

Virtual Chassis Fabric User Guide

Published
2020-06-16

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

Juniper Networks, the Juniper Networks logo, Juniper, and Junos are registered trademarks of Juniper Networks, Inc. in the United States and other countries. All other trademarks, service marks, registered marks, 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 User Guide

Copyright © 2020 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 <https://support.juniper.net/support/eula/>. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

About the Documentation | viii

Documentation and Release Notes | viii

Using the Examples in This Manual | ix

Merging a Full Example | ix

Merging a Snippet | x

Documentation Conventions | x

Documentation Feedback | xiii

Requesting Technical Support | xiii

Self-Help Online Tools and Resources | xiv

Creating a Service Request with JTAC | xiv

1

Virtual Chassis Fabric

Configuring Virtual Chassis Fabric | 2

Virtual Chassis Fabric Overview | 2

Understanding Virtual Chassis Fabric Components | 5

Spine-and-Leaf Topology | 5

Supported VCF Configurations | 6

Spine Devices | 7

Leaf Devices | 8

Routing Engine Role | 8

Linecard Role | 9

Master Routing Engine Election Process | 10

Virtual Chassis Ports (VCPs) | 10

Automatic Virtual Chassis Port (VCP) Conversion | 11

VCF Configuration Options | 11

Fabric Mode | 12

Mixed Mode | 12

Virtual Management Ethernet Interface | 13

Virtual Chassis Fabric Port Link Aggregation Group Bundles | 13

Virtual Chassis Fabric License Requirements | 14

Hardware Requirements for a Virtual Chassis Fabric | 14

Software Requirements in a Virtual Chassis Fabric	15
Understanding Virtual Chassis Fabric Configuration	17
Virtual Chassis Fabric Setup	17
Autoprovisioned Virtual Chassis Fabric Configuration	17
Preprovisioned Virtual Chassis Fabric Configuration	18
Nonprovisioned Virtual Chassis Fabric Configuration	19
Configuration File Management in a VCF	19
Logging into a Virtual Chassis Fabric	19
Understanding Interface Numbering	20
Understanding Mixed Virtual Chassis Fabric	20
Virtual Chassis Fabric Mixed and Non-mixed Support Summary	21
Understanding Mixed and Non-Mixed Virtual Chassis Fabric	21
QFX5110 VCF	22
QFX5100 VCF	22
Understanding Traffic Flow Through a Virtual Chassis Fabric	23
Smart Trunking Algorithm for Unicast Traffic Forwarding	23
Multicast Distribution Trees for Broadcast, Unknown Unicast, and Multicast Traffic	24
Adaptive Load Balancing	24
Understanding Software Upgrades in a Virtual Chassis Fabric	26
Virtual Chassis Fabric Software Basics	26
Nonstop Software Upgrade (NSSU)	26
Automatic Software Update	27
Standard Software Upgrade	28
Understanding Nonstop Software Upgrade on a Virtual Chassis Fabric	28
Benefits of NSSU	29
Requirements for Performing an NSSU for a VCF	29
How NSSU Works for a VCF	30
NSSU Limitations	32
NSSU and Junos OS Release Support	32
Overview of NSSU Configuration and Operation	32
Autoprovisioning a Virtual Chassis Fabric	33
Preprovisioning a Virtual Chassis Fabric	38

Configuring a Nonprovisioned Virtual Chassis Fabric | 45

Adding a Device to a Virtual Chassis Fabric | 49

Adding a Leaf Device to an Autoprovisioned Virtual Chassis Fabric | 50

Adding a Spine Device to an Autoprovisioned Virtual Chassis Fabric | 52

Adding a Spine or Leaf Device to a Preprovisioned Virtual Chassis Fabric | 54

Adding a Spine or Leaf Device to a Nonprovisioned Virtual Chassis Fabric | 57

Removing a Device From a Virtual Chassis Fabric | 60

Upgrading Virtual Chassis Fabric Software Using Automatic or Standard Software Update Features | 61

Automatic Software Update | 62

Standard Upgrade | 64

Upgrading a QFX5100 Switch with a USB Device to Join a QFX5110 Virtual Chassis or Virtual Chassis Fabric | 64

Identifying Compatible Software for QFX5100 Switches to Run in a QFX5110 Virtual Chassis or VCF | 65

Creating a USB Boot Device for a QFX5100 Switch | 67

Upgrading a QFX5100 Switch from Junos OS "QFX 5 Series" to "QFX 5e Series" Software Using a USB Boot Device | 69

Configuring Line-Card Upgrade Groups for Nonstop Software Upgrade | 70

How Line-card Upgrade Groups Work with Nonstop Software Upgrade | 71

Line-card Upgrade Groups Support | 71

Configure Line-Card Upgrade Groups on an EX4650 Virtual Chassis, a QFX Series Virtual Chassis or a QFX5100 VCF | 71

Configure Line-Card Upgrade Groups on Standalone EX6200 or EX8200 Switches | 72

Configure Line-Card Upgrade Groups on an EX8200 Virtual Chassis | 73

Upgrading Software on a Virtual Chassis Fabric Using Nonstop Software Upgrade | 74

Preparing the Switch for the New Software Installation | 75

Upgrading the Software Using NSSU | 77

Verifying the Member ID, Role, Status, and Neighbor Member Connections of a Virtual Chassis Fabric Member Device | 85

Verifying Virtual Chassis Port Connections in a Virtual Chassis Fabric | 86

Verifying the Virtual Chassis Fabric Mode Settings | 87

Troubleshooting Virtual Chassis Fabric | 88

Large-Scale Virtual Chassis Fabric Becomes Unstable When Logging is Enabled | 88

Virtual Chassis Port Link Does Not Form | 89

QFX5100 Leaf Device Assumes Routing Engine Role | 90

Configuration Statements for Virtual Chassis Fabric | 91

aliases (Virtual Chassis) | 92

auto-provisioned | 94

auto-sw-update | 95

enhanced-hash-key | 99

fabric-load-balance | 105

fabric-tree-root | 107

fpcs (NSSU Upgrade Groups) | 109

id | 111

inactivity-interval (Fabric Load Balance) | 112

location (Virtual Chassis) | 114

mac-persistence-timer | 115

mastership-priority | 117

member | 119

member (NSSU Upgrade Groups) | 121

no-management-vlan | 123

nssu | 124

preprovisioned | 126

rcp-count | 127

role | 129

serial-number | 132

traceoptions (Virtual Chassis) | 134

upgrade-group | 138

virtual-chassis | 140

Operational Commands for Virtual Chassis Fabric | 142

clear virtual-chassis vc-port statistics | 143

request session member | 145

request system software nonstop-upgrade | 146

request virtual-chassis mode | 158

request virtual-chassis reactivate | 162

request virtual-chassis vc-port | 163

request virtual-chassis vc-port diagnostics optics | **165**

show chassis nonstop-upgrade | **166**

show forwarding-options enhanced-hash-key | **169**

show virtual-chassis | **176**

show virtual-chassis active-topology | **185**

show virtual-chassis device-topology | **192**

show virtual-chassis login | **200**

show virtual-chassis mode | **202**

show virtual-chassis protocol adjacency | **205**

show virtual-chassis protocol database | **209**

show virtual-chassis protocol interface | **214**

show virtual-chassis protocol route | **217**

show virtual-chassis protocol statistics | **221**

show virtual-chassis vc-path | **225**

show virtual-chassis vc-port | **231**

show virtual-chassis vc-port diagnostics optics | **237**

show virtual-chassis vc-port statistics | **256**

About the Documentation

IN THIS SECTION

- Documentation and Release Notes | viii
- Using the Examples in This Manual | ix
- Documentation Conventions | x
- Documentation Feedback | xiii
- Requesting Technical Support | xiii

Use this guide to set up and configure a Virtual Chassis Fabric (VCF), which provides a low-latency, high-performance fabric architecture composed of multiple supported combinations of QFX Series and EX Series switches (VCF members) interconnected together that operate and are managed as a single network entity. Refer to switch hardware documentation for more details on switch models and physical interconnections that can be used to form a VCF.

Also, for VCF planning and configuration best practices, see [Virtual Chassis Fabric Best Practices Guide](#) (PDF only).

Documentation and Release Notes

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

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

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

Using the Examples in This Manual

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

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

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

Merging a Full Example

To merge a full example, follow these steps:

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

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

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

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

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

Merging a Snippet

To merge a snippet, follow these steps:

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

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

```
commit {
  file ex-script-snippet.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 xi](#) 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 xi defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the configure command: user@host> configure
Fixed-width text like this	Represents output that appears on the terminal screen.	user@host> show chassis alarms No alarms currently active
<i>Italic text like this</i>	<ul style="list-style-type: none"> Introduces or emphasizes important new terms. Identifies guide names. Identifies RFC and Internet draft titles. 	<ul style="list-style-type: none"> A policy <i>term</i> is a named structure that defines match conditions and actions. <i>Junos OS CLI User Guide</i> RFC 1997, <i>BGP Communities Attribute</i>

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

Convention	Description	Examples
<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@# set system domain-name <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"> To configure a stub area, include the stub statement at the [edit protocols ospf area area-id] hierarchy level. The console port is labeled CONSOLE.
< > (angle brackets)	Encloses optional keywords or variables.	stub <default-metric <i>metric</i> >;
(pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	broadcast multicast (<i>string1</i> <i>string2</i> <i>string3</i>)
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	rsvp { # Required for dynamic MPLS only
[] (square brackets)	Encloses a variable for which you can substitute one or more values.	community name members [<i>community-ids</i>]
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

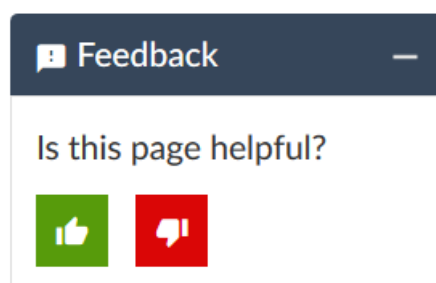
Table 2: Text and Syntax Conventions (*continued*)

Convention	Description	Examples
Bold text like this	Represents graphical user interface (GUI) items you click or select.	<ul style="list-style-type: none"> In the Logical Interfaces box, select All Interfaces. To cancel the configuration, click Cancel.
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select Protocols>Ospf .

Documentation Feedback

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

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



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

Requesting Technical Support

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

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

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

Self-Help Online Tools and Resources

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

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

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

Creating a Service Request with JTAC

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

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

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

1

PART

Virtual Chassis Fabric

Configuring Virtual Chassis Fabric | 2

Configuration Statements for Virtual Chassis Fabric | 91

Operational Commands for Virtual Chassis Fabric | 142

Configuring Virtual Chassis Fabric

IN THIS CHAPTER

- Virtual Chassis Fabric Overview | 2
- Understanding Virtual Chassis Fabric Components | 5
- Understanding Virtual Chassis Fabric Configuration | 17
- Understanding Mixed Virtual Chassis Fabric | 20
- Understanding Traffic Flow Through a Virtual Chassis Fabric | 23
- Understanding Software Upgrades in a Virtual Chassis Fabric | 26
- Understanding Nonstop Software Upgrade on a Virtual Chassis Fabric | 28
- Autoprovisioning a Virtual Chassis Fabric | 33
- Preprovisioning a Virtual Chassis Fabric | 38
- Configuring a Nonprovisioned Virtual Chassis Fabric | 45
- Adding a Device to a Virtual Chassis Fabric | 49
- Removing a Device From a Virtual Chassis Fabric | 60
- Upgrading Virtual Chassis Fabric Software Using Automatic or Standard Software Update Features | 61
- Upgrading a QFX5100 Switch with a USB Device to Join a QFX5110 Virtual Chassis or Virtual Chassis Fabric | 64
- Configuring Line-Card Upgrade Groups for Nonstop Software Upgrade | 70
- Upgrading Software on a Virtual Chassis Fabric Using Nonstop Software Upgrade | 74
- Verifying the Member ID, Role, Status, and Neighbor Member Connections of a Virtual Chassis Fabric Member Device | 85
- Verifying Virtual Chassis Port Connections in a Virtual Chassis Fabric | 86
- Verifying the Virtual Chassis Fabric Mode Settings | 87
- Troubleshooting Virtual Chassis Fabric | 88

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,

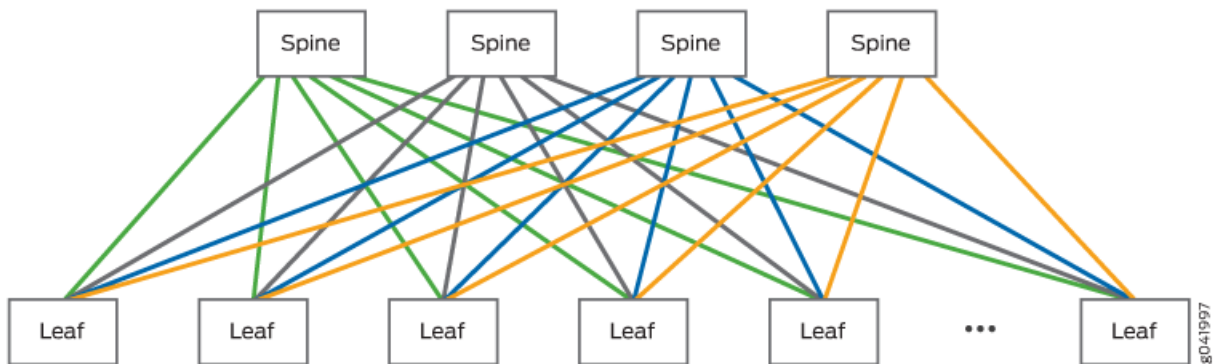
which enables you to interconnect multiple devices into a single logical device, inside of a fabric architecture. The VCF architecture is optimized to support small and medium-sized data centers that contain a mix of 1-Gbps, 10-Gbps, and 40-Gbps Ethernet interfaces.



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 3](#) for an illustration of the VCF spine-and-leaf architecture.

Figure 1: VCF Spine-and-Leaf Architecture



A VCF is based on either QFX5100 or QFX5110 switches as the spine devices, as follows:

- You can configure a VCF with QFX5100 switches as the spine devices, referred to as a QFX5100 VCF. In an optimal QFX5100 VCF configuration, the leaf devices are also QFX5100 switches. You can, however, also create a mixed QFX5100 VCF by including any combination of QFX5100, QFX3600, QFX3500, and EX4300 switches as leaf devices.

NOTE: EX4300 multigigabit model (EX4300-48MP) switches are not supported in a VCF.

- Starting in Junos OS Release 17.3R1, you can configure a QFX5110 VCF, which is a VCF with QFX5110-32Q switches as the spine devices. In an optimal QFX5110 VCF configuration, the leaf devices are also QFX5110 switches. You can, however, create a QFX5110 VCF by including any combination of QFX5110 and QFX5100 switches as leaf devices. A QFX5110 VCF that also contains QFX5100 devices is a non-mixed VCF because both types of switches run the same software image when interconnected into a VCF.

See [“Understanding Virtual Chassis Fabric Components” on page 5](#) for more information about the spine-and-leaf architecture, supported device types, and which device types are required or recommended to be in each role.

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 QFX Series Virtual Chassis are maximized, a VCF is often a logical upgrade option because it enables the system to evolve without having to remove 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.

Release History Table

Release	Description
17.3R1	Starting in Junos OS Release 17.3R1, you can configure a QFX5110 VCF, which is a VCF with QFX5110-32Q switches as the spine devices.

RELATED DOCUMENTATION

[Virtual Chassis Fabric Best Practices Guide \(PDF only\)](#)

[Network Configuration Example: MetaFabric™ Architecture 1.1: Configuring Virtual Chassis Fabric and Network Director 1.6](#)

[Understanding Virtual Chassis Fabric Components | 5](#)

[Understanding Virtual Chassis Fabric Configuration | 17](#)

[Autoprovisioning a Virtual Chassis Fabric | 33](#)

Understanding Virtual Chassis Fabric Components

IN THIS SECTION

- [Spine-and-Leaf Topology | 5](#)
- [Supported VCF Configurations | 6](#)
- [Spine Devices | 7](#)
- [Leaf Devices | 8](#)
- [Routing Engine Role | 8](#)
- [Linecard Role | 9](#)
- [Master Routing Engine Election Process | 10](#)
- [Virtual Chassis Ports \(VCPs\) | 10](#)
- [Automatic Virtual Chassis Port \(VCP\) Conversion | 11](#)
- [VCF Configuration Options | 11](#)
- [Fabric Mode | 12](#)
- [Mixed Mode | 12](#)
- [Virtual Management Ethernet Interface | 13](#)
- [Virtual Chassis Fabric Port Link Aggregation Group Bundles | 13](#)
- [Virtual Chassis Fabric License Requirements | 14](#)
- [Hardware Requirements for a Virtual Chassis Fabric | 14](#)
- [Software Requirements in a Virtual Chassis Fabric | 15](#)

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

This topic covers:

Spine-and-Leaf Topology

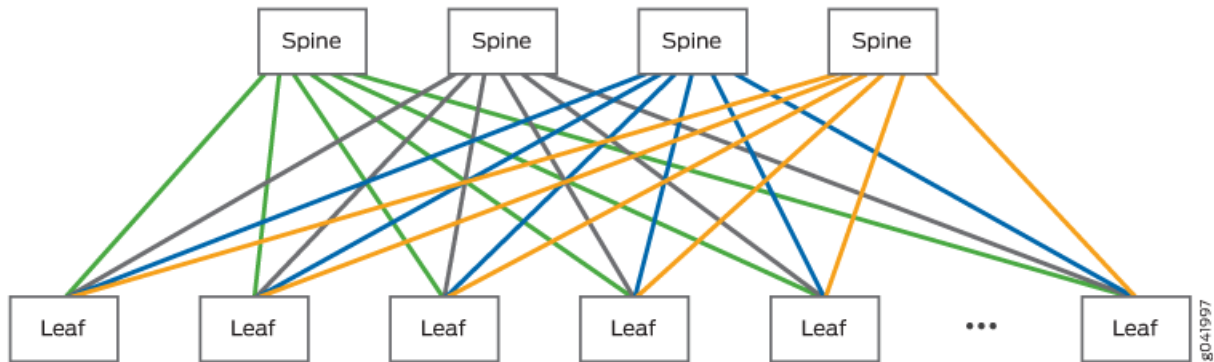
The VCF uses a spine-and-leaf architecture where each device in the fabric is either a spine device or a leaf device.

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



NOTE: A VCF topology should include at least four members—two spine devices and at least two leaf devices. For topologies with three or fewer members, use a Virtual Chassis configuration instead.

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.

Supported VCF Configurations

A VCF can be configured using QFX5100 switches as the spine devices, which is referred to as a QFX5100 VCF. Starting in Junos OS Release 17.3R1, a VCF can also be configured using QFX5110-32Q switches as the spine devices, which is referred to as a QFX5110 VCF. The following VCF configurations are supported based on either QFX5110 or QFX5100 switches as the spine members, as indicated:

- A *non-mixed QFX5100 VCF* has QFX5100 switches as spine members, and supports only QFX5100 switches as leaf members.
- A *mixed QFX5100 VCF* has QFX5100 switches as spine members, and supports any combination of EX4300, QFX3500, and QFX3600 switches, possibly with additional QFX5100 switches, as leaf members.

NOTE: EX4300 multigigabit model (EX4300-48MP) switches are not supported in a VCF.

- A *non-mixed QFX5110 VCF* or simply a *QFX5110 VCF* has QFX5110-32Q switches as spine members, and supports either only QFX5110 switches or any combination of supported QFX5100 switches and QFX5110 switches as leaf members. (Both QFX5110 and QFX5100 switches run the same software image in a VCF, and do not need to operate in mixed mode.)

Spine Devices

A spine device:

- Must be either a QFX5100 switch in a QFX5100 VCF, or a QFX5110-32Q switch in a QFX5110 VCF.
- Can be configured into the Routing Engine or linecard role.

In a VCF, at least two spine devices must be configured into the Routing Engine role. The remaining spine devices can be configured as alternate Routing Engine devices or 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 interconnected with at least two leaf devices. A VCF supports up to four spine devices.

NOTE: In a QFX5110 VCF, you must use only QFX5110-32Q switches as the spine devices.

BEST PRACTICE: In a QFX5100 VCF, 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:

- Can be a QFX5100, QFX3500, QFX3600, or EX4300 switch in a QFX5100 VCF.
- Can be a QF5110 or QFX5100 switch in a QFX5110 VCF.
- Has a direct connection to each spine device.
- Usually operates in the linecard role.

For resiliency, if you don't have additional spine devices to take on the master or backup Routing Engine role in case a spine device Routing Engine member fails, you can configure a leaf device into the Routing Engine 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.

NOTE: A VCF should include at least four members—two spine devices and at least two leaf devices. For topologies with three or fewer members, use a Virtual Chassis configuration instead.

Routing Engine Role

A VCF has two spine 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 usually 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 usually a spine device.
- Maintains a state of readiness to take over as the master 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, you usually configure at least two spine devices into the Routing Engine role. You can either explicitly configure the remaining spine devices into the linecard role, or any devices not elected as the master and backup Routing Engines will automatically operate in the linecard role (see [“Master Routing Engine Election Process” on page 10](#)).

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, you configure at least two spine devices into the Routing Engine role. You can either explicitly configure the remaining spine devices into the linecard role, or any devices not elected as the master and backup Routing Engines will automatically operate in the linecard role (see [“Master Routing Engine Election Process” on page 10](#)). A spine device that you don’t explicitly configure into Routing Engine role or that you explicitly configure into linecard role will never take on a mastership role if the master or backup Routing Engine fails.

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

Leaf devices in a VCF usually operate in the linecard role.

- In autoprovisioned configurations, the VCF assigns the linecard role to leaf devices when they are cabled into the VCF.
- In preprovisioned configurations, you must manually configure linecard role on the leaf devices and any spine devices that you don’t want to participate in the mastership election process (see [“Master Routing Engine Election Process” on page 10](#)).
- 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 (among other criteria).

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 usually a spine device.

In a preprovisioned or autoprovisioned VCF, you configure at least two spine devices into the Routing Engine role. In the mastership election 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 spine device with the highest user-configured mastership priority (255 is the highest possible value) as the master Routing Engine, and the spine device with the second highest mastership priority value as the backup Routing Engine.

A spine device with a mastership priority of 0 will always stay in the linecard role.

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

In a QFX5100 VCF, QFX3500, QFX3600, and EX4300 devices never assume the master or backup Routing Engine role.

In a QFX5110 VCF, the spine devices should be QFX5110-32Q switches, so those will always be the switches in the master or backup Routing Engine role.

We strongly recommend that you configure the mastership priority of the spine 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.

You can configure the following ports into VCPs in a VCF to interconnect switch models that support the indicated port types and speeds:

- 10-Gbps SFP+ ports in a QFX5100 VCF or a QFX5110 VCF
- 40-Gbps QSFP+ ports in either a QFX5100 VCF or a QFX5110 VCF
- 100-Gbps or 40-Gbps QSFP28 ports in a QFX5110 VCF

NOTE: Channelized interfaces cannot be configured into VCPs.

You can manually configure VCP ports, or all the ports listed above can also be automatically converted into VCPs when a new device is added to an autoprovisioned or preprovisioned VCF under certain conditions. Automatic VCP conversion is discussed in more detail in the following section.

Automatic Virtual Chassis Port (VCP) Conversion

Ports that can be VCPs are automatically converted into VCPs when:

- Link Layer Discovery Protocol (LLDP) is enabled on the interfaces on both ends of the VCP link. LLDP is enabled by default.
- The device being added to the VCF is configured into fabric mode.
- One of the devices is already part of a VCF that was autoprovisioned or preprovisioned.
- The interfaces for the ports on both ends of the link are not already configured as VCPs.

For interfaces with any of the following specifications, you must use the **request virtual-chassis vc-port delete** command to change the interface into a network interface for it to be eligible for automatic VCP conversion:

- A 40-Gbps QSFP+ port on an EX4300 switch, which is configured into a VCP by default.
- Any interface in the VCF that was a VCP that has not yet been reconfigured. 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 reconfigured into a network port.
- Any interface that has been configured into a VCP using the **request virtual-chassis vc-port set** command.

Automatic VCP conversion does not work in nonprovisioned VCFs.

Automatic VCP conversion does not automatically 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 (on either side of the VCP link) and you want the interface to function as a network interface, you must manually delete the VCP setting using the **request virtual-chassis vc-port delete** command.

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 17](#) for additional information on the VCF configuration options.

Fabric Mode

Devices must be configured in fabric mode to properly participate as a member of a VCF. As a best practice, you should configure a device into fabric mode before interconnecting it into the VCF.

Devices are not in fabric mode by default. A device that is in a Virtual Chassis or a standalone device that is not part of a VCF should never be configured into fabric mode.

In autoprovisioned or preprovisioned VCF configurations, you can manually configure a spine device into fabric mode first and then interconnect it into a VCF. Alternatively, a spine device that has been zeroized or has the factory default configuration is *automatically* configured into fabric mode when it is interconnected into a VCF, during the automatic VCP conversion process. In either case, the process of configuring the device into fabric mode requires a device reboot as the final step for the spine device to join the VCF. If configured into fabric mode automatically, the VCF reboots the device automatically; if configured into fabric mode manually, you must manually reboot the device.

In an autoprovisioned or preprovisioned VCF, when adding a leaf device that has been zeroized or has the factory default configuration, the VCF automatically sets fabric mode and reboots the leaf device when it is interconnected into the VCF, during the automatic VCP conversion process. You can avoid the downtime that accompanies the automatic reboot by setting the device into fabric mode and rebooting the leaf device before interconnecting it into the VCF.

BEST PRACTICE: For best results, we strongly recommend *manually* configuring a spine or leaf device into fabric mode and then manually rebooting the device before interconnecting it into the VCF, rather than having the reboot happen automatically, which can be perceived as an unexpected action during VCF configuration and operation.

Mixed Mode

A *mixed* VCF is a VCF that includes two or more types of member switches in supported combinations that run different software images. In a mixed VCF, you must configure all devices in the VCF into mixed mode, and the switch must be rebooted after changing the mode for the change to take effect.

A VCF can be based on either QFX5110 or QFX5100 switches as the spine members, but only a QFX5100 VCF can be a mixed VCF, when the VCF contains QFX5100 spine members and also includes EX4300, QFX3500, QFX3600, or QFX5100 switches as leaf members. A QFX5110 VCF, which must have QFX5110-32Q spine members and can have any combination of QFX5100 and QFX5110 switches as leaf members, is always considered a non-mixed VCF; both types of switches run the same software image when interconnected into a VCF, and you do not need to configure the members into mixed mode. See *Understanding Mixed EX Series and QFX Series Virtual Chassis* for more details on which switches can be combined into a mixed VCF.

The optimal QFX5110 VCF topology is to use QFX5110 switches only, and the optimal QFX5100 VCF topology is to use a non-mixed QFX5100 with QFX5100 switches only. In each of these topologies, a VCF composed entirely of the base VCF devices supports the largest breadth of features at the highest scalability while also supporting the highest number of high-speed interfaces.

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

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, so, for example, 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 (<https://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. A VCF should contain a minimum of four members—two spine devices and at least two leaf devices interconnected in a spine-and-leaf topology.

NOTE: For topologies with three or fewer members, use a Virtual Chassis configuration instead.

Spine devices in a mixed QFX5100 VCF must be QFX5100 switches, with any combination of QFX5100, QFX3600, QFX3500, or EX4300 switches as leaf 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.

Spine devices in a QFX5110 VCF must be QFX5110-32Q switches. Leaf devices in a QFX5110 VCF can be any combination of supported QFX5100 switches and QFX5110 switches.

The following QFX5110 switches are supported as leaf devices in a QFX5110 VCF:

- QFX5110-32Q
- QFX5110-48S

The following QFX5100 switches are supported as leaf devices in a QFX5110 VCF:

- QFX5100-24Q
- QFX5100-48S
- QFX5100-48T

Starting in Junos OS Release 17.3R2, QFX5100-48T switches can be included in a QFX5110 VCF.

- QFX5100-96S

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 QFX5100 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.

A QFX5110 VCF can only be set up using QFX5110 and QFX5100 switches that are running the same Junos OS image, which must be an image that includes “-qfx-5e-” in the software package filename when the Junos OS package is downloaded from the Software Center.



CAUTION: QFX5100 switches running a Junos OS image that includes “-qfx-5-” in the software package filename *must* be upgraded to a package filename that includes “-qfx-5e-” before being added to a QFX5110 VCF. See [“Upgrading a QFX5100 Switch with a USB Device to Join a QFX5110 Virtual Chassis or Virtual Chassis Fabric”](#) on page 64.

For any VCF, we recommend updating 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 26.

Release History Table

Release	Description
17.3R2	Starting in Junos OS Release 17.3R2, QFX5100-48T switches can be included in a QFX5110 VCF.
17.3R1	Starting in Junos OS Release 17.3R1, a VCF can also be configured using QFX5110-32Q switches as the spine devices, which is referred to as a <i>QFX5110 VCF</i> .

RELATED DOCUMENTATION

Network Configuration Example: MetaF abric™ Architecture 1.1: Configuring Virtual Chassis Fabric and Network Director 1.6
Autoprovisioning a Virtual Chassis Fabric 33
Preprovisioning a Virtual Chassis Fabric 38
Understanding Virtual Chassis Fabric Configuration 17
Understanding Software Upgrades in a Virtual Chassis Fabric 26
Virtual Chassis Fabric Overview 2

Understanding Virtual Chassis Fabric Configuration

IN THIS SECTION

- [Virtual Chassis Fabric Setup | 17](#)
- [Configuration File Management in a VCF | 19](#)
- [Logging into a Virtual Chassis Fabric | 19](#)
- [Understanding Interface Numbering | 20](#)

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

This topic covers:

Virtual Chassis Fabric Setup

IN THIS SECTION

- [Autoprovisioned Virtual Chassis Fabric Configuration | 17](#)
- [Preprovisioned Virtual Chassis Fabric Configuration | 18](#)
- [Nonprovisioned Virtual Chassis Fabric Configuration | 19](#)

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

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. During this process, the Virtual Chassis ports (VCPs) are configured automatically, and other parameters such as fabric and mixed mode are automatically detected and set.

For best results, a spine device in an autoprovisioned configuration should be configured into fabric mode and rebooted manually *before* being interconnected into a VCF. Otherwise, if the VCF automatically sets fabric mode for the device, the subsequent automatic device reboot might be unexpected at that point during VCF configuration and operation.

A spine device in an autoprovisioned VCF must also have the same mixed mode setting as other member devices in the VCF. Setting either fabric mode or mixed mode requires the device to be rebooted, so as a best practice, you should configure your spine device into fabric mode and at the same time, if necessary, configure mixed mode, and reboot the device manually *before* interconnecting it into the VCF.

Similar to the behavior for spine devices, a leaf device in an autoprovisioned configuration that is zeroized or in factory default configuration and not yet configured into fabric mode is automatically configured into fabric mode and rebooted during the automatic VCP conversion process when it is interconnected into a VCF. The leaf device is also automatically rebooted if the device needs to be configured into or out of mixed mode to participate in the VCF. You can optionally avoid the downtime that accompanies a leaf device reboot by manually setting the leaf device into fabric mode and into or out of mixed mode, zeroizing the device at that point if necessary, and manually rebooting the device *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 more explicitly control which devices are added to your VCF, and in what roles. At the same time, as with an autoprovisioned VCF, preprovisioned VCFs support automatic VCP conversion. As part of the VCP conversion process, when leaf devices that have been zeroized or are in factory default mode are interconnected to configured spine devices, the VCF can automatically detect and, if needed, set fabric and mixed modes. If fabric mode or mixed mode settings are automatically updated, the devices are also rebooted automatically. Alternatively, you can avoid a potentially unexpected automatic device reboot (and associated down time) by manually configuring the fabric or mixed mode setting on the device and manually rebooting it *before* interconnecting it into the VCF. For best results when adding devices to a preprovisioned VCF, we recommend manually setting fabric and mixed modes, zeroizing or restoring the factory default configuration if necessary, and manually rebooting the devices being added *before* interconnecting them into the VCF.

The disadvantage of using a preprovisioned configuration is that the configuration process requires more manual steps 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 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 | 33](#)

[Preprovisioning a Virtual Chassis Fabric | 38](#)

[Configuring a Nonprovisioned Virtual Chassis Fabric | 45](#)

Understanding Mixed Virtual Chassis Fabric

IN THIS SECTION

- [Virtual Chassis Fabric Mixed and Non-mixed Support Summary | 21](#)
- [Understanding Mixed and Non-Mixed Virtual Chassis Fabric | 21](#)

This topic describes the requirements for a mixed Virtual Chassis Fabric (VCF). A VCF can be based on either QFX5110 switches as the spine members (a QFX5110 VCF) or QFX5100 switches as the spine members (a QFX5100 VCF). A mixed VCF is any VCF that includes two or more types of member switches in supported combinations that require you to configure mixed mode for the switches to interoperate in a VCF.

Virtual Chassis Fabric Mixed and Non-mixed Support Summary

Table 3 on page 21 provides a high-level overview of the devices allowed in the Routing Engine and line-card roles of supported mixed and non-mixed VCF configurations. The switches in the Routing Engine role must be spine devices in a VCF topology. 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	QFX5110 (spine members must be QFX5110-32Q model switches)	QFX5110 QFX5100 (with a “-qfx-5e-” Junos OS image)	Yes, on two QFX5110 switches operating in master and backup Routing Engine roles	Autoprovisioning Preprovisioning Nonprovisioning (not recommended)
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 (excluding EX4300 multigigabit models)	Yes, on two QFX5100 switches operating in master and backup Routing Engine roles	Autoprovisioning Preprovisioning Nonprovisioning (not recommended)

Understanding Mixed and Non-Mixed Virtual Chassis Fabric

A VCF can be based on either QFX5110 switches as the spine members (a *QFX5110 VCF*), or QFX5100 switches as the spine members (a *QFX5100 VCF*).

A *mixed VCF* is any VCF that includes two or more types of member switches in supported combinations for which you are required to configure mixed mode on all the switches for them to interoperate in a VCF. If you are converting a non-mixed VCF into a mixed VCF by adding members that are different types of

devices in supported combinations, you must reboot the VCF to change the mixed mode setting on all members.

A *non-mixed VCF* has either all of the same type of switches or a combination of different switches that can run the same software image when in a VCF configuration. In this case you are not required set mixed mode on the member switches.

NOTE: The optimal QFX5110 VCF topology is to use QFX5110 switches only, and the optimal QFX5100 VCF topology is to use QFX5100 switches only. In each of these topologies, a VCF composed entirely of the base VCF devices supports the largest breadth of features at the highest scalability while also supporting the highest number of high-speed interfaces.

QFX5110 VCF

A QFX5110 VCF is any VCF that has QFX5110-32Q switches as the spine members, and the remaining members can be any combination of supported QFX5100 and QFX5110 member switches as leaf members. Any QFX5110 VCF, whether it has only QFX5110 members or includes both QFX5110 and QFX5100 switches, is considered to be a *non-mixed VCF* because both types of switches can run the same software image, and you do not need to configure mixed mode. The QFX5110 and QFX5100 switches must be running the same Junos OS image that includes “-qfx-5e-” in the filename when the Junos OS software package is downloaded from the Software Center.

NOTE: QFX5100 switches running a Junos OS image that includes “-qfx-5-” in the software package filename *must* be upgraded to the image filename that includes “-qfx-5e-” before being added to a QFX5110 VCF, or the VCF will not form. See [“Upgrading a QFX5100 Switch with a USB Device to Join a QFX5110 Virtual Chassis or Virtual Chassis Fabric”](#) on page 64.

Although the optimal method of configuring a QFX5110 VCF is as a VCF with QFX5110 devices only, you have the flexibility to configure a QFX5110 VCF that includes QFX5100 switches as leaf members.

NOTE: You must use QFX5110-32Q model switches as the spine members in a QFX5110 VCF.

QFX5100 VCF

A QFX5100 VCF is any VCF that has QFX5100 switches as the spine members, and the remaining members can be any combination of EX4300 (excluding EX4300 multigigabit models), QFX3500, QFX3600, or QFX5100 member switches as leaf members. If a QFX5100 VCF includes any of the other supported devices besides QFX5100 switches, it must operate as a *mixed QFX5100 VCF*, because the other devices run different software images, and you need to configure mixed mode.

Although the optimal method of configuring a QFX5100 VCF is as a non-mixed VCF with QFX5100 devices only, you have the flexibility to configure a mixed QFX5100 VCF with any combination of EX4300, QFX3500, and QFX3600 switches as leaf members.

RELATED DOCUMENTATION

[Virtual Chassis Fabric Overview | 2](#)

[Understanding Virtual Chassis Fabric Components | 5](#)

Understanding Traffic Flow Through a Virtual Chassis Fabric

IN THIS SECTION

- [Smart Trunking Algorithm for Unicast Traffic Forwarding | 23](#)
- [Multicast Distribution Trees for Broadcast, Unknown Unicast, and Multicast Traffic | 24](#)
- [Adaptive Load Balancing | 24](#)

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

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.

Starting in Junos OS Release 14.1X53-D35 and 15.1R3, you can preempt default MDT behavior and form MDTs only with specific members as root nodes. 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 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.

The recommendation to configure all and only spine devices as MDT root nodes is due to how spine devices are connected in the VCF spine-and-leaf topology, and is not influenced by the roles of the spine devices (whether a spine device is acting as a Routing Engine or line card) in the VCF.

Adaptive Load Balancing

Starting in Junos OS Release 14.1X53-D10, a VCF supports adaptive load balancing (ALB).

NOTE: Starting in Junos OS Releases 14.1X53-D46, 15.1R7, 16.1R6, 17.1R3, 17.2R2, 17.3R2, and 17.4R1, the ALB feature is deprecated to avoid potential VCF instability, so you should disable this feature when upgrading your VCF to these releases and later.

ALB 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. You use the [fabric-load-balance](#) configuration statement to enable ALB using flowlets or per-packet mode.

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.

Release History Table

Release	Description
14.1X53-D46	Starting in Junos OS Releases 14.1X53-D46, 15.1R7, 16.1R6, 17.1R3, 17.2R2, 17.3R2, and 17.4R1, the ALB feature is deprecated to avoid potential VCF instability, so you should disable this feature when upgrading your VCF to these releases and later.
14.1X53-D35	Starting in Junos OS Release 14.1X53-D35 and 15.1R3, you can preempt default MDT behavior and form MDTs only with specific members as root nodes.
14.1X53-D10	Starting in Junos OS Release 14.1X53-D10, a VCF supports adaptive load balancing (ALB).

RELATED DOCUMENTATION

Understanding Software Upgrades in a Virtual Chassis Fabric

IN THIS SECTION

- [Virtual Chassis Fabric Software Basics | 26](#)
- [Nonstop Software Upgrade \(NSSU\) | 26](#)
- [Automatic Software Update | 27](#)
- [Standard Software Upgrade | 28](#)

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

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 update all devices to the same Junos OS release before interconnecting them into a VCF. Also, as a best practice when you add 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. Adding a new member that already has a compatible software version to an existing VCF ensures that the VCF absorbs the new member gracefully without the downtime required to reboot the device after an automatic software update or needing to troubleshoot issues adding the device due to mismatched software releases.

Nonstop Software Upgrade (NSSU)

With nonstop software upgrade (NSSU), you can 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 in the VCF while the other devices not being upgraded can continue to pass traffic and perform normal operations.

NSSU is supported in a QFX5100 VCF.

NOTE: You can't use NSSU to upgrade a QFX5100 VCF from a Junos OS "-qfx-5-" image (see the package filename) to a "-qfx-5e-" image. You must first upgrade all of the QFX5100 switches to a "-qfx-5e-" image that supports NSSU. Then you can use NSSU to upgrade the VCF to a later "-qfx-5e-" Junos OS release for supported *from* and *to* release combinations. See ["Upgrading a QFX5100 Switch with a USB Device to Join a QFX5110 Virtual Chassis or Virtual Chassis Fabric" on page 64.](#)

NSSU includes a line-card upgrade group feature that you can optionally configure on a QFX5100 VCF to upgrade multiple linecard role member switches in parallel, which reduces the total time needed to complete an NSSU operation. NSSU upgrades the VCF linecard role members in the order in which you define the upgrade groups, so you can also use upgrade groups to control the upgrade sequence. You can also run NSSU without defining upgrade groups.

For additional information on NSSU in a VCF, see ["Understanding Nonstop Software Upgrade on a Virtual Chassis Fabric" on page 28.](#)

For more information on configuring upgrade groups, see ["Configuring Line-Card Upgrade Groups for Nonstop Software Upgrade" on page 70.](#)

Automatic Software Update

If you configure the automatic software update option in a VCF, when you add new devices to the VCF, the VCF automatically updates the Junos OS on the new members to the version of Junos OS running on the VCF when you cable the new devices into the VCF.

You enable automatic software updates using the **set virtual-chassis auto-sw-update** CLI command.

See ["Upgrading Virtual Chassis Fabric Software Using Automatic or Standard Software Update Features" on page 61.](#)



CAUTION: You must first upgrade a QFX5100 switch running a Junos OS software image with "-qfx-5-" in the package filename to a Junos OS software image with "-qfx-5e-" in the package filename before you can add it to a QFX5110 Virtual Chassis or VCF. The automatic software update process cannot automatically update a switch from a "-qfx-5-" image to a "-qfx-5e-" image. See ["Upgrading a QFX5100 Switch with a USB Device to Join a QFX5110 Virtual Chassis or Virtual Chassis Fabric" on page 64.](#)

After you install a QFX5100 switch with a "-qfx-5e-" Junos OS software image, the automatic software update process can successfully update the switch automatically with a different version of a "-qfx-5e-" Junos OS image to match the other members in the Virtual Chassis or VCF.

Standard Software Upgrade

To upgrade software on a VCF using the traditional update method for Junos OS, you log onto the master Routing Engine and use one of the following CLI commands:

- To initiate the upgrade on a non-mixed VCF: **request system software add**
- To initiate the upgrade on a mixed VCF where you need to specify multiple packages for the different types of switches in the VCF: **request system software add set [package-name package-name ...]**, where each *package-name* is the path to a different image for each device family in the mixed VCF.

Upgrading Junos OS on a VCF using the standard software upgrade commands disrupts traffic flow in the VCF. The process brings the entire VCF down until the upgrade completes and then brings the VCF up again.

RELATED DOCUMENTATION

[Upgrading Software on a Virtual Chassis Fabric Using Nonstop Software Upgrade | 74](#)

[Upgrading Virtual Chassis Fabric Software Using Automatic or Standard Software Update Features | 61](#)

[Adding a Device to a Virtual Chassis Fabric | 49](#)

Understanding Nonstop Software Upgrade on a Virtual Chassis Fabric

IN THIS SECTION

- [Benefits of NSSU | 29](#)
- [Requirements for Performing an NSSU for a VCF | 29](#)
- [How NSSU Works for a VCF | 30](#)
- [NSSU Limitations | 32](#)
- [NSSU and Junos OS Release Support | 32](#)
- [Overview of NSSU Configuration and Operation | 32](#)

With nonstop software upgrade (NSSU), you can upgrade the software running on all member switches in a Virtual Chassis Fabric (VCF) with minimal network traffic disruption during the upgrade. You can use NSSU with a VCF as follows:

- NSSU is supported in a non-mixed or mixed mode QFX5100 VCF with up to 20 members.

- For minimal traffic disruption, you must configure link aggregation groups (LAGs) such that the member links of each LAG reside on different VCF members. When one member link of a LAG is down, the remaining links are up, and traffic continues to flow through the LAG.

NOTE: Because NSSU upgrades the software on each VCF member one at a time, an upgrade using NSSU can take longer than an upgrade using the **request system software add** command.

You can reduce the amount of time an upgrade takes by configuring NSSU line-card upgrade groups. The members of a Virtual Chassis or VCF in an upgrade group are upgraded simultaneously. See [“Configuring Line-Card Upgrade Groups for Nonstop Software Upgrade” on page 70](#).

Benefits of NSSU

- No disruption to the control plane—NSSU uses graceful Routing Engine switchover (GRES) and nonstop active routing (NSR) to ensure no disruption occurs to the control plane. During the upgrade process, the VCF preserves interface, kernel, and routing protocol information.
- Minimal disruption to network traffic—NSSU minimizes network traffic disruption by upgrading member switches one at a time, enabling the master and backup members to maintain their master and backup roles (although mastership will change) without disrupting traffic, and permitting traffic to continue to flow through members in the linecard role that are not being upgraded.

Requirements for Performing an NSSU for a VCF

You must configure the following in the VCF before requesting an NSSU operation:

- Graceful Routing Engine switchover (GRES).
- Nonstop active routing (NSR) and nonstop bridging (NSB).

NOTE: We recommend enabling NSB when you set up a VCF with any provisioning mode—preprovisioned, auto-provisioned, or non-provisioned—to avoid losing Layer 2 control protocol adjacency during a Routing Engine switchover.

Also, when you enable NSR and NSB, commit the configuration using the **commit synchronize** CLI command.

- For minimal traffic disruption, define link aggregation groups (LAGs) such that the member LAG links reside on different VCF members.

The following conditions are also required for successful NSSU operation:

- Interconnect the VCF members in a spine-and-leaf topology, with each leaf device connected to all of the configured spine devices. This topology prevents the VCF from splitting during an NSSU operation.

NOTE: You must also configure **no-split-detection** in a two-member VCF so that the VCF doesn't split when NSSU upgrades a member.

- Use preprovisioning when you initially set up the VCF, so that you explicitly assign the Routing Engine role or line-card role to the member switches acting in each of those roles.

During an NSSU, the VCF members must maintain their roles—the master and backup must maintain their master and backup roles (although mastership will change) and those member switches must remain in their Routing Engine roles. The remaining switches must maintain their line-card roles.

- You can have only two members in the Routing Engine role in the preprovisioned configuration. The NSSU process checks the member configuration, displays a warning message if it detects that you configured more than two switches in the Routing Engine role, and stops the upgrade.

How NSSU Works for a VCF

When you request an NSSU on a VCF:

1. The VCF master verifies that:
 - The backup is online.
 - You have enabled Graceful Routing Engine switchover (GRES), nonstop active routing (NSR), and nonstop bridging (NSB).
 - The VCF has a preprovisioned configuration with only 2 members in the Routing Engine role.
2. The master transfers the new software image to the backup and remaining line-card role members in sequence using **rcp**.

Starting with Junos OS Release 14.1X53-D40, to optimize the time needed to complete an NSSU operation for a VCF, the master uses parallel **rcp** sessions to copy the new software to multiple members at a time (rather than waiting for the copy operation to complete to each member before starting to copy the software image to the next member). The number of parallel copy operations is determined by a default algorithm based on the number of members in the VCF, or on a QFX5100 VCF you can configure a specific number using the **rcp-count** configuration statement. See [rcp-count](#) for details.

NOTE: If copying the new software to any line-card role member fails, NSSU aborts the upgrade process for the entire VCF without rebooting any members, and logs the error condition. Starting with Junos OS Release 14.1X53-D40, after an NSSU copy of the new software image to a member fails, the master performs an additional error recovery measure to remove the new software from the members to which it was already transferred.

3. The master restarts the backup with the new software, and the backup resynchronizes with the master.
4. The master loads and reboots member switches that are in the line-card role, one at a time. The master waits for each member to become online and active running the new software before rebooting the next member.
 - If you configured upgrade groups, the VCF member or members in the first upgrade group load the new image and restart. When the members in that upgrade group are online again, the members in the next upgrade group load the new image and restart.
 - Traffic continues to flow through the other members during this process.
5. Rebooting continues until all active members have restarted with the new software.

NOTE: If any member fails to reboot successfully (including initial reboot of the backup), NSSU aborts the upgrade process and logs the error condition. In this case, to avoid VCF instability, you should either back out the partial upgrade by restoring the old software and rebooting the members that were already rebooted with the new software, or try to manually reboot all members with the new software that was copied to them, so all members come online again running the same version of the software.

Starting with Junos OS Release 14.1X53-D40, NSSU automatically invokes recovery measures if the reboot fails on any line-card role member, stopping the sequential reboot process and bringing down and rebooting the entire VCF. This action cleanly brings up all members at the same time running the new software, which recovers stable VCF operation more quickly than having an unstable VCF running different versions of the software trying to converge.

6. When all members that are in the line-card role have been upgraded, the master performs a graceful Routing Engine switchover, and the upgraded backup becomes the master.
7. The software on the original master is upgraded and the original master is automatically rebooted. After the original master has rejoined the VCF, you can optionally return control to it by requesting a graceful Routing Engine switchover.

NSSU Limitations

You cannot use an NSSU to downgrade the software—that is, to install an earlier version of the software than is currently running on the switch. To install an earlier software version, use the **request system software add** command.

You cannot roll back to the previous software version after you perform an upgrade using NSSU. If you need to roll back to the previous software version, you can do so by rebooting from the alternate root partition if you have not already copied the new software version into the alternate root partition.

NSSU and Junos OS Release Support

NSSU is supported on a QFX5100 VCF with up to 20 member switches in Junos OS Release 13.2X51-D20 or later.

NOTE: You can't use NSSU to upgrade a QFX5100 VCF from a Junos OS “-qfx-5-” image (see the package filename) to a “-qfx-5e-” image. You must first upgrade all of the QFX5100 switches to a “-qfx-5e-” image that supports NSSU. Then you can use NSSU to upgrade the VCF to a later “-qfx-5e-” Junos OS release for supported *from* and *to* release combinations. See [“Upgrading a QFX5100 Switch with a USB Device to Join a QFX5110 Virtual Chassis or Virtual Chassis Fabric” on page 64](#).

A VCF must first be running a Junos OS release that supports NSSU before you can perform an NSSU to upgrade it to a later release.

If a VCF is running a software version that does not support NSSU, you can upgrade the VCF using the standard CLI command to update software, **request system software add**. See [“Upgrading Virtual Chassis Fabric Software Using Automatic or Standard Software Update Features” on page 61](#), which requires VCF downtime, or the network configuration example [How to Upgrade a Four-Member QFX Series VCF](#), which minimizes service disruption during the upgrade process.

NSSU works only on VCFs with particular from and to Junos OS Releases. Contact Juniper Networks Technical Assistance Center(JTAC) to confirm supported from and to releases if you are considering upgrading your VCF using NSSU.

Overview of NSSU Configuration and Operation

You must ensure that the configuration meets the requirements described in [“Requirements for Performing an NSSU for a VCF” on page 29](#). Running NSSU itself requires no additional configuration.

You initiate an NSSU by entering the **request system software nonstop-upgrade** CLI command. For detailed instructions on how to perform an NSSU, see the topics in Related Documentation.

Release History Table

Release	Description
14.1X53-D40	Starting with Junos OS Release 14.1X53-D40, to optimize the time needed to complete an NSSU operation for a VCF, the master uses parallel rcp sessions to copy the new software to multiple members at a time (rather than waiting for the copy operation to complete to each member before starting to copy the software image to the next member).
14.1X53-D40	Starting with Junos OS Release 14.1X53-D40, after an NSSU copy of the new software image to a member fails, the master performs an additional error recovery measure to remove the new software from the members to which it was already transferred.
14.1X53-D40	Starting with Junos OS Release 14.1X53-D40, NSSU automatically invokes recovery measures if the reboot fails on any line-card role member, stopping the sequential reboot process and bringing down and rebooting the entire VCF.

RELATED DOCUMENTATION

Upgrading Software on a Virtual Chassis Fabric Using Nonstop Software Upgrade	74
Configuring Graceful Routing Engine Switchover in a Virtual Chassis	
Configuring Nonstop Bridging on Switches (CLI Procedure)	
Example: Configuring Nonstop Active Routing on Switches	
Configuring Line-Card Upgrade Groups for Nonstop Software Upgrade	70

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. See [“Understanding Virtual Chassis Fabric Components” on page 5](#) and [“Understanding Virtual Chassis Fabric Configuration” on page 17](#) for details on the supported devices that can be interconnected into a non-mixed or mixed VCF.

Before you begin:

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



CAUTION: QFX5100 switches running a Junos OS image that includes “-qfx-5-” in the software package filename *must* be upgraded to a package filename that includes “-qfx-5e-” before being added to a QFX5110 Virtual Chassis. See [“Upgrading a QFX5100 Switch with a USB Device to Join a QFX5110 Virtual Chassis or Virtual Chassis Fabric”](#) on page 64.

To configure a VCF using autoprovisioning:

1. Make a list of the serial numbers of all the spine devices in the VCF. 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* for a QFX5100 VCF, or *Locating the Serial Number on a QFX5110 Device or Component* for a QFX5110 VCF.
2. Set each device individually into fabric mode. If needed, also set the devices into mixed mode for a mixed VCF, and at the same time, request the device to reboot as part of the procedure to complete this step.

TIP: This step must be done at least for the spine devices being assigned the Routing Engine role in the VCF, but for the most predictable results, we strongly recommend you also manually set fabric mode and mixed modes for all devices (with the device reboot option) before cabling them into the VCF.

If you are configuring a non-mixed VCF:

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

If you are configuring a mixed mode VCF:

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


NOTE: A spine device whose fabric and mixed mode settings are improperly set will not properly join a mixed 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 during VCF formation as the devices reboot to commit the fabric or mixed mode settings.
- Manually correcting potential issues related to VCF formation because a 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 or preprovisioned VCF if the device is zeroized or has the factory default configuration. If the fabric or mixed mode settings are automatically changed when a leaf device is interconnected into a VCF, the leaf device automatically reboots in order to properly join the VCF. To avoid this potentially unexpected reboot and impact on VCF operation, as mentioned earlier, for best results, set the fabric and mixed modes and manually reboot each leaf device before cabling it into the VCF.

3. After the reboot completes, 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 at least two spine devices into the Routing Engine role:

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

For instance, to configure 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
```

Keep in mind that any member devices you configure into the Routing Engine role participate in the mastership election process (see [“Master Routing Engine Election Process” on page 10](#)). The VCF will elect one master and one backup member from the devices configured into this role. Member devices that are not in Routing Engine role are not eligible for mastership election. Those members automatically operate in linecard role, whether or not you explicitly configured them into linecard role. Members configured into Routing Engine role that were not elected as master or backup also automatically operate in linecard role.

You usually want spine devices acting as the master and backup members. If you don't have enough spine devices to do that, you can configure one or two leaf devices into the Routing Engine role to ensure the VCF can reassign the master and backup members if a spine device Routing Engine member fails.

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. You should cable one of the management ports, em0 or em1, on each spine device in your VCF so that 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 you commit your autoprovisioned VCF configuration, you can cable any additional supported leaf devices (in zeroized or factory default configuration) into the VCF using supported VCPs. The autoprovisioning process automatically configures the VCPs, and if needed, automatically sets mixed mode and fabric mode and reboots the device for those changes to take effect. The devices participate in the VCF with no further user intervention.

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

When adding a QFX4300 leaf device to a QFX5100 VCF, the 40-Gbps QSFP+ interfaces on EX4300 switches are configured as VCPs, by default. You must, therefore, delete the VCP configuration on the 40-Gbps QSFP+ interface using the **request virtual-chassis vc-port delete** command before interconnecting it into the VCF. Then the automatic VCP conversion process is invoked and converts the link into a VCP.

A device joins the VCF immediately without a reboot if you do not need to change the fabric mode or mixed mode settings.

9. Install the VCF feature licenses.

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

To purchase a feature license for VCF, contact your Juniper Networks sales representative (<https://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 23 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 | 17](#)

[Adding a Device to a Virtual Chassis Fabric | 49](#)

[Removing a Device From a Virtual Chassis Fabric | 60](#)

[Understanding Virtual Chassis Fabric Components | 5](#)

[Preprovisioning a Virtual Chassis Fabric | 38](#)

[Understanding Traffic Flow Through a Virtual Chassis Fabric | 23](#)

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. See “[Understanding Virtual Chassis Fabric Components](#)” on page 5 and “[Understanding Virtual Chassis Fabric Configuration](#)” on page 17 for details on the supported devices that can be interconnected into a non-mixed or mixed VCF.

Before you begin:

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



CAUTION: QFX5100 switches running a Junos OS image that includes “-qfx-5-” in the software package filename *must* be upgraded to a package filename that includes “-qfx-5e-” before being added to a QFX5110 Virtual Chassis or VCF. See [“Upgrading a QFX5100 Switch with a USB Device to Join a QFX5110 Virtual Chassis or Virtual Chassis Fabric”](#) on page 64.

