



---

# Junos<sup>®</sup> OS for EX Series Ethernet Switches

## Virtual Chassis Feature Guide for EX4600 Switches

Release

14.1X53



---

Modified: 2016-11-15

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

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

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

*Junos<sup>®</sup> OS for EX Series Ethernet Switches Virtual Chassis Feature Guide for EX4600 Switches*  
Release 14.1X53  
Copyright © 2016, Juniper Networks, Inc.  
All rights reserved.

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

#### YEAR 2000 NOTICE

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

#### END USER LICENSE AGREEMENT

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

# Table of Contents

	About the Documentation . . . . .	xi
	Documentation and Release Notes . . . . .	xi
	Supported Platforms . . . . .	xi
	Using the Examples in This Manual . . . . .	xi
	Merging a Full Example . . . . .	xii
	Merging a Snippet . . . . .	xii
	Documentation Conventions . . . . .	xiii
	Documentation Feedback . . . . .	xv
	Requesting Technical Support . . . . .	xv
	Self-Help Online Tools and Resources . . . . .	xv
	Opening a Case with JTAC . . . . .	xvi
<b>Part 1</b>	<b>Virtual Chassis</b>	
<b>Chapter 1</b>	<b>Configuring Virtual Chassis . . . . .</b>	<b>3</b>
	Understanding EX4600 Virtual Chassis . . . . .	4
	Understanding EX Series Virtual Chassis Components . . . . .	5
	Maximum Number of Switches per Virtual Chassis . . . . .	5
	Virtual Chassis Ports (VCPs) . . . . .	7
	Master Role . . . . .	8
	Backup Role . . . . .	8
	Linecard Role . . . . .	9
	Member Switch and Member ID . . . . .	9
	Mastership Priority . . . . .	10
	Mixed Virtual Chassis . . . . .	11
	Virtual Chassis Identifier (VCID) . . . . .	11
	Understanding Mixed EX Series and QFX Series Virtual Chassis or Virtual Chassis Fabric . . . . .	12
	Virtual Chassis Fabric Summary . . . . .	12
	Understanding Mixed Virtual Chassis Fabric . . . . .	13
	Virtual Chassis Summary for QFX5100, QFX3600, QFX3500, EX4600, and EX4300 Switches . . . . .	13
	Understanding the Routing Engine Role in a Mixed Virtual Chassis Using EX4300, EX4600, QFX3500, QFX3600, or QFX5100 Member Switches . . . . .	14
	Understanding EX4300, QFX3500, QFX3600, and QFX5100 Switches in a Virtual Chassis . . . . .	15
	Understanding Mixed EX4300 and EX4600 Virtual Chassis . . . . .	15
	Understanding EX4200, EX4500, and EX4550 Switches in a Mixed Virtual Chassis . . . . .	15
	Understanding How the Master in a Virtual Chassis Is Elected . . . . .	16

Understanding Software Upgrade in an EX Series Virtual Chassis . . . . .	17
Understanding Global Management of a Virtual Chassis . . . . .	18
Understanding Nonvolatile Storage in a Virtual Chassis . . . . .	21
Nonvolatile Memory Features . . . . .	21
Understanding EX Series Virtual Chassis Port Link Aggregation . . . . .	21
Understanding Split and Merge in a Virtual Chassis . . . . .	23
What Happens When a Virtual Chassis Configuration Splits . . . . .	24
Merging Virtual Chassis Configurations . . . . .	25
Understanding Automatic Software Update on Virtual Chassis Member Switches . . . . .	26
Automatic Software Update Basics . . . . .	27
Automatic Software Update Restrictions . . . . .	27
Understanding MAC Address Assignment on a Virtual Chassis . . . . .	29
Understanding High Availability on an EX Series Virtual Chassis . . . . .	30
Configuring EX4600 Switches in a Mixed or Non-Mixed Virtual Chassis (CLI Procedure) . . . . .	31
Configuring an EX4600 Virtual Chassis with a Nonprovisioned Configuration File . . . . .	32
Configuring an EX4600 Virtual Chassis with a Preprovisioned Configuration File . . . . .	34
Adding an EX4600 Switch to a Mixed or Non-mixed Virtual Chassis (CLI Procedure) . . . . .	36
Replacing a Member Switch of a Virtual Chassis Configuration (CLI Procedure) . . . . .	38
Remove, Repair, and Reinstall the Same Switch . . . . .	38
Remove a Member Switch, Replace It with a Different Switch, and Reapply the Old Configuration . . . . .	39
Remove a Member Switch and Make Its Member ID Available for Reassignment to a Different Switch . . . . .	40
Configuring Mastership of a Virtual Chassis (CLI Procedure) . . . . .	41
Configuring Mastership Using a Preprovisioned Configuration File . . . . .	41
Configuring Mastership Using a Configuration File That Is Not Preprovisioned . . . . .	42
Configuring the Timer for the Backup Member to Start Using Its Own MAC Address, as Master of a Virtual Chassis (CLI Procedure) . . . . .	43
Setting an Uplink Port on an EX Series Switch as a Virtual Chassis Port (CLI Procedure) . . . . .	43
Setting an Uplink VCP Between the Member Switches . . . . .	45
Setting an Uplink VCP on a Standalone Switch . . . . .	46
Disabling Split and Merge in a Virtual Chassis (CLI Procedure) . . . . .	47
Configuring Automatic Software Update on Virtual Chassis Member Switches (CLI Procedure) . . . . .	48
Assigning the Virtual Chassis ID to Determine Precedence During a Virtual Chassis Merge (CLI Procedure) . . . . .	49
Command Forwarding Usage with an EX Series Virtual Chassis . . . . .	49
Verifying the Member ID, Role, and Neighbor Member Connections of a Virtual Chassis Member . . . . .	54
Verifying That Virtual Chassis Ports Are Operational . . . . .	55

	Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis . . . . .	57
	Verifying That Graceful Routing Engine Switchover Is Working in the Virtual Chassis . . . . .	59
	Troubleshooting an EX Series Virtual Chassis . . . . .	60
	A Disconnected Member Switch's ID Is Not Available for Reassignment . . .	60
	Load Factory Default Does Not Commit on a Multimember Virtual Chassis . . . . .	60
	The Member ID Persists When a Member Switch Is Disconnected From a Virtual Chassis . . . . .	61
	A Member Switch Is Not Participating in a Mixed Virtual Chassis . . . . .	61
<b>Chapter 2</b>	<b>Configuration Statements for Virtual Chassis . . . . .</b>	<b>63</b>
	[edit virtual-chassis] Configuration Statement Hierarchy . . . . .	63
	Supported Statements in the [edit virtual-chassis] Hierarchy Level . . . . .	64
	Unsupported Statements in the [edit virtual-chassis] Hierarchy Level . . . . .	65
	aliases (Virtual Chassis) . . . . .	66
	alias-name (Virtual Chassis aliases) . . . . .	67
	auto-sw-update . . . . .	68
	id . . . . .	70
	location (Virtual Chassis) . . . . .	71
	mac-persistence-timer . . . . .	72
	mastership-priority . . . . .	73
	member . . . . .	75
	no-management-vlan . . . . .	76
	no-split-detection . . . . .	77
	package-name . . . . .	78
	preprovisioned . . . . .	79
	role . . . . .	80
	serial-number . . . . .	83
	serial-number (Virtual Chassis aliases) . . . . .	84
	traceoptions (Virtual Chassis) . . . . .	85
	vcp-no-hold-time . . . . .	88
	vcp-snmp-statistics . . . . .	89
	virtual-chassis . . . . .	90
<b>Chapter 3</b>	<b>Operational Commands for Virtual Chassis . . . . .</b>	<b>93</b>
	clear virtual-chassis vc-port statistics . . . . .	94
	request session member . . . . .	96
	request virtual-chassis recycle . . . . .	97
	request virtual-chassis renumber . . . . .	98
	request virtual-chassis vc-port . . . . .	99
	show snmp mib . . . . .	101
	show virtual-chassis active-topology . . . . .	104
	show virtual-chassis device-topology . . . . .	109
	show virtual-chassis protocol adjacency . . . . .	115
	show virtual-chassis protocol database . . . . .	119
	show virtual-chassis protocol interface . . . . .	123
	show virtual-chassis protocol route . . . . .	126
	show virtual-chassis protocol statistics . . . . .	129

show virtual-chassis login .....	132
show virtual-chassis .....	133
show virtual-chassis vc-path .....	138
show virtual-chassis vc-port .....	142
show virtual-chassis vc-port statistics .....	146

# List of Figures

<b>Part 1</b>	<b>Virtual Chassis</b>	
<b>Chapter 1</b>	<b>Configuring Virtual Chassis . . . . .</b>	<b>3</b>
	Figure 1: Console Session Redirection (EX4200 Virtual Chassis Pictured) . . . . .	19
	Figure 2: Management Ethernet Port Redirection to the VME Interface . . . . .	20





