine devices to maximize the number of available paths between all leaf devices, and by using as many 40-Gbps QSFP+ interfaces as Virtual Chassis ports (VCPs) as possible.

### Multicast Distribution Trees for Broadcast, Unknown Unicast, and Multicast Traffic

A VCF creates bidirectional, shared multicast distribution trees (MDTs) to choose forwarding paths for broadcast, unknown unicast, and multicast (BUM) traffic between the members of the VCF. By default, one MDT is rooted at the source for each VCF member—the VCF creates the same number of MDTs as members in the VCF, and each MDT has one of the members as its root node. The VCF topology, application of load balancing, and VCF member availability can influence how traffic is forwarded along these paths.

If you are familiar with traffic patterns and load conditions in your VCF, and want more control over how VCF MDTs are created, you can use the [fabric-tree-root](#) configuration statement to preempt the default behavior and instead have the VCF form MDTs only

with specific members as root nodes, called *fabric tree roots*. If at least one device in the VCF is available that was configured as a fabric tree root, instead of the default behavior, the VCF will form MDTs with configured fabric tree roots only. The VCF will revert to the default behavior if there are no available VCF members configured as fabric tree roots.



**NOTE:** The *fabric-tree-root* option can be used in autoprovisioned or preprovisioned VCFs only.

If you use this option to configure specific members to be fabric tree roots, we recommend that you configure *all* spine members and *only* the spine members in the VCF as fabric tree roots, for the following reasons:

- Configuring multiple spine devices as MDT root nodes prevents member switches from inadvertently returning to the default behavior (where all members become MDT root nodes) if a spine node becomes unavailable.
- In a VCF with many leaf nodes, the default MDT algorithm results in many MDTs being used when balancing traffic within the VCF. When a leaf node goes offline or is reset, the MDT with that root leaf node is no longer available, triggering interruptions in VCF traffic flow to rebalance the load based on the remaining MDTs. When the VCF is configured with only spine devices as MDT root nodes, if a leaf node becomes unavailable, the VCF continues using the same spine root MDTs without traffic disruption.

## Adaptive Load Balancing

A VCF supports adaptive load balancing (ALB), which enables the VCF smart trunking and multicast distribution algorithms to use dynamic load information on interfaces and traffic queues to make forwarding decisions within the VCF. When ALB is implemented using flowlets, traffic flows that enter the VCF are spliced into smaller flows—flowlets—and individually forwarded across the VCF to the same destination device over different paths when the inactivity time between packet bursts on the sending interface exceeds the user-configurable inactivity interval. When ALB is implemented using per-packet mode, the sending interface actively monitors all paths available between two member devices and forwards traffic through the VCF using the best available path at the moment.

Implementing ALB using flowlets is effective in environments that periodically experience extremely large traffic flows—*elephant flows*—that are substantially larger than the majority of other traffic flowing through the VCF. The VCF is better able to manage the elephant flows by splicing them into smaller flowlets using ALB.

ALB is supported on a non-mixed VCF composed entirely of QFX5100 switches only. You should enable ALB using flowlets in non-mixed VCFs in environments where a small number of traffic flows are disproportionately larger than the majority of the other traffic flows.

### Related Documentation

- [Understanding Virtual Chassis Fabric Components on page 5](#)

## Understanding Software Upgrades in a Virtual Chassis Fabric

---

This topic provides an overview of software upgrades on Virtual Chassis Fabric (VCF).

It contains the following sections:

- [Virtual Chassis Fabric Software Basics on page 22](#)
- [Nonstop Software Upgrade \(NSSU\) on page 22](#)
- [Automatic Software Update on page 22](#)
- [Traditional Software Upgrade on page 23](#)

### Virtual Chassis Fabric Software Basics

VCF is initially supported in Junos OS Release 13.2X51-D20. All devices in a VCF must be running the same version of Junos OS that supports VCF.

At initial VCF configuration, you should configure all devices to the same Junos OS release before interconnecting them into a VCF.

When you are adding a device to an existing VCF, you should update the Junos OS release on the new device to the Junos OS release running in the VCF before interconnecting it into the VCF. Updating the Junos OS on the device before interconnecting it helps ensure the device is gracefully added to the VCF, without the downtime that is required to reboot the device after an automatic software update or the troubleshooting that is required if the device isn't added to the VCF due to mismatched software releases.

Before you interconnect a device into a VCF, you should upgrade the software on the device being added to the VCF to the version of Junos OS running on the VCF.

### Nonstop Software Upgrade (NSSU)

Nonstop software upgrade (NSSU) enables you to upgrade the software running on all member devices in a VCF with minimal network traffic disruption during the upgrade.

NSSU upgrades the software on each device individually while all other devices continue normal operations.

For additional information on NSSU in a VCF, see *Understanding Nonstop Software Upgrade on a Virtual Chassis Fabric*.

### Automatic Software Update

Automatic software update automatically upgrades the Junos OS running on a device joining a VCF to the version of Junos OS running on the VCF at the moment the new device is cabled into the VCF.

Automatic software update is enabled using the **set virtual-chassis auto-sw-update** statement.

## Traditional Software Upgrade

You can upgrade software on a VCF using the traditional method of upgrading software for Junos OS by logging onto the master Routing Engine and using the **request system software add** command to initiate the upgrade on a non-mixed VCF or the **request system software add set [package-name package-name ...]** to initiate the upgrade on a mixed VCF, where *package-name* is the path to an image for one device family.

When you upgrade Junos OS on a VCF using the traditional software upgrade, the entire system is down until the upgrade is complete.

- Related Documentation**
- [Upgrading Software on a Virtual Chassis Fabric Using Nonstop Software Upgrade](#)
  - [Upgrading Software for a Virtual Chassis Fabric on page 43](#)
  - [Adding a Device to a Virtual Chassis Fabric on page 34](#)

## Autoprovisioning a Virtual Chassis Fabric

Autoprovisioning a Virtual Chassis Fabric (VCF) enables you to “plug and play” devices into your VCF after minimal initial configuration.

Update all devices to the same version of Junos OS that supports VCF. See *Upgrading Software or Installing Software on an EX Series Switch with a Single Routing Engine (CLI Procedure)*.

To configure a VCF using autoprovisioning:

1. Make a list of the serial numbers of all the spine devices in the VCF. The spine devices must be QFX5100 devices. You can configure up to four spine devices in a VCF. You can get the device's serial number in the **show virtual-chassis** output or by following the instructions in *Locating the Serial Number on a QFX5100 Device or Component*.
2. Configure each device into fabric mode. If needed, configure the devices into mixed mode.

Configure the device to reboot as part of the procedure to complete this configuration step.

Configure mixed mode if your VCF includes QFX3600, QFX3500, or EX4300 devices as leaf devices.

If you are configuring a VCF composed entirely of QFX5100 devices:

```
user@device> request virtual-chassis mode fabric local reboot
```

If you are configuring a VCF composed of two or more types of devices:

```
user@device> request virtual-chassis mode fabric mixed local reboot
```



**NOTE:** A spine device whose fabric or mixed mode setting is improperly set cannot join a VCF. You can check the mode settings by using the `show virtual-chassis mode` command.

We recommend that you set the fabric and mixed mode settings before you interconnect your spine devices into the VCF to avoid the following issues:

- Incurring downtime as the devices reboot to commit the mixed mode or fabric settings.
- Manually correcting potential issues related to VCF formation because the device did not immediately join the VCF.

You can, however, use the `request virtual-chassis mode fabric local` or `request virtual-chassis mode mixed local` commands to set a spine device into fabric or mixed mode after interconnecting your VCF.

The fabric and mixed mode settings are automatically updated for a leaf device when it is interconnected into an autoprovisioned VCF. If the fabric or mixed mode settings are changed when a leaf device is interconnected into a VCF, the leaf device reboots before joining the VCF.

3. When the reboot is complete, log in to one of the spine devices in your VCF.
4. Set the configuration mode to autoprovisioned:

```
[edit]
user@device# set virtual-chassis auto-provisioned
```

5. Configure two spine devices into the Routing Engine role:

```
[edit virtual-chassis]
user@device# set member member-id serial-number serial-number role routing-engine
```



**NOTE:** If your VCF has more than two spine devices, the remaining spine devices are placed in the line card role later in this procedure.

For instance, to configure the two spine devices with the serial numbers “SERIALNUMB00” and “SERIALNUMB01” into the Routing Engine role as members 0 and 1:

```
[edit virtual-chassis]
user@device# set member 0 serial-number SERIALNUMB00 role routing-engine
user@device# set member 1 serial-number SERIALNUMB01 role routing-engine
```

6. (Recommended) Configure a virtual management Ethernet (VME) interface for management of the VCF configuration:

```
[edit]
user@device# set interfaces vme unit 0 family inet address /ip-address/mask/
```





**NOTE:** A VME accesses the device in the master Routing Engine role using a management port, so cable management port em0 or em1 on each spine device in your VCF so the VME is available regardless of which spine device assumes the master Routing Engine role. See *Connecting a QFX Series Device to a Management Console*

7. Commit the configuration:

```
user@device# commit
```

8. Cable your VCF.

After your autoprovisioned VCF configuration is committed, you can cable any EX4300, QFX3500, QFX3600, or QFX5100 device that is zeroized or that has never been configured to a spine device using a supported SFP+ or QSFP+ interface. The device that is zeroized or in factory-default mode is added to the VCF as a leaf device. All VCPs are configured as part of this process.



**NOTE:** Mixed mode and fabric mode are checked and, if needed, set automatically on the device as part of this process. If the mixed or fabric mode has to be changed to become part of the VCF, the device reboots. The device participates in the VCF with no further user intervention after this reboot is complete.



**NOTE:** Automatic VCP conversion only works when the interfaces on both ends of the link are not configured into VCPs.

The 40-Gbps QSFP+ interfaces on EX4300 switches are configured as VCPs, by default. You must, therefore, delete the VCP on the 40-Gbps QSFP+ interface using the `request virtual-chassis vc-port delete` command before interconnecting it into the VCF in order for the link to be converted into a VCP. You can also manually configure the link into a VCP using the `request virtual-chassis vc-port set` command.

The device joins the VCF immediately without a reboot if the mixed or fabric mode setting does not need to be changed.

9. Install the VCF feature licenses.

For a VCF deployment, two license keys are recommended for redundancy—one for the device in the master role and the other for the device in the backup role.

To purchase a feature license for VCF, contact your Juniper Networks sales representative (<http://www.juniper.net/us/en/contact-us/sales-offices>). The Juniper sales representative will provide you with the feature license files and license keys. You will be asked to supply the chassis serial number of your switch; you can obtain the serial number by running the `show virtual-chassis` command.

After obtaining the licenses, follow the instructions in *Generating License Keys*.

10. (Optional) The VCF forwards broadcast, unknown unicast, and multicast (BUM) traffic among the members of the VCF using multicast distribution trees (MDTs). By default, the VCF creates MDTs for every member of the VCF with that member as the root node of an MDT. If this default MDT creation method is not optimal for your installation, you can control which members become MDT root nodes.

The **set virtual-chassis member *member-id* fabric-tree-root** configuration statement preempts the default method of creating MDTs, and specifies whether or not a member in a VCF can be an MDT root node. If this statement is configured for one or more members, MDTs are created only with the specified members as root nodes. See [“Understanding Traffic Flow Through a Virtual Chassis Fabric” on page 20](#) and **fabric-tree-root** for more details on why you might want to choose this MDT creation method instead of the default method. Note that if you decide to use this option, we recommend that you specify all the spine members (and only spine members) as MDT root nodes. In an autoprovisioned VCF, this option should be configured for all spine devices (independent of the member’s role) after the VCF is running and any additional spine device member IDs have been automatically assigned.

If desired, configure the spine devices in the VCF to be fabric MDT root nodes. For example, if you have four spine members in your VCF, where you configured the first two spine devices to be members 0 and 1, and during autoprovisioning, the two additional spine members were automatically assigned to be members 4 and 5:

```
[edit virtual-chassis]
user@device# set member 0 fabric-tree-root
user@device# set member 1 fabric-tree-root
user@device# set member 4 fabric-tree-root
user@device# set member 5 fabric-tree-root
```



**NOTE:** This option can also be configured anytime later during VCF operation if you observe internal VCF multicast traffic flow issues with default MDTs.

#### Related Documentation

- [Understanding Virtual Chassis Fabric Configuration on page 13](#)
- [Adding a Device to a Virtual Chassis Fabric on page 34](#)
- [Removing a Device From a Virtual Chassis Fabric on page 42](#)
- [Understanding Virtual Chassis Fabric Components on page 5](#)
- [Preprovisioning a Virtual Chassis Fabric on page 27](#)
- [Understanding Traffic Flow Through a Virtual Chassis Fabric on page 20](#)

## Preprovisioning a Virtual Chassis Fabric

Preprovisioning a Virtual Chassis Fabric (VCF) configuration allows you to assign the member ID and role for each device in the VCF.

Before you begin:

- Update all devices to the same version of Junos OS that supports VCF. See *Upgrading Software* or *Installing Software on an EX Series Switch with a Single Routing Engine (CLI Procedure)*.
- If automatic software update is enabled, disable automatic software update by entering the **delete chassis auto-image-upgrade** statement.

To preprovision a VCF:

1. Make a list of the serial numbers of all the devices to be connected in the VCF. You can get a device's serial number in the **show virtual-chassis** output or by following the instructions in *Locating the Serial Number on a QFX5100 Device or Component*, *Locating the Serial Number on a QFX3600 or QFX3600-I Device or Component*, *Locating the Serial Number on a QFX3500 Device or Component*, or *Locating the Serial Number on an EX4300 Switch or Component*.

2. Decide the desired role (**routing-engine** or **line-card**) for each device.

In a VCF, you configure two QFX5100 devices acting in the Routing Engine role into spine devices. All other devices—the spine devices not assuming the master or backup Routing Engine role and all leaf devices—are configured into the linecard role as leaf devices.

3. Configure each individual device into fabric mode. If needed, configure the devices into mixed mode.

Reboot each device to complete this configuration step.

Mixed mode must be configured if your VCF includes QFX3500, QFX3600, or EX4300 devices as leaf nodes.

If you are configuring a VCF composed entirely of QFX5100 devices:

```
user@device> request virtual-chassis mode fabric local reboot
```

If you are configuring a VCF that includes EX4300, QFX3500, or QFX3600 devices as leaf nodes:

```
user@device> request virtual-chassis mode fabric mixed local reboot
```



**NOTE:** A device whose fabric or mixed mode setting is improperly set cannot join a VCF. You can check the mode settings using the `show virtual-chassis mode` command.

We recommend that you set the fabric and mixed mode before you interconnect your devices into a VCF to avoid the following issues:

- Incurring downtime as the devices reboot to commit the mixed mode or fabric settings.
- Manually correcting potential issues related to VCF formation because the device did not immediately join the VCF.

You can, however, use the `request virtual-chassis mode fabric local` or `request virtual-chassis mode mixed local` commands to set a device into fabric or mixed mode after interconnecting your VCF.

4. Log in to one of your spine devices that will be configured into the Routing Engine role after the reboot has completed.

5. Specify the preprovisioned configuration mode:

```
[edit virtual-chassis]
user@device# set preprovisioned
```

6. Associate a member ID with a serial number for each device in your VCF, and configure the role for each device:

```
[edit virtual-chassis]
user@device# set member member-id serial-number serial-number role (line-card |
routing-engine)
```

Configure two spine devices into the Routing Engine role, and the additional spine devices into the linecard role if your VCF supports three or more spine devices. You must use QFX5100 devices as your spine devices.

Configure your leaf devices into the linecard role.

For instance, if you wanted to preprovision a VCF with twenty member devices that uses member 0 and 1 in the Routing Engine role, members 2 and 3 as spine devices in the line card role, and the remaining devices as leaf devices:

```
[edit virtual-chassis]
user@device# set member 0 serial-number SERIALNUMB00 role routing-engine
user@device# set member 1 serial-number SERIALNUMB01 role routing-engine
user@device# set member 2 serial-number SERIALNUMB02 role line-card
user@device# set member 3 serial-number SERIALNUMB03 role line-card
user@device# set member 4 serial-number SERIALNUMB04 role line-card
user@device# set member 5 serial-number SERIALNUMB05 role line-card
user@device# set member 6 serial-number SERIALNUMB06 role line-card
user@device# set member 7 serial-number SERIALNUMB07 role line-card
user@device# set member 8 serial-number SERIALNUMB08 role line-card
user@device# set member 9 serial-number SERIALNUMB09 role line-card
user@device# set member 10 serial-number SERIALNUMB10 role line-card
user@device# set member 11 serial-number SERIALNUMB11 role line-card
user@device# set member 12 serial-number SERIALNUMB12 role line-card
user@device# set member 13 serial-number SERIALNUMB13 role line-card
```

```

user@device# set member 14 serial-number SERIALNUMB14 role line-card
user@device# set member 15 serial-number SERIALNUMB15 role line-card
user@device# set member 16 serial-number SERIALNUMB16 role line-card
user@device# set member 17 serial-number SERIALNUMB17 role line-card
user@device# set member 18 serial-number SERIALNUMB18 role line-card
user@device# set member 19 serial-number SERIALNUMB19 role line-card

```

7. (Recommended) Configure a virtual management Ethernet (VME) interface for management of the VCF configuration:

```

[edit]
user@device# set interfaces vme unit 0 family inet address /ip-address/mask/

```



**NOTE:** A VME accesses the device in the master Routing Engine role using a management port, so cable management port em0 or em1 on each spine device in your VCF so the VME is available regardless of which spine device assumes the master Routing Engine role. See *Connecting a QFX Series Device to a Management Console*

8. (Optional) The VCF forwards broadcast, unknown unicast, and multicast (BUM) traffic among the members of the VCF using multicast distribution trees (MDTs). By default, the VCF creates MDTs for every member of the VCF with that member as the root node of an MDT. If this default MDT creation method is not optimal for your installation, you can control which members become MDT root nodes.

The **set virtual-chassis member *member-id* fabric-tree-root** configuration statement preempts the default method of creating MDTs, and specifies whether a member in a VCF can be an MDT root node. If this statement is configured for one or more members, MDTs are created only with the specified members as root nodes. See [“Understanding Traffic Flow Through a Virtual Chassis Fabric” on page 20](#) and **fabric-tree-root** for more on why you might want to choose this MDT creation method instead of the default method. Note that if you decide to use this option, we recommend that you specify all the spine members (and only spine members) as MDT root nodes.

If desired, configure the spine devices in the VCF to be fabric MDT root nodes. For example, if you are preprovisioning a VCF with members 0 through 3 as spine devices (independent of the role):

```

[edit virtual-chassis]
user@device# set member 0 fabric-tree-root
user@device# set member 1 fabric-tree-root
user@device# set member 2 fabric-tree-root
user@device# set member 3 fabric-tree-root

```



**NOTE:** This option can also be configured anytime later during VCF operation if you observe internal VCF multicast traffic flow issues with default MDTs.

9. Commit the configuration:

```

user@device# commit

```

10. (EX4300 switches using 40-Gbps QSFP+ interfaces as VCPs only) Perform one of the following tasks to convert the 40-Gbps QSFP+ interfaces into VCPs:



**NOTE:** Automatic VCP conversion only works when the interfaces on both ends of a link are not configured into VCPs.

This step is needed on EX4300 switches using 40-Gbps QSFP+ interfaces as VCPs because the 40-Gbps QSFP+ interfaces on EX4300 switches are configured as VCPs, by default.

If you are cabling the EX4300 switch to the spine switch with a DAC cable in the 40-Gbps QSFP+ interface, you must use the `request virtual-chassis vc-port set` command to manually convert the 40-Gbps QSFP+ interface on the spine device into a VCP, as described in the second bullet below.

- Delete the VCP configuration on the 40-Gbps QSFP+ interface using the `request virtual-chassis vc-port delete` command.

For instance, to delete the VCP configuration on all four QSFP+ interfaces on an EX4300-48T switch:

```
user@leaf-device# request virtual-chassis vc-port delete pic-slot 1 port 0
user@leaf-device# request virtual-chassis vc-port delete pic-slot 1 port 1
user@leaf-device# request virtual-chassis vc-port delete pic-slot 1 port 2
user@leaf-device# request virtual-chassis vc-port delete pic-slot 1 port 3
```

- Manually configure the 40-Gbps QSFP+ interface on the spine device into a VCP using the `request virtual-chassis vc-port set`. For instance:

```
user@spine-device# request virtual-chassis vc-port set pic-slot 1 port 0
user@spine-device# request virtual-chassis vc-port set pic-slot 1 port 1
user@spine-device# request virtual-chassis vc-port set pic-slot 1 port 2
user@spine-device# request virtual-chassis vc-port set pic-slot 1 port 3
```

11. Interconnect the spine device that you configured in the previous steps to all leaf devices by using the supported SFP+ and QSFP+ interfaces.



**NOTE:** The automatic Virtual Chassis port (VCP) conversion feature is enabled and automatically configures SFP+ and QSFP+ interfaces into VCPs when the VCF configuration mode is set to `preprovisioned`. You do not need to manually configure VCPs.

If you want to configure an SFP+ or QSFP+ interface into a network interface, disable LLDP on that interface. See *Configuring LLDP*.

12. Interconnect all other spine devices to all other leaf devices using the supported SFP+ and QSFP+ interfaces.
13. Install the VCF feature licenses.

For a VCF deployment, two license keys are recommended for redundancy—one for the device in the master role and the other for the device in the backup role.

To purchase a feature license for VCF, contact your Juniper Networks sales representative (<http://www.juniper.net/us/en/contact-us/sales-offices>). The Juniper sales representative will provide you with the feature license files and license keys. You will be asked to supply the chassis serial number of your switch; you can obtain the serial number by running the **show virtual-chassis** command.

After obtaining the licenses, follow the instructions in *Generating License Keys*.

#### Related Documentation

- [Understanding Virtual Chassis Fabric Configuration on page 13](#)
- [Adding a Device to a Virtual Chassis Fabric on page 34](#)
- [Removing a Device From a Virtual Chassis Fabric on page 42](#)
- [Understanding Virtual Chassis Fabric Components on page 5](#)
- [Autoprovisioning a Virtual Chassis Fabric on page 23](#)
- [Understanding Traffic Flow Through a Virtual Chassis Fabric on page 20](#)

## Configuring a Nonprovisioned Virtual Chassis Fabric



**CAUTION:** Configure your Virtual Chassis Fabric (VCF) using autoprovisioning or preprovisioning unless you have a compelling reason to use nonprovisioned configuration. You can configure all aspects of a VCF using autoprovisioned or preprovisioned configuration. The process for autoprovisioning your VCF is described in “[Autoprovisioning a Virtual Chassis Fabric](#)” on page 23 and the process for preprovisioning your VCF is described in “[Preprovisioning a Virtual Chassis Fabric](#)” on page 27.

Nonprovisioned VCF configuration is highly discouraged. Nonprovisioned VCF configuration should only be used by VCF experts in specialized scenarios.

A nonprovisioned VCF is the configuration mode used when a VCF has not been configured into autoprovisioned or preprovisioned mode.

In a nonprovisioned VCF, you configure the device roles by setting the mastership priority value of each device. If no mastership priority values are set, a master election algorithm process runs and selects the role for each device.

You must manually configure all Virtual Chassis ports (VCPs) in a nonprovisioned VCF. The automatic VCP conversion feature, which automatically configures supported 10-Gbps SFP+ links and 40-Gbps QSFP+ links into VCPs on autoprovisioned and preprovisioned VCFs, is not supported on nonprovisioned VCFs.

Update all devices to the same version of Junos OS that supports VCF. See *Upgrading Software or Installing Software on an EX Series Switch with a Single Routing Engine (CLI Procedure)*.

To configure a nonprovisioned VCF:

1. Power on the devices.
2. Configure each individual device into fabric mode. If needed, configure the devices into mixed mode.

Reboot each device to complete this configuration step.

A VCF must have QFX5100 devices in the spine role, and operates most efficiently when the leaf nodes are also QFX5100 devices. Mixed mode must be configured if your VCF also includes at least one QFX3600, QFX3500, or EX4300 device in the leaf role.

If you are configuring a VCF composed entirely of QFX5100 devices:

```
user@device> request virtual-chassis mode fabric local reboot
```

If you are configuring a VCF using at least one QFX3600, QFX3500, or EX4300 device as a leaf device:

```
user@device> request virtual-chassis mode fabric mixed local reboot
```



**NOTE:** A device whose fabric or mixed mode setting is improperly set cannot join a VCF. You can check the mode settings using the `show virtual-chassis mode` command.

We recommend setting the fabric and mixed mode settings before interconnecting your devices into a VCF to avoid the following issues:

- Incurring downtime as the devices reboot to commit the mixed mode or fabric settings.
- Manually correcting potential issues related to VCF formation because the device did not immediately join the VCF.

We strongly recommend configuring the mixed and fabric settings before you interconnect a device into a VCF. You can, however, use the `request virtual-chassis mode fabric local` or `request virtual-chassis mode mixed local` commands to set a device into fabric or mixed mode after you have interconnected your VCF.

- 
3. After the device reboots are complete, cable your spine devices to your leaf devices using supported SFP+ and QSFP+ interfaces.
  4. (Recommended) Configure a virtual management Ethernet (VME) interface for management of the VCF configuration:

[edit]

```
user@device# set interfaces vme unit 0 family inet address /ip-address/mask/
```





**NOTE:** A VME accesses the device in the master Routing Engine role using a management port, so cable management port em0 or em1 on each spine device in your VCF so the VME is available regardless of which spine device assumes the master Routing Engine role. See *Connecting a QFX Series Device to a Management Console*

5. Configure the desired SFP+ and QSFP+ interfaces into Virtual Chassis ports (VCPs):

```
user@device> request virtual-chassis vc-port set pic-slot pic-slot-number port port-number
user@device> request virtual-chassis vc-port set pic-slot pic-slot-number port port-number
```

The `show virtual-chassis vc-port` must be issued on the ports at both ends of the link in order for that link to be configured into a VCP.

6. Enter the `show virtual-chassis` command to confirm that the VCPs are operational and to learn the member ID of each member device in your VCF.

If you want to change the member ID that has been assigned to a member device, use the `request virtual-chassis renumber` command.

7. (Optional) Configure the mastership priority for each member device:

```
[edit virtual-chassis]
user@device# set member member-id mastership-priority number
```

In a nonprovisioned VCF, member roles are determined by a mastership election algorithm. The first value checked by the mastership election algorithm is the mastership priority value. The two QFX5100 devices with the highest mastership priority values assume the master and backup Routing Engine role, which must be used by the spine devices in a VCF. All other devices assume the linecard role.

QFX5100 devices assume the Routing Engine role, regardless of mastership priority settings. QFX5100 devices can also assume the linecard role.

QFX3600, QFX3500, and EX4300 devices always assume the linecard role in a VCF, regardless of the mastership priority settings.



**NOTE:** A spine device that isn't selected as master or backup Routing Engine assumes the linecard role. The spine devices should still be configured with a higher mastership priority value than the leaf devices to assure a spine device assumes the Routing Engine role when the master or backup Routing Engine fails.

If two or more devices have the same mastership priority value and are candidates for the Routing Engine role, the mastership election algorithm uses other parameters to determine which device is elected into the Routing Engine role. See *Understanding How the Master in a Virtual Chassis Is Elected*.

A device with a mastership priority of 0 never assumes the master or backup role.

For instance, to configure the mastership priority for member devices 0 through 19 in your VCF.

```
[edit virtual-chassis]
```

```
user@device# set member 0 mastership-priority 255
user@device# set member 1 mastership-priority 255
user@device# set member 2 mastership-priority 100
user@device# set member 3 mastership-priority 100
user@device# set member 4 mastership-priority 95
user@device# set member 5 mastership-priority 95
user@device# set member 6 mastership-priority 95
user@device# set member 7 mastership-priority 95
user@device# set member 8 mastership-priority 95
user@device# set member 9 mastership-priority 95
user@device# set member 10 mastership-priority 95
user@device# set member 11 mastership-priority 95
user@device# set member 12 mastership-priority 95
user@device# set member 13 mastership-priority 95
user@device# set member 14 mastership-priority 95
user@device# set member 15 mastership-priority 95
user@device# set member 16 mastership-priority 95
user@device# set member 17 mastership-priority 95
user@device# set member 18 mastership-priority 95
user@device# set member 19 mastership-priority 95
```

8. Install the VCF feature licenses.

For a VCF deployment, two license keys are recommended for redundancy—one for the device in the master role and the other for the device in the backup role.

To purchase a feature license for VCF, contact your Juniper Networks sales representative (<http://www.juniper.net/us/en/contact-us/sales-offices>). The Juniper sales representative will provide you with the feature license files and license keys. You will be asked to supply the chassis serial number of your switch; you can obtain the serial number by running the **show virtual-chassis** command.

After obtaining the licenses, follow the instructions in *Generating License Keys*.

#### Related Documentation

- [Autoprovisioning a Virtual Chassis Fabric on page 23](#)
- [Preprovisioning a Virtual Chassis Fabric on page 27](#)
- [Adding a Device to a Virtual Chassis Fabric on page 34](#)
- [Removing a Device From a Virtual Chassis Fabric on page 42](#)
- [Understanding Virtual Chassis Fabric Configuration on page 13](#)
- [Understanding Virtual Chassis Fabric Components on page 5](#)

---

## Adding a Device to a Virtual Chassis Fabric

This topic describes how to add a device to a Virtual Chassis Fabric (VCF).

It contains the following sections:

- [Adding a Leaf Device to an Autoprovisioned Virtual Chassis Fabric on page 35](#)
- [Adding a Spine Device to an Autoprovisioned Virtual Chassis Fabric on page 36](#)
- [Adding a Spine or Leaf Device to a Preprovisioned Virtual Chassis Fabric on page 38](#)
- [Adding a Spine or Leaf Device to a Nonprovisioned Virtual Chassis Fabric on page 40](#)

## Adding a Leaf Device to an Autoprovisioned Virtual Chassis Fabric

Update your device to the same version of Junos OS running on the devices in the VCF. See *Upgrading Software* or *Installing Software on an EX Series Switch with a Single Routing Engine (CLI Procedure)*.

To add a leaf device to an autoprovisioned VCF:

1. Log onto the device that you are adding to the VCF.
2. (Optional) Perform this step if you want to avoid the downtime associated with an extra reboot when your device is interconnected into your VCF. If you do not perform this step, the VCF auto-detects the fabric and mixed mode settings and, if needed, reboots the device as part of the process of changing these settings.

Configure the leaf device into fabric mode. Configure your device into mixed mode if your VCF includes QFX3600, QFX3500, or EX4300 devices as leaf devices..

If you are configuring a VCF composed entirely of QFX5100 devices:

```
user@device> request virtual-chassis mode fabric local
```

If you are configuring a VCF composed of two or more types of devices:

```
user@device> request virtual-chassis mode fabric mixed local
```

3. If the leaf device that you are adding to the VCF has not previously been configured, proceed to the next step.

If your device has been configured, zeroize your device and reboot:

```
user@device> request system zeroize
```

```
warning: System will be rebooted and may not boot without configuration
Erase all data, including configuration and log files? [yes,no] (yes) yes
```



**NOTE:** You must zeroize your device if you have previously entered one or more configuration commands, including basic configuration commands.

Your device will not join the VCF if it contains any configuration until it has been zeroized.



**NOTE:** The `request virtual-chassis mode fabric local` and `request virtual-chassis mode fabric mixed local` commands are entered in operational mode, so those settings are maintained when the device is zeroized.

You cannot use other methods to set a device into factory default mode before inserting it into a VCF if it was previously configured in another Virtual Chassis or VCF. You must use **`request system zeroize`**.

For additional information on this procedure, see *Reverting to the Default Factory Configuration for the EX Series Switch* or *Reverting to the Default Factory Configuration*.

4. (Required only if you are adding a device that turns a non-mixed VCF into a mixed VCF) Log in to the VCF and set all devices in the VCF to mixed mode, Configure all devices to reboot to complete this procedure.

```
user@device> request virtual-chassis mode mixed all-members reboot
```

The VCF experiences downtime as part of the reboot procedure.

5. Interconnect your leaf device into the existing spine devices, using at least one 10-Gbps SFP+ interface or 40-Gbps QSFP+ interface to connect to each spine device in the VCF.

An autoprovisioned VCF automatically adds a supported device in factory-default mode to the VCF when it is connected to a spine devices using a supported SFP+ or QSFP+ link. The SFP+ or QSFP+ link is automatically converted into a Virtual Chassis port (VCP) as part of this process.

No further configuration is required.

## Adding a Spine Device to an Autoprovisioned Virtual Chassis Fabric

Update your device to the same version of Junos OS running on the devices in the VCF before interconnecting it into the VCF. See *Upgrading Software* or *Installing Software on an EX Series Switch with a Single Routing Engine (CLI Procedure)*.

To add a spine device to an autoprovisioned VCF:

1. Log in to your VCF.
2. If you are replacing a spine device that is already part of the VCF, power off the spine device in the VCF.

Follow the steps in [“Removing a Device From a Virtual Chassis Fabric” on page 42](#) to remove the device from the VCF.

3. Modify the configuration.

If your new spine device is replacing an existing spine, modify the configuration to remove the old spine.

You can skip this step if you are not replacing an existing spine device.

```
[edit virtual-chassis]
```

```
user@device# delete member member-id
```

where *member-id* is the member ID of the spine that is removed from this procedure.