- If automatic software download is enabled, disable this feature 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 QFX5110 Device or Component*, *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*.

NOTE: Serial number values are case-sensitive.

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

In a VCF, you configure two supported 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. Set each device individually into fabric mode. If needed, also set the devices into mixed mode for a mixed VCF, and at the same time, request the device to reboot as part of the procedure to complete this step.

If you are configuring a non-mixed VCF:

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

If you are configuring a mixed VCF:

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

NOTE: A device whose fabric and mixed mode settings are improperly set will not properly join a mixed 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 during VCF formation as the devices reboot to commit the fabric or mixed mode 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.

The fabric and mixed mode settings are automatically updated for a leaf device when it is interconnected into an autoprovisioned or preprovisioned VCF if the device is zeroized or has the factory default configuration. If the fabric or mixed mode settings are automatically changed when a leaf device is interconnected into a VCF, the leaf device automatically reboots in order to properly join the VCF. To avoid this potentially unexpected reboot and impact on VCF operation, as mentioned earlier, for best results, set the fabric and mixed modes and manually reboot each leaf device before cabling it into the VCF.

4. After the reboot completes, log in to one of your spine devices that will be configured into the Routing Engine role.

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 supported 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.

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 23](#) 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 in a mixed QFX5100 VCF 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* already configured as 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 supported interfaces that can be VCPs.

NOTE: In autoprovisioned or preprovisioned VCFs, the automatic VCP conversion feature is enabled and automatically configures SFP+ and QSFP+ interfaces into VCPs. 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.

12. Interconnect all other spine devices to all other leaf devices using supported VCP 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 Routing Engine role and the other for the device in the backup Routing Engine role.

To purchase a feature license for VCF, contact your Juniper Networks sales representative (<https://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 | 17](#)

[Adding a Device to a Virtual Chassis Fabric | 49](#)

[Removing a Device From a Virtual Chassis Fabric | 60](#)

[Understanding Virtual Chassis Fabric Components | 5](#)

[Autoprovisioning a Virtual Chassis Fabric | 33](#)

[Understanding Traffic Flow Through a Virtual Chassis Fabric | 23](#)

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 33](#) and the process for preprovisioning your VCF is described in [“Preprovisioning a Virtual Chassis Fabric” on page 38](#).

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 *Installing Software Packages on QFX Series Devices* or *Installing Software on an EX Series Switch with a Single Routing Engine (CLI Procedure)*.



CAUTION: QFX5100 switches running a Junos OS image that includes “-qfx-5-” in the software package filename *must* be upgraded to a package filename that includes “-qfx-5e-” before being added to a QFX5110 Virtual Chassis. See [“Upgrading a QFX5100 Switch with a USB Device to Join a QFX5110 Virtual Chassis or Virtual Chassis Fabric” on page 64](#).

To configure a nonprovisioned VCF:

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

Reboot each device to complete this configuration step.

If you are configuring a non-mixed VCF:

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

If you are configuring a mixed mode VCF:

```
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 mixed 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 interfaces that can be used as VCPs.
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 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 link.

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 devices with the highest mastership priority values assume the master and backup Routing Engine role, which must be supported spine devices in a VCF. All other devices assume the linecard role.

In a QFX5100 VCF, QFX5100 switches assume the Routing Engine role, regardless of mastership priority settings. QFX5100 switches can also assume the linecard role. QFX3600, QFX3500, and EX4300 switches always assume the linecard role in a mixed QFX5100 VCF, regardless of the mastership priority settings.

In a QFX5110 VCF, spine devices must be QFX5110-32Q switches, and can also assume the linecard role as spine or leaf devices. In a QFX5110 VCF with QFX5100 switches, any QFX5100 members and other QFX5110 members must be leaf devices and assume the linecard role.

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 Routing Engine 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 Routing Engine role and the other for the device in the backup Routing Engine role.

To purchase a feature license for VCF, contact your Juniper Networks sales representative (<https://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 | 33](#)

[Preprovisioning a Virtual Chassis Fabric | 38](#)

[Adding a Device to a Virtual Chassis Fabric | 49](#)

[Removing a Device From a Virtual Chassis Fabric | 60](#)

Adding a Device to a Virtual Chassis Fabric

IN THIS SECTION

- [Adding a Leaf Device to an Autoprovisioned Virtual Chassis Fabric | 50](#)
- [Adding a Spine Device to an Autoprovisioned Virtual Chassis Fabric | 52](#)
- [Adding a Spine or Leaf Device to a Preprovisioned Virtual Chassis Fabric | 54](#)
- [Adding a Spine or Leaf Device to a Nonprovisioned Virtual Chassis Fabric | 57](#)

This topic describes how to add a device to a Virtual Chassis Fabric (VCF). See [“Understanding Virtual Chassis Fabric Components” on page 5](#) and [“Understanding Virtual Chassis Fabric Configuration” on page 17](#) for details on the supported devices that can be interconnected into a non-mixed or mixed VCF.

All devices in a VCF must be running the same or a compatible version of Junos OS, so before you begin to add a device to an existing VCF, update your device to the same version of Junos OS running on the devices in the VCF. See *Installing Software Packages on QFX Series Devices* or *Installing Software on an EX Series Switch with a Single Routing Engine (CLI Procedure)*. Then follow the applicable procedure to add the device based on how your VCF is configured.



CAUTION: QFX5100 switches running a Junos OS image that includes “-qfx-5-” in the software package filename *must* be upgraded to a package filename that includes “-qfx-5e-” before being added to a QFX5110 Virtual Chassis or VCF. See [“Upgrading a QFX5100 Switch with a USB Device to Join a QFX5110 Virtual Chassis or Virtual Chassis Fabric” on page 64](#).

This topic contains the following sections:

Adding a Leaf Device to an Autoprovisioned Virtual Chassis Fabric

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 for a mixed VCF .

NOTE: If the leaf device has not previously been configured, also specify the **reboot** option to reboot the leaf device now and apply the mode settings. Otherwise, if the leaf device has been previously configured, in the next step you zeroize and reboot the device to clear prior configuration stanzas. In that step the reboot also applies the mode settings (maintained during zeroizing), so you do not need to reboot in this step and again in the next step.

If you are configuring a non-mixed VCF:

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

If you are configuring a mixed VCF:

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

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 previously, 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 properly join the VCF as a “plug and play” device if it contains any configuration, until it has been 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**.

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.

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 step.

5. Interconnect your leaf device into the existing spine devices, using at least one interface that can be a Virtual Chassis port (VCP) to connect to each spine device in the VCF.

An autoprovisioned VCF automatically adds a supported device that is zeroized or in factory default mode to the VCF when it is connected to a spine device using a supported VCP link. Both sides of the link are automatically converted into VCPs, and fabric and mixed mode settings are detected and updated automatically if necessary, as part of this process. If fabric or mixed mode settings are updated, the newly-added leaf device is automatically rebooted to complete the configuration and join the VCF.

BEST PRACTICE: When adding a leaf device to an existing VCF, interconnect the new device to the spine member that is in the master Routing Engine role first, which is the most efficient way to synchronize the new member with the current VCF configuration and state. Interconnecting a new member only to the backup or another spine member can cause flooding of messages within the VCF as the master tries to synchronize the new member through other leaf and spine member VCP links.

After the new member is fully incorporated into the VCF, you can interconnect the remaining redundant VCP links to the backup and other spine devices without affecting traffic within the VCF.

No further configuration is required.

Adding a Spine Device to an Autoprovisioned Virtual Chassis Fabric

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 60](#) 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 23](#) 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 non-mixed VCF:

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

If you are configuring a mixed mode VCF:

```
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 interfaces that can be VCPs.

The interconnecting links are converted into VCPs automatically.

The new spine device should be operational after the cabling is completed.

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

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 60](#) 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 device that is removed in this procedure.

Add the new device to the VCF 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 23](#) 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 VCF 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 non-mixed VCF:

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

If you are configuring a mixed-mode VCF:

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

NOTE: If you are adding a device that turns a non-mixed VCF into a mixed VCF, as the next step, you must also log in to the VCF and set all of the devices in the VCF into mixed mode. This step requires a VCF reboot, which incurs some downtime.

NOTE: We recommend that you set the fabric and mixed mode settings, zeroize (if necessary), and reboot leaf devices before interconnecting them 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 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, also configuring all devices to reboot to complete this step.

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

The VCF experiences downtime during the reboot.

8. After the device reboots, interconnect the new device into the VCF using supported interfaces that can be VCPs. The interconnecting links are converted into VCPs automatically.

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

BEST PRACTICE: When adding a leaf device to an existing VCF, interconnect the new device to the spine member that is in the master Routing Engine role first, which is the most efficient way to synchronize the new member with the current VCF configuration and state. Interconnecting a new member only to the backup or another spine member can cause flooding of messages within the VCF as the master tries to synchronize the new member through other leaf and spine member VCP links.

After the new member is fully incorporated into the VCF, you can interconnect the remaining redundant VCP links to the backup and other spine devices without affecting traffic within the VCF.

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.

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 non-mixed VCF:

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

If you are configuring a mixed mode VCF:

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

NOTE: If you are adding a device that turns a non-mixed VCF into a mixed VCF, 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 interfaces that can be VCPs. Configure the interconnecting 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.

BEST PRACTICE: When adding a leaf device to an existing VCF, interconnect the new device to the spine member that is in the master Routing Engine role first, which is the most efficient way to synchronize the new member with the current VCF configuration and state.

Interconnecting a new member only to the backup or another spine member can cause flooding of messages within the VCF as the master tries to synchronize the new member through other leaf and spine member VCP links.

After the new member is fully incorporated into the VCF, you can interconnect the remaining redundant VCP links to the backup and other spine devices without affecting traffic within the VCF.

7. (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 | 60](#)

[Autoprovisioning a Virtual Chassis Fabric | 33](#)

[Preprovisioning a Virtual Chassis Fabric | 38](#)

[Understanding Virtual Chassis Fabric Configuration | 17](#)

[Understanding Virtual Chassis Fabric Components | 5](#)

[Understanding Traffic Flow Through a Virtual Chassis Fabric | 23](#)

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 QFX5100 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 mixed VCF:

```
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 | 49](#)

[Autoprovisioning a Virtual Chassis Fabric | 33](#)

[Preprovisioning a Virtual Chassis Fabric | 38](#)

[Understanding Virtual Chassis Fabric Configuration | 17](#)

[Understanding Virtual Chassis Fabric Components | 5](#)

Upgrading Virtual Chassis Fabric Software Using Automatic or Standard Software Update Features

IN THIS SECTION

● [Automatic Software Update | 62](#)

● [Standard Upgrade | 64](#)

This topic describes the procedures that you can use to update software on an existing Virtual Chassis Fabric (VCF) that require VCF downtime during the upgrade operation.

To upgrade the software running on all member switches in an operational QFX5100 VCF with minimal network traffic disruption during the upgrade, you can use nonstop software upgrade (NSSU). See [“Upgrading Software on a Virtual Chassis Fabric Using Nonstop Software Upgrade” on page 74](#).

Automatic Software Update

We recommend that you update the software on each device you include in a VCF to the same or a compatible version before initially interconnecting the VCF.

When you want to add new devices to an existing VCF, you can also update each new device to the same version of Junos OS software that is running on the VCF before cabling it into the VCF. Alternatively, you can configure the automatic software update feature, which automatically updates the software on a new device (to the same version already running on the VCF) when you cable it into the VCF.

To configure the automatic software update feature for a non-mixed VCF (a QFX5100 VCF composed entirely of QFX5100 switches, or a QFX5110 VCF composed either exclusively of QFX5110 switches or both QFX5110 and QFX5100 switches), specify the common package name for the Junos OS image that runs on all the switches in the VCF, with no other options, as follows:

```
[edit]
```

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

NOTE: A QFX5110 VCF is considered to be a non-mixed VCF whether it includes QFX5100 switches or is composed exclusively of QFX5110 switches. Because both types of switches run the same software image when interconnected into a QFX5110 VCF, you can use the **auto-sw-update** command with one software package name for all members.



CAUTION: You *must* first upgrade a QFX5100 switch running a Junos OS software image with “-qfx-5-” in the software package filename to a Junos OS software image with “-qfx-5e-” in the package filename before it can be added to a QFX5110 VCF. The automatic software update process can’t update a switch from a “-qfx-5-” image to a “-qfx-5e-” image. See [“Upgrading a QFX5100 Switch with a USB Device to Join a QFX5110 Virtual Chassis or Virtual Chassis Fabric” on page 64.](#)

After a QFX5100 switch is installed with a “-qfx-5e-” Junos OS software image, the automatic software update process can successfully update the switch automatically with a different version of a “-qfx-5e-” Junos OS image to match the other members in the VCF.

To configure the automatic software update feature on a mixed QFX5100 VCF composed of QFX5100 devices and at least one other type of device, you must specify the package name for each type of device in the mixed VCF using a separate command with the associated platform-specific option:

```
[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
```

The options specify package names for different platforms as follows:

- **qfx-5:** *package-name* is the path to a compatible Junos OS image for QFX5100 devices.
- **qfx-3:** *package-name* the path to a compatible Junos OS image for QFX3500 or QFX3600 devices.
- **ex4300:** *package-name* the path to a compatible Junos OS image for EX4300 switches.

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

/pathname/package-name

If the update process will download the software package 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 *Installing Software Packages on QFX Series Devices*.

RELATED DOCUMENTATION

[Adding a Device to a Virtual Chassis Fabric | 49](#)

[Understanding Software Upgrades in a Virtual Chassis Fabric | 26](#)

Upgrading a QFX5100 Switch with a USB Device to Join a QFX5110 Virtual Chassis or Virtual Chassis Fabric

IN THIS SECTION

- [Identifying Compatible Software for QFX5100 Switches to Run in a QFX5110 Virtual Chassis or VCF | 65](#)
- [Creating a USB Boot Device for a QFX5100 Switch | 67](#)
- [Upgrading a QFX5100 Switch from Junos OS “QFX 5 Series” to “QFX 5e Series” Software Using a USB Boot Device | 69](#)

Use this procedure to upgrade a standalone QFX5100 switch running “QFX 5 Series” Junos OS software to a “QFX 5e Series” software image so the QFX5100 switch can join a QFX5110 Virtual Chassis or Virtual Chassis Fabric (VCF).

A QFX5110 Virtual Chassis or VCF can have a combination of QFX5110 and QFX5100 switches all running compatible Junos OS “QFX 5e Series” software. This procedure explains how you identify the compatible “QFX 5e Series” install package for QFX5100 switches, create a USB boot device as the installation media, and use the boot device to upgrade the software on a QFX5100 switch to the “QFX 5e Series” image.

After installing the new software image, when you reboot the QFX5100, the switch is running a compatible software image and you can successfully add it into a QFX5110 Virtual Chassis or VCF.

NOTE: With releases prior to Junos OS Release 17.3R2 or 17.4R1, you must use this USB install method due to the differences in the boot structure and host OS software of the two types of devices.

Starting in Junos OS Releases 17.3R2 and 17.4R1, you are not required to use the USB install method to upgrade a QFX5100 switch from a “QFX 5 Series” to a “QFX 5e Series” image. Instead, you can install the “QFX 5e Series” package directly using the CLI command *request system software add*. See *Installing a Standard Software Package on QFX5000 and EX4600 Switches* for details on using the CLI install method.

If a QFX5100 switch is already running a “QFX 5e Series” image (software package filename contains the string “-qfx-5e-”), a QFX5110 Virtual Chassis or VCF can automatically update it to the right release when you add it to the Virtual Chassis or VCF. See *Understanding Software Upgrades in a Virtual Chassis* and [“Understanding Software Upgrades in a Virtual Chassis Fabric” on page 26](#).

Identifying Compatible Software for QFX5100 Switches to Run in a QFX5110 Virtual Chassis or VCF

Standalone QFX5100 switches traditionally run Junos OS “QFX 5 Series” software, and the corresponding install media and installation software package file names include the string “-qfx-5-”. For example:

```
install-media-qfx-5-17.2R1.13.tgz
jinstall-host-qfx-5-17.2R1.13-signed.tgz
```

QFX5110 switches run Junos OS “QFX 5e Series” software and use a secure-boot method at startup, so the install media and software package filenames for QFX5110 switches include “-qfx-5e-” and “secure”. For example:

```
install-media-host-usb-qfx-5e-x86-64-17.2R1.13-secure.tgz
jinstall-host-qfx-5e-x86-64-17.2R1.13-secure-signed.tgz
```

QFX5100 switches can also run “5e Series” software, but in releases prior to Junos OS Release 17.3R2 or 17.4R1 they do not use the same secure boot method as QFX5110 switches, so the “QFX 5e Series” install

media and software package filenames for QFX5100 switches include “-qfx-5e-” without the “secure” keyword. For example:

```
install-media-host-usb-qfx-5e-x86-64-17.3R1.6-signed.tgz
```

To add a QFX5100 switch to a QFX5110 Virtual Chassis or VCF, the QFX5100 must be running the same “QFX 5e Series” software version as the other switches in the Virtual Chassis or VCF. If the QFX5100 switch has a “QFX 5 Series” image, you must first upgrade it manually to a “QFX 5e Series” image using a USB boot device that does not employ the secure-boot method.

To create the USB boot device (see [“Creating a USB Boot Device for a QFX5100 Switch” on page 67](#)), use the same install media filename *without* the “secure” keyword that matches what’s running on the other QFX5110 Virtual Chassis or VCF members. For example:

If the Virtual Chassis or VCF is running the software from this install media package for QFX5110 switches (with the secure-boot method):

```
install-media-host-usb-qfx-5e-x86-64-17.3R1.6-secure-signed.tgz
```

Then the matching install media package for QFX5100 switches (without the secure-boot method) is:

```
install-media-host-usb-qfx-5e-x86-64-17.3R1.6-signed.tgz
```

After any QFX5100 switches are running a “QFX 5e Series” image, you can just use the same “jinstall-host-qfx-5e-” package file *with* the “secure” keyword to update the “QFX 5e Series” software running on all members of the Virtual Chassis or VCF, because when it starts up, the secure-boot install software determines whether or not to use the secure-boot method based on the type of switch on which it’s running. The Virtual Chassis or VCF can also successfully update any “QFX 5e Series” member switches as needed in the same way with the automatic software update feature for adding or replacing members, or even during initial Virtual Chassis or VCF configuration.

NOTE: If you remove a QFX5100 switch from a QFX5110 Virtual Chassis or VCF and want to revert the QFX5100 switch to a “QFX 5 Series” software image to run as a standalone switch, you need to reinstall the image on the switch using a USB boot device with a “qfx-5-” install media file that does not use the secure-boot method.

Creating a USB Boot Device for a QFX5100 Switch

Use the following procedure to create a USB boot device with a Junos OS “QFX 5e Series” install media package (contains “-qfx-5e-” in the package filename) for a QFX5100 switch. You can then use the USB boot device to upgrade a QFX5100 switch to run that image.

NOTE: You can create the USB boot device on the switch you want to upgrade, on another Juniper Networks switch or router, or on any PC or laptop that supports Linux. The following steps describe creating the boot device from a Junos OS device, and might differ based on the device you use to create the boot device.

Before you begin, download the installation media file from <https://www.juniper.net/customers/support/> to the device where you are creating the boot device. The install media filename should include the string “-qfx-5e-” for the same Junos OS release as the QFX5110 Virtual Chassis or VCF, but without a “secure” keyword in the filename, because you must initially upgrade QFX5100 switches without using the same secure-boot method employed by QFX5110 switches. (See “[Identifying Compatible Software for QFX5100 Switches to Run in a QFX5110 Virtual Chassis or VCF](#)” on page 65 for details.)

NOTE: The Junos OS software running on the QFX5110 members must be the “QFX 5e Series” image that uses the secure-boot method, so the install media and package filenames used on QFX5110 switches include the “secure” keyword. After initial USB installation of a “QFX 5e Series” software image on a QFX5100 switch, the same software image that supports secure boot runs on either switch model, and determines the appropriate boot method to use based on the switch on which it is running. As a result, for future updates, you can directly install the same image with the “secure” keyword on both QFX5110 and QFX5100 switches in your Virtual Chassis or VCF.

On a Junos OS device where you are creating the USB boot device:

1. Use FTP to copy the installation media file into the **/var/tmp** directory.
2. Insert the USB storage device into the USB port.
3. From the Junos OS command-line interface (CLI), start the shell:

```
user@device> start shell
%
```

4. Switch to the root account using the **su** command:

```
% su
Password: password
```

NOTE: The password is the root password for the device on which you are creating the boot media. If you logged in to the device as root, you do not need to perform this step.

5. (Optional) Before copying the installation media file to the USB device, erase the boot sector of the USB device. In some cases, depending on how the USB device was formatted previously, this step can help avoid unexpected behavior during the USB boot process. For example, enter the following command:

```
root@device% dd if=/dev/zero of=/dev/da1 count=20
```

```
20+0 records in
20+0 records out
10240 bytes (10 kB) copied, 0.008281 seconds, 1.2 MB/s
```

6. Enter the following command to copy the installation media file (see [“Identifying Compatible Software for QFX5100 Switches to Run in a QFX5110 Virtual Chassis or VCF” on page 65](#)) to the USB device:

```
root@device% dd if=/var/tmp/filename of=/dev/da1 bs=1m
```

The device writes the installation media image to the USB device. For example:

```
root@device% dd if=install-media-host-qfx-5e-17.3R1.5-domestic.img of=/dev/da0
bs=1m
1399+0 records in
1399+0 records out
1466957824 bytes transferred in 394.081902 secs (3722469 bytes/sec)
```

7. Log out of the shell:

```
root@device% exit
% exit
user@device>
```

8. Remove the USB storage device from the USB port.

You can now use the USB storage device to install the “QFX 5e Series” Junos OS software image on a QFX5100 switch.

Upgrading a QFX5100 Switch from Junos OS “QFX 5 Series” to “QFX 5e Series” Software Using a USB Boot Device

You must upgrade a standalone QFX5100 switch running “QFX 5 Series” software to “QFX 5e Series” software before the switch can join a QFX5110 Virtual Chassis or VCF. For this upgrade, you need to boot and install the software using a USB boot device.

Before you begin, have a USB boot device ready with the required Junos OS software install package as described in [“Creating a USB Boot Device for a QFX5100 Switch” on page 67](#). The upgrade process overwrites the contents of the internal flash storage on the QFX5100 switch, so if desired, before proceeding with the upgrade, save existing configuration, backup configurations, and other stored files to a remote system, server, or other storage device. Then:

1. Insert the USB boot device into a USB port on the QFX5100 switch you need to upgrade, and power-cycle the QFX5100 switch.
2. The switch comes up, booting from the USB device and running the **Juniper Linux Installer**.

NOTE: If the switch doesn’t automatically boot from the USB device, press the ESC key while the switch reboots to bring up the BIOS boot manager so you can manually select to boot from the USB device.

The installer menu prompts you to with the following options:

```
Juniper Linux Installer - (c) Juniper Networks 2014
Reboot
Install Juniper Linux Platform
Boot to host shell [debug]
```

Tab through the options to select **Install Juniper Linux Platform**, and press Enter.

3. The installer displays status messages during the install process, creates and formats the local storage partitions, and installs the host OS and Junos OS software on the switch.

After completing installation, the installer displays a list of boot options and after a few seconds automatically selects the default option to boot **Juniper Linux**. The switch boots the host OS and automatically selects and brings up Junos OS. Upon completion of the install and reboot process, the switch displays the Junos OS login prompt.

4. Log in to Junos OS on the switch, enter operational mode, and verify that the new version of software has been properly installed by running the show version command.

```
user@switch> show version
```

After the QFX5100 switch is running the “QFX 5e Series” software image, you can continue with any other configuration and setup to add the QFX5100 to a QFX5110 Virtual Chassis or VCF, such as configuring the Virtual Chassis ports (VCPs).

Release History Table

Release	Description
17.4R1	Starting in Junos OS Releases 17.3R2 and 17.4R1, you are not required to use the USB install method to upgrade a QFX5100 switch from a “QFX 5 Series” to a “QFX 5e Series” image. Instead, you can install the “QFX 5e Series” package directly using the CLI command <i>request system software add</i>

RELATED DOCUMENTATION

Understanding Mixed EX Series and QFX Series Virtual Chassis
Understanding Mixed Virtual Chassis Fabric 20
Configuring an EX4650 or a QFX Series Virtual Chassis
Understanding Virtual Chassis Fabric Configuration 17

Configuring Line-Card Upgrade Groups for Nonstop Software Upgrade

SUMMARY

You can configure line-card upgrade groups for nonstop software upgrade (NSSU) operations on supporting platforms. Line-card upgrade groups can reduce the total time required to complete an NSSU operation and enable you to control the upgrade sequence among the switches being upgraded.

IN THIS SECTION

- [How Line-card Upgrade Groups Work with Nonstop Software Upgrade | 71](#)
- [Line-card Upgrade Groups Support | 71](#)
- [Configure Line-Card Upgrade Groups on an EX4650 Virtual Chassis, a QFX Series Virtual Chassis or a QFX5100 VCF | 71](#)
- [Configure Line-Card Upgrade Groups on Standalone EX6200 or EX8200 Switches | 72](#)
- [Configure Line-Card Upgrade Groups on an EX8200 Virtual Chassis | 73](#)

How Line-card Upgrade Groups Work with Nonstop Software Upgrade

With NSSU, you can upgrade software on supporting switches with redundant Routing Engines, a Virtual Chassis, or a Virtual Chassis Fabric (VCF) using a single command with minimal disruption to network traffic.

In its default configuration, NSSU upgrades each line card in a switch or linecard role member in a Virtual Chassis or VCF one at a time. Traffic continues to flow through the other line cards or members while each one is being restarted as part of the upgrade. This behavior minimizes traffic disruption if you configure link aggregation groups (LAGs) such that the member links of each LAG reside on different line cards or members. As a result, when one member link of a LAG is down, the remaining links are up, and traffic continues to flow through the LAG.

When you configure line-card upgrade groups for NSSU, NSSU upgrades all of the devices in each upgrade group at the same time instead of sequentially, reducing the total time needed to complete the upgrade on all line cards or members.

To achieve minimal traffic disruption during an NSSU operation, you must define the line-card upgrade groups such that the member links of the LAGs reside on line cards or members that are in different upgrade groups. For information on how to configure LAGs, see *Configuring Aggregated Ethernet Links (CLI Procedure)*.

NSSU upgrades the groups in the order that they appear in the configuration (in other words, in the order you configure them). As a result, you can also define upgrade groups to control the upgrade sequence during an NSSU operation.

To configure upgrade groups, use the **upgrade-group** configuration statement in the **[edit chassis nssu]** hierarchy.

Line-card Upgrade Groups Support

The following platforms support NSSU line-card upgrade groups:

- EX4650 Virtual Chassis with more than three member switches
- QFX3500, QFX3600, and QFX5100 Virtual Chassis
- QFX5100 Virtual Chassis Fabric (VCF)
- EX6200 or EX8200 switches with redundant Routing Engines
- EX8200 Virtual Chassis

Configure Line-Card Upgrade Groups on an EX4650 Virtual Chassis, a QFX Series Virtual Chassis or a QFX5100 VCF

When you configure line-card upgrade groups on an EX4650 Virtual Chassis, a QFX Series Virtual Chassis, or a QFX5100 VCF, whose switches do not have separate line cards, you use only the **fpcs** option to specify

the Virtual Chassis or VCF member IDs that you want to include in an upgrade group. You don't need to use the **member** option.

- To create an upgrade group and add a Virtual Chassis or VCF member switch to the upgrade group, configure the upgrade group name and specify the member number using the **fpcs** option:

```
[edit chassis]
user@switch# set nssu upgrade-group group-name fpcs member-number
```

For example, to create an upgrade group called **vcf** and add linecard role member 2 to that group:

```
[edit chassis]
user@switch# set nssu upgrade-group vcf fpcs 2
```

If **vcf** already exists, this command adds member 2 to **vcf**.

- To create an upgrade group that contains multiple members in a Virtual Chassis or VCF, specify multiple member numbers enclosed in square brackets after the **fpcs** option:

```
[edit chassis]
user@switch# set nssu upgrade-group group-name fpcs [list-of-member-numbers]
```

For example, to create an upgrade group called **vc1** that contains members 1 and 2:

```
[edit chassis]
user@switch# set nssu upgrade-group vc1 fpcs [1 2]
```

Make sure you commit the configuration before starting an NSSU operation.

Configure Line-Card Upgrade Groups on Standalone EX6200 or EX8200 Switches

To configure line-card upgrade groups on a standalone EX6200 or EX8200 switch:

- To create an upgrade group and add a line card to it:

```
[edit chassis]
user@switch# set nssu upgrade-group group-name fpcs slot-number
```

For example, to create an upgrade group called **group3** and add the line card in slot 5 to it:

```
[edit chassis]
user@switch# set nssu upgrade-group group3 fpcs 5
```

If **group3** already exists, this command adds line card 5 to **group3**.

- To create an upgrade group and add multiple line cards to it:

```
[edit chassis]
user@switch# set nssu upgrade-group group-name fpcs [list-of-slot-numbers]
```

For example, to create an upgrade group called **primary** and add line cards in slots 1, 4, and 7 to it:

```
[edit chassis]
user@switch# set nssu upgrade-group primary fpcs [1 4 7]
```

If **primary** already exists, this command adds line cards in slots 1, 4, and 7 to **primary**.

SEE ALSO

Upgrading Software on an EX6200 or EX8200 Standalone Switch Using Nonstop Software Upgrade (CLI Procedure)

Configure Line-Card Upgrade Groups on an EX8200 Virtual Chassis

To configure line-card upgrade groups on an EX8200 Virtual Chassis:

- To create an upgrade group and add a line card on a Virtual Chassis member to it:

```
[edit chassis]
user@switch# set nssu upgrade-group group-name member member-id fpcs slot-number
```

For example, to create an upgrade group called **primary-ny** and add the line card on member 1 in slot 5 to it:

```
[edit chassis]
user@switch# set nssu upgrade-group primary-ny member 1 fpcs 5
```

If **primary-ny** already exists, this command adds line card 5 on member 1 to **primary-ny**.

- To create an upgrade group that contains multiple line cards on a Virtual Chassis member:

```
[edit chassis]
user@switch# set nssu upgrade-group group-name member member-id fpcs [list-of-slot-numbers]
```

For example, to create an upgrade group called **primary-ny** that contains the line cards in slots 1 and 2 on member 0 and in slots 3 and 4 on member 1:

```
[edit chassis]
user@switch# set nssu upgrade-group primary-ny member 0 fpcs [1 2]

[edit chassis]
user@switch# set nssu upgrade-group primary-ny member 1 fpcs [3 4]
```

SEE ALSO

[Upgrading Software on an EX8200 Virtual Chassis Using Nonstop Software Upgrade \(CLI Procedure\)](#)

RELATED DOCUMENTATION

[Understanding Nonstop Software Upgrade on a Virtual Chassis and Mixed Virtual Chassis](#)

[Upgrading Software on a Virtual Chassis and Mixed Virtual Chassis Using Nonstop Software Upgrade](#)

[Understanding Nonstop Software Upgrade on a Virtual Chassis Fabric | 28](#)

[Upgrading Software on a Virtual Chassis Fabric Using Nonstop Software Upgrade | 74](#)

[Understanding Nonstop Software Upgrade on EX Series Switches](#)

[Upgrading Software Using Nonstop Software Upgrade on EX Series Virtual Chassis and Mixed Virtual Chassis \(CLI Procedure\)](#)

[Example: Configuring Line-Card Upgrade Groups for Nonstop Software Upgrade on EX Series Switches](#)

Upgrading Software on a Virtual Chassis Fabric Using Nonstop Software Upgrade

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

For an overview of the benefits of NSSU and how NSSU works, see [“Understanding Nonstop Software Upgrade on a Virtual Chassis Fabric” on page 28](#).

- [Preparing the Switch for the New Software Installation | 75](#)
- [Upgrading the Software Using NSSU | 77](#)

Preparing the Switch for the New Software Installation

Before you begin software installation using NSSU:

- Ensure that the VCF is configured correctly to support NSSU:
 - Verify that you connected the VCF members in a spine-and-leaf topology with each leaf device connected to all of the configured spine devices. A spine-and-leaf topology prevents the Virtual Chassis from splitting during an NSSU.

NOTE: NSSU is supported in a QFX5100 VCF with up to 20 members.

NSSU is not supported in a QFX5110 VCF.

- Ensure the VCF is preprovisioned and you explicitly assigned the Routing Engine role or linecard role to the member switches acting in each of those roles.
 - Make sure the preprovisioned configuration has only two members in the Routing Engine role. The NSSU process checks the member configuration, displays a warning message if it detects that you configured more than two switches in the Routing Engine role, and stops the upgrade.
 - In a two-member Virtual Chassis, configure **no-split-detection** so that the VCF does not split when an NSSU upgrades a member.
 - You can optionally configure NSSU upgrade groups with multiple linecard role members in each upgrade group. NSSU upgrades the members in each upgrade group simultaneously, which can shorten the NSSU process, and upgrades the upgrade groups in the order in which you configured them.
- See [“Configuring Line-Card Upgrade Groups for Nonstop Software Upgrade” on page 70](#) for details.
- Check that members are running the same version of the software:

```
user@switch> show version
```

If the Virtual Chassis Fabric members are not running the same version of the software, use the **request system software add** CLI command to upgrade the software on the inconsistent members. If some member switches do not run the same software image, specify different package names using the **set** option: **request system software add set [package-name package-name ...] reboot**.

For example:

```
user@switch> request system software nonstop-upgrade add force-host set [jinstall-qfx5100.tgz
jinstall-qfx3500.tgz jinstall-ex-4300.tgz] reboot
```

- Verify that you enabled nonstop active routing (NSR) and graceful Routing Engine switchover (GRES). You only need to check the NSR state—if NSR is enabled, then GRES is also enabled:

```
user@switch> show task replication
Stateful Replication: Enabled
RE mode: Master

Protocol                Synchronization Status
-----                -
OSPF                    Complete
BGP                     Complete
PIM                     Complete
```

If nonstop active routing is not enabled (**Stateful Replication is Disabled**), see *Example: Configuring Nonstop Active Routing on Switches* for information on how to enable it.

- Verify that you enabled nonstop bridging (NSB), which ensures that all NSB-supported Layer 2 protocols operate seamlessly during the Routing Engine switchover that happens during NSSU.
- (Optional) Back up the system software—Junos OS, the active configuration, and log files—on each member to an external storage device with the **request system snapshot** command.

Upgrading the Software Using NSSU

This procedure describes how to upgrade the software running on all VCF members using NSSU. When the upgrade completes, all members are running the new version of the software. Because a graceful Routing Engine switchover occurs during the upgrade, the original VCF backup is the new master at the end of the NSSU process.

During NSSU, the master copies the new software image to all the members in the VCF and reboots them in turn. If copying the new software to a member fails or rebooting a member fails, NSSU aborts the upgrade process and logs the error. In this case, you must manually perform recovery measures for members left in an incompatible state, to restore all members to running the same version of the software. Starting in Junos OS Release 14.1X53-D40, NSSU automatically invokes recovery measures after either of these failures, as follows:

- if NSSU aborts due to a copy error, the new image is removed from any members to which it was already copied.
- If any member fails to reboot, NSSU automatically initiates a clean VCF restart by bringing down and rebooting the entire VCF. All members come up running the new software at the same time. This action cleanly recovers stable VCF operation more quickly than having an unstable VCF running different versions of the software trying to converge.

NOTE: Junos OS software images with enhanced automation are only supported on a non-mixed VCF with QFX5100 switches. Also, performing an NSSU from a standard Junos OS software image to a Junos OS software image with enhanced automation, or from a Junos OS software image with enhanced automation to a standard Junos OS software image is not supported.

To upgrade all members of the VCF using NSSU:

1. Download the software package. If you are upgrading the software running on a mixed mode VCF, download the software packages for each switch type.
2. Copy the software package or packages to the VCF master switch. We recommend that you copy the file to the **/var/tmp** directory on the master.
3. Log in to the VCF using the console connection of the master or the virtual management Ethernet (VME) interface. Without the console connection, you will not be able to view any CLI output during an NSSU reboot. The console connection enables you to view CLI output during an NSSU reboot and monitor the progress of the master switch reboot.
4. Start the NSSU operation:
 - To perform NSSU on a VCF composed all of one type of switch, enter:

```
user@switch> request system software nonstop-upgrade force-host /var/tmp/package-name.tgz
```

where *package-name.tgz* is, for example, *jinstall-qfx5100.tgz*.

- To perform NSSU on a mixed-mode QFX5100 VCF (QFX5100 switches with some combination of QFX3500, QFX3600, or EX4300 switches), enter:

```
user@switch> request system software nonstop-upgrade force-host set [package-name.tgz
package-name.tgz
package-name.tgz]
```

where *[package-name.tgz package-name.tgz package-name.tgz]* is, for example, *[jinstall-qfx5100.tgz jinstall-qfx3500.tgz jinstall-ex-4300.tgz]*.

As the upgrade proceeds, the VCF displays status messages similar to the following sample output for an NSSU operation on a QFX5100 VCF with four QFX5100 member switches:

```
user@switch> request system software nonstop-upgrade force-host
/var/tmp/jinstall-qfx-5-flex-14.1-20160918_x141X53_vjqfd.0-domestic.tgz no-validate

Chassis ISSU Check Done
[Sep 24 11:53:16]:ISSU: Validating Image
[Sep 24 11:53:16]:ISSU: Preparing Backup RE
Installing image on other FPC's along with the backup

[Sep 24 11:53:37]: Retrieving software images. This process can take several
minutes. Please be patient..

Retrieving version and model information from
/var/tmp/jinstall-qfx-5-flex-14.1-20160918_x141X53_vjqfd.0-domestic.tgz
Starting with package
/var/tmp/jinstall-qfx-5-flex-14.1-20160918_x141X53_vjqfd.0-domestic.tgz

Download done for package
/var/tmp/jinstall-qfx-5-flex-14.1-20160918_x141X53_vjqfd.0-domestic.tgz
Pushing bundle to fpc1
Pushing bundle to fpc4
Pushing bundle to fpc5
fpc1
WARNING:      The software that is being installed has limited support.
WARNING:      Run 'file show /etc/notices/unsupported.txt' for details.

WARNING:      The software that is being installed has limited support.
WARNING:      Run 'file show /etc/notices/unsupported.txt' for details.
```

Saving contents of boot area prior to installation

```
WARNING:      This package will load JUNOS 14.1-20160918_x141X53_vjqfd.0 software.
WARNING:      It will save JUNOS configuration files, and SSH keys
WARNING:      (if configured), but erase all other files and information
WARNING:      stored on this machine. It will attempt to preserve dumps
WARNING:      and log files, but this can not be guaranteed. This is the
WARNING:      pre-installation stage and all the software is loaded when
WARNING:      you reboot the system.
```

POST-INSTALL...

Saving the config files ...

```
NOTICE: uncommitted changes have been saved in
/var/db/config/juniper.conf.pre-install
```

```
Copying jpuppet pkg to /var/sw/pkg
```

```
Copying chef pkg to /var/sw/pkg
```

```
Pushing installation package to host...
```

```
Upgrade Host OS to 14.1-20160918_x141X53_vjqfd.0...
```

```
./jinstall-qfx-5-flex-14.1-20160918_x141X53_vjqfd.0-domestic.img.gz: OK
/boot/boot
```

```
Extracting image...
```

```
Host upgrade staging completed. Need reboot to complete upgrade installation.
```

```
WARNING: Changing next boot to SSD0
```

```
WARNING: Changing boot device enable from 0x3F to 0x3F
```

```
Install jinstall-vjunos completed
```

```
WARNING:      A REBOOT IS REQUIRED TO LOAD THIS SOFTWARE CORRECTLY. Use the
WARNING:      'request system reboot' command when software installation is
WARNING:      complete. To abort the installation, do not reboot your system,
WARNING:      instead use the 'request system software delete jinstall'
WARNING:      command as soon as this operation completes.
```

Saving state for rollback ...

fpc4

```
WARNING:      The software that is being installed has limited support.
```

```
WARNING:      Run 'file show /etc/notices/unsupported.txt' for details.
```

```
WARNING:      The software that is being installed has limited support.
```

```
WARNING:      Run 'file show /etc/notices/unsupported.txt' for details.
```

Saving contents of boot area prior to installation

```
WARNING:      This package will load JUNOS 14.1-20160918_x141X53_vjqfd.0 software.
```

```

WARNING:      It will save JUNOS configuration files, and SSH keys
WARNING:      (if configured), but erase all other files and information
WARNING:      stored on this machine. It will attempt to preserve dumps
WARNING:      and log files, but this can not be guaranteed. This is the
WARNING:      pre-installation stage and all the software is loaded when
WARNING:      you reboot the system.

POST-INSTALL...
Saving the config files ...
NOTICE: uncommitted changes have been saved in
/var/db/config/juniper.conf.pre-install
Copying jpuppet pkg to /var/sw/pkg
Copying chef pkg to /var/sw/pkg
Pushing installation package to host...
Upgrade Host OS to 14.1-20160918_xl41X53_vjqfd.0...
./jinstall-qfx-5-flex-14.1-20160918_xl41X53_vjqfd.0-domestic.img.gz: OK
/boot/boot
Extracting image...
Host upgrade staging completed. Need reboot to complete upgrade installation.
WARNING: Changing next boot to SSD0
WARNING: Changing boot device enable from 0x3F to 0x3F
Install jinstall-vjunos completed

WARNING:      A REBOOT IS REQUIRED TO LOAD THIS SOFTWARE CORRECTLY. Use the
WARNING:      'request system reboot' command when software installation is
WARNING:      complete. To abort the installation, do not reboot your system,
WARNING:      instead use the 'request system software delete jinstall'
WARNING:      command as soon as this operation completes.

Saving state for rollback ...
fpc5
WARNING:      The software that is being installed has limited support.
WARNING:      Run 'file show /etc/notices/unsupported.txt' for details.

WARNING:      The software that is being installed has limited support.
WARNING:      Run 'file show /etc/notices/unsupported.txt' for details.

Saving contents of boot area prior to installation

WARNING:      This package will load JUNOS 14.1-20160918_xl41X53_vjqfd.0 software.
WARNING:      It will save JUNOS configuration files, and SSH keys
WARNING:      (if configured), but erase all other files and information
WARNING:      stored on this machine. It will attempt to preserve dumps

```

```

WARNING:      and log files, but this can not be guaranteed. This is the
WARNING:      pre-installation stage and all the software is loaded when
WARNING:      you reboot the system.

POST-INSTALL...
Saving the config files ...
NOTICE: uncommitted changes have been saved in
/var/db/config/juniper.conf.pre-install
Copying jpuppet pkg to /var/sw/pkg
Copying chef pkg to /var/sw/pkg
Pushing installation package to host...
Upgrade Host OS to 14.1-20160918_xl41X53_vjqfd.0...
./jinstall-qfx-5-flex-14.1-20160918_xl41X53_vjqfd.0-domestic.img.gz: OK
/boot/boot
Extracting image...
Host upgrade staging completed. Need reboot to complete upgrade installation.
WARNING: Changing next boot to SSD0
WARNING: Changing boot device enable from 0x18 to 0x18
Install jinstall-vjunos completed

WARNING:      A REBOOT IS REQUIRED TO LOAD THIS SOFTWARE CORRECTLY. Use the
WARNING:      'request system reboot' command when software installation is
WARNING:      complete. To abort the installation, do not reboot your system,
WARNING:      instead use the 'request system software delete jinstall'
WARNING:      command as soon as this operation completes.

Saving state for rollback ...
[Sep 24 12:05:28]: Backup upgrade done
[Sep 24 12:05:28]: Rebooting Backup RE

Rebooting fpcl
[Sep 24 12:05:28]:ISSU: Backup RE Prepare Done
[Sep 24 12:05:28]: Waiting for Backup RE reboot
[Sep 24 12:16:44]: GRES operational
[Sep 24 12:16:44]: "Initiating Chassis In-Service-Upgrade"
Chassis ISSU Started
[Sep 24 12:16:50]:ISSU: Preparing Daemons
.
.
[Sep 24 12:17:50]:ISSU: Daemons Ready for ISSU
[Sep 24 12:17:54]:ISSU: Starting Upgrade for FRUs
.
.
.
```

•

•

.

[Sep 24 12:21:09]: FPC 4 is undergoing a software upgrade

•

•

•

•

•

•

•

•

•

•

•

•

•

•

•

```
[Sep 24 12:30:16]: FPC 4 has come up after a software upgrade
```

•

•

•

•

•

•

[Sep 24 12:34:42]: FPC 5 is undergoing a software upgrade

•

•

•

•

•

•

•

•

•

•

•

.

•

•

•


```

.
[Sep 24 12:42:10]: FPC 5 has come up after a software upgrade
.
.
.
.
.
[Sep 24 12:45:42]:ISSU: Preparing for Switchover
[Sep 24 12:45:47]:ISSU: Ready for Switchover
[Sep 24 12:45:48]: Checking In-Service-Upgrade status
  Item                Status                Reason
  FPC 0                Online
  FPC 1                Online (ISSU)
  FPC 4                Online (ISSU)
  FPC 5                Online (ISSU)
[Sep 24 12:45:48]: Going to install image on master

WARNING:      The software that is being installed has limited support.
WARNING:      Run 'file show /etc/notices/unsupported.txt' for details.


WARNING:      The software that is being installed has limited support.
WARNING:      Run 'file show /etc/notices/unsupported.txt' for details.

Saving contents of boot area prior to installation

WARNING:      This package will load JUNOS 14.1-20160918_x141X53_vjqfd.0 software.
WARNING:      It will save JUNOS configuration files, and SSH keys
WARNING:      (if configured), but erase all other files and information
WARNING:      stored on this machine. It will attempt to preserve dumps
WARNING:      and log files, but this can not be guaranteed. This is the
WARNING:      pre-installation stage and all the software is loaded when
WARNING:      you reboot the system.

POST-INSTALL...
Saving the config files ...
NOTICE: uncommitted changes have been saved in
/var/db/config/juniper.conf.pre-install
Copying jpuppet pkg to /var/sw/pkg
Copying chef pkg to /var/sw/pkg
Pushing installation package to host...
Upgrade Host OS to 14.1-20160918_x141X53_vjqfd.0...
./jinstall-qfx-5-flex-14.1-20160918_x141X53_vjqfd.0-domestic.img.gz: OK
/boot/boot

```

```

Extracting image...
Host upgrade staging completed. Need reboot to complete upgrade installation.
WARNING: Changing next boot to SSD0
WARNING: Changing boot device enable from 0x3F to 0x3F
Install jinstall-vjunos completed

WARNING:      A REBOOT IS REQUIRED TO LOAD THIS SOFTWARE CORRECTLY. Use the
WARNING:      'request system reboot' command when software installation is
WARNING:      complete. To abort the installation, do not reboot your system,
WARNING:      instead use the 'request system software delete jinstall'
WARNING:      command as soon as this operation completes.

Saving state for rollback ...
failover links
[Sep 24 12:50:22]: Rebooting Old master
Chassis NSSU Completed
[Sep 24 12:50:22]:ISSU: IDLE

```

5. Log in after the reboot of the original master switch completes. To verify that the software on all Routing Engines in the VCF members has been upgraded, enter the following command:

```
user@switch> show version
```

RELATED DOCUMENTATION

[request system software nonstop-upgrade | 146](#)

[show chassis nonstop-upgrade | 166](#)

Configuring Graceful Routing Engine Switchover in a Virtual Chassis

[Understanding Nonstop Software Upgrade on a Virtual Chassis Fabric | 28](#)

Configuring Nonstop Bridging on Switches (CLI Procedure)

Example: Configuring Nonstop Active Routing on Switches

Example: Configuring Line-Card Upgrade Groups for Nonstop Software Upgrade on EX Series Switches

Configuring Dual-Root Partitions

								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								

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 | 33](#)
- [Preprovisioning a Virtual Chassis Fabric | 38](#)

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	Type	Trunk	Status	Speed	Neighbor	
or		ID		(mbps)	ID	Interface
PIC / Port						

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 | 33](#)

[Preprovisioning a Virtual Chassis Fabric | 38](#)

Verifying the Virtual Chassis Fabric Mode Settings

Purpose

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

You configure devices in a VCF into fabric mode using the **request virtual-chassis mode fabric** command so the devices can operate properly in a VCF.

For a mixed VCF, 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

Understanding Mixed EX Series and QFX Series Virtual Chassis

[Autoprovisioning a Virtual Chassis Fabric | 33](#)

[Preprovisioning a Virtual Chassis Fabric | 38](#)

Troubleshooting Virtual Chassis Fabric

IN THIS SECTION

- [Large-Scale Virtual Chassis Fabric Becomes Unstable When Logging is Enabled | 88](#)
- [Virtual Chassis Port Link Does Not Form | 89](#)
- [QFX5100 Leaf Device Assumes Routing Engine Role | 90](#)

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

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

Problem

Description: When detailed system logging or trace operations are enabled in larger-scale VCFs, you observe significant impact on VCF stability, such as:

- Increased VCF convergence time

- Traffic interruption

Cause

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.

Solution

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 **notice** 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:

- Use the **detail** tracing option or system logging levels at or above the **error** level only for short periods of time during troubleshooting, and disable these settings after gathering enough information to begin analyzing the issue.
- 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.
- Choose remote logging rather than local logging, and avoid logging to the console.

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](#) | 2

[traceoptions](#) | 134

Junos OS System Log Configuration Statements

Junos OS System Logging Facilities and Message Severity Levels

Configuration Statements for Virtual Chassis Fabric

IN THIS CHAPTER

- aliases (Virtual Chassis) | 92
- auto-provisioned | 94
- auto-sw-update | 95
- enhanced-hash-key | 99
- fabric-load-balance | 105
- fabric-tree-root | 107
- fpcs (NSSU Upgrade Groups) | 109
- id | 111
- inactivity-interval (Fabric Load Balance) | 112
- location (Virtual Chassis) | 114
- mac-persistence-timer | 115
- mastership-priority | 117
- member | 119
- member (NSSU Upgrade Groups) | 121
- no-management-vlan | 123
- nssu | 124
- preprovisioned | 126
- rcp-count | 127
- role | 129
- serial-number | 132
- traceoptions (Virtual Chassis) | 134
- upgrade-group | 138
- virtual-chassis | 140

aliases (Virtual Chassis)

Syntax

```
aliases {
  serial-number serial-number {
    alias-name alias-name;
  }
}
```

Hierarchy Level

[edit [virtual-chassis](#)]

Release Information

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

Description

Optionally create an alias for a member switch in a Virtual Chassis or Virtual Chassis Fabric (VCF). You can more clearly identify the member switches in your Virtual Chassis or VCF by assigning text labels (alias names) to them, associating the alias name with member switch serial numbers. Aliases are for administrative purposes only and have no effect on member switch operations.

The **show virtual-chassis** command displays assigned alias names in the **Alias-Name** output field.

Options

serial-number—The permanent serial number for the member switch in the Virtual Chassis or VCF that you want to label with an alias name. Serial number values are case-sensitive.

The **show virtual-chassis** command displays the serial number for all members in the Virtual Chassis or VCF in the **Serial No** output field.

alias-name—The text label you want to assign to the member switch.

For example, if you configure alias name **dc-floor-1** in a VCF for member switch 0 with serial number AB012345678 as follows:

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

then you'll see the alias **dc-floor-1** in the Alias-Name column in the output from the **show virtual-chassis** command:

show virtual-chassis

user@switch> **show virtual-chassis**

Preprovisioned Virtual Chassis Fabric							
Fabric ID: 9d5d.5556.919a							
Fabric Mode: Enabled							
						Mstr	
Member ID	Status	Serial No	Alias-Name	Model	prio	Role	
0 (FPC 0)	Prsnt	AB0123456789	dc-floor-1	qfx5100-48s-6q	129	Master	
.							
.							
.							

- 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 33
Preprovisioning a Virtual Chassis Fabric 38
Configuring an EX4650 or a QFX Series Virtual Chassis
Understanding Virtual Chassis Fabric Components 5
Understanding Virtual Chassis Components

auto-provisioned

Syntax

```
auto-provisioned;
```

Hierarchy Level

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

Release Information

Statement introduced in Junos OS Release 13.2X51-D20 for EX Series switches and QFX Series devices in a Virtual Chassis Fabric (VCF).

Description

Enable the auto-provisioned configuration mode for a Virtual Chassis Fabric (VCF).

When a VCF is autoprovisioned, you can plug and play leaf devices that have not been configured or are zeroized into your VCF without requiring 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.

A spine or leaf device whose fabric or mixed mode setting is automatically set as part of the VCP automatic conversion process upon interconnecting it into a VCF automatically reboots to complete the configuration and allow the device to properly join the VCF. Alternatively, you can avoid a potentially unexpected device reboot by manually configuring the fabric or mixed mode setting on the device, and manually rebooting it *before* interconnecting it into the VCF.

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 | 33](#)

[Understanding Virtual Chassis Fabric Configuration | 17](#)

auto-sw-update

Syntax

```
auto-sw-update {
  (ex-4200 | ex-4300 | ex-4500 | ex-4600 | qfx-3 | qfx-5)
  package-name package-name;
}
```

Hierarchy Level

[edit [virtual-chassis](#)]

Release Information

Statements introduced in Junos OS Release 10.0 for EX Series switches.

The **ex-4200** and **ex-4500** options introduced in Junos OS Release 12.2 for EX Series switches.

Statements introduced in Junos OS Release 13.2X50-D15 for the QFX Series.

The **ex-4300**, **qfx-3**, and **qfx-5** options introduced in Junos OS Release 13.2X51-D20.

Statements introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).

The **ex-4600** option introduced in Junos OS Release 13.2X51-D25.

Description

Enable the automatic software update feature for Virtual Chassis or Virtual Chassis Fabric (VCF) configurations.

You should only use the keywords that specify a device or device family—**ex-4300**, **ex-4600**, **qfx-3**, and **qfx-5**—when configuring automatic software update on a mixed mode Virtual Chassis or Virtual Chassis Fabric (VCF). You can simply specify the *package-name* without specifying individual device keywords in non-mixed Virtual Chassis or VCF topologies.

NOTE: The automatic software update feature is not supported for a mixed mode EX4300 Virtual Chassis with a combination of EX4300 multigigabit model (EX4300-48MP) switches and other EX4300 switches.

You must enter the **auto-sw-update** 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.

The Junos OS package for an EX4500 switch updates the software for EX4500 and EX4550 switches. You do not, therefore, need to specify the **ex-4500** 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 **ex-4500** keyword once to configure automatic software update for all EX4500 and EX4550 member switches in the s37628)ame mixed Virtual Chassis.

The Junos OS package for a QFX3500 device updates the software for QFX3500 and QFX3600 devices. You do not, therefore, need to specify the **qfx-3** keyword when configuring automatic software update for a Virtual Chassis composed entirely of QFX3500 and QFX3600 devices. You also have to enter the **qfx-3** keyword only once to configure automatic software update for all QFX3500 and QFX3600 member devices in the same mixed Virtual Chassis.

The Junos OS package for a QFX5110 device updates the software for both QFX5110 and QFX5100 devices in a QFX5110 Virtual Chassis or VCF, so you do not need to specify the **qfx-5** keyword when configuring automatic software update for a QFX5110 Virtual Chassis or VCF composed of both types of devices.



CAUTION: A QFX5100 switch running a Junos OS software image with “-qfx-5-” in the package filename *must* first be upgraded to a Junos OS software image with “-qfx-5e-” in the package filename before it can be added to a mixed QFX5110 Virtual Chassis or VCF. The automatic software update process cannot update a switch from a “-qfx-5-” image to a “-qfx-5e-” image. See [“Upgrading a QFX5100 Switch with a USB Device to Join a QFX5110 Virtual Chassis or Virtual Chassis Fabric” on page 64](#).

After a QFX5100 switch is installed with a “-qfx-5e-” Junos OS software image, the automatic software update process can successfully update the switch automatically with a different version of a “-qfx-5e-” Junos OS image to match the other members in the Virtual Chassis or VCF.

The remaining statement is explained separately. See [CLI Explorer](#).

Default

The automatic software update feature is disabled.

Options

package-name *package-name*—Specify the software package name or a path to a Junos OS software image to update new members being added to a Virtual Chassis or VCF. .

- 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 file access and transfer formats:

ftp://hostname/pathname/package-name

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

http://hostname/pathname/package-name

ex-4200—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 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. Automatic software update is not supported for mixed mode EX4300 Virtual Chassis with a combination of EX4300 multigigabit model (EX4300-48MP) switches and other EX4300 switches, and this option is not applicable in that case.

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 Virtual Chassis or VCF.

qfx-5—Specify a path to a Junos OS image for a QFX5100 device when enabling automatic software update for a mixed QFX5100 Virtual Chassis or VCF. You do not need to use this option for a QFX5110 Virtual Chassis or VCF that contains both QFX5110 and QFX5100 switches because the same software image runs on either type of switch.

Required Privilege Level

system—To view this statement in the configuration.

system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

<i>Configuring Automatic Software Update on Virtual Chassis Member Switches</i>
<i>Understanding Software Upgrades in a Virtual Chassis</i>
Understanding Software Upgrades in a Virtual Chassis Fabric 26
<i>Example: Configuring Automatic Software Update on EX4200 Virtual Chassis Member Switches</i>

enhanced-hash-key

List of Syntax

[Syntax \(EX Series\) on page 99](#)

[Syntax \(QFX5000 Line of Switches\) on page 100](#)

[Syntax \(QFX10000 Series Switches\) on page 102](#)

Syntax (EX Series)