# List of Tables

	<b>About the Documentation . . . . .</b>	<b>xi</b>
	Table 1: Notice Icons . . . . .	xiii
	Table 2: Text and Syntax Conventions . . . . .	xiii
<b>Part 1</b>	<b>Virtual Chassis</b>	
<b>Chapter 1</b>	<b>Configuring Virtual Chassis . . . . .</b>	<b>3</b>
	Table 3: Maximum Member Switch Support for Virtual Chassis by Junos OS Release . . . . .	6
	Table 4: Virtual Chassis Fabric Summary . . . . .	13
	Table 5: Virtual Chassis Summary . . . . .	14
	Table 6: Maximum Member Links in LAGs Over Optical Interface VCPs . . . . .	22
	Table 7: Automatic Software Update Support . . . . .	28
	Table 8: Effects of a Routing Engine Switchover . . . . .	31
	Table 9: Commands That Can be Run on All or Specific Members of the Virtual Chassis Configuration . . . . .	50
<b>Chapter 3</b>	<b>Operational Commands for Virtual Chassis . . . . .</b>	<b>93</b>
	Table 10: show snmp mib Output Fields . . . . .	102
	Table 11: show virtual-chassis active-topology Output Fields . . . . .	104
	Table 12: show virtual-chassis device-topology Output Fields . . . . .	109
	Table 13: show virtual-chassis protocol adjacency Output Fields . . . . .	116
	Table 14: show virtual-chassis protocol database Output Fields . . . . .	119
	Table 15: show virtual-chassis protocol interface Output Fields . . . . .	124
	Table 16: show virtual-chassis protocol route Output Fields . . . . .	126
	Table 17: show virtual-chassis protocol statistics Output Fields . . . . .	129
	Table 18: show virtual-chassis Output Fields . . . . .	133
	Table 19: show virtual-chassis vc-path Output Fields . . . . .	139
	Table 20: show virtual-chassis vc-port Output Fields . . . . .	142
	Table 21: show virtual-chassis vc-port statistics Output Fields . . . . .	147



# About the Documentation

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

## Documentation and Release Notes

---

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

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

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

## Supported Platforms

---

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

- EX Series

## Using the Examples in This Manual

---

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

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

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

## Merging a Full Example

To merge a full example, follow these steps:

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

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

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

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

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

## Merging a Snippet

To merge a snippet, follow these steps:

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

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

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

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

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

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

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

For more information about the **load** command, see [CLI Explorer](#).

## Documentation Conventions

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

Table 1: Notice Icons







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

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

Table 2: Text and Syntax Conventions

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

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

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

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

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

## Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can provide feedback by using either of the following methods:

- Online feedback rating system—On any page of the Juniper Networks TechLibrary site at <http://www.juniper.net/techpubs/index.html>, simply click the stars to rate the content, and use the pop-up form to provide us with information about your experience. Alternately, you can use the online feedback form at <http://www.juniper.net/techpubs/feedback/>.
- E-mail—Send your comments to [techpubs-comments@juniper.net](mailto:techpubs-comments@juniper.net). Include the document or topic name, URL or page number, and software version (if applicable).

## Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or Partner Support Service support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

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

## Self-Help Online Tools and Resources

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

- Find CSC offerings: <http://www.juniper.net/customers/support/>
- Search for known bugs: <http://www2.juniper.net/kb/>
- Find product documentation: <http://www.juniper.net/techpubs/>
- Find solutions and answer questions using our Knowledge Base: <http://kb.juniper.net/>

- Download the latest versions of software and review release notes:  
<http://www.juniper.net/customers/csc/software/>
- Search technical bulletins for relevant hardware and software notifications:  
<http://kb.juniper.net/InfoCenter/>
- Join and participate in the Juniper Networks Community Forum:  
<http://www.juniper.net/company/communities/>
- Open a case online in the CSC Case Management tool: <http://www.juniper.net/cm/>

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

## Opening a Case with JTAC

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

- Use the Case Management tool in the CSC at <http://www.juniper.net/cm/>.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

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



## PART 1

# Virtual Chassis

- [Configuring Virtual Chassis on page 3](#)
- [Configuration Statements for Virtual Chassis on page 63](#)
- [Operational Commands for Virtual Chassis on page 93](#)



## CHAPTER 1

# Configuring Virtual Chassis

- [Understanding EX4600 Virtual Chassis on page 4](#)
- [Understanding EX Series Virtual Chassis Components on page 5](#)
- [Understanding Mixed EX Series and QFX Series Virtual Chassis or Virtual Chassis Fabric on page 12](#)
- [Understanding How the Master in a Virtual Chassis Is Elected on page 16](#)
- [Understanding Software Upgrade in an EX Series Virtual Chassis on page 17](#)
- [Understanding Global Management of a Virtual Chassis on page 18](#)
- [Understanding Nonvolatile Storage in a Virtual Chassis on page 21](#)
- [Understanding EX Series Virtual Chassis Port Link Aggregation on page 21](#)
- [Understanding Split and Merge in a Virtual Chassis on page 23](#)
- [Understanding Automatic Software Update on Virtual Chassis Member Switches on page 26](#)
- [Understanding MAC Address Assignment on a Virtual Chassis on page 29](#)
- [Understanding High Availability on an EX Series Virtual Chassis on page 30](#)
- [Configuring EX4600 Switches in a Mixed or Non-Mixed Virtual Chassis \(CLI Procedure\) on page 31](#)
- [Adding an EX4600 Switch to a Mixed or Non-mixed Virtual Chassis \(CLI Procedure\) on page 36](#)
- [Replacing a Member Switch of a Virtual Chassis Configuration \(CLI Procedure\) on page 38](#)
- [Configuring Mastership of a Virtual Chassis \(CLI Procedure\) on page 41](#)
- [Configuring the Timer for the Backup Member to Start Using Its Own MAC Address, as Master of a Virtual Chassis \(CLI Procedure\) on page 43](#)
- [Setting an Uplink Port on an EX Series Switch as a Virtual Chassis Port \(CLI Procedure\) on page 43](#)
- [Disabling Split and Merge in a Virtual Chassis \(CLI Procedure\) on page 47](#)
- [Configuring Automatic Software Update on Virtual Chassis Member Switches \(CLI Procedure\) on page 48](#)
- [Assigning the Virtual Chassis ID to Determine Precedence During a Virtual Chassis Merge \(CLI Procedure\) on page 49](#)

- [Command Forwarding Usage with an EX Series Virtual Chassis on page 49](#)
- [Verifying the Member ID, Role, and Neighbor Member Connections of a Virtual Chassis Member on page 54](#)
- [Verifying That Virtual Chassis Ports Are Operational on page 55](#)
- [Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 57](#)
- [Verifying That Graceful Routing Engine Switchover Is Working in the Virtual Chassis on page 59](#)
- [Troubleshooting an EX Series Virtual Chassis on page 60](#)

---

## Understanding EX4600 Virtual Chassis

---

EX4600 switches can act as member switches in a non-mixed Virtual Chassis—a Virtual Chassis composed entirely of EX4600 switches—as well as participate as member switches in a mixed Virtual Chassis with EX4300 switches.

EX4600 Virtual Chassis brings the Virtual Chassis flexible, scaling switch solution to the Juniper Networks EX4600 Ethernet Switch. You can connect up to ten EX4600 switches together to form one EX4600 Virtual Chassis and manage the unit as a single chassis. The advantages of connecting multiple switches into a Virtual Chassis include better-managed bandwidth at a network layer, simplified configuration and maintenance because multiple switches can be managed as a single switch, increased fault tolerance and high availability (HA) because a Virtual Chassis can remain active and network traffic can be redirected to other member switches when a single member switch fails, and a simplified Layer 2 network topology that minimizes or eliminates the need for loop prevention protocols such as Spanning Tree Protocol (STP).

The Virtual Chassis also provides a flexible model for expanding your network. If you are using an EX4600 switch or EX4600 Virtual Chassis at the access layer, for instance, and need additional access ports to support more servers, computers, phones, or other devices, you can add an EX4600 switch as a Virtual Chassis member to increase the number of access ports on your network with minimal complications to the existing network topology and switch configuration. You can add this new switch to a Virtual Chassis if the switch is installed in the same building or at a different site because the long-distance optical ports can be used to interconnect EX4600 switches into the a Virtual Chassis.

You configure an EX4600 Virtual Chassis by configuring optical interfaces connecting EX4600 switches into Virtual Chassis ports (VCPs). VCPs connect switches together to form a Virtual Chassis, and are responsible for passing all data and control traffic between member switches in the Virtual Chassis. All 40-Gigabit QSFP+ and 10-Gigabit SFP+ optical ports on an EX4600 switch can be configured into VCPs. You can increase the VCP bandwidth between any two member switches by connecting multiple VCP links between the switches. When multiple VCP links are interconnecting the same two member switches, a Link Aggregation Group (LAG) bundle is formed when the links are identical speeds. For instance, if you have four 40-Gigabit links configured as VCPs between member switches, a LAG with four member links at 160Gbps of bandwidth is formed.

10-Gigabit and 40-Gigabit links configured as VCPs cannot be members of the same LAG, however.

All models of EX4600 switches can be interconnected into the same EX4600 Virtual Chassis. EX4600 switches can also be interconnected into a mixed Virtual Chassis with EX4300 switches. EX4600 switches must be configured into the Routing Engine role in any mixed Virtual Chassis with EX4300 switches.

An EX4600 Virtual Chassis is configured, monitored, and maintained like other EX Series Virtual Chassis. See [“Understanding EX Series Virtual Chassis Components” on page 5](#).

**Related  
Documentation**

- [Understanding EX Series Virtual Chassis Components on page 5](#)
- [Understanding How the Master in a Virtual Chassis Is Elected on page 16](#)
- [Understanding EX Series Virtual Chassis Port Link Aggregation on page 21](#)
- [Understanding EX Series Virtual Chassis Configuration](#)

---

## Understanding EX Series Virtual Chassis Components

---



**NOTE:** This topic applies to all EX Series Virtual Chassis except EX8200 Virtual Chassis. See [Understanding EX8200 Virtual Chassis Components](#) for information about EX8200 Virtual Chassis.

---

This topic describes the components of EX Series Virtual Chassis—including the components of any mixed Virtual Chassis that contains EX series member switches—except EX8200 Virtual Chassis.