Add the spine device to the configuration:

```
[edit virtual-chassis]
```

```
user@device# set member member-id serial-number serial-number role [line-card | routing-engine]
```

For instance, to configure a spine device acting in the linecard role with the serial number OU81234567890 as member 3:

```
[edit virtual-chassis]
```

```
user@device# set member 3 serial-number OU81234567890 role line-card
```

The **set virtual-chassis member *member-id* fabric-tree-root** configuration statement specifies that only certain devices will be root nodes in the multicast distribution trees

(MDTs) created for directing traffic within the VCF. This configuration item preempts the default VCF behavior to create one MDT for every device in the VCF with that device as a root node. (See “[Understanding Traffic Flow Through a Virtual Chassis Fabric](#)” on page 20 and [fabric-tree-root](#) for more information about this option.) If your VCF uses this option to configure the spine devices as fabric tree roots (which is the recommended usage), then configure the new spine device as a fabric tree root as well:

```
[edit virtual-chassis]
user@device# set member member-id fabric-tree-root
```

For instance, to configure the spine device configured as member 3 as a fabric tree root node:

```
[edit virtual-chassis]
user@device# set member 3 fabric-tree-root
```

4. Commit the configuration.

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

5. Log in to the device that is going to be added to the VCF.
6. Configure the device into fabric mode. If needed, also configure the device into mixed mode.

Reboot the device to complete this configuration step.

If you are configuring a VCF composed entirely of QFX5100 devices:

```
user@device> request virtual-chassis mode fabric local reboot
```

If you are configuring a VCF composed of QFX5100 devices and at least one other type of device:

```
user@device> request virtual-chassis mode fabric mixed local reboot
```



**NOTE:** We recommend setting the fabric and mixed mode settings before interconnecting your devices into a VCF to avoid the following issues:

- Incurring downtime as the devices reboot to commit the mixed mode or fabric settings.
- Manually correcting potential issues related to VCF formation because the device did not immediately join the VCF.

You can, however, use the `request virtual-chassis mode fabric local` or `request virtual-chassis mode mixed local` commands to set a device into fabric or mixed mode after interconnecting your VCF.

7. (Required only if you are adding a device that turns a non-mixed VCF into a mixed VCF) Log in to the VCF and set all devices in the VCF to mixed mode, Configure all devices to reboot to complete this procedure.

```
user@device> request virtual-chassis mode mixed all-members reboot
```

The VCF experiences downtime as part of the reboot procedure.

8. After the device reboots, interconnect the new device into the VCF by cabling the device to the leaf devices in the VCF using supported SFP+ or QSFP+ interfaces.

The SFP+ or QSFP+ links are converted into VCPs automatically.

The new spine device should be operational once the cabling is complete.

## Adding a Spine or Leaf Device to a Preprovisioned Virtual Chassis Fabric

Update your device to the same version of Junos OS running on the devices in the VCF before interconnecting it into the VCF. See *Upgrading Software* or *Installing Software on an EX Series Switch with a Single Routing Engine (CLI Procedure)*.

To add a spine or leaf device to a preprovisioned VCF:

1. Log in to your VCF.
2. If you are replacing a device that is already part of the VCF, power off the device in the VCF.

Follow the steps in [“Removing a Device From a Virtual Chassis Fabric” on page 42](#) to remove the device from the VCF.

3. Modify the configuration.

If your new device is replacing an existing device, modify the configuration to remove the old device.

You can skip this portion of the procedure if you are not replacing an existing device.

```
[edit virtual-chassis]
user@device# delete member member-id
```

where *member-id* is the member ID of the spine that is removed from this procedure.

Add the new device to the configuration:

```
[edit virtual-chassis]
user@device# set member member-id serial-number serial-number role [line-card |
routing-engine]
```

For instance, to configure a device with the serial number OU81234567890 into the Routine Engine role as member 3:

```
[edit virtual-chassis]
user@device# set member 3 serial-number OU81234567890 role routing-engine
```

(For spine devices only) The **set virtual-chassis member *member-id* fabric-tree-root** configuration statement specifies that only certain devices will be root nodes in the multicast distribution trees (MDTs) created for directing traffic within the VCF. This configuration item preempts the default VCF behavior to create one MDT for every device in the VCF with that device as a root node. (See [“Understanding Traffic Flow Through a Virtual Chassis Fabric” on page 20](#) and **fabric-tree-root** for more information about this option.) If your VCF uses this option to configure the spine devices as fabric tree roots (which is the recommended usage), then configure the new spine device as a fabric tree root as well:

```
[edit virtual-chassis]
```

```
user@device# set member member-id fabric-tree-root
```

For instance, to configure the spine device configured as member 3 as a fabric tree root node:

```
[edit virtual-chassis]
```

```
user@device# set member 3 fabric-tree-root
```

4. Commit the configuration.

```
[edit]
```

```
user@device# commit
```

5. Log in to the device that is going to be added to the VCF.
6. Configure the device into fabric mode. If needed, also configure the device into mixed mode.

Reboot the device to complete this configuration step.

If you are configuring a VCF composed entirely of QFX5100 devices:

```
user@device> request virtual-chassis mode fabric local reboot
```

If you are configuring a VCF composed of two or more types of devices:

```
user@device> request virtual-chassis mode fabric mixed local reboot
```



**NOTE:** If you are adding a QFX3600, QFX3500, or EX4300 device to a VCF that is composed entirely of QFX5100 devices, you must also log in to the VCF and set all of the devices in the VCF into mixed mode.

Log in to the VCF and enter the **request virtual-chassis mode mixed all-members reboot** command to perform this task.

The VCF reboots and incurs downtime to complete this procedure.



**NOTE:** We recommend that you set the fabric and mixed mode settings before you interconnect your devices into a VCF to avoid the following issues:

- Incurring downtime as the devices reboot to commit the mixed mode or fabric settings.
- Manually correcting potential issues related to VCF formation because the device did not immediately join the VCF.

You can, however, use the **request virtual-chassis mode fabric local** or **request virtual-chassis mode mixed local** commands to recover a device that was not set into fabric or mixed mode before you interconnect it into your VCF.

7. (Required only if you are adding a device that turns a non-mixed VCF into a mixed VCF) Log in to the VCF and set all devices in the VCF to mixed mode, Configure all devices to reboot to complete this procedure.

```
user@device> request virtual-chassis mode mixed all-members reboot
```

The VCF experiences downtime as part of the reboot procedure.

8. After the device reboots, interconnect the new device into the VCF using supported SFP+ or QSFP+ interfaces.

The SFP+ or QSFP+ links are converted into VCPs automatically.

The new device should be operational shortly after the cabling is complete.

## Adding a Spine or Leaf Device to a Nonprovisioned Virtual Chassis Fabric

---



**CAUTION:** Configure your VCF using autoprovisioning or preprovisioning unless you have a compelling reason to use nonprovisioned configuration. You can configure all aspects of a VCF using autoprovisioned or preprovisioned configuration.

Nonprovisioned VCF configuration is highly discouraged. Nonprovisioned VCF configuration should only be used by VCF experts in specialized scenarios.

---

Update your device to the same version of Junos OS running on the devices in the VCF before interconnecting it into the VCF. See *Upgrading Software* or *Installing Software on an EX Series Switch with a Single Routing Engine (CLI Procedure)*.

To add a spine or leaf device to a nonprovisioned VCF:

1. Log in to your VCF.
2. If you are replacing a device that is already part of the VCF, power off the device in the VCF. Uncable the device once the power off is complete.

You can skip this step if you are adding a new device without replacing an existing device. You must skip this step if there is no configuration for the device that you are removing from the VCF.

If the device is configured, delete the device from the VCF configuration:

```
[edit virtual-chassis]
user@device# delete member member-id
```

where *member-id* is the member ID of the device that you are removing.

3. Log in to the device that you are going to add to the VCF.
4. Configure the device into fabric mode. If needed, also configure the device into mixed mode.

Reboot the device to complete this configuration step.

If you are configuring a VCF composed entirely of QFX5100 devices:

```
user@device> request virtual-chassis mode fabric local reboot
```

If you are configuring a VCF that includes at least one QFX3600, QFX3500, or EX4300 devices as a leaf device:

```
user@device> request virtual-chassis mode fabric mixed local reboot
```





**NOTE:** If you are adding a QFX3600, QFX3500, or EX4300 device to a VCF that is composed entirely of QFX5100 devices, you must also log in to the VCF and set all of the devices in the VCF into mixed mode.

Log in to the VCF and enter the **request virtual-chassis mode mixed all-members reboot** command to perform this task.

The VCF reboots and incurs downtime to complete this procedure.



**NOTE:** We recommend that you set the fabric and mixed mode settings before you interconnect your devices into a VCF to avoid the following issues:

- Incurring downtime as the devices reboot to commit the mixed mode or fabric settings.
- Manually correcting potential issues related to VCF formation because the device did not immediately join the VCF.

You can, however, use the **request virtual-chassis mode fabric local** or **request virtual-chassis mode mixed local** commands to set a device into fabric or mixed mode after interconnecting your VCF.

5. (Required only if you are adding a device that turns a non-mixed VCF into a mixed VCF) Log in to the VCF and set all devices in the VCF to mixed mode, Configure all devices to reboot to complete this procedure.

```
user@device> request virtual-chassis mode mixed all-members reboot
```

The VCF experiences downtime as part of the reboot procedure.

6. After the device reboots, interconnect it into the VCF using supported SFP+ or QSFP+ interfaces.

7. Configure the SFP+ or QSFP+ interfaces into Virtual Chassis ports (VCPs):

```
user@device> request virtual-chassis vc-port set pic-slot pic-slot-number port port-number
user@device> request virtual-chassis vc-port set pic-slot pic-slot-number port port-number
```

The **request virtual-chassis vc-port** must be configured on the ports at both ends of the link in order for that link to be configured into a VCP.

8. (Optional) Log in to the VCF and set the mastership priority of the new device:

```
[edit virtual-chassis]
user@device# set member member-id mastership-priority number
```

If needed, enter the **show virtual-chassis** command to learn the member ID of the new member device in the VCF.

#### Related Documentation

- [Removing a Device From a Virtual Chassis Fabric on page 42](#)
- [Autoprovisioning a Virtual Chassis Fabric on page 23](#)
- [Preprovisioning a Virtual Chassis Fabric on page 27](#)

- [Understanding Virtual Chassis Fabric Configuration on page 13](#)
- [Understanding Virtual Chassis Fabric Components on page 5](#)
- [Understanding Traffic Flow Through a Virtual Chassis Fabric on page 20](#)

---

## Removing a Device From a Virtual Chassis Fabric

---

This topic describes how to remove a device from a Virtual Chassis Fabric (VCF):

To remove a device from a VCF:

1. Power off the device that you are removing from the VCF.
2. Uncable the device that you are removing from the VCF.
3. Log in to the Virtual Management ethernet (VME) interface. Remove the device from the VCF configuration.

You can skip this step if you are removing a device that was never configured.

```
[edit virtual-chassis]
```

```
user@device# delete member member-id
```

4. Delete the Virtual Chassis port (VCP) or ports on the devices that are still in the VCF but were connected to the removed device.

```
user@device> request virtual-chassis vc-port delete pic-slot pic-slot port port-number member member-id
```

When a device is removed from a VCF, the interface on the other end of the VCP link that was connected to the removed device remains configured as a VCP.

You can check the results of this command using the [show virtual-chassis vc-port](#) command.

5. (Required only if you are removing a device that turns a mixed VCF into a homogenous VCF) Log in to the VCF and disable mixed mode for all of the devices in the VCF, Configure all devices to reboot to complete this procedure.

```
user@device> request virtual-chassis mode mixed disable all-members reboot
```

This step should only be taken if you are removing a QFX3600, QFX3500, or EX4300 device from a mixed VCF and the only devices remaining in the VCF are QFX5100 devices.

The VCF experiences downtime as part of the reboot procedure.

6. Commit the configuration.

```
[edit]
```

```
user@device# commit
```

7. Power on the device that was removed from the VCF, and log in to it.
8. (Optional, but recommended) Delete the VCP or VCPs on the device that was removed:

```
user@device> request virtual-chassis vc-port delete pic-slot pic-slot port port-number member member-id
```
9. (Optional, but recommended) Reset the fabric and mixed mode settings.

If you are removing a device that was part of a VCF composed entirely of the same device:

```
user@device> request virtual-chassis mode fabric disable reboot
```

If you are removing a device that was part of a VCF composed of two or more device types:

```
user@device> request virtual-chassis mode fabric mixed disable reboot
```

Reboot the device to complete the process.

We recommend resetting the fabric and mixed mode settings immediately after removing it from the VCF to avoid any potential issues with your device if it is placed in your network in another role.

**Related  
Documentation**

- [Adding a Device to a Virtual Chassis Fabric on page 34](#)
- [Autoprovisioning a Virtual Chassis Fabric on page 23](#)
- [Preprovisioning a Virtual Chassis Fabric on page 27](#)
- [Understanding Virtual Chassis Fabric Configuration on page 13](#)
- [Understanding Virtual Chassis Fabric Components on page 5](#)

---

## Upgrading Software for a Virtual Chassis Fabric

This topic describes the processes that can be used to update software on an operational Virtual Chassis Fabric (VCF).

You should update the software on each device before initially interconnecting your VCF. This process describes the options that are available for upgrading software after a VCF is setup.

It contains the following sections:

- [NSSU on page 43](#)
- [Automatic Software Update on page 43](#)
- [Standard Upgrade on page 44](#)

### NSSU

Nonstop software upgrade (NSSU) enables you to upgrade the software running on all member devices in a VCF with minimal network traffic disruption during the upgrade.

See *Upgrading Software on a Virtual Chassis Fabric Using Nonstop Software Upgrade*.

### Automatic Software Update

Automatic software update automatically upgrades the Junos OS running on a device joining a VCF to the version of Junos OS running on the VCF at the moment the new device is cabled into the VCF.

To configure the automatic software update feature for a VCF composed entirely of QFX5100 devices:

[edit]

```
user@device# set virtual-chassis auto-sw-update package-name package-name
```

To configure the automatic software update feature on a VCF composed of QFX5100 devices and at least one other type of device:

[edit]

```
user@device# set virtual-chassis auto-sw-update qfx-5 package-name package-name
```

```
user@device# set virtual-chassis auto-sw-update qfx-3 package-name package-name
```

```
user@device# set virtual-chassis auto-sw-update ex-4300 package-name package-name
```

where **qfx-5** specifies the path to the Junos OS used to run a QFX5100 devices, **qfx-3** specifies the path to the Junos OS used to run QFX3600 and QFX3500 devices, and **ex4300** specifies the path to the Junos OS used to run EX4300 switches.

If the software package is located on a local directory on the switch, use the following format for *package-name*:

*/pathname/package-name*

If the software package is to be downloaded and installed from a remote location, use one of the following formats:

*ftp://hostname/pathname/package-name*

*ftp://username:prompt@ftp.hostname.net/package-name*

*http://hostname/pathname/package-name*

## Standard Upgrade

You can upgrade software on a VCF using the traditional method of upgrading software for Junos OS by logging onto the master Routing Engine and using the **request system software add** command to initiate the upgrade on a non-mixed VCF or the **request system software add set [package-name package-name ...]** to initiate the upgrade on a mixed VCF, where *package-name* is the path to an image for one device family.

When you upgrade Junos OS on a VCF using the traditional software upgrade, each device in the VCF must reboot. The entire system is down until the upgrade process is complete.

For information on performing this procedure, see *Upgrading Software*.

### Related Documentation

- [Adding a Device to a Virtual Chassis Fabric on page 34](#)
- [Understanding Software Upgrades in a Virtual Chassis Fabric on page 22](#)

## Verifying the Member ID, Role, Status, and Neighbor Member Connections of a Virtual Chassis Fabric Member Device

---

**Purpose** Use this procedure to learn the current member ID, role, status, Virtual Chassis port (VCP) connections, and other information for the devices in your VCF.

Understanding the current member IDs, roles, device statuses, and VCP connections is required for routine monitoring of your VCF. You'll often need to identify this basic operational information to confirm a device or a VCP is working properly in the VCF, or how the VCF topology changed as a result of a configuration change or network error.

**Action** To display VCF status using the CLI:

#### show virtual-chassis (Virtual Chassis Fabric)

```
user@switch> show virtual-chassis
Preprovisioned Virtual Chassis Fabric
Fabric ID: 0282.5fa0.3f08
Fabric Mode: Enabled
```

					Mstr	Mixed Route Neighbor			
List	Member ID	Status	Serial No	Model	prio	Role	Mode	Mode	ID
Interface									
0 (FPC 0)	Prsnt	AB3112430001	qfx5100-48s	129	Master*	N	F	3	
vcp-255/1/0									2
vcp-255/1/1									4
vcp-255/1/2									4
vcp-255/1/3									4
1 (FPC 1)	Prsnt	AB3112230001	qfx5100-48s	129	Backup	N	F	3	
vcp-255/1/0									2
vcp-255/1/1									4
vcp-255/1/2									4
vcp-255/1/3									4
2 (FPC 2)	Prsnt	AB3112460011	qfx5100-48s	0	Linecard	N	F	1	
vcp-255/1/0									0
vcp-255/1/1									0
3 (FPC 3)	Prsnt	AB3112460011	qfx5100-48s	0	Linecard	N	F	1	
vcp-255/1/0									0
vcp-255/1/1									0
4 (FPC 4)	Prsnt	AB3112430011	qfx5100-48s	0	Linecard	N	F	1	
vcp-255/1/0									0
vcp-255/1/1									0

**Meaning** This output verifies that fabric mode is enabled and that all devices in the VCF are participating in the fabric, as shown by the **Prsnt** status output for each device.

The Neighbor ID and Interface outputs show that all VCPs are operating correctly.

**Related Documentation**

- [Autoprovisioning a Virtual Chassis Fabric on page 23](#)
- [Preprovisioning a Virtual Chassis Fabric on page 27](#)

## Verifying Virtual Chassis Port Connections in a Virtual Chassis Fabric

**Purpose** Verify the Virtual Chassis Ports (VCPs) in your Virtual Chassis Fabric (VCF).

You should use this command if you suspect a VCP link in your VCF is broken.

**Action** To display the VCPs of a device:

```
user@switch> show virtual-chassis vc-port member 4
fpc4:
```

Interface or PIC / Port	Type	Trunk ID	Status	Speed (mbps)	Neighbor ID	Interface
0/48	Auto-Configured	-1	Up	40000	0	vcp-255/0/2
0/49	Auto-Configured	-1	Up	40000	1	vcp-255/0/2
0/50	Auto-Configured	-1	Up	40000	2	vcp-255/0/2
0/51	Auto-Configured	-1	Up	40000	3	vcp-255/0/2

**Meaning** All of the VCPs on this device are up and active.

If the **Status** of an interface is **Absent** or the interface that you thought was a VCP does not appear in the command output, you likely have a problem with a link that has not been converted into a VCP. In this scenario, configure the interface on the link into a VCP using the **request virtual-chassis vc-port** command.

- Related Documentation**
- [Autoprovisioning a Virtual Chassis Fabric on page 23](#)
  - [Preprovisioning a Virtual Chassis Fabric on page 27](#)

## Verifying the Virtual Chassis Fabric Mode Settings

**Purpose** Verify the Virtual Chassis Fabric (VCF) mode settings on a device.

You must configure all devices in a VCF into fabric mode using the **request virtual-chassis mode fabric** command for the devices to operate in a VCF.

All VCFs use QFX5100 devices in the spine role. If a VCF uses a QFX3500, QFX3600, or EX4300 devices as a leaf node, you must also configure each device into mixed mode using the **request virtual-chassis mode mixed** command.

You must also configure a device out of mixed and fabric mode if it is removed from a VCF and placed into your network in a different role.

**Action** To display the current mode of a device:

```
user@switch> show virtual-chassis mode
fpc0:
```

```
-----
Current mode : Fabric with mixed devices
Future mode after reboot : Fabric with mixed devices
```

**Meaning** The output indicates that the switch is currently in mixed and fabric mode.

The output also indicates that the mode will not change when the device is rebooted without further configuration. You must reboot the device to change the fabric or mixed mode, so the **Future mode after reboot** output differs from the **Current mode** output when the mode has been changed but the device has not been rebooted.

- Related Documentation**
- [Autoprovisioning a Virtual Chassis Fabric on page 23](#)
  - [Preprovisioning a Virtual Chassis Fabric on page 27](#)

---

## Troubleshooting Virtual Chassis Fabric

This topic describes troubleshooting some common issues for a Virtual Chassis Fabric (VCF):

- [Large-Scale Virtual Chassis Fabric Becomes Unstable When Logging is Enabled on page 47](#)
- [Virtual Chassis Port Link Does Not Form on page 48](#)
- [QFX5100 Leaf Device Assumes Routing Engine Role on page 48](#)

### Large-Scale Virtual Chassis Fabric Becomes Unstable When Logging is Enabled

- |                 |  |
|-----------------|--|
| <b>Problem</b>  | <b>Description:</b> When detailed system logging or trace operations are enabled in larger-scale VCFs, you observe significant impact on VCF stability, such as: <ul style="list-style-type: none"><li>• Increased VCF convergence time</li><li>• Traffic interruption</li></ul>   |
| <b>Cause</b>    | System logging and tracing operations place a load on the master Routing Engine device in a VCF, taking processing cycles away from managing VCF operations. Logging in general, especially higher levels of logging and tracing operations, can have an impact on VCF stability.  |
| <b>Solution</b> | To help ensure good convergence and stable operation in a large-scale VCF, system logging and tracing should always be used with discretion. During normal VCF operation, system logging should be set at or below the <b>notice</b> level, and tracing options disabled. When logging or tracing is necessary to troubleshoot a particular issue, use the following guidelines to minimize impact on VCF stability: <ul style="list-style-type: none"><li>• Use the <b>detail</b> tracing option or system logging levels at or above the <b>error</b> level only for short periods of time during troubleshooting, and disable these settings after gathering enough information to begin analyzing the issue.</li><li>• Avoid logging the same level of information to more than one log file, which adds extra processing without the benefit of providing more information. Setting up logging to different files for different levels or facilities is a better option.</li><li>• Choose remote logging rather than local logging, and avoid logging to the console.</li></ul> |

## Virtual Chassis Port Link Does Not Form

**Problem**    **Description:** You connect a 40-Gbps QSFP+ port or a 10-Gbps SFP+ port between a leaf device and a spine device in an autoprovisioned or preprovisioned VCF. You expect the automatic Virtual Chassis port (VCP) conversion feature to convert the link into a VCP link, but the conversion doesn't occur.

The [show virtual-chassis vc-port](#) output indicates that the status of the interface is **Absent** or one or both of interfaces don't appear in the [show virtual-chassis vc-port](#) output.

**Cause**    If one end of a link is configured as a VCP and the other is not configured as a VCP, the VCP link does not form.

The automatic VCP conversion feature, therefore, does not work in the following situations:

- a 40-Gbps QSFP+ or 10-Gbps SFP+ interface on one end of the link is already configured as a VCP.

If you have previously removed a device from a VCF but haven't used the **request virtual-chassis vc-port delete** command to convert the interface that was connected to the removed device out of VCP mode, the interface is still configured as a VCP.

If you have removed a device from one Virtual Chassis or VCF and not changed the VCP port setting, the device being added to the VCF might also be configured as a VCP.

- a 40-Gbps QSFP+ port on an EX4300 switch, which is configured as a VCP by default, is interconnecting to a spine device.

**Solution**    Manually configure the interface that is not configured as a VCP into a VCP using the **request virtual-chassis vc-port set** command.

## QFX5100 Leaf Device Assumes Routing Engine Role

**Problem**    **Description:** A QFX5100 device configured as a leaf device assumes the Routing Engine role during VCF setup. The **show virtual-chassis** output confirms the role.

**Solution**    The device can assume the Routing Engine role for several minutes during setup before it receives the configuration from the master Routing Engine, but eventually returns to the linecard role with no user intervention.

**Related Documentation**

- [Virtual Chassis Fabric Overview on page 3](#)
- [traceoptions on page 78](#)
- *Junos OS System Log Configuration Statements*
- *Junos OS System Logging Facilities and Message Severity Levels*



## CHAPTER 2

# Configuration Statements for Virtual Chassis Fabric

- [\[edit virtual-chassis\] Configuration Statement Hierarchy on page 50](#)
- [aliases \(Virtual Chassis\) on page 52](#)
- [alias-name \(Virtual Chassis aliases\) on page 53](#)
- [auto-provisioned on page 54](#)
- [auto-sw-update on page 55](#)
- [enhanced-hash-key on page 57](#)
- [fabric-load-balance on page 60](#)
- [fabric-tree-root on page 61](#)
- [id on page 62](#)
- [inactivity-interval \(Fabric Load Balance\) on page 63](#)
- [location \(Virtual Chassis\) on page 64](#)
- [mac-persistence-timer on page 65](#)
- [mastership-priority on page 66](#)
- [member on page 68](#)
- [no-management-vlan on page 69](#)
- [no-split-detection on page 70](#)
- [package-name on page 71](#)
- [preprovisioned on page 72](#)
- [role on page 73](#)
- [serial-number on page 76](#)
- [serial-number \(Virtual Chassis aliases\) on page 77](#)
- [traceoptions \(Virtual Chassis\) on page 78](#)
- [vcp-snmp-statistics on page 81](#)
- [virtual-chassis on page 82](#)

## [edit virtual-chassis] Configuration Statement Hierarchy

This topic lists supported and unsupported configuration statements in the **[edit virtual-chassis]** hierarchy level on EX Series and QFX Series switches.

- *Supported* statements are those that you can use to configure some aspect of a software feature on the switch.
- *Unsupported* statements are those that appear in the command-line interface (CLI) on the switch, but that have no effect on switch operation if you configure them.
- Not all features are supported on all switch platforms.

For detailed information about feature support on specific EX Series or QFX Series switch platforms, see [Feature Explorer](#).

This topic lists:

- [Supported Statements in the \[edit virtual-chassis\] Hierarchy Level on page 50](#)
- [Unsupported Statements in the \[edit virtual-chassis\] Hierarchy Level on page 51](#)

### Supported Statements in the [edit virtual-chassis] Hierarchy Level

The following hierarchy shows the **[edit virtual-chassis]** configuration statements supported on EX Series or QFX Series switches:

```
virtual-chassis {
  aliases {
    serial-number serial-number {
      alias-name alias-name;
    }
  }
  auto-provisioned;
  auto-sw-update {
    (ex-4200 | ex-4300 | ex-4500 | ex-4600 | qfx-3 | qfx-5)
    package-name package-name;
  }
  fast-failover (ge | vcp disable | xe);
  graceful-restart {
    disable;
  }
  id id;
  mac-persistence-timer [minutes | disable];
  member member-id {
    fabric-tree-root;
    location location;
    mastership-priority number;
    no-management-vlan;
    role (line-card | routing-engine);
    serial-number;
  }
  no-split-detection;
  preprovisioned;
  traceoptions {
```

```

    file filename <files number> <size size> <world-readable | no-world-readable> <match
      regex>;
    flag flag ;
  }
  vc-port {
    lag-hash (packet-based | source-port-based);
  }
  vcp-no-hold-time;
  vcp-snmp-statistics;
}

```

## Unsupported Statements in the [edit virtual-chassis] Hierarchy Level

All statements in the [edit virtual-chassis] hierarchy level that are displayed in the command-line interface (CLI) on the switch are supported on the switch and operate as documented.

### Related Documentation

- [Preprovisioning a Virtual Chassis Fabric on page 27](#)
- [Autoprovisioning a Virtual Chassis Fabric on page 23](#)
- [Adding a Device to a Virtual Chassis Fabric on page 34](#)
- [Configuring a QFX Series Virtual Chassis \(CLI Procedure\)](#)
- [Configuring an EX4300 Virtual Chassis \(CLI Procedure\)](#)
- [Configuring an EX2200 Virtual Chassis \(CLI Procedure\)](#)
- [Configuring an EX3300 Virtual Chassis \(CLI Procedure\)](#)
- [Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\)](#)
- [Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\)](#)
- [Configuring an EX8200 Virtual Chassis \(CLI Procedure\)](#)

## aliases (Virtual Chassis)

---

<b>Syntax</b>	<pre>aliases {     serial-number serial-number {         alias-name alias-name;     } }</pre>
<b>Hierarchy Level</b>	[edit <a href="#">virtual-chassis</a> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 14.1X53-D10 for EX Series and QFX Series switches.
<b>Description</b>	<p>Create an alias for a member switch in a Virtual Chassis or Virtual Chassis Fabric (VCF). An alias allows you to more clearly identify the member switches in your Virtual Chassis or VCF by assigning a text label to a member switch's serial number.</p> <p>An alias is not specified for a device until the alias name is specified using the <b>alias-name</b> keyword.</p> <p>The alias appears in the <b>Alias-Name</b> field in the <b>show virtual-chassis</b> command.</p> <p>Alias usage is optional and aliases are used for administrative purposes only. Setting an alias has no effect on the operation of the member switch.</p> <p>The remaining statements are explained separately.</p>
<b>Required Privilege Level</b>	system—To view this statement in the configuration. system-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Autoprovisioning a Virtual Chassis Fabric on page 23</a></li><li>• <a href="#">Preprovisioning a Virtual Chassis Fabric on page 27</a></li><li>• <a href="#">Configuring a QFX Series Virtual Chassis (CLI Procedure)</a></li><li>• <a href="#">Understanding Virtual Chassis Fabric Components on page 5</a></li><li>• <a href="#">Understanding QFX Series Virtual Chassis Components</a></li></ul>

## alias-name (Virtual Chassis aliases)

<b>Syntax</b>	<code>alias-name <i>alias-name</i>;</code>
<b>Hierarchy Level</b>	<code>[edit <a href="#">virtual-chassis aliases serial-number</a> <i>serial-number</i>]</code>
<b>Release Information</b>	Statement introduced in Junos OS Release 14.1X53-D10 for EX Series and QFX Series switches.
<b>Description</b>	Create an alias for a member switch in a Virtual Chassis or Virtual Chassis Fabric (VCF). An alias allows you to more clearly identify the member switches in your Virtual Chassis or VCF by assigning a text label to a member switch's serial number.

The alias appears in the **Alias-Name** field in the **show virtual-chassis** command.

Alias usage is optional and aliases are used for administrative purposes only. Setting an alias has no effect on the operation of the member switch.

In the following example, the **dc-floor-1** alias name is assigned to the member switch with the serial number AB0123456789.

### set serial-number

```
[edit virtual-chassis aliases]
user@switch# set serial-number AB0123456789 alias-name dc-floor-1
```

### show virtual-chassis

```
user@switch> show virtual-chassis
Preprovisioned Virtual Chassis Fabric
Fabric ID: 9d5d.5556.919a
Fabric Mode: Enabled

Member ID  Status  Serial No  Alias-Name  Model  Mstr  prio  Role
0 (FPC 0)  Prsnt    AB0123456789  dc-floor-1  qfx5100-48s-6q  129   Master
<additional output removed for brevity>
```

<b>Options</b>	<b><i>alias-name</i></b> —The text label, or alias, assigned to the member switch by the user.
<b>Required Privilege Level</b>	system—To view this statement in the configuration. system-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Autoprovisioning a Virtual Chassis Fabric on page 23</a></li> <li>• <a href="#">Preprovisioning a Virtual Chassis Fabric on page 27</a></li> <li>• <a href="#">Configuring a QFX Series Virtual Chassis (CLI Procedure)</a></li> <li>• <a href="#">Understanding Virtual Chassis Fabric Components on page 5</a></li> <li>• <a href="#">Understanding QFX Series Virtual Chassis Components</a></li> </ul>

## auto-provisioned

---

<b>Syntax</b>	auto-provisioned;
<b>Hierarchy Level</b>	[edit <a href="#">virtual-chassis</a> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 13.2X51-D20 for EX Series switches and QFX Series devices in a Virtual Chassis Fabric (VCF).
<b>Description</b>	<p>Enable the auto-provisioned configuration mode for a Virtual Chassis Fabric (VCF).</p> <p>When a VCF is autoprovisioned, you can plug and play leaf devices that have not been configured or are zeroized into your VCF without user configuration. The leaf devices are automatically configured into the linecard role and all other VCF configuration—configuring Virtual Chassis ports (VCPs), the member ID, fabric mode, mixed mode (if applicable), and other parameters—is completed without further user action when a supported spine device interconnects to the leaf device by using a 10-Gbps SFP+ or 40-Gbps QSFP+ link that can be converted into a VCP.</p> <p>A leaf device whose fabric or mixed mode setting is changed as part of the autoprovisioning process automatically reboots. You can avoid this reboot by configuring the fabric or mixed mode setting on the leaf device before interconnecting into the VCF.</p>
<b>Required Privilege Level</b>	system—To view this statement in the configuration. system-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Autoprovisioning a Virtual Chassis Fabric on page 23</a></li><li>• <a href="#">Understanding Virtual Chassis Fabric Configuration on page 13</a></li></ul>

## auto-sw-update

<b>Syntax</b>	<pre> auto-sw-update {   (ex-4200   ex-4300   ex-4500   ex-4600   qfx-3   qfx-5)   package-name package-name; } </pre>
<b>Hierarchy Level</b>	[edit <b>virtual-chassis</b> ]
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 10.0 for EX Series switches.</p> <p>The <b>ex-4200</b> and <b>ex-4500</b> options introduced in Junos OS Release 12.2 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series.</p> <p>The <b>ex-4300</b>, <b>qfx-3</b>, and <b>qfx-5</b> options introduced in Junos OS Release 13.2X51-D20.</p> <p>The <b>ex-4600</b> option introduced in Junos OS Release 13.2X51-D25.</p>
<b>Description</b>	<p>Enable the automatic software update feature for Virtual Chassis or Virtual Chassis Fabric (VCF) configurations.</p> <p>You should only use the keywords that specify a device—<b>ex-4300</b>, <b>ex-4600</b>, <b>qfx-3</b>, and <b>qfx-5</b>—when configuring automatic software update on a mixed Virtual Chassis or Virtual Chassis Fabric (VCF). You can simply specify the <i>package-name</i> without specifying the device keywords in non-mixed Virtual Chassis or VCF topologies.</p> <p>You must enter the <b>auto-sw-update</b> statement multiple times—once for each device family in your mixed Virtual Chassis or VCF—in most scenarios when enabling the automatic software update for a mixed Virtual Chassis or VCF.</p> <p>The Junos OS package for an EX4500 switch updates the software for EX4500 and EX4550 switches. You do not, therefore, need to specify the <b>ex-4500</b> keyword when configuring automatic software update for a mixed Virtual Chassis that include EX4500 and EX4550 switches only. You also only have to enter the <b>ex-4500</b> keyword once to configure automatic software update for all EX4500 and EX4550 member switches in the same mixed Virtual Chassis.</p> <p>The Junos OS package for a QFX3500 device updates the software for QFX3500 and QFX3600 devices. You do not, therefore, need to specify the <b>qfx-3</b> keyword when configuring automatic software update for a Virtual Chassis composed entirely of QFX3500 and QFX3600 devices. You also have to enter the <b>qfx-3</b> keyword only once to configure automatic software update for all QFX3500 and QFX3600 member devices in the same mixed Virtual Chassis.</p> <p>The remaining statement is explained separately.</p>
<b>Default</b>	The automatic software update feature is disabled.
<b>Options</b>	<p><b>package-name package-name</b>—Specify a path to a Junos OS software image.</p> <p><b>ex-4200</b>—Specify a path to a Junos OS image for an EX4200 switch when enabling automatic software update for a mixed EX4200 and EX4500 Virtual Chassis, mixed</p>

EX4200 and EX4550 Virtual Chassis, or mixed EX4200, EX4500, or EX4550 Virtual Chassis.