```
enhanced-hash-key {
  ecmp-resilient-hash;
  fabric-load-balance {
    flowlet {
      inactivity-interval interval;
    }
    per-packet;
  }
  hash-mode {
    layer2-header;
    layer2-payload;
  }
  family inet {
    no-ipv4-destination-address;
    no-ipv4-source-address;
    no-l4-destination-port;
    no-l4-source-port;
    no-protocol;
    vlan-id;
  }
  family 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;
  }
}
```

Syntax (QFX5000 Line of Switches)

```

enhanced-hash-key {
  ecmp-dlb {
    assigned-flow;
    per-packet;
    flowlet inactivity-interval;
    ether-type (ipv4|ipv6|mpls);
  }
  ecmp-resilient-hash;
  fabric-load-balance {
    flowlet {
      inactivity-interval interval;
    }
    per-packet;
  }
  hash-mode {
    layer2-header;
    layer2-payload;
    gtp-header-offset offset-value;
  }
  hash-parameters {
    ecmp {
      function {
        (crc16-bisync | crc16-ccitt | crc32-hi | crc32-lo);
      }
      offset offset;
      preprocess;
    }
    lag {
      function {
        (crc16-bisync | crc16-ccitt | crc32-hi | crc32-lo);
      }
      offset offset;
      preprocess;
    }
  }
  family inet {
    gtp-tunnel-endpoint-identifier;
    no-ipv4-destination-address;
    no-ipv4-source-address;
    no-l4-destination-port;
    no-l4-source-port;
    no-protocol;
    vlan-id;
  }
}

```

```
}  
family 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;  
}  
}
```

Syntax (QFX10000 Series Switches)

```

enhanced-hash-key {
  hash-seed seed-value;
  family inet {
    gtp-tunnel-endpoint-identifier;
    no-ipv4-destination-address;
    no-ipv4-source-address;
    no-l4-destination-port;
    no-l4-source-port;
    no-incoming-port;
  }
  family inet6 {
    gtp-tunnel-endpoint-identifier;
    ipv6-flow-label;
    no-ipv6-destination-address;
    no-ipv6-source-address;
    no-l4-destination-port;
    no-l4-source-port;
    no-incoming-port;
  }
  layer2 {
    destination-mac-address
    inner-vlan-id;
    no-ether-type;
    no-vlan-id;
    source-mac-address;
  }
  no-mpls;
  gre {
    key;
    protocol;
  }
  vxlan-vnid
}

```

Hierarchy Level

[edit forwarding-options]

Release Information

Statement introduced in Junos OS Release 13.2X51-D15 for EX Series switches.

Statement introduced in Junos OS Release 13.2X51-D20 for QFX Series devices.

The **fabric-load-balance** statement introduced in Junos OS Release 14.1X53-D10.

The **fabric-load-balance** statement deprecated starting in Junos OS Releases 14.1X53-D46, 15.1R7, 16.1R6, 17.1R3, 17.2R2, 17.3R2, and 17.4R1.

The **hash-seed** statement introduced in Junos OS Release 15.1X53-D30.

The **ecmp-dlb** statement introduced in Junos OS Release 19.4R1 for QFX5120-32C and QFX5120-48Y switches.

Description

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).

NOTE: Starting in Junos OS Release 14.1X53-D46, 15.1R7, 16.1R6, 17.1R3, 17.2R2, 17.3R2, and 17.4R1, the ALB feature is deprecated. If **fabric-load-balance** is enabled in the configuration for a VCF, delete the configuration item upon upgrading Junos OS.

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.

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.

When ECMP is enabled, the hashing algorithm determines how incoming traffic is forwarded to the next-hop device.

On QFX10000 Series switches, you can configure the hash seed for load balancing. By default, the QFX10000 Series switches use the system MAC address to generate a hash seed value. You can configure the hash seed value using the **hash-seed** 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 generates a hash seed value based on the system MAC address.

The remaining statements are explained separately. See [CLI Explorer](#).

Starting in Junos OS Release 18.4R1, symmetric hashing is supported on the QFX10000 Series switches. You configure the **no-incoming-port** option under the [edit forwarding-options enhanced-hash-key] hierarchy. By default, Dynamic IP (DIP), SIP, Layer 4 source and destination ports, and the incoming port are used for hashing. You can only configure symmetric hashing at the global level.

Starting in Junos OS Release 19.4R1, the dynamic load balancing on ECMP is supported on QFX5120-32C and QFX5120-48Y switches. You can configure the **ecmp-dlb** option under the [edit forwarding-options enhanced-hash-key] hierarchy. Refer *Dynamic Load Balancing* for more details.

Required Privilege Level

interface—To view this statement in the configuration.

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

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

Understanding Passive Monitoring

Understanding Per-Packet Load Balancing

[show forwarding-options enhanced-hash-key](#) | 169

fabric-load-balance

Syntax

```
fabric-load-balance {  
  flowlet {  
    inactivity-interval interval;  
  }  
  per-packet;  
}
```

Hierarchy Level

[edit forwarding-options [enhanced-hash-key](#)]

Release Information

Statement introduced in Junos OS Release 14.1X53-D10.

Statement deprecated starting in Junos OS Releases 14.1X53-D46, 15.1R7, 16.1R6, 17.1R3, 17.2R2, 17.3R2, and 17.4R1.

Description

Enable adaptive load balancing (ALB) for a VCF, and specify how ALB is implemented.

NOTE: Starting in Junos OS Release 14.1X53-D46, 15.1R7, 16.1R6, 17.1R3, 17.2R2, 17.3R2, and 17.4R1, the ALB feature is deprecated. If **fabric-load-balance** is enabled in the configuration, delete the configuration item upon upgrading Junos OS.

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.

Default

ALB is disabled, by default.

If you do not specify a mode when enabled ALB, ALB is enabled using flowlet mode with an inactivity timer of 16 microseconds.

Options

flowlet—Implement ALB by using flowlets.

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.

The inactivity interval is 16 microseconds by default, and can be configured using the **inactivity-interval** 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.

Implementing ALB using flowlets is especially 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 elephant flows by splicing them into smaller flowlets using ALB.

per-packet—Implement ALB using per-packet mode.

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.

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](#) | 23

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 (regardless of the role—Routing Engine or linecard), 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 23 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 | 33](#)

[Preprovisioning a Virtual Chassis Fabric | 38](#)

[Adding a Device to a Virtual Chassis Fabric | 49](#)

fpcs (NSSU Upgrade Groups)

Syntax

```
fpcs (slot-number | [list-of-slot-numbers]);
```

Hierarchy Level

```
[edit chassis nssu upgrade-group group-name],  
[edit chassis nssu upgrade-group group-name member member-id]
```

Release Information

Statement introduced in Junos OS Release 10.4 for EX Series switches.

Statement introduced in Junos OS Release 13.2X51-D20 for QFX Series switches.

Statement introduced in Junos OS Release 20.1R1 for EX4650-48Y switches.

Description

Configure switch line cards, Virtual Chassis member switches, or Virtual Chassis Fabric (VCF) member switches as part of an NSSU upgrade group.

To reduce the time an NSSU takes, you can configure line-card upgrade groups for an EX6200 or EX8200 switch with redundant Routing Engines; an EX8200 Virtual Chassis; an EX4650 Virtual Chassis with more than three member switches; QFX3500, QFX3600, and QFX5100 Virtual Chassis; or a QFX5100 Virtual Chassis Fabric (VCF). NSSU upgrades the devices in the order in which you configure the upgrade groups, so you can also use upgrade groups to control the upgrade sequence.

For switches that have separate line cards, use this statement to assign one or more line cards to an NSSU upgrade group based on their line-card slot numbers.

For Virtual Chassis or VCF member switches that do not have separate line cards, use this statement to assign one or more Virtual Chassis or VCF members to an NSSU upgrade group by specifying their member IDs.

NOTE: For a Virtual Chassis or VCF, you do not use this statement with the **member** option. When to use the **member** statement hierarchy is explained next.

To configure an upgrade group that includes line cards on switches that support multiple line cards and comprise a Virtual Chassis, use this statement with the **member** option to specify the Virtual Chassis member ID and the desired line card slot number or numbers on that member switch to include in the upgrade group. Use multiple statements to add line cards from different Virtual Chassis members to the upgrade group.

Options

list-of-slot-numbers—A list of slot numbers of multiple line cards or member IDs of Virtual Chassis or VCF members to be included in the upgrade group. Separate multiple slot numbers or member IDs with spaces and enclose the list in square brackets—for example: [3 4 7].

slot-number—The slot number of a single line card or member ID of a Virtual Chassis or VCF member to be included in the upgrade group.

Required Privilege Level

interface—To view this statement in the configuration.

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

RELATED DOCUMENTATION

Example: Configuring Line-Card Upgrade Groups for Nonstop Software Upgrade on EX Series Switches

[Configuring Line-Card Upgrade Groups for Nonstop Software Upgrade | 70](#)

id

Syntax

```
id id;
```

Hierarchy Level

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

Release Information

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).

Description

Configure the alphanumeric string that identifies a Virtual Chassis or Virtual Chassis Fabric (VCF) configuration.

Options

id—Virtual Chassis ID (VCID), which uses the ISO family address format—for example, **9622.6ac8.5345**.

Required Privilege Level

system—To view this statement in the configuration.

system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Example: Assigning the Virtual Chassis ID to Determine Precedence During an EX4200 Virtual Chassis Merge

Assigning the Virtual Chassis ID to Determine Precedence During a Virtual Chassis Merge

Configuring an EX4650 or a QFX Series Virtual Chassis

[Autoprovisioning a Virtual Chassis Fabric | 33](#)

[Preprovisioning a Virtual Chassis Fabric | 38](#)

Configuring an EX8200 Virtual Chassis (CLI Procedure)

Understanding Virtual Chassis Member ID Numbering in an EX8200 Virtual Chassis

inactivity-interval (Fabric Load Balance)

Syntax

```
inactivity-interval interval;
```

Hierarchy Level

```
[edit forwarding-options enhanced-hash-key fabric-load-balance flowlet]
```

Release Information

Statement introduced in Junos OS Release 14.1X53-D10.

Statement deprecated starting in Junos OS Releases 14.1X53-D46, 15.1R7, 16.1R6, 17.1R3, 17.2R2, 17.3R2, and 17.4R1.

Description

Set the inactivity interval for adaptive load balancing (ALB) using flowlets within a VCF.

NOTE: Starting in Junos OS Release 14.1X53-D46, 15.1R7, 16.1R6, 17.1R3, 17.2R2, 17.3R2, and 17.4R1, the ALB feature is deprecated. If **fabric-load-balance** is enabled in the configuration and **inactivity-interval** is set, delete these configuration items upon upgrading Junos OS.

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.

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.

Default

ALB is disabled, by default.

If ALB is enabled without specifying a mode, ALB is enabled using flowlet mode with an inactivity interval of 16 microseconds.

If ALB is enabled using flowlet mode without specifying an inactivity interval, the inactivity interval is set to 16 microseconds.

Options

interval—The amount of time that occurs between packet bursts on a sending interface before a traffic flow is spliced into flowets.

Range: 16 microseconds (**16us**) to 32 milliseconds(**32ms**).

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](#) | 23

location (Virtual Chassis)

Syntax

```
location location;
```

Hierarchy Level

```
[edit virtual-chassis member member-id]
```

Release Information

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).

Description

Set a description of the location of the Virtual Chassis or VCF member switch or external Routing Engine.

The **Location** field is visible to users who enter the **show virtual-chassis status detail** command.

Setting this description has no effect on the operation of the member device.

Options

location—Location of the current member switch or external Routing Engine. The **location** can be any single word.

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 | 33](#)

[Preprovisioning a Virtual Chassis Fabric | 38](#)

[Configuring an EX4650 or a QFX Series Virtual Chassis](#)

[Configuring an EX2300, EX3400, or EX4300 Virtual Chassis](#)

[Configuring EX4600 Switches in a Mixed or Non-Mixed Virtual Chassis](#)

mac-persistence-timer

Syntax

```
mac-persistence-timer (minutes | disable);
```

Hierarchy Level

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

Release Information

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

Option **disable** introduced in Junos OS Release 12.2 for EX Series switches.

The maximum timer limit changed from no maximum timer limit to 60 minutes in Junos OS Release 12.2 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

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.

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.

Default

The MAC persistence timer is set to 10 minutes by default.

Options

minutes—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.

disable—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.

Required Privilege Level

system—To view this statement in the configuration.

system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Configuring the Timer for the Backup Member to Start Using Its Own MAC Address, as Master of a Virtual Chassis

[Autoprovisioning a Virtual Chassis Fabric | 33](#)

[Preprovisioning a Virtual Chassis Fabric | 38](#)

mastership-priority

Syntax

```
mastership-priority number;
```

Hierarchy Level

```
[edit virtual-chassis member member-id]
```

Release Information

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

Mastership priority option **0** 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).

Description

Assign a mastership priority value to a member of a Virtual Chassis or VCF. The mastership priority is the most important factor that determines which member assumes the master Routing Engine role in the Virtual Chassis or VCF.

A Virtual Chassis or VCF configuration has two member switches running in the Routing Engine role, one as the master and one as the backup. The remaining members operate in the linecard role. In a preprovisioned Virtual Chassis or VCF, you configure the **role** for each member by associating it with the member switch's serial number, and the Virtual Chassis assigns default mastership priorities based on the assigned role. In a non-provisioned Virtual Chassis or VCF, you can configure roles explicitly, or the mastership priorities of the members implicitly designate the role of each member based on the mastership election algorithm.

The Virtual Chassis or VCF elects the member switch with the highest mastership priority as the master Routing Engine. When two members have the same highest mastership priority value among all the members, the master election algorithm uses other factors to determine which of the two members functions as the master and which is the backup. See *Understanding How the Master in a Virtual Chassis Is Elected* for all the conditions considered in mastership election.

You should assign the *same* (highest) mastership priority value to the members you want to be the master and backup Routing Engine members to ensure reliable graceful Routing Engine switchover (GRES) operation. This and other mastership election considerations also help keep mastership from switching back and forth rapidly between the two members under failover conditions.

This statement is not used for the EX8200 Virtual Chassis, which determines mastership by external Routing Engine uptime.

A switch with a mastership priority of **0** never takes the master or backup role.

Default

128

Options

number—Mastership priority value.

Range: 0 through 255

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 33
Preprovisioning a Virtual Chassis Fabric 38
<i>Configuring an EX4650 or a QFX Series Virtual Chassis</i>
<i>Configuring an EX2300, EX3400, or EX4300 Virtual Chassis</i>
<i>Configuring EX4600 Switches in a Mixed or Non-Mixed Virtual Chassis</i>

member

Syntax

```
member member-id {
  fabric-tree-root;
  location location;
  mastership-priority number;
  no-management-vlan;
  serial-number serial-number;
  role role;
}
```

Hierarchy Level

[edit [virtual-chassis](#)]

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 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.

Default

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.

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.

Options

member-id—Identifies a specific member switch of a Virtual Chassis or VCF configuration.

The exact range for a specific Virtual Chassis or VCF depends on the number of switches allowed in the Virtual Chassis or VCF.

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.

The remaining statement options set characteristics of the Virtual Chassis or VCF member, and are explained separately.

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 33
Preprovisioning a Virtual Chassis Fabric 38
Adding a Device to a Virtual Chassis Fabric 49
<i>Configuring an EX4650 or a QFX Series Virtual Chassis</i>
<i>Configuring an EX2300, EX3400, or EX4300 Virtual Chassis</i>
<i>Configuring EX4600 Switches in a Mixed or Non-Mixed Virtual Chassis</i>
<i>Configuring an EX9200 Virtual Chassis</i>
<i>Configuring an EX4650 or a QFX Series Virtual Chassis</i>

member (NSSU Upgrade Groups)

Syntax

```
member member-id {
  fpcs (slot-number | [list-of-slot-numbers]);
}
```

Hierarchy Level

```
[edit chassis nssu upgrade-group group-name]
```

Release Information

Statement introduced in Junos OS Release 11.1 for EX Series switches.

Statement introduced in Junos OS Release 13.2X51-D20 for QFX Series switches.

Statement introduced in Junos OS Release 20.1R1 for EX4650-48Y switches.

Description

Specify the Virtual Chassis member whose line-card slot numbers you are assigning to an NSSU upgrade group.

NOTE: This statement is not applicable to Virtual Chassis or VCF member switches that do not support separate line cards. To configure Virtual Chassis or VCF member switches that do not have separate line cards into an NSSU upgrade group, use the **fpcs** statement alone, and specify the Virtual Chassis or VCF member IDs to include in the upgrade group in place of line card slot numbers.

To reduce the time an NSSU takes, you can configure NSSU line-card upgrade groups on an EX6200 or EX8200 switch with redundant Routing Engines; EX8200 Virtual Chassis; QFX3500, QFX3600, and QFX5100 Virtual Chassis; and Virtual Chassis Fabric (VCF).

To configure an upgrade group that includes line cards on different switches that support multiple line cards and comprise a Virtual Chassis, use this statement hierarchy with the **fpcs** option to first specify the Virtual Chassis member ID and then desired line card slot number or numbers on that member switch to include in the upgrade group. Use multiple statements to add line cards from different Virtual Chassis members to the upgrade group.

Options

member-id—The ID of the Virtual Chassis or VCF member switch containing one or more line cards to include in an NSSU upgrade group.

The remaining statement is explained separately. See [CLI Explorer](#).

Required Privilege Level

interface—To view this statement in the configuration.

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

RELATED DOCUMENTATION

Example: Configuring Line-Card Upgrade Groups for Nonstop Software Upgrade on EX Series Switches

[Configuring Line-Card Upgrade Groups for Nonstop Software Upgrade | 70](#)

no-management-vlan

Syntax

```
no-management-vlan;
```

Hierarchy Level

```
[edit virtual-chassis member member-id]
```

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

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.

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:

```
virtual-chassis {
  member 2 {
    no-management-vlan;
  }
}
```

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 **ifconfig** command.

Required Privilege Level

system—To view this statement in the configuration.

system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

Understanding Global Management of a Virtual Chassis

[Understanding Virtual Chassis Fabric Configuration](#) | 17

nssu

Syntax

```
nssu {
  rcp-count number;
  upgrade-group group-name {
    fpcs (slot-number | [list-of-slot-numbers]);
    member member-id {
      fpcs (slot-number | [list-of-slot-numbers]);
    }
  }
}
```

Hierarchy Level

[edit chassis]

Release Information

Statement introduced in Junos OS Release 10.4 for EX Series switches.

Statement introduced in Junos OS Release 13.2X51-D20 for QFX Series switches.

rcp-count statement introduced in Junos OS Release 14.1X53-D40 for QFX5100 switches only.

Statement introduced in Junos OS Release 20.1R1 for EX4650-48Y switches.

Description

Configure parameters that affect the nonstop software upgrade (NSSU) process.

NOTE: You use the **request system software nonstop-upgrade** command to initiate NSSU.

The **rcp-count** option (available only on QFX5100 switches) sets the number of parallel **rcp** sessions that NSSU uses to copy the new software to multiple Virtual Chassis or VCF member switches at a time.

The **upgrade-group** options define line-card upgrade groups for NSSU. When you initiate NSSU with at least one upgrade group configured, NSSU upgrades the line cards or Virtual Chassis or VCF members in each upgrade group to the new software version at the same time, in the order in which you configured them. Upgrade groups reduce the time required to complete an NSSU operation and control the order in which the line cards or members are upgraded.

Line-card upgrade groups are supported on some EX Series switches and EX Series Virtual Chassis that support NSSU and on a QFX5100 VCF.

These statements are all explained separately. You can also consult [CLI Explorer](#).

Default

If you do not configure **rcp-count**, NSSU uses a default algorithm to determine the number of parallel **rcp** sessions to use based on the number of members in the Virtual Chassis or VCF.

If you do not define any line-card upgrade groups, NSSU upgrades line cards or members of a Virtual Chassis or VCF one at a time in ascending order by slot or member number.

Required Privilege Level

interface—To view this statement in the configuration.

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

RELATED DOCUMENTATION

[Configuring Line-Card Upgrade Groups for Nonstop Software Upgrade | 70](#)

Example: Configuring Line-Card Upgrade Groups for Nonstop Software Upgrade on EX Series Switches

preprovisioned

Syntax

```
preprovisioned;
```

Hierarchy Level

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

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

Enable the preprovisioned configuration mode for a Virtual Chassis or Virtual Chassis Fabric (VCF) configuration.

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.

You must use this statement to configure an EX8200 Virtual Chassis. Nonprovisioned configuration of an EX8200 Virtual Chassis is not supported.

Required Privilege Level

system—To view this statement in the configuration.

system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

[Preprovisioning a Virtual Chassis Fabric | 38](#)

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 EX4200, EX4500, or EX4550 Virtual Chassis (CLI Procedure)

Configuring an EX8200 Virtual Chassis (CLI Procedure)

Configuring an EX9200 Virtual Chassis

Configuring an EX4650 or a QFX Series Virtual Chassis

Removing or Replacing a Member Switch of a Virtual Chassis Configuration

rcp-count

Syntax

```
rcp-count number;
```

Hierarchy Level

```
[edit chassis nssu]
```

Release Information

Statement introduced in Junos OS Release 14.1X53-D40 for QFX5100 switches only.

Description

(QFX5100 Virtual Chassis or Virtual Chassis Fabric [VCF] only) Optionally override the default algorithm that defines the number of parallel **rcp** sessions used for copying the new software image to member switches in a Virtual Chassis or VCF for nonstop software upgrade (NSSU).

You can upgrade the software running on all members of a Virtual Chassis or VCF using NSSU. At the beginning of the upgrade process, NSSU uses **rcp** to copy the new software from the master switch to each of the member switches. Starting in Junos OS Release 14.1X53-D40, to minimize the time to copy the files to all members, NSSU uses parallel **rcp** sessions to copy the software to multiple members at the same time, rather than copying sequentially to each member in turn.

By default, the number of parallel **rcp** sessions that NSSU launches at a time follows an algorithm that optimizes the transfer time based on the total number of members in the Virtual Chassis or VCF, so that the new software is transferred to about half the members in parallel up to a maximum of 8 members at one time. After the first set of parallel copy operations are complete, NSSU launches the next set of parallel copy operations, and so on as needed until all members are updated.

Although the default value is usually optimal, you can configure the **rcp-count** statement to set a specific value for the number of parallel copy sessions instead. For example, you might want to use a lower number of parallel sessions to avoid potential impact on Virtual Chassis or VCF forwarding activity during NSSU.

After copying the new software to all members using as many parallel **rcp** sessions as needed, the NSSU process continues by rebooting each member with the new software in sequence, starting with the member in the backup Routing Engine role.

NOTE: If copying the new software to any member fails, NSSU aborts the upgrade process for the entire Virtual Chassis or VCF, logs the error condition, and initiates an error recovery measure to remove the new software from the members to which it was already transferred.

Options

number—Number of parallel **rcp** sessions for NSSU to launch at a time, which overrides the default algorithm. Allowable values are 0 through 8. Specifying a value of 0 suppresses parallel copy sessions; the new software is copied to each member sequentially.

Required Privilege Level

interface—To view this statement in the configuration.

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

RELATED DOCUMENTATION

Understanding Nonstop Software Upgrade on a Virtual Chassis and Mixed Virtual Chassis

[Understanding Nonstop Software Upgrade on a Virtual Chassis Fabric](#) | 28

role

Syntax

```
role (line-card | routing-engine);
```

Hierarchy Level

```
[edit virtual-chassis member member-id]
```

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

Specify the roles of the members of the Virtual Chassis or a Virtual Chassis Fabric (VCF) in a preprovisioned Virtual Chassis. For a mixed Virtual Chassis or VCF, see *Understanding Mixed EX Series and QFX Series Virtual Chassis* or [“Understanding Mixed Virtual Chassis Fabric” on page 20](#) for any recommendations or requirements for assigning the Routing Engine role based on the types of switches comprising the Virtual Chassis or VCF.

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 also be associated with the member’s serial number (see [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 (see [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 **line-card** role and mastership priority 0 by default.

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 **line-card** role by default. In that case, a member that is functioning in a **line-card** 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 (see [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 Routing Engine 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.

In a Virtual Chassis composed of EX Series switches (except EX8200 switches) or QFX Series switches, specify two and only two members in the **routing-engine** role. 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. For a mixed Virtual Chassis or VCF, see *Understanding Mixed EX Series and QFX Series Virtual Chassis* or [“Understanding Mixed Virtual Chassis Fabric” on page 20](#) for specific recommendations or requirements for assigning the Routing Engine role based on the types of switches comprising the Virtual Chassis or VCF. The remaining switches are configured into the linecard role.

In an EX8200 Virtual Chassis, all XRE200 External Routing Engines must be in the **routing-engine** role.

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 33
Preprovisioning a Virtual Chassis Fabric 38
<i>Configuring an EX2300, EX3400, or EX4300 Virtual Chassis</i>
<i>Configuring EX4600 Switches in a Mixed or Non-Mixed Virtual Chassis</i>
<i>Configuring an EX9200 Virtual Chassis</i>
<i>Configuring an EX4650 or a QFX Series Virtual Chassis</i>
<i>Removing or Replacing a Member Switch of a Virtual Chassis Configuration</i>

serial-number

Syntax

```
serial-number serial-number;
```

Hierarchy Level

```
[edit virtual-chassis member member-id]
```

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

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. Serial number values are case-sensitive.

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.

Options

serial-number—Permanent serial number for the external Routing Engine or for the member switch.

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 | 33](#)

[Preprovisioning a Virtual Chassis Fabric | 38](#)

[Configuring an EX2300, EX3400, or EX4300 Virtual Chassis](#)

[Configuring EX4600 Switches in a Mixed or Non-Mixed Virtual Chassis](#)

[Configuring an EX9200 Virtual Chassis](#)

| *Configuring an EX4650 or a QFX Series Virtual Chassis*

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.

Required Privilege Level

system—To view this statement in the configuration.

system-control—To add this statement to the configuration.

RELATED DOCUMENTATION

<i>Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis</i>
<i>Verifying the Member ID, Role, and Neighbor Member Connections of a Virtual Chassis Member</i>
<i>Verifying That Virtual Chassis Ports Are Operational</i>
<i>Troubleshooting an EX Series Virtual Chassis</i>
Troubleshooting Virtual Chassis Fabric 88

upgrade-group

Syntax

```
upgrade-group group-name {
  fpcs (slot-number | [list-of-slot-numbers]);
  member member-id {
    fpcs (slot-number | [list-of-slot-numbers]);
  }
}
```

Hierarchy Level

[edit chassis [nssu](#)]

Release Information

Statement introduced in Junos OS Release 10.4 for EX Series switches.

Statement introduced in Junos OS Release 13.2X51-D20 for QFX Series switches.

Statement introduced in Junos OS Release 20.1R1 for EX4650-48Y switches.

Description

Assign a name to a line-card upgrade group being created for nonstop software upgrade (NSSU).

To reduce the time an NSSU takes, you can configure line-card upgrade groups on an EX6200 or EX8200 switch with redundant Routing Engines; EX8200 Virtual Chassis; EX4650 Virtual Chassis; QFX3500, QFX3600, and QFX5100 Virtual Chassis; and QFX5100 Virtual Chassis Fabric (VCF).

NSSU upgrades the groups in the order that they appear in the configuration (in other words, in the order you configure them). If you do not define any line-card upgrade groups, NSSU upgrades line cards or members of a Virtual Chassis or VCF one at a time in ascending order by slot or member number.

Options

group-name—Name of the upgrade group.

The remaining statements are explained separately. See [CLI Explorer](#).

Required Privilege Level

interface—To view this statement in the configuration.

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

RELATED DOCUMENTATION

[Configuring Line-Card Upgrade Groups for Nonstop Software Upgrade](#) | 70

Example: Configuring Line-Card Upgrade Groups for Nonstop Software Upgrade on EX Series Switches

[Upgrading Software on a Virtual Chassis Fabric Using Nonstop Software Upgrade](#) | 74

virtual-chassis

Syntax

```

virtual-chassis {
  aliases {
    serial-number serial-number {
      alias-name alias-name;
    }
  }
  auto-conversion;
  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;
    serial-number;
    role;
  }
  no-auto-conversion;
  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. See [CLI Explorer](#).