This topic covers:

- [Maximum Number of Switches per Virtual Chassis on page 5](#)
- [Virtual Chassis Ports \(VCPs\) on page 7](#)
- [Master Role on page 8](#)
- [Backup Role on page 8](#)
- [Linecard Role on page 9](#)
- [Member Switch and Member ID on page 9](#)
- [Mastership Priority on page 10](#)
- [Mixed Virtual Chassis on page 11](#)
- [Virtual Chassis Identifier \(VCID\) on page 11](#)

### Maximum Number of Switches per Virtual Chassis

The maximum number of switches that a Virtual Chassis supports varies by Virtual Chassis and can depend on the Junos OS release running on the Virtual Chassis.

[Table 3 on page 6](#) lists the maximum member switch support by Virtual Chassis and Junos OS release.

**Table 3: Maximum Member Switch Support for Virtual Chassis by Junos OS Release**

Maximum Member Switch Support	Initial Junos OS Release
EX2200 Virtual Chassis	12.2R1—Initial release. Support for up to four EX2200 member switches.
EX3300 Virtual Chassis	11.3R1—Initial release. Support for up to six EX3300 member switches  12.2R1—Support for up to ten EX3300 member switches
EX4200 Virtual Chassis	9.0R1—Initial release. Support for up to ten EX4200 member switches
EX4300 Virtual Chassis	13.2X50-D10—Initial release. Support for up to ten EX4300 member switches  13.2X50-D20—EX4300 switch support added for QFX Series Virtual Chassis and for VCF. See <i>Understanding QFX Series Virtual Chassis Components</i> or <i>Understanding Virtual Chassis Fabric Components</i> .
EX4500 Virtual Chassis	11.1R1—Initial release. Support for up to two EX4500 switches  11.4R1—Support for up to ten EX4500 member switches
EX4550 Virtual Chassis	12.2R1—Initial release. Support for up to ten EX4550 switches
EX4600 Virtual Chassis	13.2X51-D25—Initial release. Support for up to ten EX4600 switches
Mixed EX4200 and EX4500 Virtual Chassis	11.1R1—Initial release. Support for up to two EX4500 switches and up to eight EX4200 switches  11.2R1—Support for up to nine EX4200 switches  11.4R1—Support for up to nine EX4500 switches
Mixed EX4200 and EX4550 Virtual Chassis	12.2R1—Initial release. Support for up to ten total EX4200 and EX4550 switches
Mixed EX4200, EX4500, and EX4550 Virtual Chassis	12.2R1—Initial release. Support for up to ten total EX4200, EX4500, and EX4550 switches
Mixed EX4300 and EX4600 Virtual Chassis	13.2X51-D25—Initial release. Support for up to ten total EX4300 and EX4600 switches. EX4600 switches must assume routing engine role.

**Table 3: Maximum Member Switch Support for Virtual Chassis by Junos OS Release (*continued*)**

Maximum Member Switch Support	Initial Junos OS Release
Mixed EX4500 and EX4550 Virtual Chassis	12.2R1—Initial release. Support for up to ten total EX4500 and EX4550 switches
EX9200 Virtual Chassis	13.2R2—Initial release. Support for up to two EX9200 switches.

## Virtual Chassis Ports (VCPs)

You use Virtual Chassis ports (VCPs) to interconnect the member switches in a Virtual Chassis.

Some switches have dedicated VCPs. Dedicated VCPs allow you to interconnect switches without requiring any additional interface configuration.

These switches have dedicated VCPs:

- EX4200 switches, on the rear panel
- EX4500 switches, on the Virtual Chassis module
- EX4550 switches, on the Virtual Chassis module

To interconnect switches that do not have dedicated VCPs or to interconnect switches across greater distances than allowed by a dedicated-VCP connection, you configure an optical port as a VCP. You can configure those VCPs on these switches:

- EX2200 switches, through an uplink port



**NOTE:** All RJ-45 interfaces, including built-in network ports with 10/100/1000BASE-T Gigabit Ethernet connectors and 1000BASE-T RJ-45 transceivers, on EX2200 and EX2200-C switches, can also be configured into VCPs.

- EX3300 switches, through an uplink port



**NOTE:** Uplink ports 2 and 3 on EX3300 switches are configured as VCPs by default.

- EX4200 switches, through uplink module ports (SFP, SFP+, or XFP) or through an SFP+ port on the EX4200-24F switch
- EX4300 switches, through uplink ports



**NOTE:** All QSFP+ ports on an EX4300 switch are configured as VCPs by default.

- EX4500 switches, through any SFP+ port
- EX4550 switches, through any SFP+ port
- EX4600 switches, through SFP+ and QSFP+ ports.

All supported SFP, SFP+, and XFP uplink connections between EX4200, EX4500, and EX4550 switches can be configured as VCPs.

You can increase the Virtual Chassis bandwidth between member switches by configuring multiple optical ports connecting the switches as VCPs. The optical ports configured as VCPs automatically form a Link Aggregation Group (LAG) bundle. See [“Understanding EX Series Virtual Chassis Port Link Aggregation” on page 21](#).

## Master Role

The member that functions in the master role in the Virtual Chassis:

- Manages the member switches.
- Runs Junos OS for EX Series switches in a master role.
- Runs the chassis management processes and control protocols.
- Represents all the member switches interconnected within the Virtual Chassis configuration. (The hostname and other properties that you assign to this switch during setup apply to all members of the Virtual Chassis configuration.)

When an EX Series switch that supports Virtual Chassis is powered on as a standalone switch, it is considered the master member. In a Virtual Chassis, one member functions as the master and a second member functions as the backup:

- In a preprovisioned configuration, one of the two members assigned as **routing-engine** functions as the master member. The selection of which member assigned as **routing-engine** functions as master and which as backup is determined by the software based on the master election algorithm. See [“Understanding How the Master in a Virtual Chassis Is Elected” on page 16](#).
- In a configuration that is not preprovisioned, the selection of the master and backup is determined by the mastership priority value and secondary factors in the master election algorithm.

In a mixed EX4300 and EX4600 Virtual Chassis, an EX4600 switch must assume the master role.

In any mixed Virtual Chassis configuration that includes EX4200 switches, EX4500 switches, or EX4550 switches, any switch can be configured in any role in any configuration.

## Backup Role

The member that functions in the backup role in the Virtual Chassis:

- Maintains a state of readiness to take over the master role if the master fails.
- Runs Junos OS for EX Series switches in a backup role.



- 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.

You must have at least two member switches in the Virtual Chassis configuration in order to have a backup member.

- In a preprovisioned configuration, one of the two members assigned as **routing-engine** functions in the backup role. The selection of which member assigned as **routing-engine** functions as master and which as backup is determined by the software based on the master election algorithm. See ["Understanding How the Master in a Virtual Chassis Is Elected" on page 16](#).
- In a configuration that is not preprovisioned, the selection of the master and backup is determined by the mastership priority value and secondary factors in the master election algorithm.

In a mixed EX4300 and EX4600 Virtual Chassis, we strongly recommend configuring an EX4600 switch into the backup role. A mixed EX4300 and EX4600 Virtual Chassis must use an EX4600 member switch in the master role, and configuring an EX4600 switch into the backup role ensures that the Virtual Chassis remains up after a switchover event.

In any mixed Virtual Chassis configuration that includes EX4200 switches, EX4500 switches, or EX4550 switches, any switch can be configured in any role in any configuration.

## Linecard Role

A member that functions in the linecard role in the Virtual Chassis:

- Runs only a subset of Junos OS for EX Series switches.
- Does not run the chassis control protocols.
- Can detect certain error conditions (such as an unplugged cable) on any interfaces that have been configured on it through the master.

The Virtual Chassis configuration must have at least three members in order to include a linecard member.

- In a preprovisioned configuration, you can explicitly configure a member with the linecard role, which makes it ineligible for functioning as a master or backup.
- In a configuration that is not preprovisioned, the members that are not selected as master or backup function as linecard members of the Virtual Chassis configuration. The selection of the master and backup is determined by the mastership priority value and secondary factors in the master election algorithm. A switch with a mastership priority of 0 is always in the linecard role.

## Member Switch and Member ID

Each standalone EX Series switch that supports Virtual Chassis is a potential member of a Virtual Chassis configuration. When one of those switches is powered on, it receives

a member ID that can be seen by viewing the front-panel LCD or by entering the **show virtual-chassis** command. If the switch is powered on as a standalone switch, that member's member ID is always **0**. When the switch is interconnected with other switches in a Virtual Chassis configuration, its member ID is assigned by the master based on various factors, such as the order in which the switch was added to the Virtual Chassis configuration or the member ID assigned by a preprovisioned configuration. See [“Understanding How the Master in a Virtual Chassis Is Elected”](#) on page 16.

If the Virtual Chassis configuration previously included a member switch and that member was physically disconnected or removed from the Virtual Chassis configuration, its member ID is not available for assignment as part of the standard sequential assignment by the master. For example, you might have a Virtual Chassis configuration composed of member 0, member 2, and member 3, because member 1 was removed. When you add another member switch and power it on, the master assigns it as member 4.

The member ID distinguishes the member switches from one another. You use the member ID:

- To assign a mastership priority value to a member switch
- To configure interfaces for a member switch (The function is similar to that of a slot number on Juniper Networks routers.)
- To apply some operational commands to a member switch
- To display status or characteristics of a member switch

## Mastership Priority

In a configuration that is not preprovisioned, you can designate the role (master, backup, or linecard) that a member switch assumes by configuring its mastership priority (from **0** through **255**). The mastership priority value is the factor in the master election algorithm with the highest precedence for selecting the master of the Virtual Chassis configuration. A switch with a mastership priority of **0** never assumes the backup or master role.