**ex-4300**—Specify a path to a Junos OS image for an EX4300 switch when enabling automatic software update for a mixed Virtual Chassis or VCF.

**ex-4500**—Specify a path to a Junos OS image for an EX4500 switch, an EX4550 switch, or both types of switches when enabling automatic software update for a mixed EX4200 and EX4500 Virtual Chassis, mixed EX4200 and EX4550 Virtual Chassis, or mixed EX4200, EX4500, or EX4550 Virtual Chassis.

The Junos OS package for an EX4500 switch updates the software for EX4500 and EX4550 switches. Therefore, you only enter this command once to upgrade the EX4500 and EX4550 member switches in the same mixed Virtual Chassis.

The **ex-4500** keyword also does not need to be specified when configuring automatic software update for a mixed EX4500 and EX4550 Virtual Chassis.

**ex-4600**—Specify a path to a Junos OS image for an EX4600 switch when enabling automatic software update for a mixed Virtual Chassis.

**qfx-3**—Specify a path to a Junos OS image for a QFX3500, QFX3600, or both types of devices when enabling automatic software update for a mixed VCF.

**qfx-5**—Specify a path to a Junos OS image for a QFX5100 device when enabling automatic software update for a mixed VCF.

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

<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <i>Example: Configuring Automatic Software Update on EX4200 Virtual Chassis Member Switches</i></li><li>• <i>Configuring Automatic Software Update on Virtual Chassis Member Switches (CLI Procedure)</i></li><li>• <a href="#">Understanding Software Upgrades in a Virtual Chassis Fabric on page 22</a></li></ul>
------------------------------	--



## enhanced-hash-key

<b>List of Syntax</b>	<a href="#">Syntax (EX Series and QFX5100 Switch) on page 57</a> <a href="#">Syntax (QFX10002 and QFX10008 Switches) on page 57</a>
<b>Syntax (EX Series and QFX5100 Switch)</b>	<pre> enhanced-hash-key {   ecmp-resilient-hash;   fabric-load-balance {     flowlet {       inactivity-interval <i>interval</i>;     }     per-packet;   }   hash-mode {     layer2-header;     layer2-payload;   }   inet {     no-ipv4-destination-address;     no-ipv4-source-address;     no-l4-destination-port;     no-l4-source-port;     no-protocol;     vlan-id;   }   inet6 {     no-ipv6-destination-address;     no-ipv6-source-address;     no-l4-destination-port;     no-l4-source-port;     no-next-header;     vlan-id;   }   layer2 {     no-destination-mac-address;     no-ether-type;     no-source-mac-address;     vlan-id;   } } </pre>
<b>Syntax (QFX10002 and QFX10008 Switches)</b>	<pre> enhanced-hash-key {   hash-seed <i>seed-value</i>;   inet {     no-ipv4-destination-address;     no-ipv4-source-address;     no-l4-destination-port;     no-l4-source-port;   }   inet6 {     ipv6-flow-label;     no-ipv6-destination-address;     no-ipv6-source-address;     no-l4-destination-port;   } } </pre>

```

        no-l4-source-port;
    }
    layer2 {
        destination-mac-address
        inner-vlan-id;
        no-ether-type;
        no-vlan-id;
        source-mac-address;
    }
    no-mpls;
    gre {
        key;
        protocol;
    }
    vxlan-vnid
    }
}

```

<b>Hierarchy Level</b>	[edit forwarding-options]
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 13.2X51-D15 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 13.2X51-D20 for QFX Series devices.</p> <p>The <b>fabric-load-balance</b> statement introduced in Junos OS Release 14.1X53-D10.</p> <p>The <b>hash-seed</b> statement introduced in Junos OS Release 15.1X53-D30.</p>
<b>Description</b>	<p>Configure the hashing key used to hash link aggregation group (LAG) and equal-cost multipath (ECMP) traffic, or enable adaptive load balancing (ALB) in a Virtual Chassis Fabric (VCF).</p> <p>The hashing algorithm is used to make traffic-forwarding decisions for traffic entering a LAG bundle or for traffic exiting a switch when ECMP is enabled.</p> <p>For LAG bundles, the hashing algorithm determines how traffic entering a LAG bundle is placed onto the bundle's member links. The hashing algorithm tries to manage bandwidth by evenly load-balancing all incoming traffic across the member links in the bundle.</p> <p>When ECMP is enabled, the hashing algorithm determines how incoming traffic is forwarded to the next-hop device.</p> <p>On QFX10002 and QFX 10008 switches, you can configure the hash seed for load balancing.</p> <p>By default, the QFX10002 and QFX10008 switches use the system MAC address to generate a hash seed value. You can configure the hash seed value using the <b>hash-seed</b> statement at the [edit forwarding-options enhanced-hash-key] hierarchy level. Set a value between 0 and 4294967295. If you do not configure a hash seed value, the system will generate a hash seed value based on the system MAC address.</p> <p>The remaining statements are explained separately.</p>
<b>Required Privilege Level</b>	<p>interface—To view this statement in the configuration.</p> <p>interface-control—To add this statement to the configuration.</p>

- Related Documentation**
- *Configuring the Fields in the Algorithm Used To Hash LAG Bundle and ECMP Traffic (CLI Procedure)*
  - *Understanding the Algorithm Used to Hash LAG Bundle and Egress Next-Hop ECMP Traffic*

## fabric-load-balance

---

<b>Syntax</b>	<pre>fabric-load-balance {     flowlet {         inactivity-interval interval;     }     per-packet; }</pre>
<b>Hierarchy Level</b>	[edit forwarding-options <b>enhanced-hash-key</b> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 14.1X53-D10.
<b>Description</b>	<p>Enable adaptive load balancing (ALB) for a VCF, and specify how ALB is implemented.</p> <p>When ALB is enabled, the Virtual Chassis ports (VCPs) are reset. Packets are dropped and might potentially arrive out of order for a brief period of time as a result of this VCP reset. Normal operation of the VCF resumes after the VCP reset with no further user action.</p>
<b>Default</b>	<p>ALB is disabled, by default.</p> <p>If you do not specify a mode when enabled ALB, ALB is enabled using flowlet mode with an inactivity timer of 16 microseconds.</p>
<b>Options</b>	<p><b>flowlet</b>—Implement ALB by using flowlets.</p> <p>When ALB is implemented using flowlets, traffic flows that enter the VCF are spliced into smaller flows—flowlets—and individually forwarded across the VCF to the same destination device over different paths when the inactivity time between packet bursts on the sending interface exceeds the user-configurable inactivity interval.</p> <p>The inactivity interval is 16 microseconds by default, and can be configured using the <b>inactivity-interval</b> statement. You should configure the inactivity interval to ensure in-order packet delivery, so that overall performance is not negatively impacted by the packet reordering process at the receiving device. To ensure in-order packet delivery, the inactivity interval should be larger than the largest latency skew among all the paths in the VCF from any node to any other node.</p> <p>Implementing ALB using flowlets is especially effective in environments that periodically experience extremely large traffic flows—<i>elephant flows</i>—that are substantially larger than the majority of other traffic flowing through the VCF. The VCF is better able to manage elephant flows by splicing them into smaller flowlets using ALB.</p> <p><b>per-packet</b>—Implement ALB using per-packet mode.</p> <p>When per-packet mode is enabled, the VCF forwarding algorithm dynamically monitors all paths in the VCF and forwards packets to destination devices using the best available path at that moment. Flows are reordered at the destination node when per-packet mode is used to enable ALB, so some performance impact due to packet reordering is experienced.</p>

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

**Related Documentation**

- [Understanding Traffic Flow Through a Virtual Chassis Fabric on page 20](#)

## fabric-tree-root

**Syntax** fabric-tree-root;

**Hierarchy Level** [edit **virtual-chassis**] **member** *member-id*

**Release Information** Statement introduced in Junos OS Release 14.1X53-D35 for EX Series and QFX Series devices in a Virtual Chassis Fabric (VCF).

**Description** Optionally specify a member of a VCF to be a root node for creating the shared multicast distribution trees (MDTs) used in forwarding broadcast, unknown unicast, and multicast (BUM) traffic within the VCF. To configure this option, the VCF must be configured as **auto-provisioned** or **preprovisioned**.



**NOTE:** This option only affects MDTs used for directing traffic internally between member nodes of the VCF, and does not affect how BUM traffic is managed on the network outside the VCF.

Setting this option affects BUM traffic routing behavior within the VCF by changing how VCF MDT trees are created. By default, if this option is not configured for any VCF members, every member switch builds an MDT with itself as the root. When this option is configured for one or more VCF members, MDTs are only created with the specified member or members as root nodes. The number of trees created corresponds to the configured number of root nodes.

If you want to use this option in a VCF, we recommend configuring *all* the spine devices in the VCF, and *only* spine devices, as fabric tree roots. Configuring multiple spine devices as fabric tree roots prevents inadvertently reverting to the default behavior if a spine member becomes unavailable. Avoiding leaf members as fabric tree roots helps to prevent traffic interruption during load rebalancing when a leaf node goes offline or is reset.

See “[Understanding Traffic Flow Through a Virtual Chassis Fabric](#)” on page 20 for more information about how this option affects traffic flow.

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

**Related Documentation**

- [Autoprovisioning a Virtual Chassis Fabric on page 23](#)
- [Preprovisioning a Virtual Chassis Fabric on page 27](#)
- [Adding a Device to a Virtual Chassis Fabric on page 34](#)

## id

---

<b>Syntax</b>	<code>id id;</code>
<b>Hierarchy Level</b>	[edit <a href="#">virtual-chassis</a> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 9.3 for EX Series switches. Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
<b>Description</b>	Configure the alphanumeric string that identifies a Virtual Chassis or Virtual Chassis Fabric (VCF) configuration.
<b>Options</b>	<i>id</i> —Virtual Chassis ID (VCID), which uses the ISO family address format—for example, <b>9622.6ac8.5345</b> .
<b>Required Privilege Level</b>	system—To view this statement in the configuration. system-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <i>Example: Assigning the Virtual Chassis ID to Determine Precedence During an EX4200 Virtual Chassis Merge</i></li><li>• <i>Assigning the Virtual Chassis ID to Determine Precedence During a Virtual Chassis Merge (CLI Procedure)</i></li><li>• <i>Configuring a QFX Series Virtual Chassis (CLI Procedure)</i></li><li>• <a href="#">Autoprovisioning a Virtual Chassis Fabric on page 23</a></li><li>• <a href="#">Preprovisioning a Virtual Chassis Fabric on page 27</a></li><li>• <i>Configuring an EX8200 Virtual Chassis (CLI Procedure)</i></li><li>• <i>Understanding Virtual Chassis Member ID Numbering in an EX8200 Virtual Chassis</i></li></ul>

## inactivity-interval (Fabric Load Balance)

<b>Syntax</b>	<code>inactivity-interval <i>interval</i>;</code>
<b>Hierarchy Level</b>	[edit forwarding-options <a href="#">enhanced-hash-key fabric-load-balance</a> flowlet]
<b>Release Information</b>	Statement introduced in Junos OS Release 14.1X53-D10.
<b>Description</b>	<p>Set the inactivity interval for adaptive load balancing (ALB) using flowlets within a VCF.</p> <p>The inactivity interval is the amount of time that occurs between packet bursts on a sending interface before a traffic flow is spliced into smaller traffic flows—flowlets—when ALB is implemented using flowlets. The flowlets are then individually forwarded across the VCF to the same destination device over different paths.</p> <p>You should configure the inactivity interval to ensure in-order packet delivery, so that overall performance is not negatively impacted by the packet re-ordering process at the receiving device. To ensure in-order packet delivery, the inactivity interval should be larger than the largest latency skew among all the paths in the VCF from any node to any other node.</p>
<b>Default</b>	<p>ALB is disabled, by default.</p> <p>If ALB is enabled without specifying a mode, ALB is enabled using flowlet mode with an inactivity interval of 16 microseconds.</p> <p>If ALB is enabled using flowlet mode without specifying an inactivity interval, the inactivity interval is set to 16 microseconds.</p>
<b>Options</b>	<p><b><i>interval</i></b>—The amount of time that occurs between packet bursts on a sending interface before a traffic flow is spliced into flowlets.</p> <p><b>Range:</b> 16 microseconds (<b>16us</b>) to 32 milliseconds(<b>32ms</b>).</p>
<b>Required Privilege Level</b>	<p>system—To view this statement in the configuration.</p> <p>system-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Understanding Traffic Flow Through a Virtual Chassis Fabric on page 20</a></li> </ul>

## location (Virtual Chassis)

---

<b>Syntax</b>	<code>location location;</code>
<b>Hierarchy Level</b>	[edit <a href="#">virtual-chassis member member-id</a> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 11.1 for EX Series switches. Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
<b>Description</b>	<p>Set a description of the location of the Virtual Chassis or VCF member switch or external Routing Engine.</p> <p>The <b>Location</b> field is visible to users who enter the <b>show virtual-chassis status detail</b> command.</p> <p>Setting this description has no effect on the operation of the member device.</p>
<b>Options</b>	<b>location</b> —Location of the current member switch or external Routing Engine. The <b>location</b> can be any single word.
<b>Required Privilege Level</b>	system—To view this statement in the configuration. system-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Autoprovisioning a Virtual Chassis Fabric on page 23</a></li><li>• <a href="#">Preprovisioning a Virtual Chassis Fabric on page 27</a></li><li>• <i>Configuring a QFX Series Virtual Chassis (CLI Procedure)</i></li><li>• <i>Example: Configuring an EX4200 Virtual Chassis Using a Preprovisioned Configuration File</i></li><li>• <i>Example: Configuring a Preprovisioned Mixed EX4200 and EX4500 Virtual Chassis</i></li><li>• <i>Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines</i></li><li>• <i>Configuring an EX4200, EX4500, or EX4550 Virtual Chassis (CLI Procedure)</i></li><li>• <i>Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches (CLI Procedure)</i></li><li>• <i>Configuring an EX8200 Virtual Chassis (CLI Procedure)</i></li></ul>



## mac-persistence-timer

<b>Syntax</b>	<code>mac-persistence-timer [<i>minutes</i>   <b>disable</b>];</code>
<b>Hierarchy Level</b>	[edit <a href="#">virtual-chassis</a> ]
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Option <b>disable</b> introduced in Junos OS Release 12.2 for EX Series switches.</p> <p>The maximum timer limit changed from no maximum timer limit to 60 minutes in Junos OS Release 12.2 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).</p>
<b>Description</b>	<p>Specify how long the Virtual Chassis or VCF continues to use the MAC address of the switch that was originally configured in the master role as the system MAC base address after the original master switch is removed from the Virtual Chassis or VCF. The system MAC base address does not change in the event of a switchover provided the switch originally configured in the master role remains a member of the Virtual Chassis or VCF.</p> <p>The maximum timer limit is 60 minutes starting in Junos OS Release 12.2. There are no minimum or maximum timer limits in prior Junos OS releases.</p>
<b>Default</b>	The MAC persistence timer is set to 10 minutes by default.
<b>Options</b>	<p><b>minutes</b>—Time in minutes that the member switch in the backup role continues to use the system MAC base address of the old master before using its own system MAC base address after the switch in the master role is physically disconnected or removed from the Virtual Chassis or VCF.</p> <p><b>disable</b>—Disable the MAC persistence timer. The system MAC base address never changes when the MAC persistence timer is disabled, even when the switch in the master role is physically disconnected or removed from the Virtual Chassis or VCF.</p>
<b>Required Privilege Level</b>	<p><b>system</b>—To view this statement in the configuration.</p> <p><b>system-control</b>—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>Configuring the Timer for the Backup Member to Start Using Its Own MAC Address, as Master of a Virtual Chassis (CLI Procedure)</i></li> <li>• <a href="#">Autoprovisioning a Virtual Chassis Fabric on page 23</a></li> <li>• <a href="#">Preprovisioning a Virtual Chassis Fabric on page 27</a></li> </ul>

## mastership-priority

---

<b>Syntax</b>	<code>mastership-priority <i>number</i>;</code>
<b>Hierarchy Level</b>	[edit <a href="#">virtual-chassis member</a> <i>member-id</i> ]
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Mastership priority option <b>0</b> introduced in Junos OS Release 11.1 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).</p>
<b>Description</b>	<p>The mastership priority value is the most important factor in determining the role of the member switch within a nonprovisioned Virtual Chassis or VCF configuration. Other factors (see <i>Understanding How the Master in a Virtual Chassis Is Elected</i>) also affect the election of the master.</p> <p>The mastership priority value takes the highest precedence in the master election algorithm. The member switch with highest mastership priority assumes the master Routing Engine role of the Virtual Chassis or VCF. Toggling back and forth between master and backup status in failover conditions is undesirable, so we recommend that you assign the same mastership priority value to both the master and the backup. Secondary factors in the master election algorithm determine which of these two members (that is, the two members that are assigned the highest mastership priority value) functions as the master of the Virtual Chassis or VCF.</p> <p>This statement is not used for the EX8200 Virtual Chassis, which determines mastership by external Routing Engine uptime. See <i>Understanding Virtual Chassis Roles in an EX8200 Virtual Chassis</i>.</p> <p>A switch with a mastership priority of <b>0</b> never takes the master or backup role.</p>
<b>Default</b>	128
<b>Options</b>	<p><i>number</i>—Mastership priority value.</p> <p><b>Range:</b> 0 through 255</p>
<b>Required Privilege Level</b>	<p>system—To view this statement in the configuration.</p> <p>system-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Autoprovisioning a Virtual Chassis Fabric on page 23</a></li><li>• <a href="#">Preprovisioning a Virtual Chassis Fabric on page 27</a></li><li>• <a href="#">Configuring a QFX Series Virtual Chassis (CLI Procedure)</a></li><li>• <a href="#">Configuring an EX4300 Virtual Chassis (CLI Procedure)</a></li><li>• <a href="#">Example: Configuring an EX3300 Virtual Chassis with a Master and Backup</a></li><li>• <a href="#">Example: Configuring an EX4200 Virtual Chassis with a Master and Backup in a Single Wiring Closet</a></li></ul>

- *Example: Configuring an EX4200 Virtual Chassis Interconnected Across Multiple Wiring Closets*
- *Configuring an EX4200, EX4500, or EX4550 Virtual Chassis (CLI Procedure)*

## member

---

<b>Syntax</b>	<pre>member <i>member-id</i> {     fabric-tree-root;     location <i>location</i>;     mastership-priority <i>number</i>;     no-management-vlan;     serial-number <i>serial-number</i>;     role <i>role</i>; }</pre>
<b>Hierarchy Level</b>	[edit <a href="#">virtual-chassis</a> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
<b>Description</b>	Configure a switch or an XRE200 External Routing Engine as a member of a Virtual Chassis or a Virtual Chassis Fabric (VCF) with characteristics specified by the available options.
<b>Default</b>	<p>When an EX Series switch or a QFX Series devices configured in standalone mode is powered on but not interconnected through its Virtual Chassis ports (VCPs) with other member switches, its default member ID is 0.</p> <p>There is no default member ID in an EX8200 or EX9200 Virtual Chassis. An EX8200 or EX9200 Virtual Chassis must be preprovisioned, and that process configures the member IDs.</p>
<b>Options</b>	<p><b><i>member-id</i></b>—Identifies a specific member switch of a Virtual Chassis or VCF configuration.</p> <p>The exact range for a specific Virtual Chassis or VCF depends on the number of switches allowed in the Virtual Chassis or VCF.</p> <p>In an EX8200 Virtual Chassis, member IDs 0 through 7 are reserved for EX8200 member switches and member IDs 8 and 9 are reserved for the master and backup external Routing Engines.</p> <p>The remaining statement options set characteristics of the Virtual Chassis or VCF member, and are explained separately.</p>
<b>Required Privilege Level</b>	system—To view this statement in the configuration. system-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Autoprovisioning a Virtual Chassis Fabric on page 23</a></li><li>• <a href="#">Preprovisioning a Virtual Chassis Fabric on page 27</a></li><li>• <a href="#">Adding a Device to a Virtual Chassis Fabric on page 34</a></li><li>• <a href="#">Configuring a QFX Series Virtual Chassis (CLI Procedure)</a></li></ul>

- *Example: Configuring an EX4200 Virtual Chassis Using a Preprovisioned Configuration File*
- *Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines*
- *Configuring an EX3300 Virtual Chassis (CLI Procedure)*
- *Configuring an EX4200, EX4500, or EX4550 Virtual Chassis (CLI Procedure)*
- *Configuring an EX8200 Virtual Chassis (CLI Procedure)*
- *Configuring an EX9200 Virtual Chassis*
- *Configuring a QFX Series Virtual Chassis (CLI Procedure)*

## no-management-vlan

<b>Syntax</b>	no-management-vlan;
<b>Hierarchy Level</b>	[edit <a href="#">virtual-chassis member member-id</a> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
<b>Description</b>	<p>Remove the specified member's out-of-band management port from the virtual management Ethernet (VME) global management VLAN of the Virtual Chassis or VCF configuration.</p> <p>For a member that is functioning in a linecard role, you can use this configuration to reserve the member's management Ethernet port for local troubleshooting:</p> <pre>virtual-chassis {   member 2 {     no-management-vlan;   } }</pre> <p>You cannot configure the IP address for a local management Ethernet port using the CLI or the J-Web interface. To do this, you need to use the shell <b>ifconfig</b> command.</p>
<b>Required Privilege Level</b>	<p>system—To view this statement in the configuration.</p> <p>system-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>Example: Setting Up a Multimember EX4200 Virtual Chassis Access Switch with a Default Configuration</i></li> <li>• <i>Configuring the Virtual Management Ethernet Interface for Global Management of an EX Series Virtual Chassis (CLI Procedure)</i></li> <li>• <i>Understanding Global Management of a Virtual Chassis</i></li> <li>• <a href="#">Understanding Virtual Chassis Fabric Configuration on page 13</a></li> </ul>

## no-split-detection

---

<b>Syntax</b>	no-split-detection;
<b>Hierarchy Level</b>	[edit <a href="#">virtual-chassis</a> ]
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 9.3 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).</p>
<b>Description</b>	<p>Disable the split and merge feature in a Virtual Chassis or VCF configuration.</p> <p>We recommend using this statement to disable the split and merge feature when configuring a two-member Virtual Chassis. Enabling this statement on a two-member Virtual Chassis ensures that both switches remain in the correct Virtual Chassis roles in the event of a Virtual Chassis split.</p> <p>The split and merge feature is enabled by default when a Virtual Chassis is initially set up. If you set this option in a two-member Virtual Chassis and then expand the Virtual Chassis to add more members, we recommend deleting this configuration item to re-enable the split and merge feature again.</p>
<b>Default</b>	The split and merge feature is enabled.
<b>Required Privilege Level</b>	<p>system—To view this statement in the configuration.</p> <p>system-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <i>Example: Assigning the Virtual Chassis ID to Determine Precedence During an EX4200 Virtual Chassis Merge</i></li><li>• <i>Disabling Split and Merge in a Virtual Chassis (CLI Procedure)</i></li><li>• <i>Assigning the Virtual Chassis ID to Determine Precedence During a Virtual Chassis Merge (CLI Procedure)</i></li><li>• <i>Disabling Split Detection in a Virtual Chassis Configuration</i></li><li>• <i>Understanding Split and Merge in a Virtual Chassis</i></li></ul>

## package-name

---

<b>Syntax</b>	<code>package-name <i>package-name</i>;</code>
<b>Hierarchy Level</b>	[edit virtual-chassis <a href="#">auto-sw-update</a> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 10.0 for EX Series switches. Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
<b>Description</b>	Specify the software package name or location of the software package to be used by the automatic software update feature for Virtual Chassis or VCF.
<b>Default</b>	No package name is specified.
<b>Options</b>	<p><b><i>package-name</i></b>—Name of the software package or the URL to the software package to be used.</p> <ul style="list-style-type: none"> <li>If the software package is located on a local directory on the switch, use the following format for <b><i>package-name</i></b>:  <b><i>/pathname/package-name</i></b></li> <li>If the software package is to be downloaded and installed from a remote location, use one of the following formats:  <b><i>ftp://hostname/pathname/package-name</i></b> <b><i>ftp://username:prompt@ftp.hostname.net/package-name</i></b> <b><i>http://hostname/pathname/package-name</i></b></li> </ul>
<b>Required Privilege Level</b>	<p>system—To view this statement in the configuration.</p> <p>system-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li><i>Example: Configuring Automatic Software Update on EX4200 Virtual Chassis Member Switches</i></li> <li><i>Configuring Automatic Software Update on Virtual Chassis Member Switches (CLI Procedure)</i></li> <li><a href="#">Understanding Software Upgrades in a Virtual Chassis Fabric on page 22</a></li> </ul>

## preprovisioned

---

<b>Syntax</b>	preprovisioned;
<b>Hierarchy Level</b>	[edit <a href="#">virtual-chassis</a> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
<b>Description</b>	<p>Enable the preprovisioned configuration mode for a Virtual Chassis or Virtual Chassis Fabric (VCF) configuration.</p> <p>When the preprovisioned configuration mode is enabled, you cannot use the CLI or the J-Web interface to change the mastership priority or member ID of member switches.</p> <p>You must use this statement to configure an EX8200 Virtual Chassis. Nonprovisioned configuration of an EX8200 Virtual Chassis is not supported.</p>
<b>Required Privilege Level</b>	system—To view this statement in the configuration. system-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Preprovisioning a Virtual Chassis Fabric on page 27</a></li><li>• <i>Example: Configuring an EX4200 Virtual Chassis Using a Preprovisioned Configuration File</i></li><li>• <i>Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines</i></li><li>• <i>Configuring an EX4200, EX4500, or EX4550 Virtual Chassis (CLI Procedure)</i></li><li>• <i>Configuring an EX8200 Virtual Chassis (CLI Procedure)</i></li><li>• <i>Configuring an EX9200 Virtual Chassis</i></li><li>• <i>Configuring a QFX Series Virtual Chassis (CLI Procedure)</i></li><li>• <i>Replacing a Member Switch of a Virtual Chassis Configuration (CLI Procedure)</i></li></ul>



## role

<b>Syntax</b>	<code>role (line-card   routing-engine);</code>
<b>Hierarchy Level</b>	[edit <b>virtual-chassis</b> <b>preprovisioned member</b> <i>member-id</i> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
<b>Description</b>	Specify the roles of the members of the Virtual Chassis or a Virtual Chassis Fabric (VCF) in a preprovisioned Virtual Chassis.

### Virtual Chassis Fabric

Specify the role to be performed by each switch. In a VCF, two spine devices are configured into the Routing Engine role and the remaining spine devices and all leaf devices are configured into the line card role. The role must be associated with the member's serial number.

### EX Series (except EX8200 Virtual Chassis) and QFX Series Virtual Chassis

Specify the role to be performed by each member switch. Associate the role with the member's serial number.

When you use a preprovisioned configuration, you cannot modify the mastership priority or member ID of member switches through the user interfaces. The mastership priority value is generated by the software, based on the assigned role:

- A member configured as **routing-engine** is assigned the mastership priority **129**.
- A member configured as **line-card** is assigned the mastership priority **0**.
- A member listed in the preprovisioned configuration without an explicitly specified role is assigned the mastership priority **128**.

The configured role specifications are permanent. If both **routing-engine** members fail, a **line-card** member cannot take over as master of the Virtual Chassis configuration. You must delete the preprovisioned configuration to change the specified roles in a Virtual Chassis.

Explicitly configure two members as **routing-engine** and configure additional switches as members of the preprovisioned Virtual Chassis by specifying only their serial numbers. If you do not explicitly configure the role of the additional members, they function in a linecard role by default. In that case, a member that is functioning in a linecard role can take over mastership if the members functioning as master and backup (**routing-engine** role) both fail.

### EX8200 Virtual Chassis

Specify the role to be performed by each XRE200 External Routing Engine and each EX8200 member switch. Associate the role with the member's serial number. An EX8200

Virtual Chassis cannot function when both external Routing Engines, which must be configured in the **routing-engine** role, have failed.

- Options**
- **line-card**—Enables the member to be eligible to function only in the linecard role. Any member of the Virtual Chassis or VCF configuration other than the master or backup functions in the linecard role and runs only a subset of Junos OS for EX Series switches. A member functioning in the linecard role does not run the control protocols or the chassis management processes.

A Virtual Chassis must have at least three members for one member to function in the linecard role.

In an EX8200 Virtual Chassis configuration, all member switches must be in the linecard role.

- **routing-engine**—Enables the member to function as a master or backup of the Virtual Chassis or VCF configuration. The master manages all members and runs the chassis management processes and control protocols. The backup synchronizes with the master in terms of protocol states, forwarding tables, and so forth, so that it is prepared to preserve routing information and maintain network connectivity without disruption in case the master is unavailable.

(All Virtual Chassis composed of EX Series switches, except EX8200 switches, or QFX Series devices) Specify two and only two members as **routing-engine**. The software determines which of the two members assigned the **routing-engine** role functions as master, based on the master election algorithm. See *Understanding How the Master in a Virtual Chassis Is Elected*. In these Virtual Chassis, the **routing-engine** role is associated with a switch.

(EX8200 Virtual Chassis) All XRE200 External Routing Engines must be in the **routing-engine** role.