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.

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 | 33](#)

[Preprovisioning a Virtual Chassis Fabric | 38](#)

[Adding a Device to a Virtual Chassis Fabric | 49](#)

[Configuring an EX4650 or a QFX Series Virtual Chassis](#)

[Configuring an EX2300, EX3400, or EX4300 Virtual Chassis](#)

[Configuring EX4600 Switches in a Mixed or Non-Mixed Virtual Chassis](#)

[Configuring an EX9200 Virtual Chassis](#)

Operational Commands for Virtual Chassis Fabric

IN THIS CHAPTER

- clear virtual-chassis vc-port statistics | 143
- request session member | 145
- request system software nonstop-upgrade | 146
- request virtual-chassis mode | 158
- request virtual-chassis reactivate | 162
- request virtual-chassis vc-port | 163
- request virtual-chassis vc-port diagnostics optics | 165
- show chassis nonstop-upgrade | 166
- show forwarding-options enhanced-hash-key | 169
- show virtual-chassis | 176
- show virtual-chassis active-topology | 185
- show virtual-chassis device-topology | 192
- show virtual-chassis login | 200
- show virtual-chassis mode | 202
- show virtual-chassis protocol adjacency | 205
- show virtual-chassis protocol database | 209
- show virtual-chassis protocol interface | 214
- show virtual-chassis protocol route | 217
- show virtual-chassis protocol statistics | 221
- show virtual-chassis vc-path | 225
- show virtual-chassis vc-port | 231
- show virtual-chassis vc-port diagnostics optics | 237
- show virtual-chassis vc-port statistics | 256

clear virtual-chassis vc-port statistics

Syntax

```
clear virtual-chassis vc-port statistics
<all-members>
<interface-name>
<local>
<member member-id>
```

Release Information

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

The options **all-members** and **local** 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.

Description

Clear—reset to zero (0)—the traffic statistics counters on Virtual Chassis ports (VCPs).

Options

none—Clear traffic statistics for VCPs of all members of a Virtual Chassis or VCF.

all-members—(Optional) Clear traffic statistics for VCPs of all members of a Virtual Chassis or VCF.

interface-name—(Optional) Clear traffic statistics for the specified VCP.

local—(Optional) Clear traffic statistics for VCPs from the switch or external Routing Engine on which this command is entered.

member member-id—(Optional) Clear traffic statistics for VCPs from the specified member of a Virtual Chassis or VCF.

Required Privilege Level

clear

RELATED DOCUMENTATION

[show virtual-chassis vc-port statistics | 256](#)

[show virtual-chassis vc-port | 231](#)

Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis

List of Sample Output

[clear virtual-chassis vc-port statistics \(EX4200 Virtual Chassis\) on page 144](#)

[clear virtual-chassis vc-port statistics \(EX8200 Virtual Chassis\) on page 144](#)

[clear virtual-chassis vc-port statistics member 3 on page 144](#)

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

Syntax

```
request session member member-id
```

Release Information

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

Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series.

Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).

Description

Start a session with the specified member of a Virtual Chassis or a VCF.

Options

member-id—Member ID for the specific member of the Virtual Chassis or VCF.

Required Privilege Level

maintenance

RELATED DOCUMENTATION

[member](#) | 119

Understanding Virtual Chassis Components

request system software nonstop-upgrade

Syntax

```
request system software nonstop-upgrade (package-name | set [package-name package-name])
<force-host>
<no-copy>
<no-old-master-upgrade>
<reboot >
<unlink>
```

Release Information

Command introduced in Junos OS Release 10.4 for EX Series switches.

Option **set [*package-name package-name*]** added in Junos OS Release 12.1 for EX Series switches.

Command introduced in Junos OS Release 13.2X50-D20 for the QFX Series.

Command introduced in Junos OS Release 15.1X53-D55 for EX3400 switches.

Description

Perform a nonstop software upgrade (NSSU) on a switch with redundant Routing Engines or on a Virtual Chassis or Virtual Chassis Fabric (VCF). The behavior of this command depends on the type of switch, Virtual Chassis, or VCF where you run it, as follows:

- When you run this command on any of the following Virtual Chassis or VCF configurations, NSSU upgrades all members of the Virtual Chassis:
 - EX3300, EX3400, EX4200, EX4300, EX4500, EX4550, EX4600. or EX4650-48Y Virtual Chassis
 - Mixed Virtual Chassis composed of any combination of EX4200, EX4500, and EX4550 switches, or EX4300 and EX4600 switches
 - QFX3500 and QFX3600 Virtual Chassis
 - QFX5100 Virtual Chassis
 - QFX5120-48Y or QFX5120-48T Virtual Chassis
 - Fixed configuration of switches in a VCF (QFX3500/QFX3600 and QFX5100 switches)
 - Mixed VCF composed of any combination of QFX3500/QFX3600, QFX5100, and EX4300 switches

The original Virtual Chassis or VCF backup becomes the master. The new master automatically upgrades and reboots the original master, which then rejoins the Virtual Chassis or VCF as the backup.

- When you run this command on an EX6200 or EX8200 switch, NSSU upgrades both the backup and master Routing Engines. The original backup Routing Engine becomes the new master at the end of the upgrade.
 - On an EX6200 switch, NSSU automatically reboots the original master Routing Engine.

- On an EX8200 switch, NSSU does not automatically reboot the original master Routing Engine unless you specify the **reboot** option.
- When you run this command on an EX8200 Virtual Chassis, NSSU upgrades all master and backup Routing Engines in the Virtual Chassis, including the external Routing Engines. The original backup Routing Engines become the new master Routing Engines. NSSU does not automatically reboot the original master Routing Engines unless you specify the **reboot** option.

This command has the following requirements:

- All Virtual Chassis members, VCF members, and all Routing Engines must be running the same Junos OS release.
- You must enable Graceful Routing Engine switchover (GRES)..
- You must enable Nonstop active routing (NSR).

NOTE: Although not required, we recommend you also enable nonstop bridging (NSB). NSB ensures that all NSB-supported Layer 2 protocols operate seamlessly during the Routing Engine switchover during NSSU. See *Configuring Nonstop Bridging on EX Series Switches (CLI Procedure)*.

- You must run the command from the master Routing Engine on a standalone switch or from the master on a Virtual Chassis.
- For minimal traffic disruption, you must define link aggregation groups (LAGs) such that the member links reside on different Virtual Chassis or VCF member switches (or on different line cards for EX6200 and EX8200 switches and EX8200 Virtual Chassis).
- For all Virtual Chassis (except EX8200 Virtual Chassis):
 - The Virtual Chassis members must be connected in a ring topology. A ring topology prevents the Virtual Chassis from splitting during an NSSU.
 - The Virtual Chassis master and backup must be adjacent to each other in the ring topology. With adjacent placement, the master and backup are always in sync while the switches in line-card roles are rebooting.
 - The Virtual Chassis must be preprovisioned so the line-card role is explicitly assigned to member switches acting in a line-card role. During an NSSU, the master and backup member switches must maintain their Routing Engine roles (although mastership switches to the backup), and the remaining switches must maintain their line-card roles.
 - In a two-member Virtual Chassis, you must configure **no-split-detection** so the Virtual Chassis doesn't split during NSSU.
- For Virtual Chassis Fabric:

- You can only have two members preprovisioned in the Routing Engine role. If more than two Routing Engines are configured, NSSU issues a warning message and the NSSU process stops.
- The VCF members should be connected in a spine and leaf topology. A spine and leaf topology prevents the VCF from splitting during NSSU. Each leaf device must be connected to both spine devices.
- The VCF must be preprovisioned so that the line-card role has been explicitly assigned to member switches acting in a line-card role, and likewise the Routing Engine role has been explicitly assigned to the member switches acting in a Routing Engine role. During an NSSU, the master and backup member switches must maintain their Routing Engine roles (although mastership switches to the backup), and the remaining switches must maintain their line-card roles.
- You must configure **no-split-detection** in a two-member VCF so the VCF does not split during NSSU.

Options

package-name—Location of the software package or bundle to be installed. For example:

- **/var/tmp/package-name**—For a software package or bundle installed from a local directory on the switch.
- **protocol://hostname/pathname/package-name**—For a software package or bundle downloaded and installed from a remote location. Replace **protocol** with one of the following:
 - **ftp**—File Transfer Protocol.
Use **ftp://hostname/pathname/package-name**.
To specify authentication credentials, use
ftp://<username>:<password>@hostname/pathname/package-name.
To have the system prompt you for the password, specify **prompt** in place of the password.
The command displays an error message if a password is required and you do not specify the password or **prompt**.
 - **http**—Hypertext Transfer Protocol.
Use **http://hostname/pathname/package-name**.
To specify authentication credentials, use
http://<username>:<password>@hostname/pathname/package-name.
The command prompts you for a password if one is required and you didn't include it.
 - **scp**—Secure copy (available only for Canada and U.S. version).
Use **scp://hostname/pathname/package-name**.
To specify authentication credentials, use
scp://<username>:<password>@hostname/pathname/package-name.

NOTE: The **pathname** in the protocol is the relative path to the user home directory on the remote system and not the root directory.

set [*package-name package-name*]*—*(Mixed Virtual Chassis only) Locations of the different installation packages required by the different types of member switches. These packages must be for the same Junos OS release. See this command's **package-name** option for information about how to specify the installation packages.

force-host*—*(Optional) Force adding the host software package or bundle (and ignore warnings) on EX4650, QFX5100, or QFX5120 devices.

no-copy*—*(Optional) Install a software package or bundle, but do not save copies of the package or bundle files.

no-old-master-upgrade*—*(Optional) (EX8200 switches only) Upgrade the backup Routing Engine only. After the upgrade completes, the original master Routing Engine becomes the backup Routing Engine and continues running the previous software version.

reboot*—*(Optional) (EX8200 switches and EX8200 Virtual Chassis only) When you include the **reboot** option, NSSU automatically reboots the original master (new backup) Routing Engine after being upgraded to the new software. When you omit the **reboot** option, you must manually reboot the original master (new backup) Routing Engine using the **request system reboot** command.

NOTE: If you do not use the **reboot** option on an EX8200 Virtual Chassis, you must establish a connection to the console port on the Switch Fabric and Routing Engine (SRE) module or Routing Engine (RE) module to manually reboot the backup Routing Engines.

unlink*—*(Optional) Remove the software package after a successful upgrade.

Required Privilege Level

maintenance

RELATED DOCUMENTATION

[show chassis nonstop-upgrade | 166](#)

Upgrading Software on an EX6200 or EX8200 Standalone Switch Using Nonstop Software Upgrade (CLI Procedure)

Upgrading Software on an EX8200 Virtual Chassis Using Nonstop Software Upgrade (CLI Procedure)

Upgrading Software on a Virtual Chassis and Mixed Virtual Chassis Using Nonstop Software Upgrade

[Upgrading Software on a Virtual Chassis Fabric Using Nonstop Software Upgrade | 74](#)

List of Sample Output

[request system software nonstop-upgrade \(EX4200 Virtual Chassis\) on page 150](#)

[request system software nonstop-upgrade \(EX6200 Switch\) on page 152](#)

[request system software nonstop-upgrade reboot \(EX8200 Switch\) on page 153](#)

[request system software nonstop-upgrade no-old-master-upgrade \(EX8200 Switch\) on page 154](#)

[request system software nonstop-upgrade reboot \(EX8200 Virtual Chassis\) on page 155](#)

Output Fields

This command reports feedback on the status of the request. Some functions are shared between NSSU and the in-service software upgrade (ISSU) feature, so you might see what appear to be ISSU messages as well as NSSU messages in the output from this command.

Sample Output

request system software nonstop-upgrade (EX4200 Virtual Chassis)

```
user@switch> request system software nonstop-upgrade
/var/tmp/jinstall-ex-4200-12.1R5.5-domestic-signed.tgz
Chassis ISSU Check Done
ISSU: Validating Image
ISSU: Preparing Backup RE
Installing image on other FPC's along with the backup

Checking pending install on fpc1
Pushing bundle to fpc1
WARNING: A reboot is required to install the software
WARNING:      Use the 'request system reboot' command immediately
Completed install on fpc1

Checking pending install on fpc2
Pushing bundle to fpc2
WARNING: A reboot is required to install the software
WARNING:      Use the 'request system reboot' command immediately
Completed install on fpc2

Checking pending install on fpc3
Pushing bundle to fpc3
WARNING: A reboot is required to install the software
WARNING:      Use the 'request system reboot' command immediately
Completed install on fpc3

Checking pending install on fpc4
Pushing bundle to fpc4
WARNING: A reboot is required to install the software
WARNING:      Use the 'request system reboot' command immediately
```

Completed install on fpc4

Checking pending install on fpc5

Pushing bundle to fpc5

WARNING: A reboot is required to install the software

WARNING: Use the 'request system reboot' command immediately

Completed install on fpc5

Checking pending install on fpc6

Pushing bundle to fpc6

WARNING: A reboot is required to install the software

WARNING: Use the 'request system reboot' command immediately

Completed install on fpc6

Checking pending install on fpc7

Pushing bundle to fpc7

WARNING: A reboot is required to install the software

WARNING: Use the 'request system reboot' command immediately

Completed install on fpc7

Backup upgrade done

Rebooting Backup RE

Rebooting fpc1

ISSU: Backup RE Prepare Done

Waiting for Backup RE reboot

GRES operational

Initiating Chassis In-Service-Upgrade

Chassis ISSU Started

ISSU: Preparing Daemons

ISSU: Daemons Ready for ISSU

ISSU: Starting Upgrade for FRUs

ISSU: Preparing for Switchover

ISSU: Ready for Switchover

Checking In-Service-Upgrade status

Item	Status	Reason
FPC 0	Online	
FPC 1	Online	
FPC 2	Online (ISSU)	
FPC 3	Online (ISSU)	
FPC 4	Online (ISSU)	
FPC 5	Online (ISSU)	
FPC 6	Online (ISSU)	
FPC 7	Online (ISSU)	

Going to install image on master

```

WARNING: A reboot is required to install the software
WARNING:      Use the 'request system reboot' command immediately
relinquish mastership
ISSU: IDLE

*** FINAL System shutdown message from root@switch ***

System going down IMMEDIATELY

Shutdown NOW!
[pid 9336]

```

request system software nonstop-upgrade (EX6200 Switch)

```

{master}
user@switch> request system software nonstop-upgrade
/var/tmp/jinstall-ex-6200-12.2R5.5-domestic-signed.tgz
Chassis ISSU Check Done
ISSU: Validating Image
ISSU: Preparing Backup RE
Pushing bundle to re0
NOTICE: Validating configuration against
jinstall-ex-6200-12.2R5.5-domestic-signed.tgz.
NOTICE: Use the 'no-validate' option to skip this if desired.
WARNING: A reboot is required to install the software
WARNING:      Use the 'request system reboot' command immediately
Backup upgrade done
Rebooting Backup RE

Rebooting re0
ISSU: Backup RE Prepare Done
Waiting for Backup RE reboot
GRES operational
Initiating Chassis In-Service-Upgrade
Chassis ISSU Started
ISSU: Preparing Daemons
ISSU: Daemons Ready for ISSU
ISSU: Starting Upgrade for FRUs
ISSU: Preparing for Switchover
ISSU: Ready for Switchover
Checking In-Service-Upgrade status

```

Item	Status	Reason
FPC 0	Online (ISSU)	
FPC 1	Online (ISSU)	
FPC 2	Online (ISSU)	
FPC 3	Online (ISSU)	
FPC 4	Online	
FPC 5	Online	
FPC 6	Online (ISSU)	
FPC 7	Online (ISSU)	
FPC 8	Online (ISSU)	
FPC 9	Online (ISSU)	

Going to install image on master

NOTICE: Validating configuration against
jinstall-ex-6200-12.2R5.5-domestic-signed.tgz.

NOTICE: Use the 'no-validate' option to skip this if desired.

WARNING: A reboot is required to install the software

WARNING: Use the 'request system reboot' command immediately
relinquish mastership

ISSU: IDLE

Trying to relinquish mastership before rebooting...

Resolving mastership...

Complete. The other routing engine becomes the master.

*** FINAL System shutdown message from user@switch ***

System going down IMMEDIATELY

request system software nonstop-upgrade reboot (EX8200 Switch)

```
{master}
user@switch> request system software nonstop-upgrade reboot
/var/tmp/jinstall-ex-8200-10.4R1.5-domestic-signed.tgz
Chassis ISSU Check Done
ISSU: Validating Image
ISSU: Preparing Backup RE
Pushing bundle to rel
WARNING: A reboot is required to install the software
WARNING: Use the 'request system reboot' command immediately
Backup upgrade done
Rebooting Backup RE

Rebooting rel
ISSU: Backup RE Prepare Done
```

```

Waiting for Backup RE reboot
GRES operational
Initiating Chassis In-Service-Upgrade
Chassis ISSU Started
ISSU: Preparing Daemons
ISSU: Daemons Ready for ISSU
ISSU: Starting Upgrade for FRUs
ISSU: Preparing for Switchover
ISSU: Ready for Switchover
Checking In-Service-Upgrade status
  Item           Status           Reason
  FPC 0           Online (ISSU)
  FPC 2           Offline           Offlined by CLI command
  FPC 3           Online (ISSU)
Resolving mastership...
Complete. The other routing engine becomes the master.
ISSU: RE switchover Done
ISSU: Upgrading Old Master RE
WARNING: A reboot is required to install the software
WARNING:      Use the 'request system reboot' command immediately
ISSU: Old Master Upgrade Done
ISSU: IDLE
Shutdown NOW!
[pid 2635]

*** FINAL System shutdown message from user@switch ***
System going down IMMEDIATELY

```

request system software nonstop-upgrade no-old-master-upgrade (EX8200 Switch)

```

{master}
user@switch> request system software nonstop-upgrade no-old-master-upgrade
/var/tmp/jinstall-ex-8200-10.4R1.5-domestic-signed.tgz
Chassis ISSU Check Done
ISSU: Validating Image
ISSU: Preparing Backup RE
Pushing bundle to rel
WARNING: A reboot is required to install the software
WARNING:      Use the 'request system reboot' command immediately
Backup upgrade done
Rebooting Backup RE

```



```

Rebooting rel
ISSU: Backup RE Prepare Done
Waiting for Backup RE reboot
GRES operational
Initiating Chassis In-Service-Upgrade
Chassis ISSU Started
ISSU: Preparing Daemons
ISSU: Daemons Ready for ISSU
ISSU: Starting Upgrade for FRUs
ISSU: Preparing for Switchover
ISSU: Ready for Switchover
Checking In-Service-Upgrade status

```

Item	Status	Reason
FPC 0	Online (ISSU)	
FPC 1	Online (ISSU)	
FPC 2	Online (ISSU)	
FPC 3	Offline	Offlined by CLI command
FPC 4	Online (ISSU)	
FPC 5	Online (ISSU)	
FPC 6	Online (ISSU)	
FPC 7	Online (ISSU)	

```

Resolving mastership...
Complete. The other routing engine becomes the master.
ISSU: RE switchover Done
Skipping Old Master Upgrade
ISSU: IDLE

```

request system software nonstop-upgrade reboot (EX8200 Virtual Chassis)

```

{master:9}
user@external-routing-engine> request system software nonstop-upgrade reboot
/var/tmp/jinstall-ex-xre200-11.1-20101130.0-domestic-signed.tgz
Chassis ISSU Check Done
ISSU: Validating Image
ISSU: Preparing LCC Backup REs
ISSU: Preparing Backup RE
Pushing bundle /var/tmp/jinstall-ex-xre200-11.1-20101130.0-domestic-signed.tgz to
member8
-----
WARNING: A reboot is required to install the software
WARNING:      Use the 'request system reboot' command immediately
VC Backup upgrade done
Rebooting VC Backup RE

```

```

Rebooting member8
ISSU: Backup RE Prepare Done
Waiting for VC Backup RE reboot
Pushing bundle to member0-backup
Pushing bundle to member1-backup
WARNING: A reboot is required to install the software
WARNING:      Use the 'request system reboot' command immediately
WARNING: A reboot is required to install the software
WARNING:      Use the 'request system reboot' command immediately

```

```

Rebooting member0-backup
Rebooting LCC [member0-backup]

```

```

Rebooting member1-backup
Rebooting LCC [member1-backup]
ISSU: LCC Backup REs Prepare Done
GRES operational
Initiating Chassis Nonstop-Software-Upgrade
Chassis ISSU Started
ISSU: Preparing Daemons
ISSU: Daemons Ready for ISSU
ISSU: Starting Upgrade for FRUs
ISSU: Preparing for Switchover
ISSU: Ready for Switchover
Checking Nonstop-Upgrade status
member0:

```

Item	Status	Reason
FPC 0	Online (ISSU)	
FPC 1	Online (ISSU)	
FPC 2	Online (ISSU)	
FPC 5	Online (ISSU)	

```
member1:
```

Item	Status	Reason
FPC 0	Online (ISSU)	
FPC 1	Offline	Offlined due to config
FPC 2	Online (ISSU)	
FPC 3	Online (ISSU)	
FPC 4	Online (ISSU)	
FPC 5	Online (ISSU)	
FPC 7	Online (ISSU)	

member0:

Item	Status	Reason
FPC 0	Online (ISSU)	
FPC 1	Online (ISSU)	
FPC 2	Online (ISSU)	
FPC 5	Online (ISSU)	

member1:

Item	Status	Reason
FPC 0	Online (ISSU)	
FPC 1	Offline	Offlined due to config
FPC 2	Online (ISSU)	
FPC 3	Online (ISSU)	
FPC 4	Online (ISSU)	
FPC 5	Online (ISSU)	
FPC 7	Online (ISSU)	

ISSU: Upgrading Old Master RE

Pushing bundle /var/tmp/incoming-package-8200.tgz to member0-master

Pushing bundle /var/tmp/incoming-package-8200.tgz to member1-master

ISSU: RE switchover Done

WARNING: A reboot is required to install the software

WARNING: Use the 'request system reboot' command immediately

Rebooting ...

shutdown: [pid 2188]

Shutdown NOW!

ISSU: Old Master Upgrade Done

ISSU: IDLE

Shutdown NOW!

*** FINAL System shutdown message from root@ ***

System going down IMMEDIATELY

request virtual-chassis mode

Syntax

```
request virtual-chassis mode
  fabric
  mixed <ieee-clause-82>
  <disable>
  <reboot>
  <all-members> | <local> | <member member-id>
```

Release Information

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.

fabric keyword introduced in Junos OS Release 13.2X51-D20 for EX Series switches and QFX Series devices in a Virtual Chassis Fabric (VCF).

Command introduced in Junos OS Release 13.2X51-D20 for VCF.

ieee-clause-82 mixed-mode option introduced for EX4300 switches mixed with EX4300 multigigabit switches in a Virtual Chassis in Junos OS Release 18.2R1.

Description

Enable devices to operate as a Virtual Chassis Fabric (VCF) by setting fabric mode, or to interoperate in a mixed Virtual Chassis or VCF by setting mixed mode. You must reboot the standalone device, Virtual Chassis or VCF for this command to take effect, and the setting persists across subsequent reboot operations. You can use the **reboot** option to automatically reboot the device when setting fabric or mixed modes.

NOTE: For devices in a mixed VCF, you can set both mixed and fabric modes in the same command and then reboot the device once for both settings to take effect.

You must set **fabric** mode on all of the devices in a VCF.

You must set **mixed** mode on all devices in a Virtual Chassis or a VCF when the devices have differences in how they interoperate. See *Understanding Mixed EX Series and QFX Series Virtual Chassis* or [“Understanding Mixed Virtual Chassis Fabric” on page 20](#) for details on which devices can be combined to form a mixed Virtual Chassis or VCF.

NOTE: You do not need to configure mixed mode if the only devices in your Virtual Chassis are the following combinations of switches. These form a non-mixed Virtual Chassis or VCF because the devices can run the same software images:

- Only EX4500 and EX4550 switches in an EX Series Virtual Chassis.
- EX2300 multigigabit switches (EX2300-24MP and EX2300-48MP models) with any other EX2300 model switches in an EX2300 Virtual Chassis.
- Only QFX3500 and QFX3600 switches in a QFX Series Virtual Chassis.
- QFX5110 and QFX5100 switches in a QFX5110 Virtual Chassis or QFX5110 VCF.

Requirements when setting **mixed** mode include:

- EX4300 switches that are not multigigabit models combined with EX4300 multigigabit model (EX4300-48MP) switches form a mixed EX4300 Virtual Chassis. In this case, when you set mixed mode on the EX4300 non-multigigabit model member switches, you must also include the **ieee-clause-82** option on those switches. This option sets a special port mode (IEEE Clause 82) on the Virtual Chassis ports (VCPs) that enables them to communicate when interconnected with VCPs on EX4300 multigigabit switches.

NOTE: If you remove an EX4300 member switch from a mixed EX4300 Virtual Chassis with multigigabit model members, you must disable **ieee-clause-82** port mode on the switch if you want to reconfigure it as a standalone switch or use it in any other type of mixed Virtual Chassis or any non-mixed Virtual Chassis. Otherwise, the VCPs will not connect with other members in the new Virtual Chassis. To do this, use the **request virtual-chassis mode mixed ieee-clause-82 disable** command and then reboot the switch. To add the switch to any other type of mixed Virtual Chassis, enable mixed mode again without this option and reboot the switch again for the change to take effect.

- Don't enable mixed mode on a standalone device or for a member switch that is in a non-mixed Virtual Chassis or VCF. Mixed mode reduces the maximum scaling numbers for some features on the standalone switch, Virtual Chassis, or VCF to accommodate different types of devices that can be interconnected.

Best practices when setting **mixed** or **fabric** mode include:

- Before cabling a device into a mixed Virtual Chassis or a VCF, to avoid potential traffic disruptions and configuration issues, on each device we recommend setting mixed mode for a mixed Virtual Chassis or VCF and setting fabric mode for a VCF. You must then reboot the device for the change or changes to take effect.

- You can also change mode settings manually after a device has been added to a Virtual Chassis or VCF if you determine the settings are not the same on all the devices.

However, if you set only some of the devices in a mixed Virtual Chassis or VCF to mixed mode, the mixed Virtual Chassis or VCF might not form. If you experience this issue, enter the **request virtual-chassis mode mixed all-members** command to force setting mixed mode on all devices in the Virtual Chassis or VCF. You can then reboot the entire Virtual Chassis or VCF, or only the devices where the mode changed. The Virtual Chassis or VCF forms when the rebooted devices are online again.

NOTE: On switches running Junos OS Release 11.4 or later, if you don't set any of the switches in a mixed EX4200 and EX4500 Virtual Chassis to mixed mode, a mixed EX4200 and EX4500 Virtual Chassis forms where one of the switches assumes the master role, and all other switches take on the linecard role. If you see this behavior, enter the **request virtual-chassis mode mixed all-members** command to set all members to mixed mode and reboot the Virtual Chassis.

Options

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.