The default value for mastership priority is **128** for EX2200, EX3300, EX4200, EX4500, EX4550, and EX4600 switches. When a standalone switch is powered on, it receives the default mastership priority value. Because it is the only member of the Virtual Chassis configuration, it is also the master. When you interconnect a standalone switch to an existing Virtual Chassis configuration (which implicitly includes its own master), we recommend that you explicitly configure the mastership priority of the members that you want to function as the master and backup.



**NOTE:** Configuring the same mastership priority value for both the master and backup helps to ensure a smooth transition from master to backup when the master becomes unavailable. It prevents the original master from preempting control from the backup when the backup has taken control of the Virtual Chassis configuration because the original master became unavailable.

---

In a preprovisioned configuration, you assign the role of each member switch. An EX9200 switch does not have a mastership priority and an EX9200 Virtual Chassis, therefore, must be preprovisioned.

## Mixed Virtual Chassis

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

EX4300 and EX4600 switches can act as member switches in a mixed EX4300 and EX4600 Virtual Chassis. An EX4600 switch must assume the master role in a mixed EX4300 and EX4600 Virtual Chassis. We strongly recommend configuring EX4600 switches only into the Routing Engine role when you are configuring a mixed EX4300 and EX4600 Virtual Chassis to ensure only EX4600 switches assume the master role.

EX2200 and EX3300 switches cannot be a part of any mixed Virtual Chassis.

## Virtual Chassis Identifier (VCID)

All members of a Virtual Chassis configuration share one Virtual Chassis identifier (VCID). This identifier is derived from internal parameters. When you are monitoring a Virtual Chassis configuration, the VCID is displayed in certain interface views and is also part of the **show virtual-chassis** output.

### Related Documentation

- *Understanding EX8200 Virtual Chassis Components*
- *EX Series Virtual Chassis Overview*
- *Understanding EX4300 Virtual Chassis*
- *Example: Configuring an EX4200 Virtual Chassis with a Master and Backup in a Single Wiring Closet*
- *Example: Configuring an EX4500 Virtual Chassis with a Master and Backup in a Single Wiring Closet*
- *Example: Configuring an EX4200 Virtual Chassis Using a Preprovisioned Configuration File*
- [Setting an Uplink Port on an EX Series Switch as a Virtual Chassis Port \(CLI Procedure\) on page 43](#)
- *Setting an Uplink Port as a Virtual Chassis Port on an EX4500 or EX4550 Switch (CLI Procedure)*

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

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

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

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



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

This topic covers:

- [Virtual Chassis Fabric Summary on page 12](#)
- [Understanding Mixed Virtual Chassis Fabric on page 13](#)
- [Virtual Chassis Summary for QFX5100, QFX3600, QFX3500, EX4600, and EX4300 Switches on page 13](#)
- [Understanding the Routing Engine Role in a Mixed Virtual Chassis Using EX4300, EX4600, QFX3500, QFX3600, or QFX5100 Member Switches on page 14](#)
- [Understanding EX4300, QFX3500, QFX3600, and QFX5100 Switches in a Virtual Chassis on page 15](#)
- [Understanding Mixed EX4300 and EX4600 Virtual Chassis on page 15](#)
- [Understanding EX4200, EX4500, and EX4550 Switches in a Mixed Virtual Chassis on page 15](#)

### Virtual Chassis Fabric Summary

[Table 4 on page 13](#) provides a high-level overview of the permitted hardware allowed in the routing engine and line card roles of a mixed and a non-mixed VCF. The table also includes license requirements and supported configuration methods.

Table 4: Virtual Chassis Fabric Summary

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

## Understanding Mixed Virtual Chassis Fabric

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

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

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

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

Table 5 on page 14 provides a high-level overview of the permitted hardware allowed in the routing engine and line card roles of a mixed and a non-mixed Virtual Chassis for QFX5100, QFX3600, QFX3500, EX4600, and EX4300 switches. The table also includes license requirements and supported configuration methods.

Table 5: Virtual Chassis Summary

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

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

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

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

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

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

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

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

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

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

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

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

### Understanding Mixed EX4300 and EX4600 Virtual Chassis

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

The mixed Virtual Chassis supports up to ten member switches.

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

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

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

#### **Related Documentation**

- *Virtual Chassis Fabric Overview*
- *Understanding QFX Series Virtual Chassis*
- *EX Series Virtual Chassis Overview*
- *Understanding Virtual Chassis Fabric Components*

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

---

## Understanding How the Master in a Virtual Chassis Is Elected

---



**NOTE:** This topic does not apply to EX8200 Virtual Chassis. See *EX8200 Virtual Chassis Overview*.

---

All switches that are interconnected in a Virtual Chassis configuration are member switches of that Virtual Chassis. Each Virtual Chassis configuration has one member that functions as the *master* and controls the Virtual Chassis configuration.

When a Virtual Chassis configuration boots, the Juniper Networks Junos operating system (Junos OS) on the switches automatically runs a master election algorithm to determine which member switch assumes the role of master.

The algorithm proceeds from the top condition downward until the stated condition is satisfied:

1. Choose the member with the highest user-configured mastership priority (255 is the highest possible value). A switch with a mastership priority of 0 will always stay in the linecard role.
2. Choose the member that was master the last time the Virtual Chassis configuration booted.
3. Choose the member that has been included in the Virtual Chassis configuration for the longest period of time. (For this to be a deciding factor, there has to be a minimum time lapse of 1 minute between the power-ons of the individual interconnected member switches.)
4. Choose the member with the lowest MAC address.

The variations among switches and switch models do not impact the master election algorithm.

To ensure that a specific member is elected as the master:

1. Power on only the switch that you want to configure as master of the Virtual Chassis configuration.
2. Configure the mastership priority of that member to have the highest possible value (255).
3. Continue to configure other members through the master member.
4. Power on the other members.

You can also specify the switch roles by preprovisioning your Virtual Chassis. Preprovisioning a Virtual Chassis allows you to manually assign the member ID and role for each switch in the Virtual Chassis. See *Configuring an EX3300 Virtual Chassis (CLI*



*Procedure*), *Configuring an EX4200, EX4500, or EX4550 Virtual Chassis (CLI Procedure)*, or *Configuring a QFX Series Virtual Chassis (CLI Procedure)*.

**Related Documentation**

- *EX8200 Virtual Chassis Overview*
- *EX Series Virtual Chassis Overview*
- *Understanding QFX Series Virtual Chassis*
- [Understanding EX Series Virtual Chassis Components on page 5](#)
- *Understanding QFX Series Virtual Chassis Components*
- *Understanding EX Series Virtual Chassis Configuration*

## Understanding Software Upgrade in an EX Series Virtual Chassis



**NOTE:** This topic applies to all EX Series Virtual Chassis except EX8200 Virtual Chassis. See *Understanding Software Upgrades in an EX8200 Virtual Chassis* for information about EX8200 Virtual Chassis.

In a Virtual Chassis, each member switch must be running the same version of Juniper Networks Junos operating system (Junos OS).

You can install a new Junos OS release on the entire Virtual Chassis or on a particular member in the Virtual Chassis by using the same CLI command that you use to install Junos OS on standalone switches—the **request system software add** command.

You can use the automatic software update feature to automatically update the Junos OS version on member switches as you add them to a Virtual Chassis. See [“Understanding Automatic Software Update on Virtual Chassis Member Switches” on page 26](#). If you are not configuring the automatic software update feature, we recommend that you update the new member switch to the version of Junos OS running on the Virtual Chassis before adding the member switch to the Virtual Chassis.

In any mixed Virtual Chassis, the member switches must be running the same version of Junos OS. You can upgrade all member switches simultaneously by specifying a path to multiple Junos OS images in the same **request system software add** command. Multiple Junos OS images are needed to upgrade a mixed Virtual Chassis because an EX4200 switch runs a different version of Junos OS than an EX4500 or EX4550 switch. See *Installing Software on a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Switches (CLI Procedure)*.



**NOTE:** EX4500 and EX4550 switches run the same Junos OS image. You do not need to use multiple Junos OS images when updating a mixed EX4500 and EX4550 Virtual Chassis.

You can also use nonstop software upgrade (NSSU) to upgrade Junos OS on all members. NSSU provides an orderly upgrade of each member of the Virtual Chassis and takes

advantage of graceful Routing Engine switchover, nonstop active routing, and link aggregation to minimize traffic disruption during the upgrade. For more information about NSSU, see *Understanding Nonstop Software Upgrade on EX Series Switches*.