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

- Related Documentation**
- [Autoprovisioning a Virtual Chassis Fabric on page 23](#)
  - [Preprovisioning a Virtual Chassis Fabric on page 27](#)
  - *Example: Configuring an EX4200 Virtual Chassis Using a Preprovisioned Configuration File*
  - *Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines*
  - *Configuring an EX3300 Virtual Chassis (CLI Procedure)*
  - *Configuring an EX4200, EX4500, or EX4550 Virtual Chassis (CLI Procedure)*
  - *Configuring an EX8200 Virtual Chassis (CLI Procedure)*
  - *Configuring an EX9200 Virtual Chassis*
  - *Configuring a QFX Series Virtual Chassis (CLI Procedure)*
  - *Configuring a Virtual Chassis on an EX Series Switch (J-Web Procedure)*
  - *Adding a New EX4200 Switch to an Existing EX4200 Virtual Chassis (CLI Procedure)*
  - *Replacing a Member Switch of a Virtual Chassis Configuration (CLI Procedure)*

## serial-number

---

<b>Syntax</b>	<code>serial-number serial-number;</code>
<b>Hierarchy Level</b>	[edit <a href="#">virtual-chassis preprovisioned member member-id</a> ]
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).</p>
<b>Description</b>	<p>In a preprovisioned Virtual Chassis or Virtual Chassis Fabric (VCF), specify the serial number of each member switch to be included in the configuration. If you do not include the serial number within the configuration, the switch cannot be recognized as a member of a preprovisioned configuration.</p> <p>In an EX8200 Virtual Chassis configuration, specify the serial number of each XRE200 External Routing Engine and each EX8200 member switch to be included in the Virtual Chassis configuration. If you do not include the serial number within the Virtual Chassis configuration, the external Routing Engine or switch cannot be recognized as a member of the configuration.</p>
<b>Options</b>	<b>serial-number</b> —Permanent serial number for the external Routing Engine or for the member switch.
<b>Required Privilege Level</b>	<p>system—To view this statement in the configuration.</p> <p>system-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Autoprovisioning a Virtual Chassis Fabric on page 23</a></li><li>• <a href="#">Preprovisioning a Virtual Chassis Fabric on page 27</a></li><li>• <a href="#">Configuring an EX2200 Virtual Chassis (CLI Procedure)</a></li><li>• <a href="#">Configuring an EX3300 Virtual Chassis (CLI Procedure)</a></li><li>• <a href="#">Configuring an EX4300 Virtual Chassis (CLI Procedure)</a></li><li>• <a href="#">Configuring an EX4200, EX4500, or EX4550 Virtual Chassis (CLI Procedure)</a></li><li>• <a href="#">Configuring an EX8200 Virtual Chassis (CLI Procedure)</a></li><li>• <a href="#">Configuring an EX9200 Virtual Chassis</a></li><li>• <a href="#">Configuring a QFX Series Virtual Chassis (CLI Procedure)</a></li><li>• <a href="#">Configuring a Virtual Chassis on an EX Series Switch (J-Web Procedure)</a></li></ul>

## serial-number (Virtual Chassis aliases)

---

<b>Syntax</b>	<pre>serial-number <i>serial-number</i> {     <i>alias-name</i> <i>alias-name</i>; }</pre>
<b>Hierarchy Level</b>	[edit <a href="#">virtual-chassis aliases</a> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 14.1X53-D10 for EX Series and QFX Series Virtual Chassis and Virtual Chassis Fabric (VCF).
<b>Description</b>	<p>Specify the serial number that will be labeled with an alias in a Virtual Chassis or Virtual Chassis Fabric (VCF).</p> <p>The remaining statements are explained separately.</p>
<b>Options</b>	<p><b><i>serial-number</i></b>—Permanent serial number for the member switch in the Virtual Chassis or VCF.</p> <p>You can retrieve the serial number for any device in your Virtual Chassis or VCF by entering the <b>show virtual-chassis</b> command and reviewing the output in the <b>Serial No</b> field.</p>
<b>Required Privilege Level</b>	<p>system—To view this statement in the configuration.</p> <p>system-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Autoprovisioning a Virtual Chassis Fabric on page 23</a></li> <li>• <a href="#">Preprovisioning a Virtual Chassis Fabric on page 27</a></li> <li>• <a href="#">Configuring a QFX Series Virtual Chassis (CLI Procedure)</a></li> <li>• <a href="#">Understanding Virtual Chassis Fabric Components on page 5</a></li> <li>• <a href="#">Understanding QFX Series Virtual Chassis Components</a></li> </ul>

## traceoptions (Virtual Chassis)

**Syntax** traceoptions {  
     file *filename* <files *number*> <no-stamp> <replace> <size *size*> <world-readable |  
     no-world-readable>;  
     flag *flag* <detail> <disable> <receive> <send>;  
 }

**Hierarchy Level** [edit [virtual-chassis](#)]

**Release Information** Statement introduced in Junos OS Release 9.0 for EX Series switches.  
 Option **detail** added in Junos OS Release 9.2 for EX Series switches.  
 Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series.  
 Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).

**Description** Define tracing operations for the Virtual Chassis or VCF.

Each Virtual Chassis or VCF member is updated with configured tracing options, and the log file is stored locally on each device.



**NOTE:** In a large-scale VCF, use system logging and tracing with discretion, and only for troubleshooting. These operations place an extra load on the master Routing Engine device, which can impact VCF convergence time and stability. All tracing options should be disabled during normal VCF operation. To troubleshoot particular problems, selectively enable tracing options, and disable them again after collecting the desired information.

**Default** Tracing operations are disabled.

**Options** **detail**—(Optional) Generate detailed trace information for a flag.



**NOTE:** Enable tracing at the detail level only while troubleshooting a particular issue, and disable it again for normal system operation.

**disable**—(Optional) Disable a flag.

**file *filename***—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory **/var/log**.

**files *number***—(Optional) Maximum number of trace files. When a trace file named **trace-file** reaches its maximum size, it is renamed **trace-file.0**, then **trace-file.1**, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten. If you specify a maximum number of files, you also must specify a maximum file size with the **size** option.

**Range:** 2 through 1000

**Default:** 3 files

**flag *flag***—Tracing operation to perform. To specify more than one tracing operation, include multiple flag statements. You can include the following flags:

- **all**—All tracing operations.



**TIP:** The **all** flag displays a subset of logs that are generally useful in debugging issues, and for more detailed information, you can use options **all detail**. However, to avoid significantly impacting VCF stability, use this high level of tracing only for troubleshooting for a short period of time, and not during normal VCF operation.

- **auto-configuration**—Trace Virtual Chassis ports (VCPs) that have been automatically configured.
- **csn**—Trace Virtual Chassis complete sequence number (CSN) packets.
- **error**—Trace Virtual Chassis errored packets.
- **hello**—Trace Virtual Chassis hello packets.
- **krt**—Trace Virtual Chassis KRT events.
- **lsp**—Trace Virtual Chassis link-state packets.
- **lsp-generation**—Trace Virtual Chassis link-state packet generation.
- **me**—Trace Virtual Chassis ME events.
- **normal**—Trace normal events.
- **packets**—Trace Virtual Chassis packets.
- **parse**—Trace reading of the configuration.
- **psn**—Trace partial sequence number (PSN) packets.
- **route**—Trace Virtual Chassis routing information.
- **spf**—Trace Virtual Chassis SPF events.
- **state**—Trace Virtual Chassis state transitions.
- **task**—Trace Virtual Chassis task operations.

**no-stamp**—(Optional) Do not place a timestamp on any trace file.

**no-world-readable**—(Optional) Restrict file access to the user who created the file.

**receive**—(Optional) Trace received packets.

**replace**—(Optional) Replace a trace file rather than appending information to it.

**send**—(Optional) Trace transmitted packets.

**size size**—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named **trace-file** reaches its maximum size, it is renamed **trace-file.0**, then **trace-file.1**, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten. If you specify a maximum number of files, you also must specify a maximum file size with the **files** option.

**Syntax:** **xk** to specify KB, **xm** to specify MB, or **xg** to specify GB

**Range:** 10 KB through 1 GB

**Default:** 128 KB

**world-readable**—(Optional) Enable unrestricted file access.

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

<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <i>Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis</i></li><li>• <i>Verifying the Member ID, Role, and Neighbor Member Connections of a Virtual Chassis Member</i></li><li>• <i>Verifying That Virtual Chassis Ports Are Operational</i></li><li>• <i>Verifying Virtual Chassis Ports in an EX8200 Virtual Chassis</i></li><li>• <i>Troubleshooting an EX Series Virtual Chassis</i></li><li>• <a href="#">Troubleshooting Virtual Chassis Fabric on page 47</a></li></ul>
------------------------------	--



## vcp-snmp-statistics

---

<b>Syntax</b>	vcp-snmp-statistics;
<b>Hierarchy Level</b>	[edit <a href="#">virtual-chassis</a> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 15.1 for EX Series switches. Statement introduced in Junos OS Release 15.1 for the QFX Series.
<b>Description</b>	<p>Enable SNMP monitoring of the Virtual Chassis ports (VCPs) for all VCPs in the Virtual Chassis or Virtual Chassis Fabric (VCF).</p> <p>When this statement is enabled, SNMP gathers statistics on the Junos VCP MIBs. You can retrieve the statistics gathered by SNMP for these MIBs by using the <a href="#">show snmp mib</a> command with the <b>walk</b> and <b>ascii</b> options and specifying <b>jnxVirtualChassisPortInPkts</b>.</p>
<b>Default</b>	SNMP is disabled by default on devices running Junos OS.
<b>Required Privilege Level</b>	system—To view this statement in the configuration. system-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>Understanding EX4300 Virtual Chassis</i></li> <li>• <i>Understanding QFX Series Virtual Chassis</i></li> <li>• <i>Understanding EX Series Virtual Chassis Components</i></li> <li>• <i>Understanding QFX Series Virtual Chassis Components</i></li> </ul>

## virtual-chassis

```
Syntax virtual-chassis {
    aliases {
        serial-number serial-number {
            alias-name alias-name;
        }
    }
    auto-provisioned;
    auto-sw-update {
        (ex-4200 | ex-4300 | ex-4500 | ex-4600 | qfx-3 | qfx-5)
        package-name package-name;
    }
    fast-failover (ge | vcp disable | xe);
    graceful-restart {
        disable;
    }
    id id;
    mac-persistence-timer seconds;
    member member-id {
        fabric-tree-root;
        location location;
        mastership-priority number;
        no-management-vlan;
        serial-number;
        role;
    }
    no-split-detection;
    preprovisioned;
    traceoptions (Virtual Chassis) {
        file filename <files number> <size size> <world-readable | no-world-readable> <match
            regex>;
        flag flag ;
    }
    vc-port {
        lag-hash (packet-based | source-port-based);
    }
    vcp-no-hold-time;
    vcp-snmp-statistics;
}
```

Hierarchy Level [edit]

**Release Information** Statement introduced in Junos OS Release 9.0 for EX Series switches.  
Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series.  
Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).

**Description** Configure a Virtual Chassis or a Virtual Chassis Fabric (VCF).

The remaining statements are explained separately.

**Default** A standalone EX Series switch is a Virtual Chassis by default. It has a default member ID of 0, a default mastership priority of 128, and a default role as master.

A QFX Series device configured in standalone mode is a Virtual Chassis by default. It has a default member ID of 0, a default mastership priority of 128, and a default role as master.

A standalone XRE200 External Routing Engine or EX8200 switch is not part of an EX8200 Virtual Chassis until a Virtual Chassis configuration is set up.

<b>Required Privilege Level</b>	system—To view this statement in the configuration. system-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Autoprovisioning a Virtual Chassis Fabric on page 23</a></li><li>• <a href="#">Preprovisioning a Virtual Chassis Fabric on page 27</a></li><li>• <a href="#">Adding a Device to a Virtual Chassis Fabric on page 34</a></li><li>• <i>Configuring a QFX Series Virtual Chassis (CLI Procedure)</i></li><li>• <i>Example: Configuring an EX3300 Virtual Chassis with a Master and Backup</i></li><li>• <i>Example: Configuring an EX4200 Virtual Chassis with a Master and Backup in a Single Wiring Closet</i></li><li>• <i>Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines</i></li><li>• <i>Configuring an EX3300 Virtual Chassis (CLI Procedure)</i></li><li>• <i>Configuring an EX4200, EX4500, or EX4550 Virtual Chassis (CLI Procedure)</i></li><li>• <i>Configuring an EX8200 Virtual Chassis (CLI Procedure)</i></li><li>• <i>Configuring an EX9200 Virtual Chassis</i></li></ul>



## CHAPTER 3

# Operational Commands for Virtual Chassis Fabric

- clear virtual-chassis vc-port statistics
- request session member
- request virtual-chassis mode
- request virtual-chassis reactivate
- request virtual-chassis vc-port
- request virtual-chassis vc-port diagnostics optics
- show forwarding-options enhanced-hash-key
- show snmp mib
- show virtual-chassis active-topology
- show virtual-chassis device-topology
- show virtual-chassis login
- show virtual-chassis mode
- show virtual-chassis protocol adjacency
- show virtual-chassis protocol database
- show virtual-chassis protocol interface
- show virtual-chassis protocol route
- show virtual-chassis protocol statistics
- show virtual-chassis
- show virtual-chassis vc-path
- show virtual-chassis vc-port
- show virtual-chassis vc-port diagnostics optics
- show virtual-chassis vc-port statistics

## clear virtual-chassis vc-port statistics

---

<b>Syntax</b>	<code>clear virtual-chassis vc-port statistics</code> <code>&lt;all-members&gt;</code> <code>&lt;interface-name&gt;</code> <code>&lt;local&gt;</code> <code>&lt;member member-id&gt;</code>
<b>Release Information</b>	Command introduced in Junos OS Release 9.0 for EX Series switches. The options <b>all-members</b> and <b>local</b> were added in Junos OS Release 9.3 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric.
<b>Description</b>	Clear—reset to zero (0)—the traffic statistics counters on Virtual Chassis ports (VCPs).
<b>Options</b>	<b>none</b> —Clear traffic statistics for VCPs of all members of a Virtual Chassis or VCF.  <b>all-members</b> —(Optional) Clear traffic statistics for VCPs of all members of a Virtual Chassis or VCF.  <b>interface-name</b> —(Optional) Clear traffic statistics for the specified VCP.  <b>local</b> —(Optional) Clear traffic statistics for VCPs from the switch or external Routing Engine on which this command is entered.  <b>member member-id</b> —(Optional) Clear traffic statistics for VCPs from the specified member of a Virtual Chassis or VCF.
<b>Required Privilege Level</b>	clear
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">show virtual-chassis vc-port statistics on page 161</a></li><li>• <a href="#">show virtual-chassis vc-port on page 143</a></li><li>• <i>Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis</i></li></ul>
<b>List of Sample Output</b>	<a href="#">clear virtual-chassis vc-port statistics (EX4200 Virtual Chassis) on page 86</a> <a href="#">clear virtual-chassis vc-port statistics (EX8200 Virtual Chassis) on page 87</a> <a href="#">clear virtual-chassis vc-port statistics member 3 on page 87</a>

### Sample Output

#### clear virtual-chassis vc-port statistics (EX4200 Virtual Chassis)

```
user@switch> clear virtual-chassis vc-port statistics
fpc0:
-----
Statistics cleared
```

**clear virtual-chassis vc-port statistics (EX8200 Virtual Chassis)**

```
user@external-routing-engine> clear virtual-chassis vc-port statistics
```

```
member0:
```

```
-----  
Statistics cleared
```

```
member1:
```

```
-----  
Statistics cleared
```

```
member8:
```

```
-----  
Statistics cleared
```

```
member9:
```

```
-----  
Statistics cleared
```

**clear virtual-chassis vc-port statistics member 3**

```
user@switch> clear virtual-chassis vc-port statistics member 3
```

```
Cleared statistics on member 3
```

## request session member

---

<b>Syntax</b>	<code>request session member <i>member-id</i></code>
<b>Release Information</b>	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. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
<b>Description</b>	Start a session with the specified member of a Virtual Chassis or a VCF.
<b>Options</b>	<i>member-id</i> —Member ID for the specific member of the Virtual Chassis or VCF.
<b>Required Privilege Level</b>	maintenance
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">member on page 68</a></li><li>• <i>Understanding EX Series Virtual Chassis Components</i></li><li>• <i>Understanding QFX Series Virtual Chassis Components</i></li></ul>



## request virtual-chassis mode

<b>Syntax</b>	<pre>request virtual-chassis mode     fabric     mixed     &lt;disable&gt;     &lt;reboot&gt;     &lt;all-members&gt;     &lt;local&gt;     &lt;member <i>member-id</i>&gt;</pre>
<b>Release Information</b>	<p>Command introduced in Junos OS Release 11.1 for EX Series switches.</p> <p>Command introduced in Junos OS Release 13.2X51-D20 for QFX Series devices.</p> <p>The <b>fabric</b> keyword introduced in Junos OS Release 13.2X51-D20 for EX Series switches and QFX Series devices in a Virtual Chassis Fabric (VCF).</p> <p>Command introduced in Junos OS Release 13.2X51-D20 for VCF.</p>
<b>Description</b>	<p>Configure the mode for a device or multiple devices in a Virtual Chassis or a VCF.</p> <p>A device must be configured in fabric mode to participate as a member device in a VCF.</p> <p>A device must be configured in mixed mode when it is participating in a Virtual Chassis or a VCF with different types of devices.</p> <p>Do not enable the <b>request virtual-chassis mode mixed</b> command for a standalone device or for a member switch that is intended to remain in a non-mixed Virtual Chassis or VCF. Enabling this command reduces the maximum scaling numbers for some features on the switch, Virtual Chassis, or VCF.</p> <p>You do not need to configure mixed mode if the only devices in your Virtual Chassis are EX4500 and EX4550 switches.</p> <p>To avoid potential traffic disruptions and configuration issues for a mixed Virtual Chassis, we recommend configuring mixed mode on your device before cabling it into your Virtual Chassis. We recommend rebooting your device to complete this configuration procedure before interconnecting your device into the Virtual Chassis.</p> <p>To avoid potential traffic disruptions and configuration issues, we recommend configuring the fabric and, if applicable, the mixed mode settings on your device before cabling it into a VCF. We recommend rebooting your device to complete this configuration procedure before interconnecting your device into the VCF. You can change the fabric and mixed mode settings after the device has been added to a Virtual Chassis or VCF, however.</p> <p>If you set some of the devices in a mixed Virtual Chassis or VCF to mixed mode using this command but not others, the mixed Virtual Chassis or VCF might not form. If you experience this issue, enter the <b>request virtual-chassis mode mixed all-members</b> command to set the Virtual Chassis mode to mixed for all devices in the Virtual Chassis or VCF. You then need to reboot the devices that have been set into mixed mode to complete the procedure. The Virtual Chassis or VCF forms after the devices have rebooted.</p> <p>When you do not use this command to set any of the switches in a mixed EX4200 and EX4500 Virtual Chassis to mixed mode, a mixed EX4200 and EX4500 Virtual Chassis</p>

forms with one of the switches assuming the master role if the switches are running Junos OS Release 11.4 or later. All other switches in the mixed EX4200 and EX4500 Virtual Chassis are placed into the linecard role. If you experience this behavior, enter the **request virtual-chassis mode mixed all-members** command to set the Virtual Chassis mode to mixed for all switches in the Virtual Chassis. You will then need to reboot the switches to complete the procedure. The Virtual Chassis will form after all of the switches have rebooted.

The Virtual Chassis mode setting is maintained through reboots even though it is set in operational mode.

- Options**
- none**—Set the Virtual Chassis mode for all members of the Virtual Chassis or VCF.
  - all-members**—(Optional) Set the Virtual Chassis mode for all members of the Virtual Chassis or VCF.
  - disable**—Disable the Virtual Chassis fabric or mixed mode setting if it was previously enabled.
  - fabric**—Set the device into fabric mode so that the device can participate in a VCF.
  - local**—(Optional) Set the Virtual Chassis mode on the member device where the command is issued.
  - member *member-id***—(Optional) Set the Virtual Chassis mode to mixed on the specified member of the Virtual Chassis or VCF.
  - mixed**—Set the device into mixed mode so that the device can participate in a mixed Virtual Chassis or mixed VCF.



**NOTE:** You do not need to set mixed mode if the only devices in your Virtual Chassis are QFX3500 and QFX3600 devices.

You do not need to configure mixed mode if the only devices in your Virtual Chassis are EX4500 and EX4550 switches.

---

**Required Privilege Level** system-control

- Related Documentation**
- [Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\)](#)
  - [Verifying the Virtual Chassis Fabric Mode Settings on page 46](#)
  - [Verifying the Member ID, Role, and Neighbor Member Connections of a Virtual Chassis Member](#)

**List of Sample Output** [request virtual-chassis mode mixed on page 91](#)  
[request virtual-chassis mode fabric mixed reboot on page 91](#)

### Sample Output

request virtual-chassis mode mixed

```
user@switch> request virtual-chassis mode mixed
```

### Sample Output

request virtual-chassis mode fabric mixed reboot

```
user@switch> request virtual-chassis mode fabric mixed reboot
```

## request virtual-chassis reactivate

---

<b>Syntax</b>	<code>request virtual-chassis reactivate</code>
<b>Release Information</b>	Command introduced in Junos OS Release 9.3 for EX Series switches. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
<b>Description</b>	<p>Reactivate a device that has been assigned a member ID but is not currently connected to the Virtual Chassis or VCF.</p> <p>You can use this command to reactivate a device that was previously part of the Virtual Chassis or VCF but whose status is no longer <b>Prsnt</b>.</p>
<b>Required Privilege Level</b>	system-control
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <i>Verifying the Member ID, Role, and Neighbor Member Connections of a Virtual Chassis Member</i></li><li>• <i>Verifying the Member ID, Role, and Neighbor Member Connections of an EX8200 Virtual Chassis Member</i></li></ul>
<b>List of Sample Output</b>	<a href="#">request virtual-chassis reactivate on page 92</a>

### Sample Output

#### request virtual-chassis reactivate

```
user@switch> request virtual-chassis reactivate
```

## request virtual-chassis vc-port

<b>Syntax</b>	<b>request virtual-chassis vc-port</b> [ <b>set</b>   <b>delete</b> ] < <b>fpc-slot</b> <i>fpc-slot</i> > <b>pic-slot</b> <i>pic-slot</i> <b>port</b> <i>port-number</i> < <b>member</b> <i>member-id</i> >
<b>Release Information</b>	<p>Command introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Option <b>fpc-slot</b> introduced in Junos OS Release 10.4 for EX Series switches.</p> <p>Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series.</p> <p>Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).</p>
<b>Description</b>	<p>Enable or disable a port as a Virtual Chassis port (VCP).</p> <p>If you omit <b>member</b> <i>member-id</i>, this command defaults to enabling or disabling the uplink VCP or SFP network port configured as a VCP on the switch where the command is issued.</p> <p>On an EX3300 switch, uplink ports 2 and 3 are configured as VCPs by default. No other uplink ports on any other EX Series switches are configured as VCPs by default.</p> <p>You might experience a temporary traffic disruption immediately after creating or deleting a user-configured VCP in an EX8200 Virtual Chassis.</p>
<b>Options</b>	<p><b>set</b>—Set a network port into a VCP to convert a network port into a VCP.</p> <p><b>delete</b>—Delete the VCP setting on a port to convert a VCP into a network port.</p> <p><b>pic-slot</b> <i>pic-slot</i>—Number of the PIC slot for the port on the switch.</p> <p><b>port</b> <i>port-number</i>—Number of the port that is to be enabled or disabled as a VCP.</p> <p><b>member</b> <i>member-id</i>—(Optional) Enable or disable the specified VCP on the specified member of the Virtual Chassis or VCF.</p>
<b>Required Privilege Level</b>	system-control
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">request virtual-chassis vc-port</a> (dedicated port)</li> <li>• <a href="#">show virtual-chassis vc-port on page 143</a></li> <li>• <a href="#">show virtual-chassis vc-port statistics on page 161</a></li> <li>• <a href="#">clear virtual-chassis vc-port statistics on page 86</a></li> <li>• <i>Virtual Chassis Port (VCP) Interface Names in an EX8200 Virtual Chassis</i></li> <li>• <i>Understanding EX Series Virtual Chassis Components</i></li> <li>• <i>Understanding QFX Series Virtual Chassis Components</i></li> </ul>
<b>List of Sample Output</b>	<p><a href="#">request virtual-chassis vc-port set pic-slot 1 port 0 on page 94</a></p> <p><a href="#">request virtual-chassis vc-port set pic-slot 1 port 1 member 3 on page 94</a></p> <p><a href="#">request virtual-chassis vc-port delete pic-slot 1 port 1 member 3 on page 94</a></p>

## Sample Output

### `request virtual-chassis vc-port set pic-slot 1 port 0`

```
user@switch> request virtual-chassis vc-port set pic-slot 1 port 0
```

To check the results of this command, use the [show virtual-chassis vc-port](#) command.

### `request virtual-chassis vc-port set pic-slot 1 port 1 member 3`

```
user@switch> request virtual-chassis vc-port set pic-slot 1 port 1 member 3
```

To check the results of this command, use the [show virtual-chassis vc-port](#) command.

### `request virtual-chassis vc-port delete pic-slot 1 port 1 member 3`

```
user@switch> request virtual-chassis vc-port delete pic-slot 1 port 1 member 3
```

To check the results of this command, use the [show virtual-chassis vc-port](#) command.

## request virtual-chassis vc-port diagnostics optics

<b>Syntax</b>	<b>request virtual-chassis vc-port diagnostics optics</b>
<b>Release Information</b>	Command introduced in Junos OS Release 13.2X50-D10 for EX Series switches. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
<b>Description</b>	<p>Run a digital optical monitoring (DOM) scan on the optical ports configured as Virtual Chassis ports (VCPs).</p> <p>Enter the <b>show virtual-chassis vc-port diagnostics optics</b> command to view the results of the diagnostic scan.</p> <p>On certain EX Series switches, the <b>request virtual-chassis vc-port diagnostics optics</b> command must be entered to run a diagnostic scan before you can gather the <b>show virtual-chassis vc-port diagnostics optics</b> output.</p>
<b>Required Privilege Level</b>	system-control
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">show virtual-chassis vc-port diagnostics optics on page 147</a></li> </ul>

## Sample Output

### request virtual-chassis vc-port diagnostics optics

```

user@switch> request virtual-chassis vc-port diagnostics optics
fpc0:
-----
vc-port Diagnostics Optics Done

```

## show forwarding-options enhanced-hash-key

<b>Syntax</b>	<b>show forwarding-options enhanced-hash-key</b>
<b>Release Information</b>	<p>Command introduced in Junos OS Release 13.2X51-D15 for EX Series switches.</p> <p>Command introduced in Junos OS Release 13.2X51-D20 for QFX Series devices.</p> <p><b>Fabric Load Balancing Options</b> output fields introduced in Junos OS Release 14.1X53-D10.</p>
<b>Description</b>	<p>Display information about which packet fields are used by the hashing algorithm to make hashing decisions.</p> <p>You can configure the fields that are inspected by the hashing algorithm to make hashing decisions for traffic entering a LAG bundle using the <b>forwarding-options enhanced-hash-key</b> statement.</p>
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>Configuring the Fields in the Algorithm Used To Hash LAG Bundle and ECMP Traffic (CLI Procedure)</i></li> <li>• <i>Understanding the Algorithm Used to Hash LAG Bundle and Egress Next-Hop ECMP Traffic</i></li> <li>• <a href="#">enhanced-hash-key on page 57</a></li> </ul>
<b>List of Sample Output</b>	<p><a href="#">show forwarding-options enhanced-hash-key (Layer 2 Payload Hash Mode) on page 98</a></p> <p><a href="#">show forwarding-options enhanced-hash-key (Layer 2 Header Hash Mode) on page 98</a></p> <p><a href="#">show forwarding-options enhanced-hash-key (Fabric Load Balancing Options) on page 99</a></p> <p><a href="#">show forwarding-options enhanced-hash-key (QFX10002 and QFX 10008 Switches) on page 99</a></p>
<b>Output Fields</b>	<p><a href="#">Table 5 on page 96</a> lists the output fields for the <b>show forwarding-options enhanced-hash-key</b> command. Output fields are listed in the approximate order in which they first appear. Output fields vary by platform.</p>

**Table 5: show forwarding-options enhanced-hash-key Output Fields**

Field Name	Field Description
<b>Hash-Mode</b>	Current hash mode: Layer 2 header or Layer 2 payload.
<b>Protocol</b>	Indicates whether the Protocol field is or is not used by the hashing algorithm: Yes or No.
<b>Destination L4 Port</b>	Indicates whether the Destination L4 Port field is or is not used by the hashing algorithm: Yes or No.
<b>Source L4 Port</b>	Indicates whether the Source L4 Port field is or is not used by the hashing algorithm: Yes or No.



Table 5: show forwarding-options enhanced-hash-key Output Fields (*continued*)

Field Name	Field Description
<b>Destination IPv4 Addr</b>	Indicates whether the Destination IPv4 Addr field is or is not used by the hashing algorithm: Yes or No.
<b>Source IPv4 Addr</b>	Indicates whether the Source IPv4 Addr field is or is not used by the hashing algorithm: Yes or No.
<b>Vlan id</b>	Indicates whether the Vlan ID field is or is not used by the hashing algorithm: Yes or No.
<b>Inner-Vlan ID</b>	Indicates whether the inner Vlan field is or is not used by the hashing algorithm: Yes or No.
<b>Next Hdr</b>	Indicates whether the Next Hdr field is or is not used by the hashing algorithm: Yes or No.
<b>Destination IPv6 Addr</b>	Indicates whether the Destination IPv6 Addr field is or is not used by the hashing algorithm: Yes or No.
<b>Source IPv6 Addr</b>	Indicates whether the Source IPv6 Addr field is or is not used by the hashing algorithm: Yes or No.
<b>Ether Type</b>	Indicates whether the Ether Type field is or is not used by the hashing algorithm: Yes or No.
<b>Destination MAC Address</b>	Indicates whether the Destination MAC Address field is or is not used by the hashing algorithm: Yes or No.
<b>Source MAC Address</b>	Indicates whether the Source MAC Address field is or is not used by the hashing algorithm: Yes or No.
<b>Load Balancing Method</b>	Indicates the load balancing method for adaptive load balancing (ALB): flowlet or per-packet.  The load balancing method is flowlet by default, and can be configured using the <b>fabric-load-balance</b> statement.
<b>Fabric Link Scale</b>	Indicates the fabric link scale, in mbps.
<b>Inactivity Interval</b>	Indicates the fabric load balance inactivity interval, in microseconds (us).  The inactivity interval is 16 microseconds by default, and can be configured using the <b>inactivity-interval</b> statement.
<b>Hash Region Size/Trunk</b>	Indicates the hash region size, in buckets per fabric trunk.
<b>Seed</b>	A hash seed value, between 0 and 4294967295. If a hash-seed value is not configured it is automatically assigned on the QFX10002 and QFX10008 switches. A hash-seed prevents traffic polarization to same links on the next hop QFX switch when two are connected with LAG/ECMP.

Table 5: show forwarding-options enhanced-hash-key Output Fields (*continued*)

Field Name	Field Description
<b>Key</b>	Indicates whether the GRE key field is or is not used by the hashing algorithm: Yes or No.
<b>Protocol</b>	Indicates if a Generic Router Encapsulation (GRE) endpoint over routes was dynamically learned by a routing protocol such as RIP or OSPF.
<b>MPLS Enabled</b>	Indicates if MPLS is enabled under L2 switching.
<b>VXLAN VNID</b>	A 24-bit virtual network identifier (VNID) that uniquely identifies the Virtual Extensible Local Area Networks (VXLAN) segment.

## Sample Output

### show forwarding-options enhanced-hash-key (Layer 2 Payload Hash Mode)

```

user@switch> show forwarding-options enhanced-hash-key
Slot 0

Current Hash Settings
-----
Hash-Mode                               : layer2-payload

inet Hash settings-
-----
inet packet fields
  Protocol                               : Yes
  Destination L4 Port                    : Yes
  Source L4 Port                         : Yes
  Destination IPv4 Addr                  : Yes
  Source IPv4 Addr                       : Yes
  Vlan id                                : No

inet6 Hash settings-
-----
inet6 packet fields
  Next Hdr                              : Yes
  Destination L4 Port                    : Yes
  Source L4 Port                         : Yes
  Destination IPv6 Addr                  : Yes
  Source IPv6 Addr                       : Yes
  Vlan id                                : No

```

### show forwarding-options enhanced-hash-key (Layer 2 Header Hash Mode)

```

user@switch> show forwarding-options enhanced-hash-key
Slot 0

Current Hash Settings
-----

Hash-Mode                               : layer2-header

```

## layer2 Hash settings-

-----  
layer2 packet fields

Ether Type	: Yes
Destination MAC Address	: Yes
Source MAC Address	: Yes
VLAN ID	: No

**show forwarding-options enhanced-hash-key (Fabric Load Balancing Options)**

```
user@switch> show forwarding-options enhanced-hash-key
<some output removed for brevity>
```

## Fabric Load Balancing Options

```
-----
Load Balancing Method : Flowlet
Fabric Link Scale      : 40960 (mbps)
Inactivity Interval   : 16 (us)
Hash Region Size/Trunk : 1024 (buckets)
```

**show forwarding-options enhanced-hash-key (QFX10002 and QFX 10008 Switches)**

```
user@switch> show forwarding-options enhanced-hash-key
Slot 0
```

Seed value for Hash function	0: 3626023417
Seed value for Hash function	1: 3626023417
Seed value for Hash function	2: 3626023417
Seed value for Hash function	3: 3626023417

## Inet settings:

```
-----
IPV4 dest address:  Yes
IPV4 source address: Yes
L4 Dest Port:      Yes
L4 Source Port:    Yes
```

## Inet6 settings:

```
-----
IPV6 dest address:  Yes
IPV6 source address: Yes
L4 Dest Port:      Yes
L4 Source Port:    Yes
```

## L2 settings:

```
-----
Dest Mac address:  No
Source Mac address: No
Vlan Id:           Yes
Inner-vlan Id:     No
Incoming port:     Yes
```

## GRE settings:

```
-----
Key:               No
Protocol:          No
```

## MPLS settings:

```
-----
MPLS Enabled:      Yes
```

## VXLAN settings:

```
-----
VXLAN VNID:        No
```



## show snmp mib

<b>Syntax</b>	<code>show snmp mib (get   get-next   walk) (ascii   decimal) <i>object-id</i></code>
<b>Release Information</b>	<p>Command introduced before Junos OS Release 7.4.</p> <p>Command introduced in Junos OS Release 9.0 for EX Series switches.</p> <p><b>ascii</b> and <b>decimal</b> options introduced in Junos OS Release 9.6.</p> <p><b>ascii</b> and <b>decimal</b> options introduced in Junos OS Release 9.6 for EX Series switches.</p> <p>Command introduced in Junos OS Release 11.1 for the QFX Series.</p> <p>Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.</p>
<b>Description</b>	Display local Simple Network Management Protocol (SNMP) Management Information Base (MIB) object values.
<b>Options</b>	<p><b>get</b>—Retrieve and display one or more SNMP object values.</p> <p><b>get-next</b>—Retrieve and display the next SNMP object values.</p> <p><b>walk</b>—Retrieve and display the SNMP object values that are associated with the requested object identifier (OID). When you use this option, the Junos OS displays the objects below the subtree that you specify.</p> <p><b>ascii</b>—Display the SNMP object's string indices as an ASCII-key representation.</p> <p><b>decimal</b>—Display the SNMP object values in the decimal (default) format. The <b>decimal</b> option is the default option for this command. Therefore, issuing the <b>show snmp mib (get   get-next   walk) decimal object-id</b> and the <b>show snmp mib (get   get-next   walk) object-id</b> commands display the same output.</p> <p><b>object-id</b>—The object can be represented by a sequence of dotted integers (such as 1.3.6.1.2.1.2) or by its subtree name (such as <b>interfaces</b>). When entering multiple objects, enclose the objects in quotation marks.</p>
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration.
<b>List of Sample Output</b>	<p><a href="#">show snmp mib get on page 102</a></p> <p><a href="#">show snmp mib get (Multiple Objects) on page 102</a></p> <p><a href="#">show snmp mib get (Layer 2 Policer) on page 102</a></p> <p><a href="#">show snmp mib get-next on page 102</a></p> <p><a href="#">show snmp mib get-next (Specify an OID) on page 102</a></p> <p><a href="#">show snmp mib walk on page 102</a></p> <p><a href="#">show snmp mib walk (QFX Series) on page 102</a></p> <p><a href="#">show snmp mib walk decimal on page 103</a></p> <p><a href="#">show snmp mib walk (ASCII) on page 103</a></p> <p><a href="#">show snmp mib walk (Multiple Indices) on page 103</a></p> <p><a href="#">show snmp mib walk decimal (Multiple Indices) on page 103</a></p>
<b>Output Fields</b>	<a href="#">Table 6 on page 102</a> describes the output fields for the <b>show snmp mib</b> command. Output fields are listed in the approximate order in which they appear.