The **request virtual-chassis mode mixed disable** command does not also automatically disable the mixed mode **ieee-clause-82** port mode option. To disable **ieee-clause-82** port mode on the device, you must include that option in the disable command as follows:

request virtual-chassis mode mixed ieee-clause-82 disable

fabric—Set the device into fabric mode so it can participate as a member device in a VCF.

local—(Optional) Set the Virtual Chassis mode only on the member device where the command is issued.

member *member-id*—(Optional) Set the Virtual Chassis mode on the specified member of the Virtual Chassis or VCF.

mixed <ieee-clause-82> —Set the device into mixed mode so that the device can participate in a mixed Virtual Chassis or mixed VCF.

You must set mixed mode with the **ieee-clause-82** option on EX4300 switches that are not multigigabit models when mixing them with EX4300 multigigabit (EX4300-48MP) model switches in an EX4300 Virtual Chassis.

reboot—After applying the mode change specified by the other options, reboot the device automatically.

Required Privilege Level

system-control

RELATED DOCUMENTATION

Understanding Mixed EX Series and QFX Series Virtual Chassis

[Understanding Mixed Virtual Chassis Fabric | 20](#)

Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches (CLI Procedure)

Configuring an EX2300, EX3400, or EX4300 Virtual Chassis

Configuring EX4600 Switches in a Mixed or Non-Mixed Virtual Chassis

Configuring an EX4650 or a QFX Series Virtual Chassis

Adding a New Switch to an Existing EX2300, EX3400, or EX4300 Virtual Chassis

Removing or Replacing a Member Switch of a Virtual Chassis Configuration

[Verifying the Virtual Chassis Fabric Mode Settings | 87](#)

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 161](#)

[request virtual-chassis mode fabric mixed reboot on page 161](#)

[request virtual-chassis mode mixed \(with IEEE Clause 82 mode for EX4300 switches mixed with EX4300 multigigabit switches in a Virtual Chassis\) on page 161](#)

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
```

Sample Output

```
request virtual-chassis mode mixed (with IEEE Clause 82 mode for EX4300 switches mixed with EX4300 multigigabit switches in a Virtual Chassis)
```

```
user@switch> request virtual-chassis mode ieee-clause-82 mixed
```

request virtual-chassis reactivate

Syntax

```
request virtual-chassis reactivate
```

Release Information

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).

Description

Reactivate a device that has been assigned a member ID but is not currently connected to the Virtual Chassis or VCF.

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 **Prsnt**.

Required Privilege Level

system-control

RELATED DOCUMENTATION

Verifying the Member ID, Role, and Neighbor Member Connections of a Virtual Chassis Member

Verifying the Member ID, Role, and Neighbor Member Connections of an EX8200 Virtual Chassis Member

List of Sample Output

[request virtual-chassis reactivate on page 162](#)

Sample Output

```
request virtual-chassis reactivate
```

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


request virtual-chassis vc-port

Syntax

```
request virtual-chassis vc-port (set | delete)
<fpc-slot fpc-slot>
pic-slot pic-slot
port port-number
<member member-id | local>
```

Release Information

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

Option **fpc-slot** 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).

Description

Set a port to operate as a Virtual Chassis port (VCP), or delete the VCP setting on a port. See *Virtual Chassis Port Options* for details on which ports you can set as VCPs on different switches. After setting a port as a VCP, you can't use the port for any other purpose unless you remove the VCP setting.

If you don't include the **member member-id** option, this command defaults to setting the port as a VCP or deleting the VCP setting on the switch where you run the command. You can alternatively set the **local** option to ensure the command applies to the specified port on the local switch where you run it.

You might experience a temporary traffic disruption immediately after creating or deleting a user-configured VCP in an EX8200 Virtual Chassis.

Some switches have ports that are dedicated VCPs (you can only use them as VCPs and for no other purpose) or that are configured as VCPs by default (set as VCPs in the default factory configuration). You do not need to explicitly set those ports as VCPs to use them to interconnect the switch into a Virtual Chassis. See *Virtual Chassis Port Options* for details.

If you don't need a default-configured VCP to interconnect Virtual Chassis member switches, you can run this command with the **delete** option to remove the VCP setting and use the port as a regular uplink or network port. If you want to use the port as a VCP again later, then you need to run this command with the **set** option to reapply the VCP setting.

Options

set—Set a port as a VCP to convert an uplink or network port into a VCP.

delete—Delete the VCP setting on a port to convert a VCP back into an uplink or network port.

pic-slot pic-slot—Number of the PIC slot for the port on the switch.

port port-number—Number of the port that is to be enabled or disabled as a VCP.

member *member-id*—(Optional) Enable or disable the specified VCP on the specified member of the Virtual Chassis or VCF.

local—(Optional) Enable or disable the specified VCP on the local switch where you run the command.

Required Privilege Level

system-control

RELATED DOCUMENTATION

request virtual-chassis vc-port (Dedicated VCP)

[show virtual-chassis vc-port | 231](#)

[show virtual-chassis vc-port statistics | 256](#)

[clear virtual-chassis vc-port statistics | 143](#)

Virtual Chassis Port (VCP) Interface Names in an EX8200 Virtual Chassis

Understanding Virtual Chassis Components

List of Sample Output

[request virtual-chassis vc-port set pic-slot 1 port 0 on page 164](#)

[request virtual-chassis vc-port set pic-slot 1 port 1 member 3 on page 164](#)

[request virtual-chassis vc-port delete pic-slot 1 port 1 member 3 on page 164](#)

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

Syntax

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

Release Information

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).

Description

Run a digital optical monitoring (DOM) scan on the optical ports configured as Virtual Chassis ports (VCPs).

Enter the **show virtual-chassis vc-port diagnostics optics** command to view the results of the diagnostic scan.

On certain EX Series switches, you must enter the **request virtual-chassis vc-port diagnostics optics** command to run a diagnostic scan before you can gather the **show virtual-chassis vc-port diagnostics optics** output.

Required Privilege Level

system-control

RELATED DOCUMENTATION

[show virtual-chassis vc-port diagnostics optics](#) | 237

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 chassis nonstop-upgrade

Syntax

```
show chassis nonstop-upgrade
```

Release Information

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.

Description

(EX6200 switches, EX8200 switches, EX8200 Virtual Chassis, QFX3500 and QFX3600 Virtual Chassis, and Virtual Chassis Fabric only) Display the status of the line cards or Virtual Chassis members in the linecard role after the most recent nonstop software upgrade (NSSU). This command must be issued on the master Routing Engine.

Required Privilege Level

view

RELATED DOCUMENTATION

request system software nonstop-upgrade 146
<i>Upgrading Software on an EX6200 or EX8200 Standalone Switch Using Nonstop Software Upgrade (CLI Procedure)</i>
<i>Upgrading Software on a Virtual Chassis and Mixed Virtual Chassis Using Nonstop Software Upgrade</i>
Upgrading Software on a Virtual Chassis Fabric Using Nonstop Software Upgrade 74
<i>Upgrading Software on an EX8200 Virtual Chassis Using Nonstop Software Upgrade (CLI Procedure)</i>

List of Sample Output

- [show chassis nonstop-upgrade \(EX8200 Switch\) on page 167](#)
- [show chassis nonstop-upgrade \(EX8200 Virtual Chassis\) on page 167](#)
- [show chassis nonstop-upgrade \(Virtual Chassis Fabric\) on page 168](#)

Output Fields

[Table 4 on page 167](#) lists the output fields for the **show chassis nonstop-upgrade** command. Output fields are listed in the approximate order in which they appear.

Table 4: show chassis nonstop-upgrade Output Fields

Field Name	Field Description
Item	Line card slot number.
Status	State of line card: <ul style="list-style-type: none"> • Error—Line card is in an error state. • Offline—Line card is powered down. • Online—Line card is online and running.
Reason	Reason for the state (if the line card is offline).

Sample Output

show chassis nonstop-upgrade (EX8200 Switch)

```
user@switch> show chassis nonstop-upgrade
```

Item	Status	Reason
FPC 0	Online	
FPC 1	Online	
FPC 2	Online	
FPC 3	Offline	Offlined by CLI command
FPC 4	Online	
FPC 5	Online	
FPC 6	Online	
FPC 7	Online	

show chassis nonstop-upgrade (EX8200 Virtual Chassis)

```
user@external-routing-engine> show chassis nonstop-upgrade
```

```
member0:
```

Item	Status	Reason
FPC 0	Online	
FPC 1	Online	
FPC 2	Online	
FPC 5	Online	

```
member1:
```

Item	Status	Reason
FPC 0	Online	
FPC 1	Offline	Offlined due to config
FPC 2	Online	
FPC 3	Online	
FPC 4	Online	
FPC 5	Online	
FPC 7	Online	

show chassis nonstop-upgrade (Virtual Chassis Fabric)

Item	Status	Reason
FPC 0	Online	
FPC 1	Online	
FPC 2	Online	
FPC 3	Online	
FPC 4	Online	
FPC 5	Online	

show forwarding-options enhanced-hash-key

Syntax

```
show forwarding-options enhanced-hash-key
```

Release Information

Command introduced in Junos OS Release 13.2X51-D15 for EX Series switches.

Command introduced in Junos OS Release 13.2X51-D20 for QFX Series devices.

Fabric Load Balancing Options output fields introduced in Junos OS Release 14.1X53-D10.

Incoming port output field introduced in Junos OS Release 18.4R1 for QFX10000 Series switches.

Command introduced in Junos OS Release 19.4R1 for QFX5120-32C and QFX5120-48Y switches.

The **ecmp-dlb** statement introduced in Junos OS evolved Release 19.4R2 for QFX5220 switches.

Description

Display information about which packet fields are used by the hashing algorithm to make hashing decisions.

You can configure the fields that are inspected by the hashing algorithm to make hashing decisions for traffic entering a LAG bundle using the **forwarding-options enhanced-hash-key** statement.

Required Privilege Level

view

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

List of Sample Output

[show forwarding-options enhanced-hash-key \(Layer 2 Payload Hash Mode\) on page 172](#)

[show forwarding-options enhanced-hash-key \(Layer 2 Header Hash Mode\) on page 173](#)

[show forwarding-options enhanced-hash-key \(Fabric Load Balancing Options\) on page 173](#)

[show forwarding-options enhanced-hash-key \(Dynamic Load Balancing Options\) on page 173](#)

[show forwarding-options enhanced-hash-key \(QFX10000 Series Switches\) on page 174](#)

Output Fields

[Table 5 on page 170](#) lists the output fields for the **show forwarding-options enhanced-hash-key** command.

Output fields are listed in the approximate order in which they first appear. Output fields vary by platform.

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

Field Name	Field Description
Hash-Mode	Current hash mode: Layer 2 header or Layer 2 payload.
Protocol	Indicates whether the Protocol field is or is not used by the hashing algorithm: Yes or No.
Destination L4 Port	Indicates whether the Destination L4 Port field is or is not used by the hashing algorithm: Yes or No.
Source L4 Port	Indicates whether the Source L4 Port field is or is not used by the hashing algorithm: Yes or No.
Destination IPv4 Addr	Indicates whether the Destination IPv4 Addr field is or is not used by the hashing algorithm: Yes or No.
Source IPv4 Addr	Indicates whether the Source IPv4 Addr field is or is not used by the hashing algorithm: Yes or No.
Incoming port	Indicates whether the incoming port number (interface) is or is not used by the hashing algorithm. Yes or No NOTE: When passive monitoring is enabled on a QFX10000 Series switch interface, the inet , inet6 and L2 fields are all set to No.
Vlan id	Indicates whether the Vlan ID field is or is not used by the hashing algorithm: Yes or No.
Inner-Vlan ID	Indicates whether the inner Vlan field is or is not used by the hashing algorithm: Yes or No.
Next Hdr	Indicates whether the Next Hdr field is or is not used by the hashing algorithm: Yes or No.
Destination IPv6 Addr	Indicates whether the Destination IPv6 Addr field is or is not used by the hashing algorithm: Yes or No.
Source IPv6 Addr	Indicates whether the Source IPv6 Addr field is or is not used by the hashing algorithm: Yes or No.
Ether Type	Indicates whether the Ether Type 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
Destination MAC Address	Indicates whether the Destination MAC Address field is or is not used by the hashing algorithm: Yes or No.
Source MAC Address	Indicates whether the Source MAC Address field is or is not used by the hashing algorithm: Yes or No.
Load Balancing Method for ALB	<p>Indicates the load balancing method for adaptive load balancing (ALB): flowlet or per-packet.</p> <p>The load balancing method is flowlet by default, and can be configured using the fabric-load-balance statement.</p>
Load Balancing Method for DLB (QFX5120-32C, QFX5120-48Y, and QFX5220 switches)	<p>Indicates various Dynamic Load Balancing (DLB) modes:</p> <ul style="list-style-type: none"> • Flowlet • Assigned flow • Per-packet <p>Refer <i>Dynamic Load Balancing</i> for more details.</p>
Fabric Link Scale	Indicates the fabric link scale, in mbps.
Inactivity Interval	<p>Indicates the fabric load balance inactivity interval, in microseconds (us).</p> <p>The inactivity interval is 16 microseconds by default, and can be configured using the inactivity-interval statement.</p>
Hash Region Size/Trunk	Indicates the hash region size, in buckets per fabric trunk.
Seed	A hash seed value, between 0 and 4294967295. If a hash-seed value is not configured it is automatically assigned on the QFX10000 Series switches. A hash-seed prevents traffic polarization to same links on the next hop QFX switch when two are connected with LAG/ECMP.
Key	Indicates whether the GRE key field is or is not used by the hashing algorithm: Yes or No.
Protocol	Indicates if a Generic Router Encapsulation (GRE) endpoint over routes was dynamically learned by a routing protocol such as RIP or OSPF.
MPLS Enabled	Indicates if MPLS is enabled under L2 switching.

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

Field Name	Field Description
VXLAN VNID	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 (Dynamic Load Balancing Options)

```
user@switch> show forwarding-options enhanced-hash-key ecmp-dlb
```

```
Slot 0
Current RTAG7 Settings
-----

Hash-Mode                               : layer2-payload
inet RTAG7 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 RTAG7 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

```

```

ECMP Load Balancing Options
-----

```

```

  Load Balancing Method   : Flowlet
  Inactivity Interval     : 64 (us)

```

show forwarding-options enhanced-hash-key (QFX10000 Series Switches)

```

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

```

```

Slot 0

```

```

Seed value for Hash function    0: 2301323130
Seed value for Hash function    1: 2301323130
Seed value for Hash function    2: 2301323130
Seed value for Hash function    3: 2301323130

```

```

Inet settings:
-----

```

```

  IPV4 dest address:    Yes
  IPV4 source address:  Yes
  L4 Dest Port:         Yes
  L4 Source Port:       Yes
  Incoming port:        No

```

```

Inet6 settings:
-----

```

```
        IPV6 dest address:    Yes
        IPV6 source address:  Yes
        L4 Dest Port:         Yes
        L4 Source Port:       Yes
        Incoming port:        No
L2 settings:
-----
        Dest Mac address:     No
        Source Mac address:   No
        Vlan Id:              Yes
        Inner-vlan Id:        No
        Incoming port:        No
GRE settings:
-----
        Key:                  No
        Protocol:              No
MPLS settings:
-----
MPLS Enabled:                Yes

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

show virtual-chassis

Syntax

```
show virtual-chassis <status>
```

Release Information

Command introduced in Junos OS Release 9.2 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).

Fabric ID, **Fabric Mode**, and **Route Mode** output fields introduced in Junos OS Release 13.2X51-D20.

Alias-Name output field introduced in Junos OS Release 14.1X53-D10.

Description

Display information about all members of the Virtual Chassis or VCF.

Options

none—Display information about all Virtual Chassis or VCF member devices.

status—Same output as for **show virtual-chassis** without any options.

Required Privilege Level

view

RELATED DOCUMENTATION

[show virtual-chassis active-topology | 185](#)

[show virtual-chassis protocol adjacency | 205](#)

[show virtual-chassis vc-path | 225](#)

Understanding Mixed EX Series and QFX Series Virtual Chassis

[Understanding Mixed Virtual Chassis Fabric | 20](#)

Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis

List of Sample Output

[show virtual-chassis \(EX2300 multigigabit model—EX2300-24MP and EX2300-48MP—Virtual Chassis\) on page 179](#)

[show virtual-chassis \(EX2300 and EX2300 multigigabit model \(EX2300-24MP, EX2300-48MP\) Virtual Chassis\) on page 180](#)

[show virtual-chassis \(EX4200 Virtual Chassis\) on page 180](#)

[show virtual-chassis \(Mixed EX4300 multigigabit model—EX4300-48MP—Virtual Chassis\) on page 181](#)

[show virtual-chassis \(EX8200 Virtual Chassis\) on page 181](#)

[show virtual-chassis \(QFX5110 Virtual Chassis\) on page 182](#)

[show virtual-chassis \(QFX5200 Virtual Chassis\) on page 183](#)

[show virtual-chassis \(QFX5100 Virtual Chassis Fabric\) on page 183](#)

Output Fields

Table 6 on page 177 lists the output fields for the **show virtual-chassis** command. Output fields are listed in the approximate order in which they appear.

Table 6: show virtual-chassis Output Fields

Field Name	Field Description
Fabric ID	(VCF only) Assigned ID used to identify the VCF.
Fabric Mode	(VCF only) Mode of the VCF: Enabled, Disabled, or Mixed.
Preprovisioned Virtual Chassis or Preprovisioned Virtual Chassis Fabric	Virtual Chassis or VCF is configured using preprovisioning.
Virtual Chassis ID	Assigned ID that applies to the entire Virtual Chassis or VCF.
Virtual Chassis Mode	<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"> • Enabled—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). • Disabled—The switch does not support the Virtual Chassis feature. <p>NOTE: Switches that support the Virtual Chassis feature do not display this value. Even if a Virtual Chassis is not currently configured, those switches display Enabled in this field.</p> <ul style="list-style-type: none"> • Mixed—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).

Table 6: show virtual-chassis Output Fields (*continued*)

Field Name	Field Description
Member ID	<p>Assigned member ID and FPC:</p> <ul style="list-style-type: none"> 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. 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.
Status	<p>For a nonprovisioned configuration:</p> <ul style="list-style-type: none"> Prsnt for a member that is currently connected to the Virtual Chassis or VCF configuration. NotPrsnt for a member ID that has been assigned but is not currently connected. <p>For a preprovisioned configuration:</p> <ul style="list-style-type: none"> Prsnt for a member that is specified in the preprovisioned configuration file and is currently connected to the Virtual Chassis or VCF. Unprvsnd for a member that is interconnected with the Virtual Chassis or VCF configuration but is not specified in the preprovisioned configuration file.
Serial No	Serial number of the member device.
Alias-Name	<p>The user-configured alias of the member device.</p> <p>The Alias-Name 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 alias-name statement in the [edit virtual-chassis aliases serial-number serial-number] hierarchy.</p>
Model	Model number of the member device.
Mastership Priority or Mstr prio	Mastership priority value of the member device.
Role	<p>Role of the member device: master, backup, or linecard.</p> <p>An asterisk (*) following the Role denotes the member device on which the show virtual-chassis <status> command was issued.</p>

Table 6: show virtual-chassis Output Fields (*continued*)

Field Name	Field Description
Mixed Mode	<p>Mixed mode configuration status:</p> <ul style="list-style-type: none"> • Y for a member device configured in mixed mode. • N for a member device not configured in mixed mode. • NA for a member device that cannot be configured in mixed mode.
Route Mode	<p>The route mode of the member device:</p> <ul style="list-style-type: none"> • F for fabric (VCF) mode • VC for Virtual Chassis mode
Location	<p>Location of the member device.</p> <p>If this field is empty, the location field was not set for the device.</p>
Neighbor List ID and Interface	Member ID of the neighbor member to which this member's Virtual Chassis port (VCP) is connected, and the VCP interface name.

Sample Output

show virtual-chassis (EX2300 multigigabit model—EX2300-24MP and EX2300-48MP—Virtual Chassis)

user@switch> show virtual-chassis

```

Virtual Chassis ID: cdef.789a.bc12
Virtual Chassis Mode: Enabled

Member ID  Status  Serial No  Model          Mstr    Mixed Route  Neighbor List
prio Role   Mode  Mode  ID  Interface
0 (FPC 0)  Prsnt    ...        ex2300-24mp   128    Backup      N    VC    3  vcp-255/1/0
                                     1  vcp-255/1/2
1 (FPC 1)  Prsnt    ...        ex2300-48mp   128    Master*     N    VC    2  vcp-255/1/4
                                     0  vcp-255/1/1
2 (FPC 2)  Prsnt    ...        ex2300-24mp   128    Linecard    N    VC    3  vcp-255/1/0
                                     1  vcp-255/1/1
3 (FPC 3)  Prsnt    ...        ex2300-48mp   128    Linecard    N    VC    0  vcp-255/1/4
                                     2  vcp-255/1/5

```

show virtual-chassis (EX2300 and EX2300 multigigabit model (EX2300-24MP, EX2300-48MP) Virtual Chassis)

user@switch> show virtual-chassis status

```

Preprovisioned Virtual Chassis
Virtual Chassis ID: 9876.5432.abcd
Virtual Chassis Mode: Enabled

List
Member ID  Status  Serial No  Model          Mstr      Role      Mode Mode  ID  Interface
0 (FPC 0)  Prsnt   ...       ex2300-24mp    129      Backup    N   VC    1  vcp-255/1/3
                                     3  vcp-255/1/0
1 (FPC 1)  Prsnt   ...       ex2300-24p     129      Master*   N   VC    2  vcp-255/1/0
                                     0  vcp-255/1/1
2 (FPC 2)  Prsnt   ...       ex2300-24p      0       Linecard  N   VC    3  vcp-255/1/2
                                     1  vcp-255/1/3
3 (FPC 3)  Prsnt   ...       ex2300-48mp     0       Linecard  N   VC    2  vcp-255/1/5
                                     0  vcp-255/1/0

```

show virtual-chassis (EX4200 Virtual Chassis)

user@switch> show virtual-chassis

```

Virtual Chassis ID: 00ab.cdef.1234
Virtual Chassis Mode: Enabled

Mastership      Mixed  Neighbor List
Member ID  Status  Serial No  Model          priority  Role  Mode  ID  Interface
0 (FPC 0)  Prsnt   ...       ex4200-24t     249      Master*  N    8  vcp-0
                                     1  vcp-1
1 (FPC 1)  Prsnt   ...       ex4200-24t     248      Backup   N    0  vcp-0
                                     2  vcp-1
2 (FPC 2)  Prsnt   ...       ex4200-48p     247      Linecard N    1  vcp-0
                                     3  vcp-1
3 (FPC 3)  Prsnt   ...       ex4200-24t     246      Linecard N    2  vcp-0
                                     4  vcp-1
4 (FPC 4)  Prsnt   ...       ex4200-48p     245      Linecard N    3  vcp-0
                                     5  vcp-1
5 (FPC 5)  Prsnt   ...       ex4200-48t     244      Linecard N    4  vcp-0

```

```

6 (FPC 6)  Prsnt  ...      ex4200-48t  243  Linecard  N  5  vcp-0
6 (FPC 6)  Prsnt  ...      ex4200-48t  243  Linecard  N  7  vcp-1
7 (FPC 7)  Prsnt  ...      ex4200-24f  242  Linecard  N  6  vcp-0
7 (FPC 7)  Prsnt  ...      ex4200-24f  242  Linecard  N  8  vcp-1
8 (FPC 8)  Prsnt  ...      ex4200-24f  241  Linecard  N  7  vcp-0
8 (FPC 8)  Prsnt  ...      ex4200-24f  241  Linecard  N  0  vcp-1

Member ID for next new member: 9 (FPC 9)

```

show virtual-chassis (Mixed EX4300 multigigabit model—EX4300-48MP—Virtual Chassis)

user@switch> show virtual-chassis

```

Preprovisioned Virtual Chassis
Virtual Chassis ID: abcd.ef00.1234
Virtual Chassis Mode: Mixed

Member ID  Status  Serial No  Model          Mstr    Mixed Route Neighbor List
prio  Role  Mode  Mode  ID Interface
0 (FPC 0)  Prsnt   ...      ex4300-48mp  129  Master*  Y  VC  1  vcp-255/1/0
1 (FPC 1)  Prsnt   ...      ex4300-48mp  129  Backup   Y  VC  3  vcp-255/1/3
2 (FPC 2)  Prsnt   ...      ex4300-48mp   0  Linecard Y  VC  0  vcp-255/1/1
3 (FPC 3)  Prsnt   ...      ex4300-48t   0  Linecard Y  VC  2  vcp-255/1/3
          1  vcp-255/1/3
          3  vcp-255/1/1
          2  vcp-255/1/2

```

show virtual-chassis (EX8200 Virtual Chassis)

user@external-routing-engine> show virtual-chassis

```

Virtual Chassis ID: cdc1.1212.efef
Virtual Chassis Mode: Enabled

Member ID  Status  Serial No  Model  Mastership  Neighbor List
prio  Role  ID Interface
0 (FPC 0-15)  Prsnt   ...      ex8216   0  Linecard  8  vcp-0/0
          8  vcp-0/1

```

```

1 (FPC 16-31) Prsnt ... ex8208 0 Linecard 1 vcp-4/0/4
8 (FPC 128-143) Prsnt ... ex-xre 128 Master 8 vcp-0/0
0 vcp-3/0/4
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 ... ex-xre 128 Backup* 8 vcp-1/0
8 vcp-1/2
8 vcp-1/3
8 vcp-1/3

```

show virtual-chassis (QFX5110 Virtual Chassis)

user@switch> show virtual-chassis

```

Preprovisioned Virtual Chassis
Virtual Chassis ID: abab.1212.cdcd
Virtual Chassis Mode: Enabled

Mstr      Mixed Route Neighbor List
MemberID  Status Serial No  Model      prio  Role   Mode  Mode  ID Interface

0 (FPC 0) Prsnt ... qfx5110-32q 129 Backup Y VC 9 vcp-255/0/12
1 (FPC 1) Prsn ... qfx5110-32q 129 Master* Y VC 1 vcp-255/0/16
2 vcp-255/0/16
2 (FPC 2) Prsnt ... qfx5110-32q 0 Linecard Y VC 0 vcp-255/0/17
3 vcp-255/0/11
3 (FPC 3) Prsnt ... qfx5110-32q 0 Linecard Y VC 1 vcp-255/0/14
4 vcp-255/0/14
4 (FPC 4) Prsnt ... qfx5110-32q 0 Linecard Y VC 2 vcp-255/0/31
5 vcp-255/0/10
3 vcp-255/0/5
5 (FPC 5) Prsnt ... qfx5100e-24q-2p 0 Linecard Y VC 6 vcp-255/0/12
4 vcp-255/0/20

```

```

6 (FPC 6) Prsnt ...      qfx5110-48s-4c  0  Linecard  Y  VC      7 vcp-255/0/10
                                     5 vcp-255/0/49
7 (FPC 7) Prsnt ...      qfx5100e-48s-6q  0  Linecard  Y  VC      6 vcp-255/0/18
                                     8 vcp-255/0/31
8 (FPC 8) Prsnt ...      qfx5110-48s-4c  0  Linecard  Y  VC      7 vcp-255/0/21
                                     9 vcp-255/0/49
9 (FPC 9) Prsnt ...      qfx5110-32q      0  Linecard  Y  VC      8 vcp-255/0/13
                                     0 vcp-255/0/17

```

show virtual-chassis (QFX5200 Virtual Chassis)

user@switch> show virtual-chassis

```

Virtual Chassis ID: abab.1212.cdcd
Virtual Chassis Mode: Enabled

```

		Serial		Mstr		Mixed Route Neighbor List			
Member ID	Status	No	Model	prio	Role	Mode	Mode	ID	Interface
0 (FPC 0)	Prsnt	...	qfx5200-32c-r	128	Master*	N	VC	1	vcp-255/0/19
								2	vcp-255/0/26
1 (FPC 1)	Prsnt	...	qfx5200-32c-r	128	Linecard	N	VC	0	vcp-255/0/20
								2	vcp-255/0/16
2 (FPC 2)	Prsnt	...	qfx5200-32c-32q	128	Backup	N	VC	0	vcp-255/0/15
								1	vcp-255/0/18

show virtual-chassis (QFX5100 Virtual Chassis Fabric)

user@switch> show virtual-chassis

```

Preprovisioned Virtual Chassis Fabric
Fabric ID: 0123.abcd.4567
Fabric Mode: Enabled

```

		Serial		Mstr		Mixed Route Neighbor List			
Member ID	Status	No	Model	prio	Role	Mode	Mode	ID	Interface
0 (FPC 0)	Prsnt	...	qfx5100-48s	129	Master*	N	F	3	vcp-255/1/0

[illegible]

show virtual-chassis active-topology

Syntax

```
show virtual-chassis active-topology
<all-members | local | member member-id>
```

Release Information

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

Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series.

Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).

Description

Display the active topology of the Virtual Chassis or VCF with next-hop reachability information.

Options

none—Display the active topology of the member switch where you enter this command.

all-members—(Optional) Display the active topology of all members of the Virtual Chassis or VCF.

local—(Optional) Display the active topology of the switch or external Routing Engine where you enter this command.

member *member-id*—(Optional) Display the active topology of the specified member of the Virtual Chassis or VCF.

Required Privilege Level

view

RELATED DOCUMENTATION

Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis

List of Sample Output

[show virtual-chassis active-topology \(EX4200 Virtual Chassis\) on page 186](#)

[show virtual-chassis active-topology \(EX8200 Virtual Chassis\) on page 187](#)

[show virtual-chassis active-topology \(Virtual Chassis Fabric\) on page 188](#)

Output Fields

[Table 7 on page 186](#) lists the output fields for the **show virtual-chassis active-topology** 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
Destination ID	Specifies the member ID of the destination.
Next-hop	Specifies the member ID and Virtual Chassis port (VCP) of the next hop to which packets for the destination ID are forwarded. The next hop can be more than one device in a VCF.

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

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

Syntax

```
show virtual-chassis device-topology
<all-members | local | member member-id>
```

Release Information

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).

Description

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.

Options

- none**—Display the device topology for all members of the Virtual Chassis or VCF.
- all-members**—(Optional) Display the device topology for all members of the Virtual Chassis or VCF.
- local**—(Optional) Display the device topology for the switch or external Routing Engine where you enter this command.
- member *member-id***—(Optional) Display the device topology for the specified member of the Virtual Chassis or VCF.

Required Privilege Level

clear

RELATED DOCUMENTATION

- [Understanding Virtual Chassis Port Link Aggregation](#)
- [Understanding EX8200 Virtual Chassis Topologies](#)

Output Fields

[Table 8 on page 192](#) lists the output fields for the **show virtual-chassis device-topology** 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
Member	Assigned member ID.

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

Field Name	Field Description
Device	Assigned device ID. For an EX8200 Virtual Chassis, the member ID and the device ID are always identical.
Status	The status of the device within the Virtual Chassis or VCF. Outputs include: <ul style="list-style-type: none"> • Prsnt—Device is currently connected to and participating in the Virtual Chassis or VCF. • NotPrsnt—Device is assigned but is not currently connected.
System ID	System ID of the device. The system ID of the device is the device's MAC address.
Member (Neighbor List)	Assigned member ID of the neighbor device.
Device (Neighbor List)	Assigned device ID of the neighbor device. For an EX8200 Virtual Chassis, the member ID and the device ID are always identical.
Interface (Neighbor List)	The interface connecting the device to the neighbor.

Sample Output

show virtual-chassis device-topology

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

```
member0:
```

```
-----
                                Neighbor List
Member  Device  Status  System ID      Member  Device  Interface
   0      0    Prsnt   0021.59f7.d000    8       8      vcp-0/0
   1      1    Prsnt   0026.888d.6800    1       1      vcp-4/0/1
   1      1    Prsnt   0026.888d.6800    8       8      vcp-0/0
   1      1    Prsnt   0026.888d.6800    9       9      vcp-0/1
   1      1    Prsnt   0026.888d.6800    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:

				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

member8:

				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

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

8	8	Prsnt	0000.4a75.9b7c	0	0	vcp-3/0/4
				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

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

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
				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

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
				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

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
				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

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
				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

show virtual-chassis login

Syntax

```
show virtual-chassis login
```

Release Information

Command introduced 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 (VCF).

Description

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.

You might need this information for tracing or troubleshooting purposes.

Required Privilege Level

view

RELATED DOCUMENTATION

[request session member | 145](#)

Understanding Global Management of a Virtual Chassis

List of Sample Output

[show virtual-chassis login \(Direct Login to the Master Console Port\) on page 200](#)

[show virtual-chassis login \(Backup Console Session Redirected to the Master Console Port\) on page 200](#)

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

Syntax

```
show virtual-chassis mode
<all-members | local | member member-id>
```

Release Information

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).

Current mode and **Future mode after reboot** fields introduced in Junos OS Release 13.2X51-D20.

Description

Display the Virtual Chassis or Virtual Chassis Fabric (VCF) mixed mode status.

Options

none—Display the Virtual Chassis or VCF mixed mode status for the device where you enter the command.

all-members—(Optional) Display the Virtual Chassis or VCF mixed mode status for all member devices in the Virtual Chassis or VCF.

local—(Optional) Display the Virtual Chassis or VCF mixed mode status for the device where you enter the command.

member *member-id*—(Optional) Display the Virtual Chassis or VCF mixed mode status for the specified member device.

Required Privilege Level

view

RELATED DOCUMENTATION

[request virtual-chassis mode | 158](#)

[Verifying the Virtual Chassis Fabric Mode Settings | 87](#)

[Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\)](#)

List of Sample Output

[show virtual-chassis mode \(EX4200\) on page 203](#)

[show virtual-chassis mode \(QFX5100\) on page 204](#)

Output Fields

[Table 9 on page 203](#) lists the output fields for the **show virtual-chassis mode** command.

Table 9: show virtual-chassis mode Output Fields

Field Name	Field Description
Mixed Mode	Specifies the mixed mode status of the member switch. Mixed mode is either Enabled or Disabled .
Current mode	<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 Current mode and Future mode after reboot 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"> • Fabric with mixed devices—Fabric mode and mixed mode are enabled. • Fabric with similar devices—Fabric mode is enabled and mixed mode is disabled. • Virtual Chassis with mixed devices—Fabric mode is disabled and mixed mode is enabled. • Virtual Chassis with similar devices—Fabric mode is disabled and mixed mode is disabled.
Future mode after reboot	<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 Current mode and Future mode after reboot 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"> • Fabric with mixed devices—Fabric mode and mixed mode are enabled. • Fabric with similar devices—Fabric mode is enabled and mixed mode is disabled. • Virtual Chassis with mixed devices—Fabric mode is disabled and mixed mode is enabled. • Virtual Chassis with similar devices—Fabric mode is disabled and mixed mode is disabled.

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

Syntax

```
show virtual-chassis protocol adjacency
<brief | detail | extensive>
<all-members | local | member member-id>
<system-id>
```

Release Information

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).

Description

Display the Virtual Chassis Control Protocol (VCCP) adjacency statistics in the Virtual Chassis or VCF for all hardware devices.

Options

none—Display VCCP adjacency statistics in brief form for all members of the Virtual Chassis or VCF.

brief | detail | extensive—(Optional) Display the specified level of output. Using the **brief** option is equivalent to entering the command with no options (the default). The **detail** and **extensive** options provide identical displays.

all-members—(Optional) Display VCCP adjacency statistics in brief form for all members of the Virtual Chassis or VCF.

local—(Optional) Display VCCP adjacency statistics for the switch or external Routing Engine on which this command is entered.

member *member-id*—(Optional) Display VCCP adjacency statistics for the specified member of the Virtual Chassis or VCF.

system-id—(Optional) Display VCCP adjacency statistics for the specified member of the Virtual Chassis or VCF.

Required Privilege Level

clear

RELATED DOCUMENTATION

Understanding Virtual Chassis Port Link Aggregation

Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis

List of Sample Output

[show virtual-chassis protocol adjacency on page 206](#)

[show virtual-chassis protocol adjacency detail on page 207](#)

Output Fields

Table 10 on page 206 lists the output fields for the **show virtual-chassis protocol adjacency** command. Output fields are listed in the approximate order in which they appear.

Table 10: show virtual-chassis protocol adjacency Output Fields

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

Syntax

```
show virtual-chassis protocol database
<brief | detail | extensive>
<all-members | local | member member-id>
```

Release Information

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).

Description

Display the Virtual Chassis Control Protocol (VCCP) database statistics for all hardware devices within the Virtual Chassis or VCF.

Options

none—Display VCCP database statistics in brief form for all members of the Virtual Chassis or VCF.

brief | detail | extensive—(Optional) Display the specified level of output. Using the **brief** option is equivalent to entering the command with no options (the default). The **detail** option provides more output than the **brief** option. The **extensive** option provides all output and is most useful for customer support personnel.

all-members—(Optional) Display VCCP database statistics in brief form for all members of the Virtual Chassis or VCF.

local—(Optional) Display VCCP database statistics for the switch or external Routing Engine where you enter this command.

member *member-id*—(Optional) Display VCCP database statistics for the specified member of the Virtual Chassis or VCF.

Required Privilege Level

clear

RELATED DOCUMENTATION

Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis

Understanding Virtual Chassis Components

List of Sample Output

[show virtual-chassis protocol database on page 210](#)

[show virtual-chassis protocol database detail on page 211](#)

Output Fields

Table 11 on page 210 lists the output fields for the **show virtual-chassis protocol database** 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
Sequence	Sequence number of the LSP.	All levels
Checksum	Checksum value of the LSP.	All levels
Lifetime	Remaining lifetime of the LSP, in seconds.	All levels
Neighbor	MAC address of the neighbor on the advertising system.	detail
Interface	Virtual Chassis port (VCP) interface name.	detail
Metric	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

Syntax

```
show virtual-chassis protocol interface
<brief | detail>
<all-members | local | member member-id>
<interface-name>
```

Release Information

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).

Description

Display information about Virtual Chassis Control Protocol (VCCP) statistics for VCCP-enabled interfaces within the Virtual Chassis or VCF.

Options

none—Display the VCCP interface statistics in brief form for all members of the Virtual Chassis or VCF.

brief | detail —(Optional) Display the specified level of output. Using the **brief** option is equivalent to entering the command with no options (the default). The **detail** option provides more output than the **brief** option.

all-members—(Optional) Display VCCP interface statistics for all members of the Virtual Chassis or VCF.

interface-name—(Optional) Display VCCP interface statistics for the specified interface.

local—(Optional) Display VCCP interface statistics for the switch or external Routing Engine where you enter this command.

member *member-id*—(Optional) Display VCCP interface statistics for the specified member of the Virtual Chassis or VCF.

Required Privilege Level

clear

RELATED DOCUMENTATION

[Understanding EX Series Virtual Chassis](#)

[Understanding QFX Series Virtual Chassis](#)

[Understanding Virtual Chassis Ports in an EX8200 Virtual Chassis](#)

List of Sample Output[show virtual-chassis protocol interface on page 215](#)**Output Fields**

[Table 12 on page 215](#) lists the output fields for the **show virtual-chassis protocol interface** 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
Interface	Name of the VCP.	All levels
State	State of the link. Outputs include: <ul style="list-style-type: none"> • Up—The link is up. • Down—The link is down. 	All levels
Metric	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

Syntax

```
show virtual-chassis protocol route
<all-members | local | member member-id>
<destination-id>
```

Release Information

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).

Description

Display the unicast and multicast Virtual Chassis Control Protocol (VCCP) routing tables within the Virtual Chassis or VCF.

Options

none—Display the unicast and multicast routing tables for all members of the Virtual Chassis.

all-members—(Optional) Display the unicast and multicast routing tables for all members of the Virtual Chassis or VCF.

destination-id—(Optional) Display the unicast and multicast routing tables to the specified destination member ID for each member of the Virtual Chassis or VCF.

local—(Optional) Display the unicast and multicast routing tables on the device where you enter this command.

member *member-id*—(Optional) Display the unicast and multicast routing tables for the specified member of the Virtual Chassis or VCF.

Required Privilege Level

clear

RELATED DOCUMENTATION

Understanding EX Series Virtual Chassis

Understanding QFX Series Virtual Chassis

Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis

List of Sample Output

[show virtual-chassis protocol route on page 218](#)

Output Fields

Table 13 on page 218 lists the output fields for the **show virtual-chassis protocol route** 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
Dev	MAC address of the member storing the VCCP routing table.
Version	Version of the shortest-path-first algorithm that generated the routing table.
System ID	MAC address of the device.
Version	Version of the shortest-path-first (SPF) algorithm that generated the route.
Metric	The metric number to get to that device.
Interface	Name of the Virtual Chassis port (VCP) interface connecting the devices.
Via	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

Syntax

```
show virtual-chassis protocol statistics
<all-members | local | member member-id>
<interface-name>
```

Release Information

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).

Description

Display the Virtual Chassis Control Protocol (VCCP) statistics for all hardware devices within the Virtual Chassis or VCF.

Options

none—Display VCCP statistics for all members of the Virtual Chassis or VCF.

all-members—(Optional) Display VCCP statistics for all members of the Virtual Chassis or VCF.

interface-name—(Optional) Display VCCP statistics for the specified interface.

local—(Optional) Display VCCP statistics for the switch or external Routing Engine where you run this command.

member *member-id*—(Optional) Display VCCP statistics for the specified member of the Virtual Chassis or VCF.

Required Privilege Level

clear

RELATED DOCUMENTATION

Understanding EX Series Virtual Chassis

Understanding QFX Series Virtual Chassis

Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis

List of Sample Output

[show virtual-chassis protocol statistics on page 222](#)

Output Fields

Table 14 on page 222 lists the output fields for the **show virtual-chassis protocol interface** 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
PDU type	Protocol data unit type.
Received	Number of PDUs received since VCCP started or since the statistics were set to zero.
Processed	Number of PDUs received minus the number of PDUs dropped.
Drops	Number of PDUs dropped.
Sent	Number of PDUs transmitted since VCCP started or since the statistics were set to zero.
Rexmit	Number of PDUs retransmitted since VCCP started or since the statistics were set to zero.
Total Packets Received	Number of PDUs received since VCCP started or since the statistics were set to zero.
Total Packets Sent	Number of PDUs sent since VCCP started or since the statistics were set to zero.
LSP queue length	Number of link-state PDUs waiting in the queue for processing. This value is almost always 0.
SPF runs	Number of shortest-path-first (SPF) calculations that have been performed.
Fragments Rebuilt	Number of link-state PDU fragments that the local system has computed.
LSP Regenerations	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.
Purges initiated	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 vc-path

Syntax

```
show virtual-chassis vc-path source-interface interface-name destination-interface interface-name
```

Release Information

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.

Description

Show the forwarding path a packet takes when going from a source interface to a destination interface in a Virtual Chassis or VCF configuration.

Starting in Junos OS Releases 14.1X53-D40, 15.1R5, and 16.1R3, when used in VCF configurations, this command displays additional information about next hops, including when the forwarding path has multiple possible next hops within the VCF.

Options

- source-interface *interface-name***—Name of the interface from which the packet originates in the Virtual Chassis or VCF
- destination-interface *interface-name***—Name of the interface to which the packet is being delivered in the Virtual Chassis or VCF

Required Privilege Level

view

RELATED DOCUMENTATION

Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis
Understanding EX Series Virtual Chassis
Understanding QFX Series Virtual Chassis
EX8200 Virtual Chassis Overview

List of Sample Output

- [show virtual-chassis vc-path source-interface destination-interface \(Virtual Chassis\) on page 227](#)
- [show virtual-chassis vc-path source-interface destination-interface \(Virtual Chassis Fabric\) on page 227](#)

Output Fields

[Table 15 on page 226](#) lists the output fields for the **show virtual-chassis vc-path** command. Output fields are listed in the approximate order in which they appear.

Some output field names and display order differ between the output for a Virtual Chassis and for a VCF. When this command is used to display forwarding paths in a VCF, additional fields are included in the output to show details of multiple possible next hops. The differences are described in [Table 15 on page 226](#) and shown in sample output for each mode.

Table 15: show virtual-chassis vc-path Output Fields

Field Name	Field Description
Hop	<p>The hop number along the path between the source and destination interfaces. The first hop entry (Hop 0) 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.</p> <p>For VCF output, each Hop entry also shows information about multiple possible next hops towards the destination. See Next-hop PFE, Interface, and Bandwidth output field descriptions for details.</p>
Member (Virtual Chassis) Member-ID (VCF)	<p>The Virtual Chassis or VCF member ID of the switch that contains the Packet Forwarding Engine for each hop through which the packet passes.</p>
PFE-Device (Virtual Chassis) PFE (VCF)	<p>The number of the Packet Forwarding Engine in each Virtual Chassis or VCF member through which a packet passes.</p> <p>For Virtual Chassis output, 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> <p>VCF output is expanded on multiple rows to show more information about multiple possible next hops—each hop entry is followed by one or more Next-hop PFE output fields with interface name and bandwidth information for each possible next hop.</p>
Next-hop PFE (VCF only)	<p>One or more possible next-hop Packet Forwarding Engine numbers for VCF member Member-ID. Each Next-hop entry includes the next-hops's interface name and bandwidth.</p>
Interface	<p>The name of the interface through which the Packet Forwarding Engines are connected. The interface for the first hop (Hop 0) is always the source interface.</p> <p>VCF output provides more information about multiple possible next hops for each hop entry, listing the Interface and the interface Bandwidth for each possible Next-hop PFE.</p>


```

      2 vcp-255/0/2
8 (FPC 8) Prsnt ... qfx5100-24q-aa 0 Linecard N F 1 vcp-255/0/1

      2 vcp-255/0/2
9 (FPC 9) Prsnt ... qfx5100-24q-aa 0 Linecard N F 1 vcp-255/0/1

      2 vcp-255/0/2
10 (FPC 10)Prsnt ... qfx5100-24q-aa 0 Linecard N F 1 vcp-255/0/1

      2 vcp-255/0/2
11 (FPC 11)Prsnt ... qfx5100-24q-aa 0 Linecard N F 1 vcp-255/0/1

      2 vcp-255/0/2
12 (FPC 12)Prsnt ... qfx5100-24q-aa 0 Linecard N F 1 vcp-255/0/1

      2 vcp-255/0/2

```

user@switch> **show virtual-chassis vc-path source-interface xe-12/0/25 destination-interface xe-5/0/25**

```

Fabric forwarding path from xe-12/0/25 (PFE 12) to xe-5/0/25 (PFE 5)
Hop 0 Member-ID 12 PFE 12
  Next-hop PFE 1
    Interface vcp-255/0/1.32768 Bandwidth 40
  Next-hop PFE 2
    Interface vcp-255/0/2.32768 Bandwidth 40
Hop 1 Member-ID 1 PFE 1
  Next-hop PFE 5
    Interface vcp-255/0/5.32768 Bandwidth 40
Hop 1 Member-ID 2 PFE 2
  Next-hop PFE 5
    Interface vcp-255/0/5.32768 Bandwidth 40
Hop 2 Member-ID 5 PFE 5

```

user@switch> **show virtual-chassis vc-path source-interface xe-12/0/25 destination-interface xe-1/0/25**

```

Fabric forwarding path from xe-12/0/25 (PFE 12) to xe-1/0/25 (PFE 1)
Hop 0 Member-ID 12 PFE 12
  Next-hop PFE 1

```

Interface vcp-255/0/1.32768					Bandwidth	40
Hop	1	Member-ID	1	PFE	1	

show virtual-chassis vc-port

Syntax

```
show virtual-chassis vc-port
<all-members | local | member member-id>
```

Release Information

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

Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series.

Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).

Description

Display the status of the Virtual Chassis ports (VCPs), including dedicated VCPs, default-configured VCPs, and uplink ports configured as VCPs, if present.

Options

none—Display the operational status of all VCPs of the member switch where you enter the command.

all-members—(Optional) Display the operational status of all VCPs on all members of the Virtual Chassis or VCF.

local—(Optional) Display the operational status of the switch or external Routing Engine where you enter this command.

member *member-id*—(Optional) Display the operational status of all VCPs for the specified member of the Virtual Chassis or VCF.

Required Privilege Level

view

RELATED DOCUMENTATION

[show virtual-chassis vc-port statistics](#) | 256

Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis

List of Sample Output

[show virtual-chassis vc-port \(EX4200 Virtual Chassis\) on page 233](#)

[show virtual-chassis vc-port \(EX8200 Virtual Chassis\) on page 234](#)

[show virtual-chassis vc-port all-members on page 235](#)

Output Fields

Table 16 on page 232 lists the output fields for the **show virtual-chassis vc-port** command. Output fields are listed in the approximate order in which they appear.

Table 16: show virtual-chassis vc-port Output Fields

Field Name	Field Description
fpcnumber	The FPC number is the same as the member ID.
Interface or PIC/Port	<p>VCP name.</p> <ul style="list-style-type: none"> • The dedicated VCPs in an EX4200 or EX4500 Virtual Chassis are vcp-0 and vcp-1. The dedicated VCPs in an EX4550 Virtual Chassis are VCP-1/0, VCP-1/1, VCP-2/0, and VCP-2/1. • Optical ports set as VCPs are named 1/0 and 1/1, representing the PIC number and the port number. • The native VCP (port 0) on an XRE200 External Routing Engine in an EX8200 Virtual Chassis is named vcp-0. • The VCPs on each Virtual Chassis Control Interface (VCCI) module in an XRE200 External Routing Engine are named using the vcp-slot-number/port-number convention; for instance, vcp-1/0. • The VCPs on EX8200 member switches are named using the vcp-slot-number/pic-number/interface-number convention; for instance, vcp-3/0/2. • A 255 as the first number in your port number indicates that your VCP is part of a Link Aggregation group (LAG) bundle. For example, vcp-255/1/0 indicates that the dedicated VCP named vcp-1/0 is part of a LAG bundle, and vcp-255/1/0 represents an uplink port previously named xe-0/1/0 that is now part of a VCP LAG bundle.
Type	<p>Type of VCP:</p> <ul style="list-style-type: none"> • Dedicated—Rear panel dedicated VCP on an EX4200 or EX4300 multigigabit model switch, a Virtual Chassis module port on EX4500 or EX4550 switch, or any VCP link connected to an XRE200 External Routing Engine in an EX8200 Virtual Chassis. • Configured—Optical port configured as a VCP, which includes default-configured VCPs (set in the default factory configuration) or those that are set by a user. • Auto-Configured—Optical port automatically converted into a VCP. <p>See <i>Setting an Uplink Port on an EX Series or QFX Series Switch as a Virtual Chassis Port</i> or <i>Configuring an EX4650 or a QFX Series Virtual Chassis</i> for information about configuring VCPs, and <i>Automatic Virtual Chassis Port (VCP) Conversion</i> for details on how a port is automatically converted into a VCP.</p>

Table 16: show virtual-chassis vc-port Output Fields (*continued*)

Field Name	Field Description
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"> • absent—Interface is not a VCP link. • down—VCP link is down. • up—VCP link is up.
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		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

Syntax

```
show virtual-chassis vc-port diagnostics optics
<all-members | local | member member-id>
<interface-name>
```

Release Information

Command introduced in Junos OS Release 12.2 for EX Series switches.

Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).

Description

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.

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.

On some EX Series switches, the **request virtual-chassis vc-port diagnostics optics** command must be entered to run a diagnostic scan before you can gather the **show virtual-chassis vc-port diagnostics optics** output.

Options

none—Display diagnostics information for transceivers installed in VCPs of all members of a Virtual Chassis or VCF.

all-members—(Optional) Display diagnostics information for transceivers installed in VCPs of all members of a Virtual Chassis or VCF.

interface-name—(Optional) Display diagnostics information for the transceiver installed in a specified VCP.

local—(Optional) Display diagnostics information for transceivers installed in VCPs on the switch or external Routing Engine where you enter this command.

member *member-id*—(Optional) Display diagnostics information for transceivers installed in VCPs on a specified member of a Virtual Chassis or VCF.

Required Privilege Level

view

RELATED DOCUMENTATION

[show virtual-chassis vc-port](#) | 231

Installing a Transceiver

Removing a Transceiver

Junos OS Ethernet Interfaces Configuration Guide

List of Sample Output

[show virtual-chassis vc-port diagnostics optics](#) on page 241

[show virtual-chassis vc-port diagnostics optics \(interface-name\)](#) on page 248

[show virtual-chassis vc-port diagnostics optics local](#) on page 251

[show virtual-chassis vc-port diagnostics optics \(member member-id\)](#) on page 253

Output Fields

[Table 17 on page 238](#) 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 17: 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> .

Table 17: show virtual-chassis vc-port diagnostics optics Output Fields (continued)

Field Name	Field Description
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> .
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.

Table 17: show virtual-chassis vc-port diagnostics optics Output Fields (continued)

Field Name	Field Description
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.
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.

Table 17: show virtual-chassis vc-port diagnostics optics Output Fields (continued)

Field Name	Field Description
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

Syntax

```
show virtual-chassis vc-port statistics
<all-members | local | member member-id>
<brief | detail | extensive >
<interface-name>
```

Release Information

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

The options **all-members**, **brief**, **detail**, **extensive**, and **local** 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 (VCF).

Description

Display the traffic statistics collected on Virtual Chassis ports (VCPs).

Options

none—Display traffic statistics for VCPs of all members of a Virtual Chassis or VCF.

brief | detail | extensive—(Optional) Display the specified level of output. Using the **brief** option is equivalent to entering the command with no options (the default). The **detail** and **extensive** options provide identical displays.

all-members—(Optional) Display traffic statistics for VCPs of all members of a Virtual Chassis or VCF

interface-name—(Optional) Display traffic statistics for the specified VCP.

local—(Optional) Display traffic statistics for VCPs on the switch or external Routing Engine where you enter this command.

member *member-id*—(Optional) Display traffic statistics for VCPs on the specified member of a Virtual Chassis or VCF.

Required Privilege Level

view

RELATED DOCUMENTATION

[clear virtual-chassis vc-port statistics | 143](#)

[show virtual-chassis vc-port | 231](#)

List of Sample Output[show virtual-chassis vc-port statistics on page 260](#)[show virtual-chassis vc-port statistics \(EX8200 Virtual Chassis\) on page 261](#)[show virtual-chassis vc-port statistics brief on page 261](#)[show virtual-chassis vc-port statistics extensive on page 262](#)[show virtual-chassis vc-port statistics member 0 on page 264](#)**Output Fields**

[Table 17 on page 238](#) 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 18: show virtual-chassis vc-port statistics Output Fields

Field Name	Field Description	Level of Output
fpcnumber	(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
member number	(EX8200 Virtual Chassis only) Member ID of the Virtual Chassis member.	All levels
Interface	VCP name.	brief
Input Octets/Packets	Number of octets and packets received on the VCP.	brief, member , none
Output Octets/Packets	Number of octets and packets transmitted on the VCP.	brief, member , none
master: number	Member ID of the master Routing Engine.	All levels
Port	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.	detail, extensive
Total octets	Total number of octets received and transmitted on the VCP.	detail, extensive
Total packets	Total number of packets received and transmitted on the VCP.	detail, extensive

Table 18: show virtual-chassis vc-port statistics Output Fields (*continued*)

Field Name	Field Description	Level of Output
Unicast packets	Number of unicast packets received and transmitted on the VCP.	detail, extensive
Broadcast packets	Number of broadcast packets received and transmitted on the VCP.	detail, extensive
Multicast packets	Number of multicast packets received and transmitted on the VCP.	detail, extensive
MAC control frames	Number of media access control (MAC) control frames received and transmitted on the VCP.	detail, extensive
CRC alignment errors	<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"> • Invalid FCS with an integral number of octets (FCS error) • Invalid FCS with a nonintegral number of octets (alignment error) 	detail, extensive
Oversize packets	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.	detail, extensive
Undersize packets	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..	detail, extensive

Table 18: show virtual-chassis vc-port statistics Output Fields (*continued*)

Field Name	Field Description	Level of Output
Jabber packets	<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>NOTE: 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>	detail, extensive
Fragments received	<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 runts (which are normal occurrences caused by collisions) and noise hits are counted.</p>	detail, extensive
Ifout errors	Number of outbound packets received on the VCP that could not be transmitted because of errors.	detail, extensive
Packet drop events	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.	detail, extensive
64 octets frames	Number of packets received on the VCP (including invalid packets) that were 64 octets in length (excluding framing bits, but including FCS octets).	detail, extensive
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

Table 18: show virtual-chassis vc-port statistics Output Fields (continued)

Field Name	Field Description	Level of Output
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}