**Related  
Documentation**

- [Understanding Software Upgrades in an EX8200 Virtual Chassis](#)
- [Understanding EX Series Virtual Chassis Components on page 5](#)
- [Understanding Automatic Software Update on Virtual Chassis Member Switches on page 26](#)
- *Installing Software on an EX Series Switch with a Single Routing Engine (CLI Procedure)*
- *Upgrading Software on an EX3300, EX4200, EX4300, EX4500 and EX4550 Virtual Chassis, and Mixed Virtual Chassis Using Nonstop Software Upgrade (CLI Procedure)*

---

## Understanding Global Management of a Virtual Chassis

---

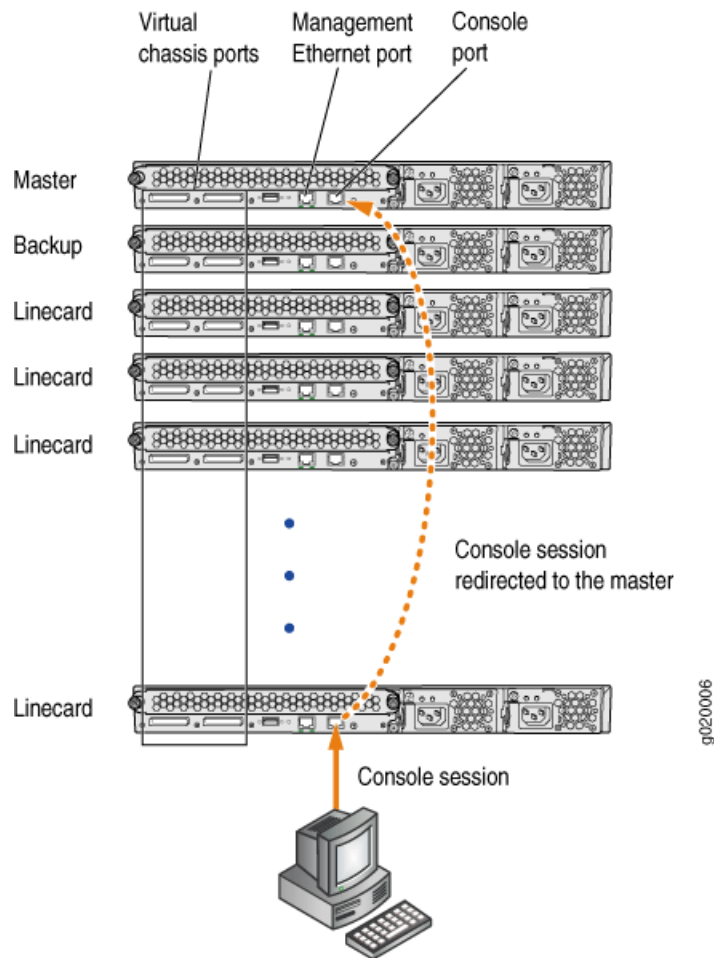


**NOTE:** This topic does not apply to EX8200 Virtual Chassis. See *Understanding Global Management of an EX8200 Virtual Chassis*.

A Virtual Chassis is composed of multiple switches, and it, therefore, has multiple console ports and multiple out-of-band management Ethernet ports located on the switches.

You can connect a PC or laptop directly to a console port of any member switch to set up and configure the Virtual Chassis. When you connect to the console port of any member switch, the console session is redirected to the master switch, as shown in [Figure 1 on page 19](#).

Figure 1: Console Session Redirection (EX4200 Virtual Chassis Pictured)

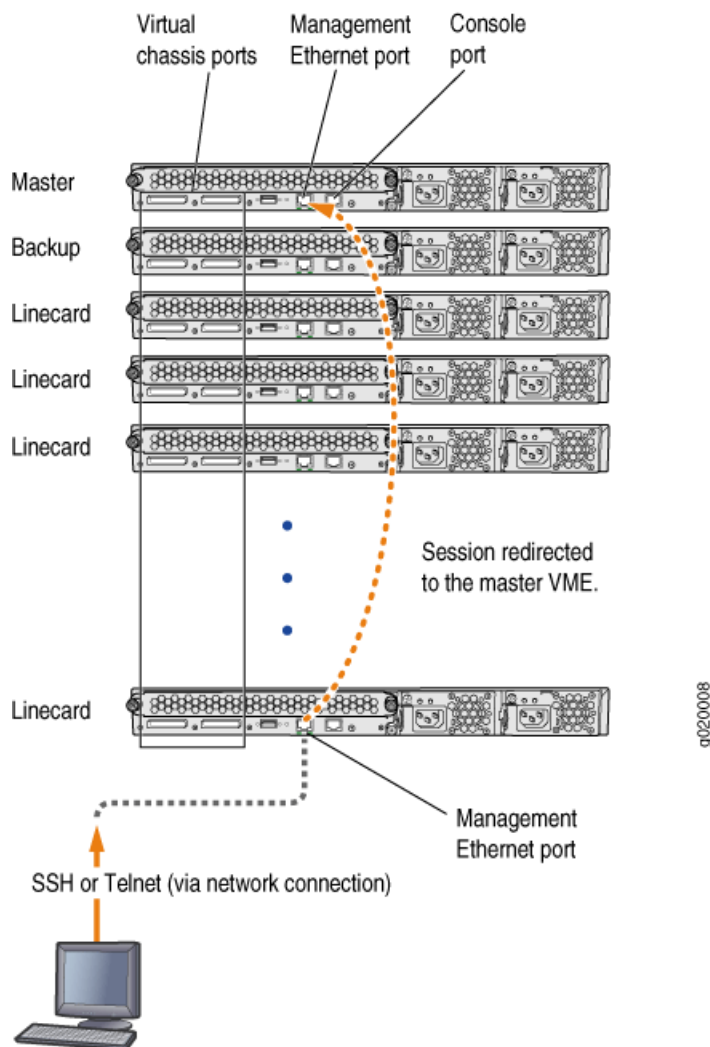


If the master becomes unavailable, the console session is disconnected from the old master and a new session is established with the newly elected master.

An out-of-band management Ethernet port is often referred to simply as a management Ethernet port. It uses a dedicated management channel for device maintenance and allows a system administrator to monitor and manage the switch by remote control.

The Virtual Chassis configuration can be managed remotely through SSH or Telnet 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 switches. When you connect to the Virtual Chassis configuration using the VME interface's IP address, the connection is redirected to the master member as shown in [Figure 2 on page 20](#).

Figure 2: Management Ethernet Port Redirection to the VME Interface



If the master management Ethernet link is unavailable, the session is redirected through the backup management Ethernet link. If there is no active management Ethernet link on the backup, the VME interface chooses a management Ethernet link on one of the linecard members, selecting the linecard member with the lowest member ID as its first choice.

You can configure an IP address for the VME global management interface at any time.

You can perform remote configuration and administration of all members of the Virtual Chassis configuration through the VME interface.

#### Related Documentation

- [Understanding Global Management of an EX8200 Virtual Chassis](#)
- [Understanding EX Series Virtual Chassis Components on page 5](#)
- [Understanding QFX Series Virtual Chassis Components](#)

- *Example: Configuring an EX4200 Virtual Chassis with a Master and Backup in a Single Wiring Closet*
- *Configuring the Virtual Management Ethernet Interface for Global Management of an EX Series Virtual Chassis (CLI Procedure)*

## Understanding Nonvolatile Storage in a Virtual Chassis

---



**NOTE:** This topic applies to all EX Series Virtual Chassis except EX8200 Virtual Chassis. See *Understanding File Storage in an EX8200 Virtual Chassis* for information about EX8200 Virtual Chassis.

The EX Series or QFX Series switches store the Juniper Networks Junos operating system (Junos OS) system files in internal flash memory. In the Virtual Chassis configurations, both the master and the backup switch store the configuration information for all the member switches.

- [Nonvolatile Memory Features on page 21](#)

### Nonvolatile Memory Features

Junos OS optimizes the way the Virtual Chassis stores its configuration if a member switch or the Virtual Chassis configuration is shut down improperly:

- If the master is not available, the backup switch takes on the role of the master and its internal flash memory takes over as the alternate location for maintaining nonvolatile configuration memory.
- If a member switch is taken offline for repair, the master stores the configuration of the member switch.

#### Related Documentation

- [Understanding File Storage in an EX8200 Virtual Chassis](#)
- [Understanding QFX Series Virtual Chassis](#)
- [Command Forwarding Usage with an EX Series Virtual Chassis on page 49](#)

## Understanding EX Series Virtual Chassis Port Link Aggregation

---



**NOTE:** This topic applies to all EX Series Virtual Chassis except EX8200 Virtual Chassis. See *Understanding Virtual Chassis Port Link Aggregation in an EX8200 Virtual Chassis* for information about EX8200 Virtual Chassis.

You can combine physical Ethernet ports belonging to different member switches of a Virtual Chassis configuration to form a logical point-to-point link, known as a *link aggregation group (LAG)* or *bundle*. A LAG provides more bandwidth than a single Ethernet link can provide. Additionally, link aggregation provides network redundancy by

load-balancing traffic across all available links. If one of the links fails, the system automatically load-balances traffic across all remaining links.

Similarly, if a Virtual Chassis member switch that has LAG member interfaces on multiple member switches fails for any reason, the traffic traversing the LAG can be redirected through the active member switch. This setup has benefits for failover purposes and can be especially beneficial in cases when a member switch needs to be inactive for some time, such as during a software upgrade using NSSU.

You can configure any optical uplink port that can be used to connect EX2200, EX3300, EX4200, EX4300, EX4500, or EX4550 switches together into a Virtual Chassis port (VCP). You can configure multiple optical uplink interfaces between two member switches in the same Virtual Chassis as VCPs. If you have configured two or more optical ports as VCPs connecting the same member switches, the optical uplink ports configured as VCPs automatically form a LAG provided the optical uplink ports are configured to operate at the same link speeds. Each LAG is assigned a positive-integer identifier called a *trunk ID*.

On EX2200 and EX2200-C switches only, you can also configure the RJ-45 interfaces, including built-in network ports with 10/100/1000BASE-T Gigabit Ethernet connectors and 1000BASE-T RJ-45 transceivers, into VCPs. On EX2200 and EX2200-C switches, a LAG that includes up to 8 interfaces configured as VCPs automatically forms. The LAG bundles includes all interfaces configured as VCPs, regardless of whether the interfaces are optical transceiver interfaces, RJ-45 transceiver interfaces, or built-in network ports with 10/100/1000BASE-T Gigabit Ethernet connectors.