Table 6: show snmp mib Output Fields

Field Name	Field Description
<i>name</i>	Object name and numeric instance value.
<i>object value</i>	Object value. The Junos OS translates OIDs into the corresponding object names.

## Sample Output

### show snmp mib get

```
user@host> show snmp mib get sysObjectID.0
sysObjectID.0 = jnxProductNameM20
```

### show snmp mib get (Multiple Objects)

```
user@host> show snmp mib get ?sysObjectID.0 sysUpTime.0?
sysObjectID.0 = jnxProductNameM20
sysUpTime.0 = 1640992
```

### show snmp mib get (Layer 2 Policer)

```
user@host> show snmp mib get ifInOctets.25970
ifInOctets.25970 = 7545720
```

### show snmp mib get-next

```
user@host> show snmp mib get-next jnxMibs
jnxBoxClass.0 = jnxProductLineM20.0
```

### show snmp mib get-next (Specify an OID)

```
user@host> show snmp mib get-next 1.3.6.1
sysDescr.0 = Juniper Networks, Inc. m20 internet router, kernel
Junos OS Release: 2004-1 Build date: build date UTC Copyright (c) 1996-2004 Juniper
Networks, Inc.
```

### show snmp mib walk

```
user@host> show snmp mib walk system
sysDescr.0 = Juniper Networks, Inc. m20 internet router, kernel
Junos OS Release #0: 2004-1 Build date: build date UTC Copyright (c) 1996-2004
Juniper Networks, Inc.
sysObjectID.0 = jnxProductNameM20
sysUpTime.0 = 1640992
sysContact.0 = Your contact
sysName.0 = my router
sysLocation.0 = building 1
sysServices.0 = 4
```

### show snmp mib walk (QFX Series)

```
user@switch> show snmp mib walk system
sysDescr.0 = Juniper Networks, Inc. qfx3500s internet router, kernel JUNOS
11.1-20100926.0 #0: 2010-09-26 06:17:38 UTC Build date: 2010-09-26 06:00:10
sysObjectID.0 = jnxProductQFX3500
sysUpTime.0 = 138980301
sysContact.0 = System Contact
```

```

sysName.0      = LabQFX3500
sysLocation.0 = Lab
sysServices.0 = 4

```

#### show snmp mib walk decimal

```

user@host show snmp mib walk decimal jnxUtilData
jnxUtilCounter32Value.102.114.101.100 = 100

```

#### show snmp mib walk (ASCII)

```

show snmp mib walk ascii jnxUtilData
jnxUtilCounter32Value."fred" = 100

```

#### show snmp mib walk (Multiple Indices)

```

show snmp mib walk ascii jnxFWCounterByteCount
jnxFWCounterByteCount."fe-1/3/0.0-i"."CLASS_BE-fe-1/3/0.0-i".2 = 0
jnxFWCounterByteCount."fe-1/3/0.0-i"."CLASS_CC-fe-1/3/0.0-i".2 = 0
jnxFWCounterByteCount."fe-1/3/0.0-i"."CLASS_RT-fe-1/3/0.0-i".2 = 0
.....

```

#### show snmp mib walk decimal (Multiple Indices)

```

show snmp mib walk decimal jnxFWCounterByteCount
jnxFWCounterByteCount."fe-1/3/0.0-i"."CLASS_BE-fe-1/3/0.0-i".2 = 0
jnxFWCounterByteCount."fe-1/3/0.0-i"."CLASS_CC-fe-1/3/0.0-i".2 = 0
jnxFWCounterByteCount."fe-1/3/0.0-i"."CLASS_RT-fe-1/3/0.0-i".2 = 0
.....

```

## show virtual-chassis active-topology

<b>Syntax</b>	show virtual-chassis active-topology <all-members> <local> <member <i>member-id</i> >
<b>Release Information</b>	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. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
<b>Description</b>	Display the active topology of the Virtual Chassis or VCF with next-hop reachability information.
<b>Options</b>	<p><b>none</b>—Display the active topology of the member switch where the command is issued.</p> <p><b>all-members</b>—(Optional) Display the active topology of all members of the Virtual Chassis or VCF.</p> <p><b>local</b>—(Optional) Display the active topology of the switch or external Routing Engine on which this command is entered.</p> <p><b>member <i>member-id</i></b>—(Optional) Display the active topology of the specified member of the Virtual Chassis or VCF.</p>
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li><i>Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis</i></li> <li><i>Understanding EX Series Virtual Chassis Configuration</i></li> </ul>
<b>List of Sample Output</b>	<a href="#">show virtual-chassis active-topology (EX4200 Virtual Chassis) on page 105</a> <a href="#">show virtual-chassis active-topology (EX8200 Virtual Chassis) on page 105</a> <a href="#">show virtual-chassis active-topology (Virtual Chassis Fabric) on page 106</a>
<b>Output Fields</b>	Table 7 on page 104 lists the output fields for the <b>show virtual-chassis active-topology</b> command. Output fields are listed in the approximate order in which they appear.

Table 7: show virtual-chassis active-topology Output Fields

Field Name	Field Description
<b>Destination ID</b>	Specifies the member ID of the destination.
<b>Next-hop</b>	<p>Specifies the member ID and Virtual Chassis port (VCP) of the next hop to which packets for the destination ID are forwarded.</p> <p>The next hop can be more than one device in a VCF.</p>



## Sample Output

### show virtual-chassis active-topology (EX4200 Virtual Chassis)

```

user@switch> show virtual-chassis active-topology
 1                      1(vcp-1)

 2                      1(vcp-1)

 3                      1(vcp-1)

 4                      1(vcp-1)

 5                      8(vcp-0) 1(vcp-1)

 6                      8(vcp-0)

 7                      8(vcp-0)

 8                      8(vcp-0)

```

### show virtual-chassis active-topology (EX8200 Virtual Chassis)

```

user@external-routing-engine> show virtual-chassis active-topology
member0:

```

Destination ID	Next-hop
1	1(vcp-4/0/4.32768)
8	8(vcp-0/0.32768)
9	8(vcp-0/0.32768)

```
member1:
```

Destination ID	Next-hop
0	0(vcp-3/0/4.32768)
8	8(vcp-0/0.32768)
9	8(vcp-0/0.32768)

```
member8:
```

Destination ID	Next-hop
0	0(vcp-1/1.32768)
1	1(vcp-1/2.32768)
9	9(vcp-2/1.32768)

member9:

Destination ID	Next-hop
0	8(vcp-1/2.32768)
1	8(vcp-1/2.32768)
8	8(vcp-1/2.32768)

#### show virtual-chassis active-topology (Virtual Chassis Fabric)

user@device> show virtual-chassis active-topology  
fpc0:

Destination ID	Next-hop
1 6(vcp-255/0/1.32768)	4(vcp-255/0/2.32768) 5(vcp-255/0/3.32768)
2 6(vcp-255/0/1.32768)	4(vcp-255/0/2.32768) 5(vcp-255/0/3.32768)
3 6(vcp-255/0/1.32768)	4(vcp-255/0/2.32768) 5(vcp-255/0/3.32768)
4	4(vcp-255/0/2.32768)
5	5(vcp-255/0/3.32768)
6	6(vcp-255/0/1.32768)

fpc1:

Destination ID	Next-hop
0 6(vcp-255/0/1.32768)	4(vcp-255/0/2.32768) 5(vcp-255/0/3.32768)
2 6(vcp-255/0/1.32768)	4(vcp-255/0/2.32768) 5(vcp-255/0/3.32768)
3 6(vcp-255/0/1.32768)	4(vcp-255/0/2.32768) 5(vcp-255/0/3.32768)
4	4(vcp-255/0/2.32768)
5	5(vcp-255/0/3.32768)
6	6(vcp-255/0/1.32768)

fpc2:

Destination ID	Next-hop
0 6(vcp-255/0/1.32768)	4(vcp-255/0/2.32768) 5(vcp-255/0/3.32768)
1 6(vcp-255/0/1.32768)	4(vcp-255/0/2.32768) 5(vcp-255/0/3.32768)
3 6(vcp-255/0/1.32768)	4(vcp-255/0/2.32768) 5(vcp-255/0/3.32768)
4	4(vcp-255/0/2.32768)
5	5(vcp-255/0/3.32768)
6	6(vcp-255/0/1.32768)

fpc3:

Destination ID	Next-hop
0 6(vcp-255/0/1.32768)	4(vcp-255/0/2.32768) 5(vcp-255/0/3.32768)
1 6(vcp-255/0/1.32768)	4(vcp-255/0/2.32768) 5(vcp-255/0/3.32768)
2 6(vcp-255/0/1.32768)	4(vcp-255/0/2.32768) 5(vcp-255/0/3.32768)
4	4(vcp-255/0/2.32768)
5	5(vcp-255/0/3.32768)
6	6(vcp-255/0/1.32768)

fpc4:

Destination ID	Next-hop
0	0(vcp-255/0/48.32768)
1	1(vcp-255/0/49.32768)
2	2(vcp-255/0/50.32768)
3	3(vcp-255/0/51.32768)
5 0(vcp-255/0/48.32768)	3(vcp-255/0/51.32768) 2(vcp-255/0/50.32768) 1(vcp-255/0/49.32768)
6 0(vcp-255/0/48.32768)	3(vcp-255/0/51.32768) 2(vcp-255/0/50.32768) 1(vcp-255/0/49.32768)

fpc5:

Destination ID	Next-hop
0	0(vcp-255/0/48.32768)

1	1(vcp-255/0/49.32768)	
2	2(vcp-255/0/50.32768)	
3	3(vcp-255/0/51.32768)	
4	3(vcp-255/0/51.32768)	2(vcp-255/0/50.32768)
0(vcp-255/0/48.32768)	1(vcp-255/0/49.32768)	
6	3(vcp-255/0/51.32768)	2(vcp-255/0/50.32768)
0(vcp-255/0/48.32768)	1(vcp-255/0/49.32768)	

fpc6:

Destination ID	Next-hop
0	0(vcp-255/0/0.32768)
1	1(vcp-255/0/1.32768)
2	2(vcp-255/0/2.32768)
3	3(vcp-255/0/3.32768)
4	3(vcp-255/0/3.32768) 2(vcp-255/0/2.32768)
0(vcp-255/0/0.32768)	1(vcp-255/0/1.32768)
5	3(vcp-255/0/3.32768) 2(vcp-255/0/2.32768)
0(vcp-255/0/0.32768)	1(vcp-255/0/1.32768)

## show virtual-chassis device-topology

<b>Syntax</b>	show virtual-chassis device-topology <all-members> <local> <member <i>member-id</i> >
<b>Release Information</b>	Command introduced in Junos OS Release 10.4 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
<b>Description</b>	Display the device topology—the member and system IDs, the VCP numbers, and device status—for all hardware devices in the Virtual Chassis or VCF.
<b>Options</b>	<p><b>none</b>—Display the device topology for all members of the Virtual Chassis or VCF.</p> <p><b>all-members</b>—(Optional) Display the device topology for all members of the Virtual Chassis or VCF.</p> <p><b>local</b>—(Optional) Display the device topology for the switch or external Routing Engine on which this command is entered.</p> <p><b>member <i>member-id</i></b>—(Optional) Display the device topology for the specified member of the Virtual Chassis or VCF.</p>
<b>Required Privilege Level</b>	clear
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>Understanding EX Series Virtual Chassis Port Link Aggregation</i></li> <li>• <i>Understanding EX8200 Virtual Chassis Topologies</i></li> </ul>
<b>Output Fields</b>	<a href="#">Table 8 on page 109</a> lists the output fields for the <b>show virtual-chassis device-topology</b> command. Output fields are listed in the approximate order in which they appear.

**Table 8: show virtual-chassis device-topology Output Fields**

Field Name	Field Description
<b>Member</b>	Assigned member ID.
<b>Device</b>	Assigned device ID.  For an EX8200 Virtual Chassis, the member ID and the device ID are always identical.
<b>Status</b>	The status of the device within the Virtual Chassis or VCF. Outputs include: <ul style="list-style-type: none"> <li>• <b>Prsnt</b>—Device is currently connected to and participating in the Virtual Chassis or VCF.</li> <li>• <b>NotPrsnt</b>—Device is assigned but is not currently connected.</li> </ul>

Table 8: show virtual-chassis device-topology Output Fields (*continued*)

Field Name	Field Description
<b>System ID</b>	System ID of the device.  The system ID of the device is the device's MAC address.
<b>Member (Neighbor List)</b>	Assigned member ID of the neighbor device.
<b>Device (Neighbor List)</b>	Assigned device ID of the neighbor device.  For an EX8200 Virtual Chassis, the member ID and the device ID are always identical.
<b>Interface (Neighbor List)</b>	The interface connecting the device to the neighbor.

## Sample Output

### show virtual-chassis device-topology

```
user@switch> show virtual-chassis device-topology
```

```
member0:
```

```
-----
Member  Device  Status  System ID      Neighbor List
                                Member  Device  Interface
0        0        Prsnt   0021.59f7.d000  8        8        vcp-0/0
                                1        1        vcp-4/0/1
1        1        Prsnt   0026.888d.6800  8        8        vcp-0/0
                                9        9        vcp-0/1
                                0        0        vcp-3/0/4
8        8        Prsnt   0000.4a75.9b7c  9        9        vcp-1/0
                                0        0        vcp-1/1
                                1        1        vcp-1/2
9        9        Prsnt   0000.73e9.9a57  8        8        vcp-1/0
                                1        1        vcp-1/1
```

```
member1:
```

```
-----
Member  Device  Status  System ID      Neighbor List
                                Member  Device  Interface
0        0        Prsnt   0021.59f7.d000  8        8        vcp-0/0
                                1        1        vcp-4/0/1
1        1        Prsnt   0026.888d.6800  8        8        vcp-0/0
                                9        9        vcp-0/1
                                0        0        vcp-3/0/4
8        8        Prsnt   0000.4a75.9b7c  9        9        vcp-1/0
                                0        0        vcp-1/1
                                1        1        vcp-1/2
9        9        Prsnt   0000.73e9.9a57  8        8        vcp-1/0
                                1        1        vcp-1/1
```

```
member8:
```

```
-----
Member  Device  Status  System ID      Neighbor List
                                Member  Device  Interface
```

0	0	Prsnt	0021.59f7.d000	8	8	vcp-0/0
				1	1	vcp-4/0/1
1	1	Prsnt	0026.888d.6800	8	8	vcp-0/0
				9	9	vcp-0/1
				0	0	vcp-3/0/4
8	8	Prsnt	0000.4a75.9b7c	9	9	vcp-1/0
				0	0	vcp-1/1
				1	1	vcp-1/2
9	9	Prsnt	0000.73e9.9a57	8	8	vcp-1/0
				1	1	vcp-1/1

member9:

				Neighbor List		
Member	Device	Status	System ID	Member	Device	Interface
0	0	Prsnt	0021.59f7.d000	8	8	vcp-0/0
				1	1	vcp-4/0/1
1	1	Prsnt	0026.888d.6800	8	8	vcp-0/0
				9	9	vcp-0/1
				0	0	vcp-3/0/4
8	8	Prsnt	0000.4a75.9b7c	9	9	vcp-1/0
				0	0	vcp-1/1
				1	1	vcp-1/2
9	9	Prsnt	0000.73e9.9a57	8	8	vcp-1/0
				1	1	vcp-1/1

#### show virtual-chassis device-topology (Virtual Chassis Fabric)

user@device> show virtual-chassis device-topology  
fpc0:

				Neighbor List		
Member	Device	Status	System ID	Member	Device	Interface
0	0	Prsnt	100e.7eb6.a900	4	4	vcp-255/0/2
				5	5	vcp-255/0/3
				6	6	vcp-255/0/1
1	1	Prsnt	100e.7eb8.3a40	4	4	vcp-255/0/2
				5	5	vcp-255/0/3
				6	6	vcp-255/0/1
2	2	Prsnt	100e.7eb5.d700	4	4	vcp-255/0/2
				5	5	vcp-255/0/3
				6	6	vcp-255/0/1
3	3	Prsnt	100e.7eb5.c440	4	4	vcp-255/0/2
				5	5	vcp-255/0/3
				6	6	vcp-255/0/1
4	4	Prsnt	100e.7eb5.7e40	3	3	vcp-255/0/51
				2	2	vcp-255/0/50
				0	0	vcp-255/0/48
				1	1	vcp-255/0/49
5	5	Prsnt	100e.7eb5.80c0	3	3	vcp-255/0/51
				2	2	vcp-255/0/50
				1	1	vcp-255/0/49
				0	0	vcp-255/0/48
6	6	Prsnt	100e.7eb6.3b00	3	3	vcp-255/0/3
				2	2	vcp-255/0/2
				0	0	vcp-255/0/0
				1	1	vcp-255/0/1

fpc1:

Neighbor List

Member	Device	Status	System ID	Member	Device	Interface
0	0	Prsnt	100e.7eb6.a900	4	4	vcp-255/0/2
				5	5	vcp-255/0/3
				6	6	vcp-255/0/1
1	1	Prsnt	100e.7eb8.3a40	4	4	vcp-255/0/2
				5	5	vcp-255/0/3
				6	6	vcp-255/0/1
2	2	Prsnt	100e.7eb5.d700	4	4	vcp-255/0/2
				5	5	vcp-255/0/3
				6	6	vcp-255/0/1
3	3	Prsnt	100e.7eb5.c440	4	4	vcp-255/0/2
				5	5	vcp-255/0/3
				6	6	vcp-255/0/1
4	4	Prsnt	100e.7eb5.7e40	3	3	vcp-255/0/51
				2	2	vcp-255/0/50
				0	0	vcp-255/0/48
				1	1	vcp-255/0/49
5	5	Prsnt	100e.7eb5.80c0	3	3	vcp-255/0/51
				2	2	vcp-255/0/50
				1	1	vcp-255/0/49
				0	0	vcp-255/0/48
6	6	Prsnt	100e.7eb6.3b00	3	3	vcp-255/0/3
				2	2	vcp-255/0/2
				0	0	vcp-255/0/0
				1	1	vcp-255/0/1

fpc2:

Neighbor List						
Member	Device	Status	System ID	Member	Device	Interface
0	0	Prsnt	100e.7eb6.a900	4	4	vcp-255/0/2
				5	5	vcp-255/0/3
				6	6	vcp-255/0/1
1	1	Prsnt	100e.7eb8.3a40	4	4	vcp-255/0/2
				5	5	vcp-255/0/3
				6	6	vcp-255/0/1
2	2	Prsnt	100e.7eb5.d700	4	4	vcp-255/0/2
				5	5	vcp-255/0/3
				6	6	vcp-255/0/1
3	3	Prsnt	100e.7eb5.c440	4	4	vcp-255/0/2
				5	5	vcp-255/0/3
				6	6	vcp-255/0/1
4	4	Prsnt	100e.7eb5.7e40	3	3	vcp-255/0/51
				2	2	vcp-255/0/50
				0	0	vcp-255/0/48
				1	1	vcp-255/0/49
5	5	Prsnt	100e.7eb5.80c0	3	3	vcp-255/0/51
				2	2	vcp-255/0/50
				1	1	vcp-255/0/49
				0	0	vcp-255/0/48
6	6	Prsnt	100e.7eb6.3b00	3	3	vcp-255/0/3
				2	2	vcp-255/0/2
				0	0	vcp-255/0/0
				1	1	vcp-255/0/1

fpc3:

Neighbor List						
Member	Device	Status	System ID	Member	Device	Interface
0	0	Prsnt	100e.7eb6.a900	4	4	vcp-255/0/2
				5	5	vcp-255/0/3



1	1	Prsnt	100e.7eb8.3a40	6	6	vcp-255/0/1
				4	4	vcp-255/0/2
				5	5	vcp-255/0/3
2	2	Prsnt	100e.7eb5.d700	6	6	vcp-255/0/1
				4	4	vcp-255/0/2
				5	5	vcp-255/0/3
3	3	Prsnt	100e.7eb5.c440	6	6	vcp-255/0/1
				4	4	vcp-255/0/2
				5	5	vcp-255/0/3
4	4	Prsnt	100e.7eb5.7e40	6	6	vcp-255/0/1
				3	3	vcp-255/0/51
				2	2	vcp-255/0/50
5	5	Prsnt	100e.7eb5.80c0	0	0	vcp-255/0/48
				1	1	vcp-255/0/49
				3	3	vcp-255/0/51
				2	2	vcp-255/0/50
				1	1	vcp-255/0/49
6	6	Prsnt	100e.7eb6.3b00	0	0	vcp-255/0/48
				3	3	vcp-255/0/3
				2	2	vcp-255/0/2
				0	0	vcp-255/0/0
				1	1	vcp-255/0/1

fpc4:

				Neighbor List		
Member	Device	Status	System ID	Member	Device	Interface
0	0	Prsnt	100e.7eb6.a900	4	4	vcp-255/0/2
				5	5	vcp-255/0/3
				6	6	vcp-255/0/1
1	1	Prsnt	100e.7eb8.3a40	4	4	vcp-255/0/2
				5	5	vcp-255/0/3
				6	6	vcp-255/0/1
2	2	Prsnt	100e.7eb5.d700	4	4	vcp-255/0/2
				5	5	vcp-255/0/3
				6	6	vcp-255/0/1
3	3	Prsnt	100e.7eb5.c440	4	4	vcp-255/0/2
				5	5	vcp-255/0/3
				6	6	vcp-255/0/1
4	4	Prsnt	100e.7eb5.7e40	3	3	vcp-255/0/51
				2	2	vcp-255/0/50
				0	0	vcp-255/0/48
5	5	Prsnt	100e.7eb5.80c0	1	1	vcp-255/0/49
				3	3	vcp-255/0/51
				2	2	vcp-255/0/50
				1	1	vcp-255/0/49
				0	0	vcp-255/0/48
6	6	Prsnt	100e.7eb6.3b00	3	3	vcp-255/0/3
				2	2	vcp-255/0/2
				0	0	vcp-255/0/0
				1	1	vcp-255/0/1

fpc5:

				Neighbor List		
Member	Device	Status	System ID	Member	Device	Interface
0	0	Prsnt	100e.7eb6.a900	4	4	vcp-255/0/2
				5	5	vcp-255/0/3
				6	6	vcp-255/0/1
1	1	Prsnt	100e.7eb8.3a40	4	4	vcp-255/0/2
				5	5	vcp-255/0/3

2	2	Prsnt	100e.7eb5.d700	6	6	vcp-255/0/1
				4	4	vcp-255/0/2
				5	5	vcp-255/0/3
3	3	Prsnt	100e.7eb5.c440	6	6	vcp-255/0/1
				4	4	vcp-255/0/2
				5	5	vcp-255/0/3
4	4	Prsnt	100e.7eb5.7e40	6	6	vcp-255/0/1
				3	3	vcp-255/0/51
				2	2	vcp-255/0/50
5	5	Prsnt	100e.7eb5.80c0	0	0	vcp-255/0/48
				1	1	vcp-255/0/49
				3	3	vcp-255/0/51
6	6	Prsnt	100e.7eb6.3b00	2	2	vcp-255/0/50
				1	1	vcp-255/0/49
				0	0	vcp-255/0/48
				3	3	vcp-255/0/3
				2	2	vcp-255/0/2
				0	0	vcp-255/0/0
				1	1	vcp-255/0/1

fpc6:

				Neighbor List		
Member	Device	Status	System ID	Member	Device	Interface
0	0	Prsnt	100e.7eb6.a900	4	4	vcp-255/0/2
				5	5	vcp-255/0/3
				6	6	vcp-255/0/1
1	1	Prsnt	100e.7eb8.3a40	4	4	vcp-255/0/2
				5	5	vcp-255/0/3
				6	6	vcp-255/0/1
2	2	Prsnt	100e.7eb5.d700	4	4	vcp-255/0/2
				5	5	vcp-255/0/3
				6	6	vcp-255/0/1
3	3	Prsnt	100e.7eb5.c440	4	4	vcp-255/0/2
				5	5	vcp-255/0/3
				6	6	vcp-255/0/1
4	4	Prsnt	100e.7eb5.7e40	3	3	vcp-255/0/51
				2	2	vcp-255/0/50
				0	0	vcp-255/0/48
5	5	Prsnt	100e.7eb5.80c0	1	1	vcp-255/0/49
				3	3	vcp-255/0/51
				2	2	vcp-255/0/50
6	6	Prsnt	100e.7eb6.3b00	1	1	vcp-255/0/49
				0	0	vcp-255/0/48
				3	3	vcp-255/0/3
				2	2	vcp-255/0/2
				0	0	vcp-255/0/0
				1	1	vcp-255/0/1

## show virtual-chassis login

<b>Syntax</b>	<b>show virtual-chassis login</b>
<b>Release Information</b>	<p>Command introduced in Junos OS Release 9.3 for EX Series switches.</p> <p>Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series.</p> <p>Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).</p>
<b>Description</b>	<p>Supply the address of the host that logged into the Virtual Chassis or VCF, or identify the location of the member switch that redirected the current session to a different member switch.</p> <p>You might need this information for tracing or troubleshooting purposes.</p>
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">request session member on page 88</a></li> <li>• <i>Understanding Global Management of a Virtual Chassis</i></li> </ul>
<b>List of Sample Output</b>	<p><a href="#">show virtual-chassis login (Direct Login to the Master Console Port) on page 115</a></p> <p><a href="#">show virtual-chassis login (Backup Console Session Redirected to the Master Console Port) on page 115</a></p>

### Sample Output

#### show virtual-chassis login (Direct Login to the Master Console Port)

```
user@switch> show virtual-chassis login
Current login session initiated from host 192.0.2.3
```

#### show virtual-chassis login (Backup Console Session Redirected to the Master Console Port)

```
user@switch> show virtual-chassis login
Current login session initiated from host backup
```

## show virtual-chassis mode

<b>Syntax</b>	<b>show virtual-chassis mode</b> <all-members> <local> <member <i>member-id</i> >
<b>Release Information</b>	Command introduced in Junos OS Release 11.1 for EX Series switches. Command introduced in Junos OS Release 13.2X51-D20 for QFX Series devices. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF). <b>Current mode</b> and <b>Future mode after reboot</b> fields introduced in Junos OS Release 13.2X51-D20.
<b>Description</b>	Display the Virtual Chassis or Virtual Chassis Fabric (VCF) mixed mode status.
<b>Options</b>	<p><b>none</b>—Display the Virtual Chassis or VCF mixed mode status for the device on which the command is entered.</p> <p><b>all-members</b>—(Optional) Display the Virtual Chassis or VCF mixed mode status for all member devices in the Virtual Chassis or VCF.</p> <p><b>local</b>—(Optional) Display the Virtual Chassis or VCF mixed mode status for the device on which the command is entered.</p> <p><b>member <i>member-id</i></b>—(Optional) Display the Virtual Chassis or VCF mixed mode status for the specified member device..</p>
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">request virtual-chassis mode on page 89</a></li> <li>• <a href="#">Verifying the Virtual Chassis Fabric Mode Settings on page 46</a></li> <li>• <a href="#">Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches (CLI Procedure)</a></li> </ul>
<b>List of Sample Output</b>	<a href="#">show virtual-chassis mode (EX4200) on page 117</a> <a href="#">show virtual-chassis mode (QFX5100) on page 117</a>
<b>Output Fields</b>	<a href="#">Table 9 on page 116</a> lists the output fields for the <b>show virtual-chassis mode</b> command.

Table 9: show virtual-chassis mode Output Fields

Field Name	Field Description
<b>Mixed Mode</b>	Specifies the mixed mode status of the member switch. Mixed mode is either <b>Enabled</b> or <b>Disabled</b> .