You can create an optical VCP LAG connecting any two member switches in any Virtual Chassis, including VCP LAG connections interconnecting different switch models in a mixed Virtual Chassis.

On an EX2200 switch only, you can also configure all RJ-45 interfaces, including built-in network ports with 10/100/1000BASE-T Gigabit Ethernet connectors and 1000BASE-T RJ-45 transceivers, on EX2200 and EX2200-C switches, into VCPs. The RJ-45 interfaces also automatically form a LAG when configured into VCPs.

[Table 6 on page 22](#) provides the maximum member link limit for each optical VCP LAG.

**Table 6: Maximum Member Links in LAGs Over Optical Interface VCPs**

Member Switch 1	Member Switch 2	Maximum Member Links in VCP LAG
EX2200 Switch	EX2200 Switch	8
EX3300 Switch	EX3300 Switch	8
EX4200 Switch	EX4200 Switch	8
EX4200 Switch	EX4500 Switch	8
EX4200 Switch	EX4550 Switch	8
EX4500 Switch	EX4500 Switch	8

**Table 6: Maximum Member Links in LAGs Over Optical Interface VCPs (*continued*)**

Member Switch 1	Member Switch 2	Maximum Member Links in VCP LAG
EX4500 Switch	EX4550 Switch	8
EX4550 Switch	EX4550 Switch	8

A LAG over uplink VCPs provides higher overall bandwidth for forwarding traffic between the member switches connected by the optical VCPs, faster management communications, and greater redundancy of operations among the members than would be available without the LAG. A LAG over uplink VCPs provides an additional Virtual Chassis link throughput for the switches.

See [“Setting an Uplink Port on an EX Series Switch as a Virtual Chassis Port \(CLI Procedure\)” on page 43](#) for information about configuring uplink ports as VCPs.



**NOTE:** The interfaces that are included within a bundle or LAG are sometimes referred to as *member interfaces*. Do not confuse this term with *member switches*, which refers to switches that are interconnected as a Virtual Chassis. It is possible to create a LAG that is composed of member interfaces that are located in different member switches of a Virtual Chassis.

#### Related Documentation

- [Understanding Virtual Chassis Port Link Aggregation in an EX8200 Virtual Chassis](#)
- [EX Series Virtual Chassis Overview](#)
- [Understanding Aggregated Ethernet Interfaces and LACP](#)
- [Example: Configuring Aggregated Ethernet High-Speed Uplinks Between an EX4200 Virtual Chassis Access Switch and an EX4200 Virtual Chassis Distribution Switch](#)
- [Example: Configuring Aggregated Ethernet High-Speed Uplinks with LACP Between an EX4200 Virtual Chassis Access Switch and an EX4200 Virtual Chassis Distribution Switch](#)
- [Example: Configuring an EX4200 Virtual Chassis Interconnected Across Multiple Wiring Closets](#)
- [Example: Connecting EX4500 Member Switches in a Virtual Chassis Across Wiring Closets](#)
- [Example: Configuring Link Aggregation Groups Using EX4200 Uplink Virtual Chassis Ports](#)

## Understanding Split and Merge in a Virtual Chassis

In a Virtual Chassis, two or more switches are connected together to form a unit that is managed as a single chassis. If there is a disruption to the Virtual Chassis configuration due to member switches failing or being removed from the configuration, the Virtual Chassis configuration splits into two separate Virtual Chassis. This situation could cause disruptions in the network if the two separate configurations share common resources, such as global IP addresses. The split and merge feature provides a method to prevent

the separate Virtual Chassis configurations from adversely affecting the network and also allows the two parts to merge back into a single Virtual Chassis configuration.



**NOTE:** If a Virtual Chassis configuration splits into separate parts, we recommend that you resolve the problem that caused the Virtual Chassis configuration to split as soon as possible.

You can also use this feature to merge two active but separate Virtual Chassis that have not previously been part of the same configuration into one Virtual Chassis configuration.



**NOTE:** The split and merge feature is enabled by default on EX Series and QFX Series Virtual Chassis. You can disable the split and merge feature by using the `set virtual-chassis no-split-detection` command.

This topic describes:

- [What Happens When a Virtual Chassis Configuration Splits on page 24](#)
- [Merging Virtual Chassis Configurations on page 25](#)

## What Happens When a Virtual Chassis Configuration Splits

When a Virtual Chassis configuration splits into two separate Virtual Chassis configurations, the individual member switches detect this topology change and run the master election algorithm to select a new master for each of the two Virtual Chassis configurations. The new masters then determine whether their Virtual Chassis configuration remains active. One of the configurations remains active based on the following:

- It contains both the stable master and the stable backup (that is, the master and backup from the original Virtual Chassis configuration before the split).
- It contains the stable master and the configuration is greater than half the Virtual Chassis size.
- It contains the stable backup and is at least half the Virtual Chassis size.

In accordance with the rules given in the second and third list items, if the Virtual Chassis configuration splits into two equal parts and the stable master and stable backup are in different parts, then the part that contains the stable backup becomes active.



**NOTE:** The number of members in the Virtual Chassis configuration includes all member switches connected to date minus the number whose Virtual Chassis member IDs have been recycled (that is, made available for reassignment). Therefore, the size of the Virtual Chassis configuration increases when a new member switch is detected and decreases when a member switch's ID is recycled.



These rules ensure that only one of the two separate Virtual Chassis configurations created by the split remains active. The member switches in the inactive Virtual Chassis configuration remain in a linecard role. For the inactive members to become active again, one of the following things must happen:

- The problem that caused the original Virtual Chassis configuration to split is resolved, allowing the two Virtual Chassis configurations to merge.
- You load the factory default configuration on the inactive members, which causes the inactive members to function as standalone switches or become part of a different Virtual Chassis configuration.



**NOTE:** When you remove a member switch from a Virtual Chassis configuration, we recommend that you recycle the member ID using the `request virtual-chassis recycle` command.

## Merging Virtual Chassis Configurations

There are two scenarios in which separate Virtual Chassis merge:

- A Virtual Chassis configuration that had split into two is now merging back into a single configuration because the problem that had caused it to split has been resolved.
- You want to merge two Virtual Chassis that had not previously been configured together.

Every Virtual Chassis configuration has a unique ID (VCID) that is automatically assigned when the Virtual Chassis configuration is formed. You can also explicitly assign a VCID using the `set virtual-chassis id` command. A VCID that you assign takes precedence over automatically assigned VCIDs.

When you reconnect the separate Virtual Chassis configurations or connect them for the first time, the members determine whether or not the separate Virtual Chassis configurations can merge. The members use the following rules to determine whether a merge is possible:

- If the Virtual Chassis configurations have the same VCID, then the configurations can merge. If the two Virtual Chassis were formed as the result of a split, they have the same VCID.
- If the VCIDs are different, then the two configurations can merge only if both are active (inactive configurations cannot merge, ensuring that members removed from one Virtual Chassis configuration do not become members of another Virtual Chassis configuration). If the configurations to merge are both active and one of them has a user-configured VCID, this ID becomes the ID of the merged Virtual Chassis. If neither Virtual Chassis has a user-configured VCID, then the VCID of the configuration with the highest mastership priority becomes the ID of the merged Virtual Chassis. The resulting merged Virtual Chassis configuration is active.

When you connect two Virtual Chassis configurations, the following events occur:

1. Connecting the two split Virtual Chassis configurations triggers the shortest-path-first (SPF) algorithm. The SPF algorithm computes the network topology and then triggers the master election algorithm. The master election algorithm waits for the members to synchronize the topology information before running.
2. The master election algorithm merges the VCIDs of all the members.
3. Each member runs the master election algorithm to select a master and a backup from among all members with the same VCIDs. For more information, see [“Understanding How the Master in a Virtual Chassis Is Elected”](#) on page 16.
4. The master determines whether the Virtual Chassis configuration is active or inactive. (See [“What Happens When a Virtual Chassis Configuration Splits”](#) on page 24.)
5. If the Virtual Chassis configuration is active, the master assigns roles to all members. If the Virtual Chassis configuration is inactive, the master assigns all members the role of linecard.
6. When the other members receive their role from the master, they change their role to backup or linecard. They also use the active or inactive state information sent by the master to set their own state to active or inactive and to construct the Virtual Chassis member list from the information sent by the master.
7. If the Virtual Chassis state is active, the master waits for messages from the members indicating that they have changed their roles to the assigned roles, and then the master changes its own role to master.



**NOTE:** When you merge two Virtual Chassis that had not previously been part of the same Virtual Chassis configuration, any configuration settings (such as the settings for Telnet and FTP services, graceful Routing Engine switchover (GRES), fast failover, VLANs, and so on) that exist on the new master become the configuration settings for all members of the new Virtual Chassis, overwriting any other configuration settings.

**Related  
Documentation**

- [Disabling Split and Merge in a Virtual Chassis \(CLI Procedure\)](#) on page 47
- [Assigning the Virtual Chassis ID to Determine Precedence During a Virtual Chassis Merge \(CLI Procedure\)](#) on page 49
- *Example: Assigning the Virtual Chassis ID to Determine Precedence During an EX4200 Virtual Chassis Merge*
- [Understanding EX Series Virtual Chassis Configuration](#)
- [Understanding QFX Series Virtual Chassis](#)

---

## Understanding Automatic Software Update on Virtual Chassis Member Switches

You can use the automatic software update feature to automatically update the Juniper Networks Junos operating system (Junos OS) version on prospective member switches as you add them to an EX Series or QFX Series Virtual Chassis.