Table 9: show virtual-chassis mode Output Fields (*continued*)

Field Name	Field Description
<b>Current mode</b>	<p>Specifies the current mixed and fabric mode settings running on the member device or devices.</p> <p>A device reboot is required to change the fabric or mixed mode. The <b>Current mode</b> and <b>Future mode after reboot</b> are different when the mode has been changed but the device has not been rebooted.</p> <p>Outputs include:</p> <ul style="list-style-type: none"> <li>• <b>Fabric with mixed devices</b>—Fabric mode and mixed mode are enabled.</li> <li>• <b>Fabric with similar devices</b>—Fabric mode is enabled and mixed mode is disabled.</li> <li>• <b>Virtual Chassis with mixed devices</b>—Fabric mode is disabled and mixed mode is enabled.</li> <li>• <b>Virtual Chassis with similar devices</b>—Fabric mode is disabled and mixed mode is disabled.</li> </ul>
<b>Future mode after reboot</b>	<p>Specifies the mixed and fabric mode settings running on the member device or devices.</p> <p>A device reboot is required to change the fabric or mixed mode. The <b>Current mode</b> and <b>Future mode after reboot</b> are different when the mode has been changed but the device has not been rebooted.</p> <p>Outputs include:</p> <ul style="list-style-type: none"> <li>• <b>Fabric with mixed devices</b>—Fabric mode and mixed mode are enabled.</li> <li>• <b>Fabric with similar devices</b>—Fabric mode is enabled and mixed mode is disabled.</li> <li>• <b>Virtual Chassis with mixed devices</b>—Fabric mode is disabled and mixed mode is enabled.</li> <li>• <b>Virtual Chassis with similar devices</b>—Fabric mode is disabled and mixed mode is disabled.</li> </ul>

## Sample Output

### show virtual-chassis mode (EX4200)

```
user@switch>show virtual-chassis mode
fpc0:
-----
Mixed Mode: Disabled
```

## Sample Output

### show virtual-chassis mode (QFX5100)

```
user@switch>show virtual-chassis mode
fpc0:
-----
Current mode : Fabric with similar devices
Future mode after reboot : Fabric with similar devices

fpc1:
-----
Current mode : Fabric with similar devices
Future mode after reboot : Fabric with similar devices

fpc2:
-----
Current mode : Fabric with similar devices
Future mode after reboot : Fabric with similar devices

fpc3:
```

-----  
Current mode : Fabric with similar devices  
Future mode after reboot : Fabric with similar devices

fpc4:

-----  
Current mode : Fabric with similar devices  
Future mode after reboot : Fabric with similar devices

## show virtual-chassis protocol adjacency

<b>Syntax</b>	<pre>show virtual-chassis protocol adjacency &lt;brief   detail   extensive&gt; &lt;all-members&gt; &lt;local&gt; &lt;member member-id&gt; &lt;system-id&gt;</pre>
<b>Release Information</b>	<p>Command introduced in Junos OS Release 10.4 for EX Series switches.</p> <p>Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series.</p> <p>Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).</p>
<b>Description</b>	Display the Virtual Chassis Control Protocol (VCCP) adjacency statistics in the Virtual Chassis or VCF for all hardware devices.
<b>Options</b>	<p><b>none</b>—Display VCCP adjacency statistics in brief form for all members of the Virtual Chassis or VCF.</p> <p><b>brief   detail   extensive</b>—(Optional) Display the specified level of output. Using the <b>brief</b> option is equivalent to entering the command with no options (the default). The <b>detail</b> and <b>extensive</b> options provide identical displays.</p> <p><b>all-members</b>—(Optional) Display VCCP adjacency statistics in brief form for all members of the Virtual Chassis or VCF.</p> <p><b>local</b>—(Optional) Display VCCP adjacency statistics for the switch or external Routing Engine on which this command is entered.</p> <p><b>member member-id</b>—(Optional) Display VCCP adjacency statistics for the specified member of the Virtual Chassis or VCF.</p> <p><b>system-id</b>—(Optional) Display VCCP adjacency statistics for the specified member of the Virtual Chassis or VCF.</p>
<b>Required Privilege Level</b>	clear
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>Understanding EX Series Virtual Chassis Port Link Aggregation</i></li> <li>• <i>Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis</i></li> </ul>
<b>List of Sample Output</b>	<p><a href="#">show virtual-chassis protocol adjacency on page 120</a></p> <p><a href="#">show virtual-chassis protocol adjacency detail on page 121</a></p>
<b>Output Fields</b>	<p><a href="#">Table 10 on page 120</a> lists the output fields for the <b>show virtual-chassis protocol adjacency</b> command. Output fields are listed in the approximate order in which they appear.</p>

Table 10: show virtual-chassis protocol adjacency Output Fields

Field Name	Field Description	Level of Output
<b>Interface</b>	Name of the Virtual Chassis port (VCP) interface.	All levels
<b>System</b>	The MAC address of the device on the receiving side of the VCP link.	All levels
<b>State</b>	State of the link. Outputs include: <ul style="list-style-type: none"> <li>• <b>Up</b>—The link is up.</li> <li>• <b>Down</b>—The link is down.</li> <li>• <b>New</b>—The link is new.</li> <li>• <b>One-way</b>—The link is transmitting traffic in one direction.</li> <li>• <b>Initializing</b>—The link is initializing.</li> <li>• <b>Rejected</b>—The link is rejected.</li> </ul>	All levels
<b>Hold, Expires in</b>	Remaining holdtime of the adjacency.	All levels
<b>Priority</b>	Priority to become the designated intermediary system.	detail
<b>Up/Down Transitions</b>	Count of adjacency status transition changes from up to down or down to up.	detail
<b>Last transition</b>	Time of the last up/down transition.	detail

## Sample Output

### show virtual-chassis protocol adjacency

```
user@switch> show virtual-chassis protocol adjacency
```

```
member0:
```

```
-----
Interface      System      State      Hold (secs)
vcp-0/0.32768  0000.4a75.9b7c Up          57
vcp-0/1.32768  0000.4a75.9b7c Up          59
vcp-4/0/1.32768 0026.888d.6800 Up          57
```

```
member1:
```

```
-----
Interface      System      State      Hold (secs)
vcp-0/0.32768  0000.4a75.9b7c Up          58
vcp-0/1.32768  0000.73e9.9a57 Up          59
vcp-3/0/4.32768 0021.59f7.d000 Up          58
```

```
member8:
```

```
-----
Interface      System      State      Hold (secs)
vcp-1/0.32768  0000.73e9.9a57 Up          58
vcp-1/1.32768  0021.59f7.d000 Up          58
vcp-1/2.32768  0026.888d.6800 Up          59
vcp-2/0.32768  0021.59f7.d000 Up          59
```

```
member9:
```

```
-----
Interface      System      State      Hold (secs)
```



vcp-1/0.32768	0000.4a75.9b7c Up	58
vcp-1/1.32768	0026.888d.6800 Up	59

### show virtual-chassis protocol adjacency detail

```
user@switch> show virtual-chassis protocol adjacency detail
```

```
member0:
```

```
-----
0000.4a75.9b7c
  interface-name: vcp-0/0.32768, State: Up, Expires in 57 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 19:26:37 ago
```

```
0000.4a75.9b7c
  interface-name: vcp-0/1.32768, State: Up, Expires in 59 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 19:26:37 ago
```

```
0026.888d.6800
  interface-name: vcp-4/0/1.32768, State: Up, Expires in 59 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 22:06:39 ago
```

```
member1:
```

```
-----
0000.4a75.9b7c
  interface-name: vcp-0/0.32768, State: Up, Expires in 59 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 19:26:38 ago
```

```
0000.73e9.9a57
  interface-name: vcp-0/1.32768, State: Up, Expires in 58 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 22:17:36 ago
```

```
0021.59f7.d000
  interface-name: vcp-3/0/4.32768, State: Up, Expires in 58 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 22:06:39 ago
```

```
member8:
```

```
-----
0000.73e9.9a57
  interface-name: vcp-1/0.32768, State: Up, Expires in 58 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 19:26:38 ago
```

```
0021.59f7.d000
  interface-name: vcp-1/1.32768, State: Up, Expires in 59 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 19:26:38 ago
```

```
0026.888d.6800
  interface-name: vcp-1/2.32768, State: Up, Expires in 59 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 19:26:38 ago
```

```
0021.59f7.d000
  interface-name: vcp-2/0.32768, State: Up, Expires in 57 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 19:26:38 ago
```

```
member9:
```

```
-----
0000.4a75.9b7c
  interface-name: vcp-1/0.32768, State: Up, Expires in 59 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 19:26:38 ago
```

```
0026.888d.6800
  interface-name: vcp-1/1.32768, State: Up, Expires in 58 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 22:17:36 ago
```

## show virtual-chassis protocol database

<b>Syntax</b>	show virtual-chassis protocol database <brief   detail   extensive> <all-members> <local> <member <i>member-id</i> >
<b>Release Information</b>	Command introduced in Junos OS Release 10.4 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
<b>Description</b>	Display the Virtual Chassis Control Protocol (VCCP) database statistics for all hardware devices within the Virtual Chassis or VCF.
<b>Options</b>	<p><b>none</b>—Display VCCP database statistics in brief form for all members of the Virtual Chassis or VCF.</p> <p><b>brief   detail   extensive</b>—(Optional) Display the specified level of output. Using the <b>brief</b> option is equivalent to entering the command with no options (the default). The <b>detail</b> option provides more output than the <b>brief</b> option. The <b>extensive</b> option provides all output and is most useful for customer support personnel.</p> <p><b>all-members</b>—(Optional) Display VCCP database statistics in brief form for all members of the Virtual Chassis or VCF.</p> <p><b>local</b>—(Optional) Display VCCP database statistics for the switch or external Routing Engine on which this command is entered.</p> <p><b>member <i>member-id</i></b>—(Optional) Display VCCP database statistics for the specified member of the Virtual Chassis or VCF.</p>
<b>Required Privilege Level</b>	clear
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis</i></li> <li>• <i>Understanding EX Series Virtual Chassis Components</i></li> <li>• <i>Understanding QFX Series Virtual Chassis Components</i></li> </ul>
<b>List of Sample Output</b>	<a href="#">show virtual-chassis protocol database on page 124</a> <a href="#">show virtual-chassis protocol database detail on page 125</a>
<b>Output Fields</b>	Table 11 on page 123 lists the output fields for the <b>show virtual-chassis protocol database</b> command. Output fields are listed in the approximate order in which they appear.

Table 11: show virtual-chassis protocol database Output Fields

Field Name	Field Description	Level of Output
LSP ID	Link-state protocol (LSP) data unit identifier.	All levels

Table 11: show virtual-chassis protocol database Output Fields (*continued*)

Field Name	Field Description	Level of Output
<b>Sequence</b>	Sequence number of the LSP.	All levels
<b>Checksum</b>	Checksum value of the LSP.	All levels
<b>Lifetime</b>	Remaining lifetime of the LSP, in seconds.	All levels
<b>Neighbor</b>	MAC address of the neighbor on the advertising system.	detail
<b>Interface</b>	Virtual Chassis port (VCP) interface name.	detail
<b>Metric</b>	Metric of the prefix or neighbor.	detail

The **extensive** output was omitted from this list. The **extensive** output is useful for customer support personnel only.

## Sample Output

### show virtual-chassis protocol database

```
user@switch> show virtual-chassis protocol database
```

```
member0:
```

```
-----
LSP ID          Sequence Checksum Lifetime
0000.4a75.9b7c.00-00 0x1dd80 0xc2e3 116
0000.73e9.9a57.00-00 0xf361 0x27e8 113
0021.59f7.d000.00-00 0x16882 0x3993 118
0026.888d.6800.00-00 0x1691f 0x82b7 116
  4 LSPs
```

```
member1:
```

```
-----
LSP ID          Sequence Checksum Lifetime
0000.4a75.9b7c.00-00 0x1dd80 0xc2e3 116
0000.73e9.9a57.00-00 0xf361 0x27e8 114
0021.59f7.d000.00-00 0x16883 0x289 116
0026.888d.6800.00-00 0x1691f 0x82b7 118
  4 LSPs
```

```
member8:
```

```
-----
LSP ID          Sequence Checksum Lifetime
0000.4a75.9b7c.00-00 0x1dd80 0xc2e3 118
0000.73e9.9a57.00-00 0xf361 0x27e8 114
0021.59f7.d000.00-00 0x16883 0x289 116
0026.888d.6800.00-00 0x16920 0xa335 116
  4 LSPs
```

```
member9:
```

```
-----
LSP ID          Sequence Checksum Lifetime
0000.4a75.9b7c.00-00 0x1dd80 0xc2e3 116
0000.73e9.9a57.00-00 0xf361 0x27e8 116
0021.59f7.d000.00-00 0x16883 0x289 114
```

```
0026.888d.6800.00-00      0x16920   0xa335      116
4 LSPs
```

### show virtual-chassis protocol database detail

```
user@switch> show virtual-chassis protocol database detail
```

```
member0:
```

```
-----
0000.4a75.9b7c.00-00 Sequence: 0x1ddbc, Checksum: 0x3111, Lifetime: 115 secs
Neighbor: 0000.73e9.9a57.00 Interface: vcp-1/0.32768 Metric: 150
Neighbor: 0021.59f7.d000.00 Interface: vcp-1/1.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-1/2.32768 Metric: 150
```

```
0000.73e9.9a57.00-00 Sequence: 0xf381, Checksum: 0xe065, Lifetime: 114 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-1/0.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-1/1.32768 Metric: 150
```

```
0021.59f7.d000.00-00 Sequence: 0x168af, Checksum: 0x8b0b, Lifetime: 118 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-0/0.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-4/0/1.32768 Metric: 15
```

```
0026.888d.6800.00-00 Sequence: 0x1694e, Checksum: 0xca97, Lifetime: 115 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-0/0.32768 Metric: 150
Neighbor: 0000.73e9.9a57.00 Interface: vcp-0/1.32768 Metric: 150
Neighbor: 0021.59f7.d000.00 Interface: vcp-3/0/4.32768 Metric: 15
```

```
member1:
```

```
-----
0000.4a75.9b7c.00-00 Sequence: 0x1ddbc, Checksum: 0x3111, Lifetime: 115 secs
Neighbor: 0000.73e9.9a57.00 Interface: vcp-1/0.32768 Metric: 150
Neighbor: 0021.59f7.d000.00 Interface: vcp-1/1.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-1/2.32768 Metric: 150
```

```
0000.73e9.9a57.00-00 Sequence: 0xf381, Checksum: 0xe065, Lifetime: 116 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-1/0.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-1/1.32768 Metric: 150
```

```
0021.59f7.d000.00-00 Sequence: 0x168af, Checksum: 0x8b0b, Lifetime: 116 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-0/0.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-4/0/1.32768 Metric: 15
```

```
0026.888d.6800.00-00 Sequence: 0x1694e, Checksum: 0xca97, Lifetime: 117 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-0/0.32768 Metric: 150
Neighbor: 0000.73e9.9a57.00 Interface: vcp-0/1.32768 Metric: 150
Neighbor: 0021.59f7.d000.00 Interface: vcp-3/0/4.32768 Metric: 15
```

```
member8:
```

```
-----
0000.4a75.9b7c.00-00 Sequence: 0x1ddbd, Checksum: 0xfd83, Lifetime: 118 secs
Neighbor: 0000.73e9.9a57.00 Interface: vcp-1/0.32768 Metric: 150
Neighbor: 0021.59f7.d000.00 Interface: vcp-1/1.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-1/2.32768 Metric: 150
```

```
0000.73e9.9a57.00-00 Sequence: 0xf381, Checksum: 0xe065, Lifetime: 115 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-1/0.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-1/1.32768 Metric: 150
```

```
0021.59f7.d000.00-00 Sequence: 0x168af, Checksum: 0x8b0b, Lifetime: 116 secs
```

```
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-0/0.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-4/0/1.32768 Metric: 15

0026.888d.6800.00-00 Sequence: 0x1694e, Checksum: 0xca97, Lifetime: 115 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-0/0.32768 Metric: 150
Neighbor: 0000.73e9.9a57.00 Interface: vcp-0/1.32768 Metric: 150
Neighbor: 0021.59f7.d000.00 Interface: vcp-3/0/4.32768 Metric: 15

member9:
-----

0000.4a75.9b7c.00-00 Sequence: 0x1ddbd, Checksum: 0xfd83, Lifetime: 116 secs
Neighbor: 0000.73e9.9a57.00 Interface: vcp-1/0.32768 Metric: 150
Neighbor: 0021.59f7.d000.00 Interface: vcp-1/1.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-1/2.32768 Metric: 150

0000.73e9.9a57.00-00 Sequence: 0xf381, Checksum: 0xe065, Lifetime: 117 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-1/0.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-1/1.32768 Metric: 150

0021.59f7.d000.00-00 Sequence: 0x168af, Checksum: 0x8b0b, Lifetime: 113 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-0/0.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-4/0/1.32768 Metric: 15

0026.888d.6800.00-00 Sequence: 0x1694f, Checksum: 0xa61a, Lifetime: 116 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-0/0.32768 Metric: 150
Neighbor: 0000.73e9.9a57.00 Interface: vcp-0/1.32768 Metric: 150
Neighbor: 0021.59f7.d000.00 Interface: vcp-3/0/4.32768 Metric: 15
```

## show virtual-chassis protocol interface

<b>Syntax</b>	<pre>show virtual-chassis protocol interface &lt;brief   detail&gt; &lt;all-members&gt; &lt;interface-name&gt; &lt;local&gt; &lt;member member-id&gt;</pre>
<b>Release Information</b>	<p>Command introduced in Junos OS Release 10.4 for EX Series switches.</p> <p>Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series.</p> <p>Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).</p>
<b>Description</b>	Display information about Virtual Chassis Control Protocol (VCCP) statistics for VCCP-enabled interfaces within the Virtual Chassis or VCF.
<b>Options</b>	<p><b>none</b>—Display the VCCP interface statistics in brief form for all members of the Virtual Chassis or VCF.</p> <p><b>brief   detail</b> —(Optional) Display the specified level of output. Using the <b>brief</b> option is equivalent to entering the command with no options (the default). The <b>detail</b> option provides more output than the <b>brief</b> option.</p> <p><b>all-members</b>—(Optional) Display VCCP interface statistics for all members of the Virtual Chassis or VCF.</p> <p><b>interface-name</b>—(Optional) Display VCCP interface statistics for the specified interface.</p> <p><b>local</b>—(Optional) Display VCCP interface statistics for the switch or external Routing Engine on which this command is entered.</p> <p><b>member member-id</b>—(Optional) Display VCCP interface statistics for the specified member of the Virtual Chassis or VCF.</p>
<b>Required Privilege Level</b>	clear
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>EX Series Virtual Chassis Overview</i></li> <li>• <i>Understanding QFX Series Virtual Chassis</i></li> <li>• <i>Understanding Virtual Chassis Ports in an EX8200 Virtual Chassis</i></li> <li>• <i>Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis</i></li> </ul>
<b>List of Sample Output</b>	<a href="#">show virtual-chassis protocol interface on page 128</a>
<b>Output Fields</b>	<a href="#">Table 12 on page 128</a> lists the output fields for the <b>show virtual-chassis protocol interface</b> command. Output fields are listed in the approximate order in which they appear.

Table 12: show virtual-chassis protocol interface Output Fields

Field Name	Field Description	Level of Output
<b>Interface</b>	Name of the VCP.	All levels
<b>State</b>	State of the link. Outputs include: <ul style="list-style-type: none"> <li>• <b>Up</b>—The link is up.</li> <li>• <b>Down</b>—The link is down.</li> </ul>	All levels
<b>Metric</b>	Metric of the prefix or neighbor.	All levels

## Sample Output

### show virtual-chassis protocol interface

```
user@switch> show virtual-chassis protocol interface
```

```
member0:
```

```
-----
IS-IS interface database:
```

Interface	State	Metric
vcp-0/0.32768	Up	150
vcp-0/1.32768	Up	150
vcp-4/0/1.32768	Up	15
vcp-4/0/7.32768	Down	15

```
member1:
```

```
-----
IS-IS interface database:
```

Interface	State	Metric
vcp-0/0.32768	Up	150
vcp-0/1.32768	Up	150
vcp-3/0/4.32768	Up	15

```
member8:
```

```
-----
IS-IS interface database:
```

Interface	State	Metric
vcp-0/0.32768	Down	150
vcp-1/0.32768	Up	150
vcp-1/1.32768	Up	150
vcp-1/2.32768	Up	150
vcp-1/3.32768	Down	150
vcp-2/0.32768	Up	150
vcp-2/1.32768	Down	150
vcp-2/2.32768	Down	150
vcp-2/3.32768	Down	150

```
member9:
```

```
-----
IS-IS interface database:
```

Interface	State	Metric
vcp-0/0.32768	Down	150
vcp-1/0.32768	Up	150
vcp-1/1.32768	Up	150
vcp-1/2.32768	Down	150
vcp-1/3.32768	Down	150





## show virtual-chassis protocol route

<b>Syntax</b>	show virtual-chassis protocol route <all-members> <destination-id> <local> <member member-id>
<b>Release Information</b>	Command introduced in Junos OS Release 10.4 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
<b>Description</b>	Display the unicast and multicast Virtual Chassis Control Protocol (VCCP) routing tables within the Virtual Chassis or VCF.
<b>Options</b>	<p><b>none</b>—Display the unicast and multicast routing tables for all members of the Virtual Chassis.</p> <p><b>all-members</b>—(Optional) Display the unicast and multicast routing tables for all members of the Virtual Chassis or VCF.</p> <p><b>destination-id</b>—(Optional) Display the unicast and multicast routing tables to the specified destination member ID for each member of the Virtual Chassis or VCF.</p> <p><b>local</b>—(Optional) Display the unicast and multicast routing tables on the device where this command is entered.</p> <p><b>member member-id</b>—(Optional) Display the unicast and multicast routing tables for the specified member of the Virtual Chassis or VCF.</p>
<b>Required Privilege Level</b>	clear
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>EX Series Virtual Chassis Overview</i></li> <li>• <i>Understanding QFX Series Virtual Chassis</i></li> <li>• <i>Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis</i></li> </ul>
<b>List of Sample Output</b>	<a href="#">show virtual-chassis protocol route on page 131</a>
<b>Output Fields</b>	<a href="#">Table 13 on page 130</a> lists the output fields for the <b>show virtual-chassis protocol route</b> command. Output fields are listed in the approximate order in which they appear.

**Table 13: show virtual-chassis protocol route Output Fields**

Field Name	Field Description
<b>Dev</b>	MAC address of the member storing the VCCP routing table.
<b>Version</b>	Version of the shortest-path-first algorithm that generated the routing table.

Table 13: show virtual-chassis protocol route Output Fields (*continued*)

Field Name	Field Description
<b>System ID</b>	MAC address of the device.
<b>Version</b>	Version of the shortest-path-first (SPF) algorithm that generated the route.
<b>Metric</b>	The metric number to get to that device.
<b>Interface</b>	Name of the Virtual Chassis port (VCP) interface connecting the devices.
<b>Via</b>	MAC address of the next-hop device, if applicable.

## Sample Output

### show virtual-chassis protocol route

```

user@switch> show virtual-chassis protocol route
member0:
-----
Dev 0021.59f7.d000 ucast routing table          Current version: 21
-----
System ID      Version  Metric Interface  Via
0000.4a75.9b7c    21      150 vcp-0/1.32768 0000.4a75.9b7c
0000.73e9.9a57    21      165 vcp-4/0/1.32768 0026.888d.6800
0021.59f7.d000    21        0
0026.888d.6800    21      15 vcp-4/0/1.32768 0026.888d.6800

Dev 0021.59f7.d000 mcast routing table          Current version: 21
-----
System ID      Version  Metric Interface  Via
0000.4a75.9b7c    21
0000.73e9.9a57    21
0021.59f7.d000    21      vcp-4/0/1.32768
                                vcp-0/1.32768
0026.888d.6800    21

member1:
-----
Dev 0026.888d.6800 ucast routing table          Current version: 25
-----
System ID      Version  Metric Interface  Via
0000.4a75.9b7c    25      150 vcp-0/0.32768 0000.4a75.9b7c
0000.73e9.9a57    25      150 vcp-0/1.32768 0000.73e9.9a57
0021.59f7.d000    25      15 vcp-3/0/4.32768 0021.59f7.d000
0026.888d.6800    25        0

Dev 0026.888d.6800 mcast routing table          Current version: 25
-----
System ID      Version  Metric Interface  Via
0000.4a75.9b7c    25
0000.73e9.9a57    25      vcp-3/0/4.32768
0021.59f7.d000    25      vcp-0/1.32768
0026.888d.6800    25      vcp-3/0/4.32768
                                vcp-0/0.32768

```

vcp-0/1.32768

member8:

-----

Dev 0000.4a75.9b7c ucast routing table                      Current version: 39

-----

System ID	Version	Metric	Interface	Via
0000.4a75.9b7c	39	0		
0000.73e9.9a57	39	150	vcp-1/0.32768	0000.73e9.9a57
0021.59f7.d000	39	150	vcp-2/0.32768	0021.59f7.d000
0026.888d.6800	39	150	vcp-1/2.32768	0026.888d.6800

Dev 0000.4a75.9b7c mcast routing table                      Current version: 39

-----

System ID	Version	Metric	Interface	Via
0000.4a75.9b7c	39		vcp-1/0.32768	
			vcp-2/0.32768	
			vcp-1/2.32768	
0000.73e9.9a57	39			
0021.59f7.d000	39			
0026.888d.6800	39			

member9:

-----

Dev 0000.73e9.9a57 ucast routing table                      Current version: 31

-----

System ID	Version	Metric	Interface	Via
0000.4a75.9b7c	31	150	vcp-1/0.32768	0000.4a75.9b7c
0000.73e9.9a57	31	0		
0021.59f7.d000	31	165	vcp-1/1.32768	0026.888d.6800
0026.888d.6800	31	150	vcp-1/1.32768	0026.888d.6800

Dev 0000.73e9.9a57 mcast routing table                      Current version: 31

-----

System ID	Version	Metric	Interface	Via
0000.4a75.9b7c	31			
0000.73e9.9a57	31		vcp-1/0.32768	
			vcp-1/1.32768	
0021.59f7.d000	31			
0026.888d.6800	31			

## show virtual-chassis protocol statistics

<b>Syntax</b>	show virtual-chassis protocol statistics <all-members> <interface-name> <local> <member member-id>
<b>Release Information</b>	Command introduced in Junos OS Release 10.4 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
<b>Description</b>	Display the Virtual Chassis Control Protocol (VCCP) statistics for all hardware devices within the Virtual Chassis or VCF.
<b>Options</b>	<p><b>none</b>—Display VCCP statistics for all members of the Virtual Chassis or VCF.</p> <p><b>all-members</b>—(Optional) Display VCCP statistics for all members of the Virtual Chassis or VCF.</p> <p><b>interface-name</b>—(Optional) Display VCCP statistics for the specified interface.</p> <p><b>local</b>—(Optional) Display VCCP statistics for the switch or external Routing Engine on which this command is entered.</p> <p><b>member member-id</b>—(Optional) Display VCCP statistics for the specified member of the Virtual Chassis or VCF.</p>
<b>Required Privilege Level</b>	clear
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>EX Series Virtual Chassis Overview</i></li> <li>• <i>Understanding QFX Series Virtual Chassis</i></li> <li>• <i>Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis</i></li> </ul>
<b>List of Sample Output</b>	<a href="#">show virtual-chassis protocol statistics on page 134</a>
<b>Output Fields</b>	<a href="#">Table 14 on page 133</a> lists the output fields for the <b>show virtual-chassis protocol interface</b> command. Output fields are listed in the approximate order in which they appear.

Table 14: show virtual-chassis protocol statistics Output Fields

Field Name	Field Description
<b>PDU type</b>	Protocol data unit type.
<b>Received</b>	Number of PDUs received since VCCP started or since the statistics were set to zero.
<b>Processed</b>	Number of PDUs received minus the number of PDUs dropped.

Table 14: show virtual-chassis protocol statistics Output Fields (*continued*)

Field Name	Field Description
<b>Drops</b>	Number of PDUs dropped.
<b>Sent</b>	Number of PDUs transmitted since VCCP started or since the statistics were set to zero.
<b>Rexmit</b>	Number of PDUs retransmitted since VCCP started or since the statistics were set to zero.
<b>Total Packets Received</b>	Number of PDUs received since VCCP started or since the statistics were set to zero.
<b>Total Packets Sent</b>	Number of PDUs sent since VCCP started or since the statistics were set to zero.
<b>LSP queue length</b>	Number of link-state PDUs waiting in the queue for processing. This value is almost always 0.
<b>SPF runs</b>	Number of shortest-path-first (SPF) calculations that have been performed.
<b>Fragments Rebuilt</b>	Number of link-state PDU fragments that the local system has computed.
<b>LSP Regenerations</b>	Number of link-state PDUs that have been regenerated. A link-state PDU is regenerated when it is nearing the end of its lifetime and it has not changed.
<b>Purges initiated</b>	Number of purges that the system initiated. A purge is initiated if the software determines that a link-state PDU must be removed from the network.

## Sample Output

### show virtual-chassis protocol statistics

```

user@switch> show virtual-chassis protocol statistics
member0:
-----
IS-IS statistics for 0021.59f7.d000:
PDU type      Received    Processed      Drops      Sent      Rexmit
LSP            8166        8166           0         4551         0
HELLO          1659        1659           0         1693         0
CSNP             2            2             0            3         0
PSNP           1909        1909           0         2293         0
Unknown         0            0             0            0         0
Totals        11736       11736           0         8540         0

Total packets received: 11736 Sent: 8540

LSP queue length: 0 Drops: 0
SPF runs: 9
Fragments rebuilt: 1640
LSP regenerations: 1
Purges initiated: 0

member1:
-----
IS-IS statistics for 0026.888d.6800:

```

PDU type	Received	Processed	Drops	Sent	Rexmit
LSP	10909	10909	0	12088	0
HELLO	1877	1877	0	2251	0
CSNP	3	3	0	3	0
PSNP	3846	3846	0	3732	0
Unknown	0	0	0	0	0
Totals	16635	16635	0	18074	0

Total packets received: 16635 Sent: 18074

LSP queue length: 0 Drops: 0  
 SPF runs: 13  
 Fragments rebuilt: 1871  
 LSP regenerations: 2  
 Purges initiated: 0

member8:

IS-IS statistics for 0000.4a75.9b7c:

PDU type	Received	Processed	Drops	Sent	Rexmit
LSP	7935	7935	0	14865	0
HELLO	2695	2695	0	7124	0
CSNP	4	4	0	4	0
PSNP	4398	4398	0	3666	0
Unknown	0	0	0	0	0
Totals	15032	15032	0	25659	0

Total packets received: 15032 Sent: 25659

LSP queue length: 0 Drops: 0  
 SPF runs: 26  
 Fragments rebuilt: 2666  
 LSP regenerations: 4  
 Purges initiated: 0

member9:

IS-IS statistics for 0000.73e9.9a57:

PDU type	Received	Processed	Drops	Sent	Rexmit
LSP	10800	10800	0	6327	0
HELLO	1492	1492	0	2356	0
CSNP	2	2	0	2	0
PSNP	2683	2683	0	3149	0
Unknown	0	0	0	0	0
Totals	14977	14977	0	11834	0

Total packets received: 14977 Sent: 11834

LSP queue length: 0 Drops: 0  
 SPF runs: 19  
 Fragments rebuilt: 1510  
 LSP regenerations: 6  
 Purges initiated: 0

## show virtual-chassis

<b>Syntax</b>	<b>show virtual-chassis</b> <b>&lt;status&gt;</b>
<b>Release Information</b>	<p>Command introduced in Junos OS Release 9.2 for EX Series switches.</p> <p>Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series.</p> <p>Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).</p> <p><b>Fabric ID</b>, <b>Fabric Mode</b>, and <b>Route Mode</b> output fields introduced in Junos OS Release 13.2X51-D20.</p> <p><b>Alias-Name</b> output field introduced in Junos OS Release 14.1X53-D10.</p>
<b>Description</b>	Display information about all members of the Virtual Chassis or VCF.
<b>Options</b>	<p><b>none</b>—Display information about all Virtual Chassis or VCF member devices.</p> <p><b>status</b>—Same output as for <b>show virtual-chassis</b>.</p>
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">show virtual-chassis active-topology on page 104</a></li> <li>• <a href="#">show virtual-chassis protocol adjacency on page 119</a></li> <li>• <a href="#">show virtual-chassis vc-path on page 141</a></li> <li>• <i>Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis</i></li> </ul>
<b>List of Sample Output</b>	<p><a href="#">show virtual-chassis (EX4200 Virtual Chassis) on page 138</a></p> <p><a href="#">show virtual-chassis (EX8200 Virtual Chassis) on page 139</a></p> <p><a href="#">show virtual-chassis (Virtual Chassis Fabric) on page 139</a></p>
<b>Output Fields</b>	<p><a href="#">Table 15 on page 136</a> lists the output fields for the <b>show virtual-chassis</b> command. Output fields are listed in the approximate order in which they appear.</p>

**Table 15: show virtual-chassis Output Fields**

Field Name	Field Description
<b>Fabric ID</b>	Assigned ID used to identify the VCF.
<b>Fabric Mode</b>	Mode of the VCF: Enabled, Disabled, or Mixed.
<b>Virtual Chassis ID</b>	Assigned ID that applies to the entire Virtual Chassis or VCF.



Table 15: show virtual-chassis Output Fields (*continued*)

Field Name	Field Description
<b>Virtual Chassis Mode</b>	<p>Mode of the Virtual Chassis or VCF. This field indicates support for the Virtual Chassis feature and, if a Virtual Chassis is configured, if it is a mixed or homogenous Virtual Chassis. Values can be:</p> <ul style="list-style-type: none"> <li>• <b>Enabled</b>—The platform supports the Virtual Chassis feature. If a Virtual Chassis is currently configured, this is a homogenous Virtual Chassis (all members are the same type of switch).</li> <li>• <b>Disabled</b>—The switch does not support the Virtual Chassis feature.</li> </ul> <p><b>NOTE:</b> Switches that support the Virtual Chassis feature do not display this value. Even if a Virtual Chassis is not currently configured, those switches display <b>Enabled</b> in this field.</p> <ul style="list-style-type: none"> <li>• <b>Mixed</b>—The platform supports the Virtual Chassis feature, and is configured as a mixed mode Virtual Chassis (members consist of more than one type of switch).</li> </ul>
<b>Member ID</b>	<p>Assigned member ID and FPC:</p> <ul style="list-style-type: none"> <li>• On all EX Series Virtual Chassis except EX8200 Virtual Chassis, and on a VCF, the FPC number refers to the member ID assigned to the switch.</li> <li>• On EX8200 Virtual Chassis, member IDs are numbered 0 through 9. The FPC number indicates the slot number of the line card within the Virtual Chassis. The FPC number on member 0 is always 0 through 15. The FPC number on member 1 is always 16 through 31. The FPC number on member 2 is always 32 through 47; and so on for the members.</li> </ul>
<b>Status</b>	<p>For a nonprovisioned configuration:</p> <ul style="list-style-type: none"> <li>• <b>Prsnt</b> for a member that is currently connected to the Virtual Chassis or VCF configuration.</li> <li>• <b>NotPrsnt</b> for a member ID that has been assigned but is not currently connected.</li> </ul> <p>For a preprovisioned configuration:</p> <ul style="list-style-type: none"> <li>• <b>Prsnt</b> for a member that is specified in the preprovisioned configuration file and is currently connected to the Virtual Chassis or VCF.</li> <li>• <b>Unprvsnd</b> for a member that is interconnected with the Virtual Chassis or VCF configuration but is not specified in the preprovisioned configuration file.</li> </ul>
<b>Serial No</b>	Serial number of the member device.
<b>Alias-Name</b>	<p>The user-configured alias of the member device.</p> <p>The <b>Alias-Name</b> field appears only if an alias has been configured for at least one device in the Virtual Chassis or VCF. Aliases are configured using the <b>alias-name</b> statement in the <code>[edit virtual-chassis aliases serial-number serial-number]</code> hierarchy.</p>
<b>Model</b>	Model number of the member device.
<b>Mastership Priority</b>	Mastership priority value of the member device.
<b>Role</b>	Role of the member device: master, backup, or linecard.

Table 15: show virtual-chassis Output Fields (*continued*)

Field Name	Field Description
<b>Mixed Mode</b>	Mixed mode configuration status: <ul style="list-style-type: none"> <li>• <b>Y</b> for a member device configured in mixed mode.</li> <li>• <b>N</b> for a member device not configured in mixed mode.</li> <li>• <b>NA</b> for a member device that cannot be configured in mixed mode.</li> </ul>
<b>Route Mode</b>	The route mode of the member device: fabric (F) or Virtual Chassis (V).
<b>Location</b>	Location of the member device.  If this field is empty, the location field was not set for the device.
<b>Neighbor List</b>	Member ID of the neighbor member to which this member's Virtual Chassis port (VCP) is connected.