This topic includes:

- [Automatic Software Update Basics on page 27](#)
- [Automatic Software Update Restrictions on page 27](#)

## Automatic Software Update Basics

When you have configured automatic software update on a Virtual Chassis, the Junos OS version is updated on the new member switch when you add it to the Virtual Chassis. The new member switch immediately joins the Virtual Chassis configuration and is put in the active state.

For a standalone switch to join an existing Virtual Chassis, it must be running the same version of Junos OS that is running on the Virtual Chassis master. When the master in a Virtual Chassis detects that a new switch has been added to the configuration, it checks the software version on the new switch. If the software version on the new switch is not the same as the version running on the master, the master keeps the new switch in the inactive state. If you have not enabled the automatic software update feature, you have to manually install the correct software version on each prospective member switch as it is added to the Virtual Chassis.

## Automatic Software Update Restrictions

You cannot use automatic software update in certain scenarios, and you must ensure that the software release version on the Virtual Chassis is supported by the release on the prospective member switch.

You cannot use the automatic software update feature to update software for a prospective member switch in the following scenarios:

- The Virtual Chassis was preprovisioned and is running Junos OS Release 10.4R2 or earlier.
- You configured the **mastership-priority** command to manually configure the mastership priority of at least one Virtual Chassis member switch and the Virtual Chassis was running Junos OS Release 10.4R2 or earlier when you committed this configuration.
- The Junos OS versions on the Virtual Chassis and the prospective member switch are different versions of the same major Junos OS release. For instance, if a Virtual Chassis is running Junos OS Release 10.4R1, the prospective member switch cannot be updated using automatic software update if it is running Junos OS Release 10.4R2, 10.4R3, or any other Junos OS Release 10.4 release version.

The automatic software update feature also has a Junos OS release dependency between the release that is already running on the Virtual Chassis and the release that is running on the prospective member switch.

[Table 7 on page 28](#) summarizes automatic software update support for each Junos OS release combination.

Table 7: Automatic Software Update Support

Virtual Chassis Junos OS Release	Supported Junos OS Releases for Prospective Member Switches
All versions of Junos OS 9.0 through 9.6	All versions of Junos OS 9.0 through 9.6 Junos OS Releases 10.0R1 through 10.0R4 All versions of Junos OS Release 10.1 Junos OS Releases 10.2R1 through 10.2R3 Junos OS Releases 10.3R1 through 10.3R3
Junos OS Releases 10.0R1 through 10.0R4	All versions of Junos OS 9.0 through 9.6 All versions of Junos OS Release 10.1 Junos OS Releases 10.2R1 through 10.2R3 Junos OS Releases 10.3R1 through 10.3R3
Junos OS Release 10.0R5 and later 10.0 releases	Junos OS Release 10.2R4 and later 10.2 releases Junos OS Release 10.3R4 and later 10.3 releases All versions of Junos OS Release 10.4 All versions of Junos OS Release 11.1
All versions of Junos OS Release 10.1	All versions of Junos OS 9.0 through 9.6 Junos OS Releases 10.0R1 through 10.0R4 Junos OS Releases 10.2R1 through 10.2R3 Junos OS Releases 10.3R1 through 10.3R3
Junos OS Releases 10.2R1 through 10.2R3	All versions of Junos OS 9.0 through 9.6 Junos OS Releases 10.0R1 through 10.0R4 All versions of Junos OS Release 10.1 Junos OS Releases 10.3R1 through 10.3R3
Junos OS Release 10.2R4 and later 10.2 releases	Junos OS Release 10.0R5 Junos OS Release 10.3R4 and later 10.3 releases All versions of Junos OS Release 10.4 All versions of Junos OS Release 11.1
Junos OS Releases 10.3R1 through 10.3R3	All versions of Junos OS 9.0 through 9.6 Junos OS Releases 10.0R1 through 10.0R4 All versions of Junos OS Release 10.1 Junos OS Releases 10.2R1 through 10.2R3
Junos OS Release 10.3R4 and later 10.3 releases	Junos OS Release 10.0R5 All versions of Junos OS Release 10.4 All versions of Junos OS Release 11.1
Junos OS Releases 10.4R1 through 10.4R3	All versions of Junos OS 9.0 through 9.6 Junos OS Releases 10.0R1 through 10.0R4 All versions of Junos OS Release 10.1 Junos OS Releases 10.2R1 through 10.2R3 Junos OS Releases 10.3R1 through 10.3R3
Junos OS Release 10.4R4 and later 10.4 releases	Junos OS Release 10.0R5 Junos OS Release 10.2R4 and later 10.2 releases Junos OS Release 10.3R4 and later 10.3 releases All versions of Junos OS Release 11.1

Table 7: Automatic Software Update Support (*continued*)

Virtual Chassis Junos OS Release	Supported Junos OS Releases for Prospective Member Switches
Junos OS Release 11.1R1	All versions of Junos OS Release 10.4 Junos OS Release 11.2 and later Junos OS releases
Junos OS Release 11.1R2 and later Junos OS releases	Junos OS Release 10.0R5 Junos OS Release 10.2R4 and later 10.2 releases Junos OS Release 10.3R4 and later 10.3 releases Junos OS Release 11.2 and later Junos OS releases

**Related Documentation**

- [Understanding Software Upgrade in an EX Series Virtual Chassis on page 17](#)
- [Understanding Software Upgrades in a QFX Series Virtual Chassis](#)
- [Example: Configuring Automatic Software Update on EX4200 Virtual Chassis Member Switches](#)
- [Configuring Automatic Software Update on Virtual Chassis Member Switches \(CLI Procedure\) on page 48](#)

## Understanding MAC Address Assignment on a Virtual Chassis

In a Virtual Chassis, multiple switches—each with its own set of interfaces with unique MAC addresses—are connected together to form one chassis that can be managed as a single switch. The MAC address assigned to each network-facing interface on the switch changes when the switch joins a Virtual Chassis. Because all Layer 2 traffic decisions are based on an interface's MAC address, understanding MAC address assignment is important to understanding how network traffic is forwarded and received by the Virtual Chassis. For additional information about how a network uses MAC addresses to forward and receive traffic, see *Understanding Bridging and VLANs on EX Series Switches*.

When a Virtual Chassis is formed, the MAC address of the switch in the master role becomes the system MAC base address. The Virtual Chassis assigns the system MAC base address as the MAC address for all Layer 3 interfaces within the Virtual Chassis. The Virtual Chassis also assigns the system MAC base address to the virtual management Ethernet (VME) interface and to all of the virtual LANs (VLANs) in the Virtual Chassis.

The system MAC base address does not change in the event of a switchover if the switch that was originally configured in the master role remains a member of the Virtual Chassis. If the switch that was originally configured in the master role is removed from the Virtual Chassis, the MAC address of the current member switch in the master role is assigned as the system MAC base address after the MAC persistence timer interval has expired. You can configure the MAC persistence timer interval.

For Layer 2 and aggregated Ethernet interfaces, the Virtual Chassis assigns a unique MAC address that is derived from the member switch MAC address to each interface. The assignment of a unique MAC address to each network interface helps ensure that functions that require MAC address differentiation—such as redundant trunk groups

(RTGs), Link Aggregation Control Protocol (LACP), and general monitoring functions—can function properly.



**NOTE:** Unique MAC address assignment for Layer 2 and aggregated Ethernet interfaces in a Virtual Chassis was introduced in Junos OS Release 11.3. The same MAC address could be assigned to interfaces on different member switches in the same Virtual Chassis prior to this release.

If you reconfigure a Layer 2 interface into a Layer 3 interface, or the reverse, within a Virtual Chassis, the MAC address of that interface changes accordingly.

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

**Related  
Documentation**

- *Understanding MAC Address Assignment in an EX Series Switch*
- [Configuring the Timer for the Backup Member to Start Using Its Own MAC Address, as Master of a Virtual Chassis \(CLI Procedure\) on page 43](#)
- *EX Series Virtual Chassis Overview*
- *EX8200 Virtual Chassis Overview*
- *Understanding QFX Series Virtual Chassis*

---

## Understanding High Availability on an EX Series Virtual Chassis

---

You increase your network's high availability (HA) when you interconnect a Juniper Networks EX Series Ethernet switch into a Virtual Chassis. A Virtual Chassis is more fault tolerant than a standalone EX series switch because it remains up when a single member switch fails.

You can further improve HA by configuring the HA features available for your EX Series Virtual Chassis. You can, for instance, configure Link Aggregation Groups (LAG) bundles to include member links on multiple member switches in the same Virtual Chassis. This configuration increases fault tolerance because traffic traversing the LAG can be redirected to an active member switch when a single member switch fails.

A Virtual Chassis has dual Routing Engines—the switch in the master role and the switch in the backup role—and therefore supports many HA features not supported on standalone EX Series switches. For a complete list of High Availability features available for your EX Series Virtual Chassis, see *EX Series Virtual Chassis Software Features Overview*.

Many HA features for the EX Series Virtual Chassis are designed to improve network resiliency after a Routing Engine switchover. [Table 8 on page 31](#) describes the effects of a Routing Engine switchover when no high availability features are enabled and when some High Availability features are enabled.