## Sample Output

### show virtual-chassis (EX4200 Virtual Chassis)

```

user@switch> show virtual-chassis
Virtual Chassis ID: 0019.e250.47a0
Virtual Chassis Mode: Enabled

```

Member ID	Status	Serial No	Model	Mastership priority	Role	Mixed Mode	Neighbor List ID	Interface
0 (FPC 0)	Prsnt	AK0207360276	ex4200-24t	249	Master*	N	8	vcp-0
							1	vcp-1
1 (FPC 1)	Prsnt	AK0207360281	ex4200-24t	248	Backup	N	0	vcp-0
							2	vcp-1
2 (FPC 2)	Prsnt	AJ0207391130	ex4200-48p	247	Linecard	N	1	vcp-0
							3	vcp-1
3 (FPC 3)	Prsnt	AK0207360280	ex4200-24t	246	Linecard	N	2	vcp-0
							4	vcp-1
4 (FPC 4)	Prsnt	AJ0207391113	ex4200-48p	245	Linecard	N	3	vcp-0
							5	vcp-1
5 (FPC 5)	Prsnt	BP0207452204	ex4200-48t	244	Linecard	N	4	vcp-0
							6	vcp-1
6 (FPC 6)	Prsnt	BP0207452222	ex4200-48t	243	Linecard	N	5	vcp-0
							7	vcp-1
7 (FPC 7)	Prsnt	BR0207432028	ex4200-24f	242	Linecard	N	6	vcp-0

```

8 vcp-1
8 (FPC 8) Prsnt BR0207431996 ex4200-24f 241 Linecard N 7 vcp-0
0 vcp-1

```

Member ID for next new member: 9 (FPC 9)

#### show virtual-chassis (EX8200 Virtual Chassis)

```
user@external-routing-engine> show virtual-chassis
```

Virtual Chassis ID: c806.0842.de51

Virtual Chassis Mode: Enabled

Member ID	Status	Serial No	Model	Mastership priority	Role	Neighbor List ID Interface
0 (FPC 0-15)	Prsnt	BA0908380001	ex8216	0	Linecard	8 vcp-0/0 8 vcp-0/1 1 vcp-4/0/4
1 (FPC 16-31)	Prsnt	BT0909411634	ex8208	0	Linecard	8 vcp-0/0 0 vcp-3/0/4
8 (FPC 128-143)	Prsnt	062009000021	ex-xre	128	Master	9 vcp-1/0 1 vcp-1/2 9 vcp-1/3 0 vcp-2/0 9 vcp-2/1 0 vcp-1/1
9 (FPC 144-159)	Prsnt	062009000022	ex-xre	128	Backup*	8 vcp-1/0 8 vcp-1/2 8 vcp-1/3 8 vcp-1/3

#### show virtual-chassis (Virtual Chassis Fabric)

```
user@switch> show virtual-chassis
```

Preprovisioned Virtual Chassis Fabric

Fabric ID: 0282.5fa0.3f08

Fabric Mode: Enabled

List	Member ID	Status	Serial No	Model	Mstr prio	Role	Mixed Route Mode	Neighbor Mode	ID
Interface	0 (FPC 0)	Prsnt	AB3112430001	qfx5100-48s	129	Master*	N	F	3
vcp-255/1/0									
vcp-255/1/1									2
vcp-255/1/2									4
vcp-255/1/3									4
1 (FPC 1)	Prsnt	AB3112230001	qfx5100-48s	129	Backup	N	F	3	
vcp-255/1/0									2
vcp-255/1/1									4
vcp-255/1/2									4
vcp-255/1/3									
2 (FPC 2)	Prsnt	AB3112460011	qfx5100-48s	0	Linecard	N	F	1	
vcp-255/1/0									0
vcp-255/1/1									

3 (FPC 3) Prsnt	AB3112460011 qfx5100-48s	0	Linecard	N	F	1
vcp-255/1/0						0
vcp-255/1/1						
4 (FPC 4) Prsnt	AB3112430011 qfx5100-48s	0	Linecard	N	F	1
vcp-255/1/0						0
vcp-255/1/1						

## show virtual-chassis vc-path

<b>Syntax</b>	<b>show virtual-chassis vc-path source-interface <i>interface-name</i> destination-interface <i>interface-name</i></b>
<b>Release Information</b>	Command introduced in Junos OS Release 9.6 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series.
<b>Description</b>	Show the forwarding path a packet takes when going from a source interface to a destination interface in a Virtual Chassis or VCF configuration.
<b>Options</b>	<p><b>source-interface <i>interface-name</i></b>—Name of the interface from which the packet originates in the Virtual Chassis or VCF</p> <p><b>destination-interface <i>interface-name</i></b>—Name of the interface to which the packet is being delivered in the Virtual Chassis or VCF</p>
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis</i></li> <li>• <i>Understanding EX Series Virtual Chassis Configuration</i></li> <li>• <i>EX8200 Virtual Chassis Overview</i></li> </ul>
<b>List of Sample Output</b>	<a href="#">show virtual-chassis vc-path source-interface destination-interface on page 142</a>
<b>Output Fields</b>	<a href="#">Table 16 on page 141</a> lists the output fields for the <b>show virtual-chassis vc-path</b> command. Output fields are listed in the approximate order in which they appear.

**Table 16: show virtual-chassis vc-path Output Fields**

Field Name	Field Description
<b>Hop</b>	The hop number along the path between the source and destination interfaces. The first hop entry ( <b>Hop 0</b> ) is the packet's source, intermediate hop information represents transitions through the members within the Virtual Chassis or VCF, and the last hop entry represents arrival at the packet's destination.
<b>Member</b>	The Virtual Chassis or VCF member ID of the switch that contains the Packet Forwarding Engine for each hop through which the packet passes.
<b>PFE-Device</b>	<p>The number of the Packet Forwarding Engine in each Virtual Chassis or VCF member through which a packet passes.</p> <p>The Packet Forwarding Engine in each row is the next hop of the preceding Packet Forwarding Engine, including intermediate transitions through members within the Virtual Chassis.</p>
<b>Interface</b>	The name of the interface through which the Packet Forwarding Engines are connected. The interface for the first hop ( <b>Hop 0</b> ) is always the source interface.

## Sample Output

show virtual-chassis vc-path source-interface destination-interface

```
user@switch> show virtual-chassis vc-path source-interface ge-0/0/0 destination-interface
ge-1/0/1
vc-path from ge-0/0/0 to ge-1/0/1
Hop      Member    PFE-Device    Interface
0         0          1              ge-0/0/0
1         0          0              internal-1/24
2         1          3              vcp-0
3         1          4              ge-1/0/1
```

## show virtual-chassis vc-port

<b>Syntax</b>	show virtual-chassis vc-port <all-members> <local> <member <i>member-id</i> >
<b>Release Information</b>	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. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
<b>Description</b>	Display the status of the Virtual Chassis ports (VCPs), including both the dedicated VCPs and the uplink ports configured as VCPs.
<b>Options</b>	<p><b>none</b>—Display the operational status of all VCPs of the member switch where the command is issued.</p> <p><b>all-members</b>—(Optional) Display the operational status of all VCPs on all members of the Virtual Chassis or VCF.</p> <p><b>local</b>—(Optional) Display the operational status of the switch or external Routing Engine on which this command is entered.</p> <p><b>member <i>member-id</i></b>—(Optional) Display the operational status of all VCPs for the specified member of the Virtual Chassis or VCF.</p>
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">show virtual-chassis vc-port statistics on page 161</a></li> <li>• <i>Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis</i></li> <li>• <i>Verifying Virtual Chassis Ports in an EX8200 Virtual Chassis</i></li> </ul>
<b>List of Sample Output</b>	<a href="#">show virtual-chassis vc-port (EX4200 Virtual Chassis) on page 145</a> <a href="#">show virtual-chassis vc-port (EX8200 Virtual Chassis) on page 145</a> <a href="#">show virtual-chassis vc-port all-members on page 146</a>
<b>Output Fields</b>	Table 17 on page 143 lists the output fields for the <b>show virtual-chassis vc-port</b> command. Output fields are listed in the approximate order in which they appear.

Table 17: show virtual-chassis vc-port Output Fields

Field Name	Field Description
<i>fpcnumber</i>	The FPC number is the same as the member ID.

Table 17: show virtual-chassis vc-port Output Fields (*continued*)

Field Name	Field Description
Interface or PIC/Port	<p>VCP name.</p> <ul style="list-style-type: none"> <li>The dedicated VCPs in an EX4200 or EX4500 Virtual Chassis are <b>vcp-0</b> and <b>vcp-1</b>. The dedicated VCPs in an EX4550 Virtual Chassis are <b>VCP-1/0</b>, <b>VCP-1/1</b>, <b>VCP-2/0</b>, and <b>VCP-2/1</b>.</li> <li>Optical ports set as VCPs are named 1/0 and 1/1, representing the PIC number and the port number.</li> <li>The native VCP (port 0) on an XRE200 External Routing Engine in an EX8200 Virtual Chassis is named <b>vcp-0</b>.</li> <li>The VCPs on each Virtual Chassis Control Interface (VCCI) module in an XRE200 External Routing Engine are named using the <b>vcp-slot-number/port-number</b> convention; for instance, <b>vcp-1/0</b>.</li> <li>The VCPs on EX8200 member switches are named using the <b>vcp-slot-number/pic-number/interface-number</b> convention; for instance, <b>vcp-3/0/2</b>.</li> <li>A <b>255</b> as the first number in your port number indicates that your VCP is part of a Link Aggregation group (LAG) bundle. For instance, a display of <b>vcp-255/1/0</b> indicates that the dedicated VCP named <b>vcp-1/0</b> is part of a LAG bundle. A display of <b>vcp-255/1/0</b> indicates that an uplink port that was previously named <b>xe-0/1/0</b> is now part of a VCP LAG bundle.</li> </ul>
Type	<p>Type of VCP:</p> <ul style="list-style-type: none"> <li><b>Dedicated</b>—The rear panel VCP on an EX4200, EX4500, or EX4550 switch, or any VCP link connected to an XRE200 External Routing Engine in an EX8200 Virtual Chassis.</li> <li><b>Configured</b>—Optical port configured as a VCP.</li> <li><b>Auto-Configured</b>—Optical port autoconfigured as a VCP.</li> </ul> <p>See <i>Setting an Uplink Port on an EX Series Switch as a Virtual Chassis Port (CLI Procedure)</i> or <i>Setting a 10-Gigabit Ethernet Port as a Virtual Chassis Port in an EX8200 Virtual Chassis (CLI Procedure)</i> for information about configuring VCPs.</p>
Trunk ID	<p>A positive-number ID assigned to a link aggregation group (LAG) formed by the Virtual Chassis. The trunk ID value is –1 if no trunk is formed. A LAG between uplink VCPs requires that the link speed be the same on connected interfaces and that at least two VCPs on one member be connected to at least two VCPs on the other member in an EX4200 or EX4500 Virtual Chassis.</p> <p>Dedicated VCP LAGs are assigned trunk IDs 1 and 2. Trunk IDs for LAGs formed with uplink VCPs therefore have values of 3 or greater.</p> <p>The trunk ID value changes if the link-adjacency state between LAG members changes; trunk membership is then allocated or deallocated.</p>
Status	<p>Interface status:</p> <ul style="list-style-type: none"> <li><b>absent</b>—Interface is not a VCP link.</li> <li><b>down</b>—VCP link is down.</li> <li><b>up</b>—VCP link is up.</li> </ul>
Speed (mbps)	Speed of the interface in megabits per second.
Neighbor ID/Interface	The Virtual Chassis member ID and interface of a VCP on a member that is connected to the interface or PIC/Port field in the same row as this interface.



## Sample Output

### show virtual-chassis vc-port (EX4200 Virtual Chassis)

```
user@switch> show virtual-chassis vc-port
```

```
fpc0:
```

Interface or PIC / Port	Type	Trunk ID	Status	Speed (mbps)	Neighbor ID	Interface
vcp-0	Dedicated	1	Up	32000	1	vcp-1
vcp-1	Dedicated	2	Up	32000	0	vcp-0
1/0	Auto-Configured	3	Up	1000	2	vcp-255/1/0
1/0	Auto-Configured	3	Up	1000	2	vcp-255/1/1

### show virtual-chassis vc-port (EX8200 Virtual Chassis)

```
user@external-routing-engine> show virtual-chassis vc-port
```

```
member0:
```

Interface or Slot/PIC/Port	Type	Trunk ID	Status	Speed (mbps)	Neighbor ID	Interface
vcp-0/0	Dedicated	-1	Up	1000	8	vcp-1/1
vcp-0/1	Dedicated	-1	Up	1000	8	vcp-2/0
4/0/4	Configured	-1	Up	10000	1	vcp-3/0/4
4/0/7	Configured	-1	Down	10000		
4/0/3	Configured		Absent			
4/0/2	Configured		Absent			
4/0/5	Configured		Absent			
4/0/6	Configured		Absent			
4/0/1	Configured		Absent			
4/0/0	Configured		Absent			

```
member1:
```

Interface or Slot/PIC/Port	Type	Trunk ID	Status	Speed (mbps)	Neighbor ID	Interface
vcp-0/0	Dedicated	-1	Up	1000	8	vcp-1/2
3/0/0	Configured	-1	Down	10000		
3/0/1	Configured	-1	Down	10000		
3/0/4	Configured	-1	Up	10000	0	vcp-4/0/4
3/0/5	Configured		Absent			
4/0/5	Configured		Absent			
4/0/4	Configured		Absent			

```
member8:
```

Interface or Slot/PIC/Port	Type	Trunk ID	Status	Speed (mbps)	Neighbor ID	Interface
vcp-0/0	Dedicated	-1	Down	1000		
vcp-1/0	Dedicated	-1	Up	1000	9	vcp-1/0
vcp-1/1	Dedicated	-1	Up	1000	0	vcp-0/0
vcp-1/2	Dedicated	-1	Up	1000	1	vcp-0/0
vcp-1/3	Dedicated	-1	Up	1000	9	vcp-1/3
vcp-2/0	Dedicated	-1	Up	1000	0	vcp-0/1
vcp-2/1	Dedicated	-1	Up	1000	9	vcp-1/2
vcp-2/2	Dedicated	-1	Down	1000		

```
vcp-2/3      Dedicated      -1   Down      1000
```

```
member9:
```

```
-----
Interface    Type           Trunk  Status    Speed    Neighbor
or           or              ID      (mbps)    ID  Interface
Slot/PIC/Port
vcp-0/0      Dedicated      -1     Disabled  1000
vcp-1/0      Dedicated      -1     Up        1000      8   vcp-1/0
vcp-1/1      Dedicated      -1     Down      1000
vcp-1/2      Dedicated      -1     Up        1000      8   vcp-2/1
vcp-1/3      Dedicated      -1     Up        1000      8   vcp-1/3
```

### show virtual-chassis vc-port all-members

```
user@switch> show virtual-chassis vc-port all-members
```

```
fpc0:
```

```
-----
Interface    Type           Trunk  Status    Speed    Neighbor
or           or              ID      (mbps)    ID  Interface
PIC / Port
vcp-0        Dedicated      1      Up        32000    1   vcp-1
vcp-1        Dedicated      2      Up        32000    0   vcp-0
1/0          Auto-Configured 3      Up        1000     2   vcp-255/1/0
1/1          Auto-Configured 3      Up        1000     2   vcp-255/1/1
```

```
fpc1:
```

```
-----
Interface    Type           Trunk  Status    Speed    Neighbor
or           or              ID      (mbps)    ID  Interface
PIC / Port
vcp-0        Dedicated      1      Up        32000    0   vcp-1
vcp-1        Dedicated      2      Up        32000    0   vcp-0
1/0          Auto-Configured -1     Up        1000     3   vcp-255/1/0
```

```
fpc2:
```

```
-----
Interface    Type           Trunk  Status    Speed    Neighbor
or           or              ID      (mbps)    ID  Interface
PIC / Port
vcp-0        Dedicated      1      Up        32000    3   vcp-1
vcp-1        Dedicated      2      Up        32000    3   vcp-0
1/0          Auto-Configured 3      Up        1000     0   vcp-255/1/0
1/1          Auto-Configured 3      Up        1000     0   vcp-255/1/1
```

```
fpc3:
```

```
-----
Interface    Type           Trunk  Status    Speed    Neighbor
or           or              ID      (mbps)    ID  Interface
PIC / Port
vcp-0        Dedicated      1      Up        32000    2   vcp-0
vcp-1        Dedicated      2      Up        32000    2   vcp-1
1/0          Auto-Configured -1     Up        1000     1   vcp-255/1/0
```

## show virtual-chassis vc-port diagnostics optics

<b>Syntax</b>	<pre>show virtual-chassis vc-port diagnostics optics &lt;all-members&gt; &lt;interface-name&gt; &lt;local&gt; &lt;member member-id&gt;</pre>
<b>Release Information</b>	<p>Command introduced in Junos OS Release 12.2 for EX Series switches.</p> <p>Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).</p>
<b>Description</b>	<p>Display diagnostics data and alarms for Ethernet optical transceivers installed in ports configured as Virtual Chassis Ports (VCPs) in an EX Series switches. The information provided by this command is known as digital optical monitoring (DOM) information.</p> <p>Thresholds that trigger a high alarm, low alarm, high warning, or low warning are set by the transponder vendors. Generally, a high alarm or low alarm indicates that a transceiver is not operating properly. DOM information can be used to diagnose why a transceiver is not working.</p> <p>On some EX Series switches, the <b>request virtual-chassis vc-port diagnostics optics</b> command must be entered to run a diagnostic scan before you can gather the <b>show virtual-chassis vc-port diagnostics optics</b> output.</p>
<b>Options</b>	<p><b>none</b>—Display diagnostics information for transceivers installed in VCPs of all members of a Virtual Chassis or VCF.</p> <p><b>all-members</b>—(Optional) Display diagnostics information for transceivers installed in VCPs of all members of a Virtual Chassis or VCF.</p> <p><b>interface-name</b>—(Optional) Display diagnostics information for the transceiver installed in a specified VCP.</p> <p><b>local</b>—(Optional) Display diagnostics information for transceivers installed in VCPs on the switch or external Routing Engine on which this command is entered.</p> <p><b>member member-id</b>—(Optional) Display diagnostics information for transceivers installed in VCPs on a specified member of a Virtual Chassis or VCF.</p>
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">show virtual-chassis vc-port on page 143</a></li> <li>• <i>Installing a Transceiver in a Switch</i></li> <li>• <i>Removing a Transceiver from a Switch</i></li> <li>• <i>Junos OS Ethernet Interfaces Configuration Guide</i></li> </ul>
<b>List of Sample Output</b>	<p><a href="#">show virtual-chassis vc-port diagnostics optics on page 150</a></p> <p><a href="#">show virtual-chassis vc-port diagnostics optics (interface-name) on page 155</a></p>

[show virtual-chassis vc-port diagnostics optics local on page 157](#)

[show virtual-chassis vc-port diagnostics optics \(member member-id\) on page 159](#)

**Output Fields** [Table 18 on page 148](#) lists the output fields for the **show virtual-chassis vc-port diagnostics optics** command. Output fields are listed in the approximate order in which they appear.

**Table 18: show virtual-chassis vc-port diagnostics optics Output Fields**

Field Name	Field Description
FPC	Displays the FPC slot number.
Virtual chassis port	Displays the name of the VCP.
Laser bias current	Displays the magnitude of the laser bias power setting current, in milliamperes (mA). The laser bias provides direct modulation of laser diodes and modulates currents.
Laser output power	Displays the laser output power, in milliwatts (mW) and decibels referred to 1.0 mW (dBm).
Module temperature	Displays the temperature, in Celsius and Fahrenheit.
Module voltage	Displays the voltage, in Volts.
Receiver signal average optical power	Displays the receiver signal average optical power, in milliwatts (mW) and decibels referred to 1.0 mW (dBm).
Laser bias current high alarm	Displays whether the laser bias power setting high alarm is <i>On</i> or <i>Off</i> .
Laser bias current low alarm	Displays whether the laser bias power setting low alarm is <i>On</i> or <i>Off</i> .
Laser bias current high warning	Displays whether the laser bias power setting high warning is <i>On</i> or <i>Off</i> .
Laser bias current low warning	Displays whether the laser bias power setting low warning is <i>On</i> or <i>Off</i> .
Laser output power high alarm	Displays whether the laser output power high alarm is <i>On</i> or <i>Off</i> .
Laser output power low alarm	Displays whether the laser output power low alarm is <i>On</i> or <i>Off</i> .
Laser output power high warning	Displays whether the laser output power high warning is <i>On</i> or <i>Off</i> .
Laser output power low warning	Displays whether the laser output power low warning is <i>On</i> or <i>Off</i> .
Module temperature high alarm	Displays whether the module temperature high alarm is <i>On</i> or <i>Off</i> .
Module temperature low alarm	Displays whether the module temperature low alarm is <i>On</i> or <i>Off</i> .
Module temperature high warning	Displays whether the module temperature high warning is <i>On</i> or <i>Off</i> .
Module temperature low warning	Displays whether the module temperature low warning is <i>On</i> or <i>Off</i> .

Table 18: show virtual-chassis vc-port diagnostics optics Output Fields (*continued*)

Field Name	Field Description
Module voltage high alarm	Displays whether the module voltage high alarm is <i>On</i> or <i>Off</i> .
Module voltage low alarm	Displays whether the module voltage low alarm is <i>On</i> or <i>Off</i> .
Module voltage high warning	Displays whether the module voltage high warning is <i>On</i> or <i>Off</i> .
Module voltage low warning	Displays whether the module voltage low warning is <i>On</i> or <i>Off</i> .
Laser rx power high alarm	Displays whether the receive laser power high alarm is <i>On</i> or <i>Off</i> .
Laser rx power low alarm	Displays whether the receive laser power low alarm is <i>On</i> or <i>Off</i> .
Laser rx power high warning	Displays whether the receive laser power high warning is <i>On</i> or <i>Off</i> .
Laser rx power low warning	Displays whether the receive laser power low warning is <i>On</i> or <i>Off</i> .
Laser bias current high alarm threshold	Displays the vendor-specified threshold for the laser bias current high alarm.
Laser bias current low alarm threshold	Displays the vendor-specified threshold for the laser bias current low alarm.
Laser bias current high warning threshold	Displays the vendor-specified threshold for the laser bias current high warning.
Laser bias current low warning threshold	Displays the vendor-specified threshold for the laser bias current low warning.
Laser output power high alarm threshold	Displays the vendor-specified threshold for the laser output power high alarm.
Laser output power low alarm threshold	Displays the vendor-specified threshold for the laser output power low alarm.
Laser output power high warning threshold	Displays the vendor-specified threshold for the laser output power high warning.
Laser output power low warning threshold	Displays the vendor-specified threshold for the laser output power low warning.
Module temperature high alarm threshold	Displays the vendor-specified threshold for the module temperature high alarm.
Module temperature low alarm threshold	Displays the vendor-specified threshold for the module temperature low alarm.
Module temperature high warning threshold	Displays the vendor-specified threshold for the module temperature high warning.
Module temperature low warning threshold	Displays the vendor-specified threshold for the module temperature low warning.

Table 18: show virtual-chassis vc-port diagnostics optics Output Fields (*continued*)

Field Name	Field Description
Module voltage high alarm threshold	Displays the vendor-specified threshold for the module voltage high alarm.
Module voltage low alarm threshold	Displays the vendor-specified threshold for the module voltage low alarm.
Module voltage high warning threshold	Displays the vendor-specified threshold for the module voltage high warning.
Module voltage low warning threshold	Displays the vendor-specified threshold for the module voltage low warning.
Laser rx power high alarm threshold	Displays the vendor-specified threshold for the laser rx power high alarm.
Laser rx power low alarm threshold	Displays the vendor-specified threshold for the laser rx power low alarm.
Laser rx power high warning threshold	Displays the vendor-specified threshold for the laser rx power high warning.
Laser rx power low warning threshold	Displays the vendor-specified threshold for the laser rx power low warning.

## Sample Output

### show virtual-chassis vc-port diagnostics optics

```

user@switch> show virtual-chassis vc-port diagnostics optics
fpc0:
-----
Virtual chassis port: vcp-0
  Optical diagnostics                : N/A
Virtual chassis port: vcp-1
  Optical diagnostics                : N/A

fpc1:
-----
Virtual chassis port: vcp-0
  Optical diagnostics                : N/A
Virtual chassis port: vcp-1
  Optical diagnostics                : N/A

fpc2:
-----
Virtual chassis port: vcp-2/0
  Optical diagnostics                : N/A
Virtual chassis port: vcp-2/1
  Optical diagnostics                : N/A
Virtual chassis port: vcp-255/0/14
  Optical diagnostics                : N/A
Virtual chassis port: vcp-255/0/15
  Optical diagnostics                : N/A
Virtual chassis port: vcp-255/0/24
  Laser bias current                 : 4.130 mA
  Laser output power                 : 0.2450 mW / -6.11 dBm
  Module temperature                 : 32 degrees C / 90 degrees F
  Module voltage                     : 3.3530 V
  Receiver signal average optical power : 0.0971 mW / -10.13 dBm
  Laser bias current high alarm      : Off
  Laser bias current low alarm       : Off

```

```

Laser bias current high warning      : Off
Laser bias current low warning       : Off
Laser output power high alarm        : Off
Laser output power low alarm         : Off
Laser output power high warning      : Off
Laser output power low warning       : Off
Module temperature high alarm        : Off
Module temperature low alarm         : Off
Module temperature high warning      : Off
Module temperature low warning       : Off
Module voltage high alarm            : Off
Module voltage low alarm             : Off
Module voltage high warning          : Off
Module voltage low warning           : Off
Laser rx power high alarm            : Off
Laser rx power low alarm             : Off
Laser rx power high warning          : Off
Laser rx power low warning           : Off
Laser bias current high alarm threshold : 14.998 mA
Laser bias current low alarm threshold : 0.998 mA
Laser bias current high warning threshold : 14.000 mA
Laser bias current low warning threshold : 1.198 mA
Laser output power high alarm threshold : 0.7940 mW / -1.00 dBm
Laser output power low alarm threshold : 0.0790 mW / -11.02 dBm
Laser output power high warning threshold : 0.6300 mW / -2.01 dBm
Laser output power low warning threshold : 0.0990 mW / -10.04 dBm
Module temperature high alarm threshold : 85 degrees C / 185 degrees F
Module temperature low alarm threshold : -10 degrees C / 14 degrees F
Module temperature high warning threshold : 80 degrees C / 176 degrees F
Module temperature low warning threshold : -5 degrees C / 23 degrees F
Module voltage high alarm threshold : 3.600 V
Module voltage low alarm threshold : 3.000 V
Module voltage high warning threshold : 3.499 V
Module voltage low warning threshold : 3.099 V
Laser rx power high alarm threshold : 1.5848 mW / 2.00 dBm
Laser rx power low alarm threshold : 0.0100 mW / -20.00 dBm
Laser rx power high warning threshold : 1.2589 mW / 1.00 dBm
Laser rx power low warning threshold : 0.0125 mW / -19.03 dBm
Virtual chassis port: vcp-255/0/3
Laser bias current                  : 5.428 mA
Laser output power                  : 0.4760 mW / -3.22 dBm
Module temperature                  : 28 degrees C / 83 degrees F
Module voltage                      : 3.3440 V
Receiver signal average optical power : 0.4002 mW / -3.98 dBm
Laser bias current high alarm       : Off
Laser bias current low alarm        : Off
Laser bias current high warning     : Off
Laser bias current low warning      : Off
Laser output power high alarm       : Off
Laser output power low alarm        : Off
Laser output power high warning     : Off
Laser output power low warning      : Off
Module temperature high alarm       : Off
Module temperature low alarm        : Off
Module temperature high warning     : Off
Module temperature low warning      : Off
Module voltage high alarm           : Off
Module voltage low alarm            : Off
Module voltage high warning         : Off
Module voltage low warning          : Off
Laser rx power high alarm           : Off

```

```

Laser rx power low alarm           : Off
Laser rx power high warning        : Off
Laser rx power low warning         : Off
Laser bias current high alarm threshold : 10.500 mA
Laser bias current low alarm threshold : 2.000 mA
Laser bias current high warning threshold : 9.000 mA
Laser bias current low warning threshold : 2.500 mA
Laser output power high alarm threshold : 1.4120 mW / 1.50 dBm
Laser output power low alarm threshold : 0.0740 mW / -11.31 dBm
Laser output power high warning threshold : 0.7070 mW / -1.51 dBm
Laser output power low warning threshold : 0.1860 mW / -7.30 dBm
Module temperature high alarm threshold : 75 degrees C / 167 degrees F
Module temperature low alarm threshold : -5 degrees C / 23 degrees F
Module temperature high warning threshold : 70 degrees C / 158 degrees F
Module temperature low warning threshold : 0 degrees C / 32 degrees F
Module voltage high alarm threshold : 3.630 V
Module voltage low alarm threshold : 2.970 V
Module voltage high warning threshold : 3.465 V
Module voltage low warning threshold : 3.135 V
Laser rx power high alarm threshold : 1.5849 mW / 2.00 dBm
Laser rx power low alarm threshold : 0.0407 mW / -13.90 dBm
Laser rx power high warning threshold : 0.7943 mW / -1.00 dBm
Laser rx power low warning threshold : 0.1023 mW / -9.90 dBm

```

fpc3:

-----  
Virtual chassis port: vcp-255/0/2

```

Laser bias current           : 7.876 mA
Laser output power           : 0.5330 mW / -2.73 dBm
Module temperature            : 26 degrees C / 78 degrees F
Module voltage                : 3.3060 V
Receiver signal average optical power : 0.4885 mW / -3.11 dBm
Laser bias current high alarm : Off
Laser bias current low alarm  : Off
Laser bias current high warning : Off
Laser bias current low warning : Off
Laser output power high alarm  : Off
Laser output power low alarm   : Off
Laser output power high warning : Off
Laser output power low warning : Off
Module temperature high alarm  : Off
Module temperature low alarm   : Off
Module temperature high warning : Off
Module temperature low warning : Off
Module voltage high alarm      : Off
Module voltage low alarm       : Off
Module voltage high warning    : Off
Module voltage low warning     : Off
Laser rx power high alarm      : Off
Laser rx power low alarm       : Off
Laser rx power high warning    : Off
Laser rx power low warning     : Off
Laser bias current high alarm threshold : 14.500 mA
Laser bias current low alarm threshold : 3.500 mA
Laser bias current high warning threshold : 14.500 mA
Laser bias current low warning threshold : 3.500 mA
Laser output power high alarm threshold : 1.8620 mW / 2.70 dBm
Laser output power low alarm threshold : 0.0740 mW / -11.31 dBm
Laser output power high warning threshold : 0.7410 mW / -1.30 dBm
Laser output power low warning threshold : 0.1860 mW / -7.30 dBm
Module temperature high alarm threshold : 75 degrees C / 167 degrees F

```



```

Module temperature low alarm threshold : -5 degrees C / 23 degrees F
Module temperature high warning threshold : 70 degrees C / 158 degrees F
Module temperature low warning threshold : 0 degrees C / 32 degrees F
Module voltage high alarm threshold : 3.630 V
Module voltage low alarm threshold : 2.970 V
Module voltage high warning threshold : 3.465 V
Module voltage low warning threshold : 3.135 V
Laser rx power high alarm threshold : 1.9952 mW / 3.00 dBm
Laser rx power low alarm threshold : 0.0407 mW / -13.90 dBm
Laser rx power high warning threshold : 0.7943 mW / -1.00 dBm
Laser rx power low warning threshold : 0.1023 mW / -9.90 dBm
Virtual chassis port: vcp-255/0/3
Laser bias current : 5.052 mA
Laser output power : 0.5030 mW / -2.98 dBm
Module temperature : 24 degrees C / 75 degrees F
Module voltage : 3.2890 V
Receiver signal average optical power : 0.5028 mW / -2.99 dBm
Laser bias current high alarm : Off
Laser bias current low alarm : Off
Laser bias current high warning : Off
Laser bias current low warning : Off
Laser output power high alarm : Off
Laser output power low alarm : Off
Laser output power high warning : Off
Laser output power low warning : Off
Module temperature high alarm : Off
Module temperature low alarm : Off
Module temperature high warning : Off
Module temperature low warning : Off
Module voltage high alarm : Off
Module voltage low alarm : Off
Module voltage high warning : Off
Module voltage low warning : Off
Laser rx power high alarm : Off
Laser rx power low alarm : Off
Laser rx power high warning : Off
Laser rx power low warning : Off
Laser bias current high alarm threshold : 10.500 mA
Laser bias current low alarm threshold : 2.000 mA
Laser bias current high warning threshold : 9.000 mA
Laser bias current low warning threshold : 2.500 mA
Laser output power high alarm threshold : 1.4120 mW / 1.50 dBm
Laser output power low alarm threshold : 0.0740 mW / -11.31 dBm
Laser output power high warning threshold : 0.7070 mW / -1.51 dBm
Laser output power low warning threshold : 0.1860 mW / -7.30 dBm
Module temperature high alarm threshold : 75 degrees C / 167 degrees F
Module temperature low alarm threshold : -5 degrees C / 23 degrees F
Module temperature high warning threshold : 70 degrees C / 158 degrees F
Module temperature low warning threshold : 0 degrees C / 32 degrees F
Module voltage high alarm threshold : 3.630 V
Module voltage low alarm threshold : 2.970 V
Module voltage high warning threshold : 3.465 V
Module voltage low warning threshold : 3.135 V
Laser rx power high alarm threshold : 1.5849 mW / 2.00 dBm
Laser rx power low alarm threshold : 0.0407 mW / -13.90 dBm
Laser rx power high warning threshold : 0.7943 mW / -1.00 dBm
Laser rx power low warning threshold : 0.1023 mW / -9.90 dBm
Virtual chassis port: vcp-255/0/4
Laser bias current : 7.978 mA
Laser output power : 0.5460 mW / -2.63 dBm
Module temperature : 24 degrees C / 76 degrees F

```

```

Module voltage                                     : 3.3060 V
Receiver signal average optical power             : 0.6305 mW / -2.00 dBm
Laser bias current high alarm                     : Off
Laser bias current low alarm                      : Off
Laser bias current high warning                   : Off
Laser bias current low warning                    : Off
Laser output power high alarm                     : Off
Laser output power low alarm                      : Off
Laser output power high warning                   : Off
Laser output power low warning                    : Off
Module temperature high alarm                     : Off
Module temperature low alarm                      : Off
Module temperature high warning                   : Off
Module temperature low warning                    : Off
Module voltage high alarm                         : Off
Module voltage low alarm                          : Off
Module voltage high warning                       : Off
Module voltage low warning                        : Off
Laser rx power high alarm                         : Off
Laser rx power low alarm                         : Off
Laser rx power high warning                       : Off
Laser rx power low warning                        : Off
Laser bias current high alarm threshold           : 14.500 mA
Laser bias current low alarm threshold            : 3.500 mA
Laser bias current high warning threshold         : 14.500 mA
Laser bias current low warning threshold          : 3.500 mA
Laser output power high alarm threshold           : 1.8620 mW / 2.70 dBm
Laser output power low alarm threshold            : 0.0740 mW / -11.31 dBm
Laser output power high warning threshold         : 0.7410 mW / -1.30 dBm
Laser output power low warning threshold          : 0.1860 mW / -7.30 dBm
Module temperature high alarm threshold           : 75 degrees C / 167 degrees F
Module temperature low alarm threshold             : -5 degrees C / 23 degrees F
Module temperature high warning threshold         : 70 degrees C / 158 degrees F
Module temperature low warning threshold          : 0 degrees C / 32 degrees F
Module voltage high alarm threshold               : 3.630 V
Module voltage low alarm threshold                : 2.970 V
Module voltage high warning threshold             : 3.465 V
Module voltage low warning threshold              : 3.135 V
Laser rx power high alarm threshold               : 1.9952 mW / 3.00 dBm
Laser rx power low alarm threshold                : 0.0407 mW / -13.90 dBm
Laser rx power high warning threshold             : 0.7943 mW / -1.00 dBm
Laser rx power low warning threshold              : 0.1023 mW / -9.90 dBm

```

fpc4:

```

-----
Virtual chassis port: vcp-0
  Optical diagnostics                               : N/A
Virtual chassis port: vcp-1
  Optical diagnostics                               : N/A
Virtual chassis port: vcp-255/0/4
  Laser bias current                               : 7.860 mA
  Laser output power                               : 0.5370 mW / -2.70 dBm
  Module temperature                               : 24 degrees C / 75 degrees F
  Module voltage                                   : 3.2920 V
  Receiver signal average optical power            : 0.6271 mW / -2.03 dBm
  Laser bias current high alarm                    : Off
  Laser bias current low alarm                     : Off
  Laser bias current high warning                  : Off
  Laser bias current low warning                   : Off
  Laser output power high alarm                    : Off
  Laser output power low alarm                     : Off
  Laser output power high warning                  : Off

```

```

Laser output power low warning      : Off
Module temperature high alarm       : Off
Module temperature low alarm        : Off
Module temperature high warning     : Off
Module temperature low warning      : Off
Module voltage high alarm           : Off
Module voltage low alarm            : Off
Module voltage high warning         : Off
Module voltage low warning          : Off
Laser rx power high alarm           : Off
Laser rx power low alarm            : Off
Laser rx power high warning         : Off
Laser rx power low warning          : Off
Laser bias current high alarm threshold : 14.500 mA
Laser bias current low alarm threshold : 3.500 mA
Laser bias current high warning threshold : 14.500 mA
Laser bias current low warning threshold : 3.500 mA
Laser output power high alarm threshold : 1.8620 mW / 2.70 dBm
Laser output power low alarm threshold : 0.0740 mW / -11.31 dBm
Laser output power high warning threshold : 0.7410 mW / -1.30 dBm
Laser output power low warning threshold : 0.1860 mW / -7.30 dBm
Module temperature high alarm threshold : 75 degrees C / 167 degrees F
Module temperature low alarm threshold : -5 degrees C / 23 degrees F
Module temperature high warning threshold : 70 degrees C / 158 degrees F
Module temperature low warning threshold : 0 degrees C / 32 degrees F
Module voltage high alarm threshold : 3.630 V
Module voltage low alarm threshold : 2.970 V
Module voltage high warning threshold : 3.465 V
Module voltage low warning threshold : 3.135 V
Laser rx power high alarm threshold : 1.9952 mW / 3.00 dBm
Laser rx power low alarm threshold : 0.0407 mW / -13.90 dBm
Laser rx power high warning threshold : 0.7943 mW / -1.00 dBm
Laser rx power low warning threshold : 0.1023 mW / -9.90 dBm

```

### show virtual-chassis vc-port diagnostics optics (interface-name)

```

user@external-routing-engine> show virtual-chassis vc-port diagnostics optics vcp-255/0/3
fpc0:
-----

```

```

fpc1:
-----

```

```

fpc2:
-----

```

```

Virtual chassis port: vcp-255/0/3
Laser bias current      : 5.448 mA
Laser output power      : 0.4770 mW / -3.21 dBm
Module temperature      : 28 degrees C / 82 degrees F
Module voltage          : 3.3450 V
Receiver signal average optical power : 0.3973 mW / -4.01 dBm
Laser bias current high alarm : Off
Laser bias current low alarm  : Off
Laser bias current high warning : Off
Laser bias current low warning : Off
Laser output power high alarm : Off
Laser output power low alarm  : Off
Laser output power high warning : Off
Laser output power low warning : Off
Module temperature high alarm : Off
Module temperature low alarm  : Off

```

```

Module temperature high warning      : Off
Module temperature low warning       : Off
Module voltage high alarm            : Off
Module voltage low alarm             : Off
Module voltage high warning          : Off
Module voltage low warning           : Off
Laser rx power high alarm            : Off
Laser rx power low alarm             : Off
Laser rx power high warning          : Off
Laser rx power low warning           : Off
Laser bias current high alarm threshold : 10.500 mA
Laser bias current low alarm threshold : 2.000 mA
Laser bias current high warning threshold : 9.000 mA
Laser bias current low warning threshold : 2.500 mA
Laser output power high alarm threshold : 1.4120 mW / 1.50 dBm
Laser output power low alarm threshold : 0.0740 mW / -11.31 dBm
Laser output power high warning threshold : 0.7070 mW / -1.51 dBm
Laser output power low warning threshold : 0.1860 mW / -7.30 dBm
Module temperature high alarm threshold : 75 degrees C / 167 degrees F
Module temperature low alarm threshold : -5 degrees C / 23 degrees F
Module temperature high warning threshold : 70 degrees C / 158 degrees F
Module temperature low warning threshold : 0 degrees C / 32 degrees F
Module voltage high alarm threshold : 3.630 V
Module voltage low alarm threshold : 2.970 V
Module voltage high warning threshold : 3.465 V
Module voltage low warning threshold : 3.135 V
Laser rx power high alarm threshold : 1.5849 mW / 2.00 dBm
Laser rx power low alarm threshold : 0.0407 mW / -13.90 dBm
Laser rx power high warning threshold : 0.7943 mW / -1.00 dBm
Laser rx power low warning threshold : 0.1023 mW / -9.90 dBm

```

fpc3:

-----  
Virtual chassis port: vcp-255/0/3

```

Laser bias current      : 5.040 mA
Laser output power      : 0.5020 mW / -2.99 dBm
Module temperature      : 24 degrees C / 74 degrees F
Module voltage          : 3.2870 V
Receiver signal average optical power : 0.5073 mW / -2.95 dBm
Laser bias current high alarm : Off
Laser bias current low alarm  : Off
Laser bias current high warning : Off
Laser bias current low warning : Off
Laser output power high alarm : Off
Laser output power low alarm  : Off
Laser output power high warning : Off
Laser output power low warning : Off
Module temperature high alarm : Off
Module temperature low alarm  : Off
Module temperature high warning : Off
Module temperature low warning : Off
Module voltage high alarm : Off
Module voltage low alarm  : Off
Module voltage high warning : Off
Module voltage low warning : Off
Laser rx power high alarm : Off
Laser rx power low alarm  : Off
Laser rx power high warning : Off
Laser rx power low warning : Off
Laser bias current high alarm threshold : 10.500 mA
Laser bias current low alarm threshold : 2.000 mA

```

```

Laser bias current high warning threshold : 9.000 mA
Laser bias current low warning threshold  : 2.500 mA
Laser output power high alarm threshold   : 1.4120 mW / 1.50 dBm
Laser output power low alarm threshold    : 0.0740 mW / -11.31 dBm
Laser output power high warning threshold : 0.7070 mW / -1.51 dBm
Laser output power low warning threshold  : 0.1860 mW / -7.30 dBm
Module temperature high alarm threshold   : 75 degrees C / 167 degrees F
Module temperature low alarm threshold    : -5 degrees C / 23 degrees F
Module temperature high warning threshold : 70 degrees C / 158 degrees F
Module temperature low warning threshold  : 0 degrees C / 32 degrees F
Module voltage high alarm threshold       : 3.630 V
Module voltage low alarm threshold        : 2.970 V
Module voltage high warning threshold     : 3.465 V
Module voltage low warning threshold      : 3.135 V
Laser rx power high alarm threshold       : 1.5849 mW / 2.00 dBm
Laser rx power low alarm threshold        : 0.0407 mW / -13.90 dBm
Laser rx power high warning threshold     : 0.7943 mW / -1.00 dBm
Laser rx power low warning threshold      : 0.1023 mW / -9.90 dBm

```

fpc4:

-----

#### show virtual-chassis vc-port diagnostics optics local

```

user@switch> show virtual-chassis vc-port diagnostics optics local
Virtual chassis port: vcp-2/0
  Optical diagnostics : N/A
Virtual chassis port: vcp-2/1
  Optical diagnostics : N/A
Virtual chassis port: vcp-255/0/14
  Optical diagnostics : N/A
Virtual chassis port: vcp-255/0/15
  Optical diagnostics : N/A
Virtual chassis port: vcp-255/0/24
  Laser bias current : 4.130 mA
  Laser output power : 0.2450 mW / -6.11 dBm
  Module temperature : 32 degrees C / 90 degrees F
  Module voltage     : 3.3530 V
  Receiver signal average optical power : 0.0961 mW / -10.17 dBm
  Laser bias current high alarm : Off
  Laser bias current low alarm  : Off
  Laser bias current high warning : Off
  Laser bias current low warning : Off
  Laser output power high alarm : Off
  Laser output power low alarm  : Off
  Laser output power high warning : Off
  Laser output power low warning : Off
  Module temperature high alarm : Off
  Module temperature low alarm  : Off
  Module temperature high warning : Off
  Module temperature low warning : Off
  Module voltage high alarm : Off
  Module voltage low alarm  : Off
  Module voltage high warning : Off
  Module voltage low warning : Off
  Laser rx power high alarm : Off
  Laser rx power low alarm  : Off
  Laser rx power high warning : Off
  Laser rx power low warning : Off
  Laser bias current high alarm threshold : 14.998 mA
  Laser bias current low alarm threshold  : 0.998 mA

```

```
Laser bias current high warning threshold : 14.000 mA
Laser bias current low warning threshold  : 1.198 mA
Laser output power high alarm threshold   : 0.7940 mW / -1.00 dBm
Laser output power low alarm threshold    : 0.0790 mW / -11.02 dBm
Laser output power high warning threshold : 0.6300 mW / -2.01 dBm
Laser output power low warning threshold  : 0.0990 mW / -10.04 dBm
Module temperature high alarm threshold   : 85 degrees C / 185 degrees F
Module temperature low alarm threshold    : -10 degrees C / 14 degrees F
Module temperature high warning threshold : 80 degrees C / 176 degrees F
Module temperature low warning threshold  : -5 degrees C / 23 degrees F
Module voltage high alarm threshold        : 3.600 V
Module voltage low alarm threshold         : 3.000 V
Module voltage high warning threshold      : 3.499 V
Module voltage low warning threshold       : 3.099 V
Laser rx power high alarm threshold        : 1.5848 mW / 2.00 dBm
Laser rx power low alarm threshold         : 0.0100 mW / -20.00 dBm
Laser rx power high warning threshold      : 1.2589 mW / 1.00 dBm
Laser rx power low warning threshold       : 0.0125 mW / -19.03 dBm
Virtual chassis port: vcp-255/0/3
Laser bias current                        : 5.426 mA
Laser output power                        : 0.4760 mW / -3.22 dBm
Module temperature                        : 28 degrees C / 83 degrees F
Module voltage                            : 3.3450 V
Receiver signal average optical power     : 0.3955 mW / -4.03 dBm
Laser bias current high alarm             : Off
Laser bias current low alarm              : Off
Laser bias current high warning           : Off
Laser bias current low warning            : Off
Laser output power high alarm             : Off
Laser output power low alarm              : Off
Laser output power high warning           : Off
Laser output power low warning            : Off
Module temperature high alarm             : Off
Module temperature low alarm              : Off
Module temperature high warning           : Off
Module temperature low warning            : Off
Module voltage high alarm                 : Off
Module voltage low alarm                  : Off
Module voltage high warning               : Off
Module voltage low warning                : Off
Laser rx power high alarm                 : Off
Laser rx power low alarm                  : Off
Laser rx power high warning               : Off
Laser rx power low warning                : Off
Laser bias current high alarm threshold   : 10.500 mA
Laser bias current low alarm threshold    : 2.000 mA
Laser bias current high warning threshold : 9.000 mA
Laser bias current low warning threshold  : 2.500 mA
Laser output power high alarm threshold   : 1.4120 mW / 1.50 dBm
Laser output power low alarm threshold    : 0.0740 mW / -11.31 dBm
Laser output power high warning threshold : 0.7070 mW / -1.51 dBm
Laser output power low warning threshold  : 0.1860 mW / -7.30 dBm
Module temperature high alarm threshold   : 75 degrees C / 167 degrees F
Module temperature low alarm threshold    : -5 degrees C / 23 degrees F
Module temperature high warning threshold : 70 degrees C / 158 degrees F
Module temperature low warning threshold  : 0 degrees C / 32 degrees F
Module voltage high alarm threshold        : 3.630 V
Module voltage low alarm threshold         : 2.970 V
Module voltage high warning threshold      : 3.465 V
Module voltage low warning threshold       : 3.135 V
Laser rx power high alarm threshold        : 1.5849 mW / 2.00 dBm
```

```

Laser rx power low alarm threshold      : 0.0407 mW / -13.90 dBm
Laser rx power high warning threshold   : 0.7943 mW / -1.00 dBm
Laser rx power low warning threshold    : 0.1023 mW / -9.90 dBm

```

### show virtual-chassis vc-port diagnostics optics (member member-id)

```

user@switch> show virtual-chassis vc-port diagnostics optics member 2
fpc2:

```

```

-----
Virtual chassis port: vcp-2/0
  Optical diagnostics                : N/A
Virtual chassis port: vcp-2/1
  Optical diagnostics                : N/A
Virtual chassis port: vcp-255/0/14
  Optical diagnostics                : N/A
Virtual chassis port: vcp-255/0/15
  Optical diagnostics                : N/A
Virtual chassis port: vcp-255/0/24
  Laser bias current                 : 4.130 mA
  Laser output power                 : 0.2450 mW / -6.11 dBm
  Module temperature                 : 31 degrees C / 88 degrees F
  Module voltage                     : 3.3530 V
  Receiver signal average optical power : 0.0961 mW / -10.17 dBm
  Laser bias current high alarm      : Off
  Laser bias current low alarm       : Off
  Laser bias current high warning    : Off
  Laser bias current low warning     : Off
  Laser output power high alarm      : Off
  Laser output power low alarm       : Off
  Laser output power high warning    : Off
  Laser output power low warning     : Off
  Module temperature high alarm      : Off
  Module temperature low alarm       : Off
  Module temperature high warning    : Off
  Module temperature low warning     : Off
  Module voltage high alarm          : Off
  Module voltage low alarm           : Off
  Module voltage high warning        : Off
  Module voltage low warning         : Off
  Laser rx power high alarm          : Off
  Laser rx power low alarm           : Off
  Laser rx power high warning        : Off
  Laser rx power low warning         : Off
  Laser bias current high alarm threshold : 14.998 mA
  Laser bias current low alarm threshold : 0.998 mA
  Laser bias current high warning threshold : 14.000 mA
  Laser bias current low warning threshold : 1.198 mA
  Laser output power high alarm threshold : 0.7940 mW / -1.00 dBm
  Laser output power low alarm threshold : 0.0790 mW / -11.02 dBm
  Laser output power high warning threshold : 0.6300 mW / -2.01 dBm
  Laser output power low warning threshold : 0.0990 mW / -10.04 dBm
  Module temperature high alarm threshold : 85 degrees C / 185 degrees F
  Module temperature low alarm threshold : -10 degrees C / 14 degrees F
  Module temperature high warning threshold : 80 degrees C / 176 degrees F
  Module temperature low warning threshold : -5 degrees C / 23 degrees F
  Module voltage high alarm threshold : 3.600 V
  Module voltage low alarm threshold : 3.000 V
  Module voltage high warning threshold : 3.499 V
  Module voltage low warning threshold : 3.099 V
  Laser rx power high alarm threshold : 1.5848 mW / 2.00 dBm

```

```
Laser rx power low alarm threshold      : 0.0100 mW / -20.00 dBm
Laser rx power high warning threshold   : 1.2589 mW / 1.00 dBm
Laser rx power low warning threshold    : 0.0125 mW / -19.03 dBm
Virtual chassis port: vcp-255/0/3
Laser bias current                      : 5.418 mA
Laser output power                      : 0.4770 mW / -3.21 dBm
Module temperature                      : 28 degrees C / 83 degrees F
Module voltage                          : 3.3450 V
Receiver signal average optical power   : 0.3964 mW / -4.02 dBm
Laser bias current high alarm           : Off
Laser bias current low alarm            : Off
Laser bias current high warning         : Off
Laser bias current low warning          : Off
Laser output power high alarm           : Off
Laser output power low alarm            : Off
Laser output power high warning         : Off
Laser output power low warning          : Off
Module temperature high alarm           : Off
Module temperature low alarm            : Off
Module temperature high warning         : Off
Module temperature low warning          : Off
Module voltage high alarm               : Off
Module voltage low alarm                : Off
Module voltage high warning             : Off
Module voltage low warning              : Off
Laser rx power high alarm               : Off
Laser rx power low alarm                : Off
Laser rx power high warning             : Off
Laser rx power low warning              : Off
Laser bias current high alarm threshold : 10.500 mA
Laser bias current low alarm threshold  : 2.000 mA
Laser bias current high warning threshold : 9.000 mA
Laser bias current low warning threshold : 2.500 mA
Laser output power high alarm threshold : 1.4120 mW / 1.50 dBm
Laser output power low alarm threshold  : 0.0740 mW / -11.31 dBm
Laser output power high warning threshold : 0.7070 mW / -1.51 dBm
Laser output power low warning threshold : 0.1860 mW / -7.30 dBm
Module temperature high alarm threshold : 75 degrees C / 167 degrees F
Module temperature low alarm threshold  : -5 degrees C / 23 degrees F
Module temperature high warning threshold : 70 degrees C / 158 degrees F
Module temperature low warning threshold : 0 degrees C / 32 degrees F
Module voltage high alarm threshold      : 3.630 V
Module voltage low alarm threshold       : 2.970 V
Module voltage high warning threshold    : 3.465 V
Module voltage low warning threshold     : 3.135 V
Laser rx power high alarm threshold      : 1.5849 mW / 2.00 dBm
Laser rx power low alarm threshold       : 0.0407 mW / -13.90 dBm
Laser rx power high warning threshold    : 0.7943 mW / -1.00 dBm
Laser rx power low warning threshold     : 0.1023 mW / -9.90 dBm
```



## show virtual-chassis vc-port statistics

<b>Syntax</b>	<pre>show virtual-chassis vc-port statistics &lt;all-members&gt; &lt;brief   detail   extensive &gt; &lt;interface-name&gt; &lt;local&gt; &lt;member member-id&gt;</pre>
<b>Release Information</b>	<p>Command introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>The options <b>all-members</b>, <b>brief</b>, <b>detail</b>, <b>extensive</b>, and <b>local</b> were added in Junos OS Release 9.3 for EX Series switches.</p> <p>Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series.</p> <p>Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).</p>
<b>Description</b>	Display the traffic statistics collected on Virtual Chassis ports (VCPs).
<b>Options</b>	<p><b>none</b>—Display traffic statistics for VCPs of all members of a Virtual Chassis or VCF.</p> <p><b>brief   detail   extensive</b>—(Optional) Display the specified level of output. Using the <b>brief</b> option is equivalent to entering the command with no options (the default). The <b>detail</b> and <b>extensive</b> options provide identical displays.</p> <p><b>all-members</b>—(Optional) Display traffic statistics for VCPs of all members of a Virtual Chassis or VCF.</p> <p><b>interface-name</b>—(Optional) Display traffic statistics for the specified VCP.</p> <p><b>local</b>—(Optional) Display traffic statistics for VCPs on the switch or external Routing Engine on which this command is entered.</p> <p><b>member member-id</b>—(Optional) Display traffic statistics for VCPs on the specified member of a Virtual Chassis or VCF.</p>
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">clear virtual-chassis vc-port statistics on page 86</a></li> <li>• <a href="#">show virtual-chassis vc-port on page 143</a></li> <li>• <i>Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis</i></li> <li>• <i>Verifying Virtual Chassis Ports in an EX8200 Virtual Chassis</i></li> </ul>
<b>List of Sample Output</b>	<p><a href="#">show virtual-chassis vc-port statistics on page 164</a></p> <p><a href="#">show virtual-chassis vc-port statistics (EX8200 Virtual Chassis) on page 165</a></p> <p><a href="#">show virtual-chassis vc-port statistics brief on page 165</a></p> <p><a href="#">show virtual-chassis vc-port statistics extensive on page 165</a></p> <p><a href="#">show virtual-chassis vc-port statistics member 0 on page 167</a></p>

**Output Fields** Table 18 on page 148 lists the output fields for the **show virtual-chassis vc-port statistics** command. Output fields are listed in the approximate order in which they appear.

**Table 19: show virtual-chassis vc-port statistics Output Fields**

Field Name	Field Description	Level of Output
<b>fpcnumber</b>	(All Virtual Chassis except EX8200 Virtual Chassis. VCF) ID of the Virtual Chassis member. The FPC number is the same as the member ID.	All levels
<b>member number</b>	(EX8200 Virtual Chassis only) Member ID of the Virtual Chassis member.	All levels
<b>Interface</b>	VCP name.	<b>brief</b>
<b>Input Octets/Packets</b>	Number of octets and packets received on the VCP.	<b>brief, member, none</b>
<b>Output Octets/Packets</b>	Number of octets and packets transmitted on the VCP.	<b>brief, member, none</b>
<b>master: number</b>	Member ID of the master Routing Engine.	All levels
<b>Port</b>	VCP for which RX (Receive) statistics, TX (Transmit) statistics, or both are reported by the VCP subsystem during a sampling interval—since the statistics counter was last cleared.	<b>detail, extensive</b>
<b>Total octets</b>	Total number of octets received and transmitted on the VCP.	<b>detail, extensive</b>
<b>Total packets</b>	Total number of packets received and transmitted on the VCP.	<b>detail, extensive</b>
<b>Unicast packets</b>	Number of unicast packets received and transmitted on the VCP.	<b>detail, extensive</b>
<b>Broadcast packets</b>	Number of broadcast packets received and transmitted on the VCP.	<b>detail, extensive</b>
<b>Multicast packets</b>	Number of multicast packets received and transmitted on the VCP.	<b>detail, extensive</b>
<b>MAC control frames</b>	Number of media access control (MAC) control frames received and transmitted on the VCP.	<b>detail, extensive</b>

Table 19: show virtual-chassis vc-port statistics Output Fields (*continued*)

Field Name	Field Description	Level of Output
<b>CRC alignment errors</b>	<p>Number of packets received on the VCP that had a length—excluding framing bits, but including frame check sequence (FCS) octets—of between 64 and 1518 octets, inclusive, and had one of the following errors:</p> <ul style="list-style-type: none"> <li>Invalid FCS with an integral number of octets (FCS error)</li> <li>Invalid FCS with a nonintegral number of octets (alignment error)</li> </ul>	<b>detail, extensive</b>
<b>Oversize packets</b>	Number of packets received on the VCP that were longer than 1518 octets (excluding framing bits, but including FCS octets) but were otherwise well formed.	<b>detail, extensive</b>
<b>Undersize packets</b>	Number of packets received on the VCP that were shorter than 64 octets (excluding framing bits but including FCS octets) and were otherwise well formed..	<b>detail, extensive</b>
<b>Jabber packets</b>	<p>Number of packets received on the VCP that were longer than 1518 octets—excluding framing bits, but including FCS octets—and that had either an FCS error or an alignment error.</p> <p><b>NOTE:</b> This definition of <i>jabber</i> is different from the definition in IEEE-802.3 section 8.2.1.5 (10Base5) and section 10.3.1.4 (10Base2). These documents define <i>jabber</i> as the condition in which any packet exceeds 20 ms. The allowed range to detect jabber is between 20 ms and 150 ms.</p>	<b>detail, extensive</b>
<b>Fragments received</b>	<p>Number of packets received on the VCP that were shorter than 64 octets (excluding framing bits, but including FCS octets), and had either an FCS error or an alignment error.</p> <p>Fragment frames normally increment because both runs (which are normal occurrences caused by collisions) and noise hits are counted.</p>	<b>detail, extensive</b>
<b>Ifout errors</b>	Number of outbound packets received on the VCP that could not be transmitted because of errors.	<b>detail, extensive</b>
<b>Packet drop events</b>	Number of outbound packets received on the VCP that were dropped, rather than being encapsulated and sent out of the switch as fragments. The packet drop counter is incremented if a temporary shortage of packet memory causes packet fragmentation to fail.	<b>detail, extensive</b>
<b>64 octets frames</b>	Number of packets received on the VCP (including invalid packets) that were 64 octets in length (excluding framing bits, but including FCS octets).	<b>detail, extensive</b>

Table 19: show virtual-chassis vc-port statistics Output Fields (*continued*)

Field Name	Field Description	Level of Output
65–127 octets frames	Number of packets received on the VCP (including invalid packets) that were between 65 and 127 octets in length, inclusive (excluding framing bits, but including FCS octets).	detail, extensive
128–255 octets frames	Number of packets received on the VCP (including invalid packets) that were between 128 and 255 octets in length, inclusive (excluding framing bits, but including FCS octets).	detail, extensive
256–511 octets frames	Number of packets received on the VCP (including invalid packets) that were between 256 and 511 octets in length, inclusive (excluding framing bits, but including FCS octets).	detail, extensive
512–1023 octets frames	Number of packets received on the VCP (including invalid packets) that were between 512 and 1023 octets in length, inclusive (excluding framing bits, but including FCS octets).	detail, extensive
1024–1518 octets frames	Number of packets received on the VCP (including invalid packets) that were between 1024 and 1518 octets in length, inclusive (excluding framing bits, but including FCS octets).	detail, extensive
Rate packets per second	Number of packets per second received and transmitted on the VCP.	detail, extensive
Rate bytes per second	Number of bytes per second received and transmitted on the VCP.	detail, extensive

## Sample Output

### show virtual-chassis vc-port statistics

```
user@switch> show virtual-chassis vc-port statistics
fpc0:
```

```
-----
Interface          Input  Octets/Packets      Output  Octets/Packets
internal-0/24       0      / 0                0      / 0
internal-0/25       0      / 0                0      / 0
internal-1/26       0      / 0                0      / 0
internal-1/27       0      / 0                0      / 0
vcp-0               0      / 0                0      / 0
vcp-1               0      / 0                0      / 0
internal-0/26       0      / 0                0      / 0
internal-0/27       0      / 0                0      / 0
internal-1/24       0      / 0                0      / 0
internal-1/25       0      / 0                0      / 0
```

```
{master:0}
```

## show virtual-chassis vc-port statistics (EX8200 Virtual Chassis)

```

user@external-routing-engine> show virtual-chassis vc-port statistics
member0:
-----
Interface          Input Octets/Packets      Output Octets/Packets
vcp-4/0/4          43171238 / 48152          47687133 / 51891
vcp-4/0/7          0 / 0                     0 / 0

member1:
-----
Interface          Input Octets/Packets      Output Octets/Packets
vcp-3/0/0          0 / 0                     0 / 0
vcp-3/0/1          0 / 0                     0 / 0
vcp-3/0/4          47695376 / 51899          43180556 / 48160

member8:
-----

member9:
-----

```

## show virtual-chassis vc-port statistics brief

```

user@switch> show virtual-chassis vc-port statistics brief
fpc0:
-----
Interface          Input Octets/Packets      Output Octets/Packets
internal-0/24       0 / 0                     0 / 0
internal-0/25       0 / 0                     0 / 0
internal-1/26       0 / 0                     0 / 0
internal-1/27       0 / 0                     0 / 0
vcp-0               0 / 0                     0 / 0
vcp-1               0 / 0                     0 / 0
internal-0/26       0 / 0                     0 / 0
internal-0/27       0 / 0                     0 / 0
internal-1/24       0 / 0                     0 / 0
internal-1/25       0 / 0                     0 / 0

{master:0}

```

## show virtual-chassis vc-port statistics extensive

```

user@switch> show virtual-chassis vc-port statistics extensive
fpc0:
-----

```

	RX	TX
Port: internal-0/24		
Total octets:	0	0
Total packets:	0	0
Unicast packets:	0	0
Broadcast packets:	0	0
Multicast packets:	0	0
MAC control frames:	0	0
CRC alignment errors:	0	
Oversize packets:	0	
Undersize packets:	0	
Jabber packets:	0	
Fragments received:	0	

```
Ifout errors: 0
Packet drop events: 0
64 octets frames: 0
65-127 octets frames: 0
128-255 octets frames: 0
256-511 octets frames: 0
512-1023 octets frames: 0
1024-1518 octets frames: 0
Rate packets per second: 0 0
Rate bytes per second: 0 0

...

Port: vcp-0
Total octets: 0 0
Total packets: 0 0
Unicast packets: 0 0
Broadcast packets: 0 0
Multicast packets: 0 0
MAC control frames: 0 0
CRC alignment errors: 0
Oversize packets: 0
Undersize packets: 0
Jabber packets: 0
Fragments received: 0
Ifout errors: 0
Packet drop events: 0
64 octets frames: 0
65-127 octets frames: 0
128-255 octets frames: 0
256-511 octets frames: 0
512-1023 octets frames: 0
1024-1518 octets frames: 0
Rate packets per second: 0 0
Rate bytes per second: 0 0

Port: vcp-1
Total octets: 0 0
Total packets: 0 0
Unicast packets: 0 0
Broadcast packets: 0 0
Multicast packets: 0 0
MAC control frames: 0 0
CRC alignment errors: 0
Oversize packets: 0
Undersize packets: 0
Jabber packets: 0
Fragments received: 0
Ifout errors: 0
Packet drop events: 0
64 octets frames: 0
65-127 octets frames: 0
128-255 octets frames: 0
256-511 octets frames: 0
512-1023 octets frames: 0
1024-1518 octets frames: 0
Rate packets per second: 0 0
Rate bytes per second: 0 0

...
```

```
{master:0}
```

### show virtual-chassis vc-port statistics member 0

```
user@switch>show virtual-chassis vc-port statistics member 0  
fpc0:
```

```
-----  
Interface          Input  Octets/Packets      Output  Octets/Packets  
internal-0/24       0      / 0              0      / 0  
internal-0/25       0      / 0              0      / 0  
internal-1/26       0      / 0              0      / 0  
internal-1/27       0      / 0              0      / 0  
vcp-0               0      / 0              0      / 0  
vcp-1               0      / 0              0      / 0  
internal-0/26       0      / 0              0      / 0  
internal-0/27       0      / 0              0      / 0  
internal-1/24       0      / 0              0      / 0  
internal-1/25       0      / 0              0      / 0
```

```
{master:0}
```





## PART 2

# Index

- [Index on page 171](#)



# Index

## Symbols

#, comments in configuration statements.....	xiv
( ), in syntax descriptions.....	xiv
< >, in syntax descriptions.....	xiv
[ ], in configuration statements.....	xiv
{ }, in configuration statements.....	xiv
(pipe), in syntax descriptions.....	xiv

## B

braces, in configuration statements.....	xiv
brackets	
angle, in syntax descriptions.....	xiv
square, in configuration statements.....	xiv

## C

clear virtual-chassis vc-port statistics	
command.....	86
comments, in configuration statements.....	xiv
conventions	
text and syntax.....	xiii
curly braces, in configuration statements.....	xiv
customer support.....	xv
contacting JTAC.....	xv

## D

documentation	
comments on.....	xv

## F

font conventions.....	xiii
-----------------------	------

## I

id statement, virtual chassis	
virtual chassis ID.....	62

## M

mac-persistence-timer statement.....	65
manuals	
comments on.....	xv
mastership-priority statement.....	66
member statement.....	68

## MIBs

SNMP object values, displaying.....	101
-------------------------------------	-----

## N

no-management-vlan statement.....	69
no-split-detection statement, virtual chassis	
virtual chassis no-split-detection.....	70

## P

parentheses, in syntax descriptions.....	xiv
preprovisioned statement.....	72

## R

request session member command.....	88
request virtual-chassis vc-port (uplink port)	
command.....	93
role.....	73
role statement.....	73

## S

serial-number statement.....	76
show snmp mib command.....	101
show virtual-chassis command.....	136
show virtual-chassis vc-port command.....	104, 143
show virtual-chassis vc-port diagnostics optics	
command.....	147
show virtual-chassis vc-port statistics	
command.....	161
SNMP	
MIB object values, displaying.....	101
support, technical See technical support	
syntax conventions.....	xiii

## T

technical support	
contacting JTAC.....	xv
traceoptions statement	
Virtual Chassis.....	78

## V

Virtual Chassis	
active topology.....	104
member ID.....	136
members of.....	68
session.....	88
uplink VCP.....	93
Virtual Chassis ports.....	136
Virtual Chassis configuration	
Virtual Chassis port.....	143

Virtual Chassis ports	
clear statistics.....	86
diagnostics.....	147
statistics.....	161
virtual-chassis statement.....	82