**Table 8: Effects of a Routing Engine Switchover**

High Availability Feature	Effect of Routing Engine Switchover
No HA features enabled	Kernel and forwarding state information is not preserved to the backup Routing Engine. A convergence process that requires all interfaces on the Virtual Chassis to be taken offline has to be performed before the Virtual Chassis returns online. The switchover can take several minutes and the Virtual Chassis does not send or receive traffic until the switchover is complete.
Graceful Routing Engine switchover (GRES) enabled	Kernel and forwarding state information is preserved on both Routing Engines, so the convergence process does not occur and the switchover happens quickly with minimal traffic loss.
Nonstop active routing (NSR), Nonstop bridging (NSB), or both enabled	<p>Layer 2 protocols that are supported by NSB are not disrupted by a Routing Engine switchover when NSB is enabled. Layer 2 protocol information for all active Layer 2 protocols is stored on both Routing Engines when NSB is enabled.</p> <p>Layer 3 protocols that are supported by NSR are not disrupted by a Routing Engine switchover when NSR is enabled. Layer 3 protocol information for all active Layer 3 protocols is stored on both Routing Engines when NSR is enabled.</p>
Graceful Protocol Restart enabled	Traffic is not interrupted during the switchover. Interface and kernel information is preserved. Graceful restart protocol extensions quickly collect and restore routing information for supported protocols from the neighboring devices.

- Related Documentation**
- [EX Series Virtual Chassis Overview](#)
  - [High Availability Features for EX Series Switches Overview](#)

## Configuring EX4600 Switches in a Mixed or Non-Mixed Virtual Chassis (CLI Procedure)

This topic provides information about configuring a EX4600 Virtual Chassis, or a mixed Virtual Chassis that includes EX4600 switches.

In an EX4600 Virtual Chassis, you interconnect the EX4600 switches by using uplink ports configured as Virtual Chassis ports (VCPs). The 40-Gbps QSFP+ and 10-Gbps SFP+ uplink ports can be configured into VCPs. The uplink ports can be used to connect switches that are several miles apart, so you can connect switches in different buildings into the same Virtual Chassis.

You must configure QSFP+ or SFP+ uplink module ports into VCPs to create an EX4600 Virtual Chassis, or to interconnect EX4300 and EX4600 switches to form a mixed EX4300 and EX4600 Virtual Chassis. If you need additional VCP bandwidth between two member switches, you can configure additional ports as VCPs between the member switches. The ports that have identical speeds become links in a link aggregation group (LAG) to provide resiliency to the Virtual Chassis; for instance, if you had two QSFP+ ports and two SFP+ ports configured as VCPs connecting to the same switch, you would have two LAGs—one LAG with two 40Gbps QSFP+ port member links and another with two 10Gbps SFP+ port member links—between the member switches.



**NOTE:** A Virtual Chassis configuration has two Routing Engines—the master switch and the backup switch. Therefore, we recommend that you always use `commit synchronize` rather than simply `commit` to save configuration changes made for a Virtual Chassis. This ensures that the configuration changes are saved on both Routing Engines.

An EX4600 Virtual Chassis can be configured with either:

- A nonprovisioned configuration—The master sequentially assigns a member ID to other member switches. The role is determined by the mastership priority value and other factors in the master election algorithm.
- A preprovisioned configuration—You can deterministically control the member ID and role assigned to a member switch by tying the member switch to its serial number.

This topic includes:

- [Configuring an EX4600 Virtual Chassis with a Nonprovisioned Configuration File on page 32](#)
- [Configuring an EX4600 Virtual Chassis with a Preprovisioned Configuration File on page 34](#)

## Configuring an EX4600 Virtual Chassis with a Nonprovisioned Configuration File

You can use nonprovisioned configuration to configure an EX4600 Virtual Chassis.

To configure the Virtual Chassis using a nonprovisioned configuration:



**NOTE:** You can configure a EX4600 Virtual Chassis while the cables are or are not physically connected.

1. Power on only the switch that you plan to use as the master switch.

If you are configuring a mixed EX4300 and EX4600 Virtual Chassis, you must use an EX4600 switch. A mixed EX4300 and EX4600 Virtual Chassis that uses an EX4300 in the master routing engine role is an unsupported configuration.

2. (Required for a mixed Virtual Chassis only) Set the master switch into mixed mode, and reboot the switch to complete the configuration:



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

3. If you are configuring a mixed Virtual Chassis, wait for the reboot to complete before performing this step.

Specify the identification parameters for the switch by completing the initial configuration. See *Configuring an EX4600 Switch*.



**NOTE:** The properties that you specify for the master switch apply to the entire Virtual Chassis configuration.

4. (Optional) Configure the master switch with the virtual management Ethernet (VME) interface for out-of-band management of the Virtual Chassis:

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

5. (Required for mixed EX4300 and EX4600 Virtual Chassis. Optional for all other Virtual Chassis) Configure mastership priority for the member switches:



**NOTE:** If you are configuring a mixed EX4300 and EX4600 Virtual Chassis, configure the EX4600 switches with the highest mastership priorities to ensure EX4300 switches do not assume the master role. A mixed EX4300 and EX4600 switch operating with an EX4300 switch in the master role is an unsupported configuration.

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

```
user@switch# set member 0 mastership-priority 255
```

```
user@switch# set member 1 mastership-priority 255
```

6. (Optional. Recommended for a two-member Virtual Chassis) On the master switch, disable the split and merge feature:

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

```
user@switch# set no-split-detection
```

7. Power on the other member switches.

8. (Required for a mixed Virtual Chassis only) Set each individual switch into mixed mode, and reboot the switch to complete the configuration:

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

9. On each individual member switch, configure the ports that will be used to interconnect the member switches into VCPs using the following command:

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

where *pic-slot-number* is the PIC slot number.

For instance, if you wanted to set port 0 on the QSFP+ interface on PIC slot 2 as a VCP:

```
user@switch> request virtual-chassis vc-port set pic-slot 2 port 0 local
```

The VCPs automatically bundle into a Link Aggregation Group when two or more interfaces of the same speed are configured into VCPs between the same two member switches. See [“Understanding EX Series Virtual Chassis Port Link Aggregation” on page 21](#).



**NOTE:** If you do not edit the Virtual Chassis configuration file, a nonprovisioned configuration is generated by default. The mastership priority value for each member switch is 128. The master role is selected by default. You can change the role that is performed by the members by modifying the mastership priority. See “[Configuring Mastership of a Virtual Chassis \(CLI Procedure\)](#)” on page 41. We recommend that you specify the same mastership priority value for the desired master and backup members. In this example, the highest possible mastership priority has been assigned to two members. However, the member that was powered on first has higher prioritization according to the master election algorithm. See “[Understanding How the Master in a Virtual Chassis Is Elected](#)” on page 16. The other members use the default mastership priority in this example, which configures them to function in the role of linecard.



**NOTE:** If you want to change the member ID that the master has assigned to a member switch, use the [request virtual-chassis renumber](#) command.

## Configuring an EX4600 Virtual Chassis with a Preprovisioned Configuration File

Preprovisioning a Virtual Chassis configuration allows you to assign the member ID and role for each switch in the Virtual Chassis.

To configure a Virtual Chassis using a preprovisioned configuration:



**NOTE:** You can configure a EX4600 Virtual Chassis while the cables are or are not physically connected.

1. Make a list of the serial numbers of all the switches to be connected in a Virtual Chassis configuration.
2. Note the desired role (**routing-engine** or **line-card**) of each switch. If you configure the member with a **routing-engine** role, it is eligible to function in the master or backup role. If you configure the member with a **line-card** role, it is not eligible to function in the master or backup role.

If you are configuring a mixed EX4300 and EX4600 Virtual Chassis, EX4300 switches must be configured into the **line-card** role only. A mixed EX4300 and EX4600 Virtual Chassis

3. Power on only the switch that you plan to use as the master switch.
4. (Required for a mixed Virtual Chassis only) Set the master switch into mixed mode, and reboot the switch to complete the configuration:



**NOTE:** You do not have to complete this step if you are configuring a Virtual Chassis that includes QFX3500 and QFX3600 switches only.

You must complete this step if your Virtual Chassis includes a mix of QFX5100, EX4300, and QFX3500 or QFX3600 switches.

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

5. If you are configuring a mixed Virtual Chassis, wait for the reboot to complete before performing this step.

Specify the identification parameters for the switch by completing the initial configuration. See *Configuring an EX4600 Switch*.



**NOTE:** The properties that you specify for the master switch apply to the entire Virtual Chassis configuration.

6. (Optional) Configure the master switch with the virtual management Ethernet (VME) interface for out-of-band management of the Virtual Chassis:

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

7. Specify the preprovisioned configuration mode:

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

```
user@switch# set preprovisioned
```

8. Specify all the members that you want included in the Virtual Chassis, listing each switch's serial number with the desired member ID and role:

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

```
user@switch# set member 0 serial-number abc123 role routing-engine
```

```
user@switch# set member 1 serial-number def456 role routing-engine
```

```
user@switch# set member 2 serial-number ghi789 role line-card
```

```
user@switch# set member 3 serial-number jkl012 role line-card
```

9. (Optional. Recommended for a two-member Virtual Chassis) Disable the split and merge feature:

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

```
user@switch# set no-split-detection
```

10. Power on the other member switches. The member IDs and roles have been determined by the configuration, so you can power on the member switches in any order.

11. (Required if you are configuring a mixed Virtual Chassis) Set each individual switch into mixed mode, and reboot the switch to complete the configuration:

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

12. On each individual member switch, configure the ports that will be used to interconnect the member switches into VCPs using the following command:

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

where *pic-slot-number* is the PIC slot number.

For instance, if you wanted to set port 0 on the QSFP+ interface on PIC slot 2 as a VCP:

```
user@switch> request virtual-chassis vc-port set pic-slot 2 port 0 local
```

The VCPs automatically bundle into a Link Aggregation Group when two or more interfaces of the same speed are configured into VCPs between the same two member switches. See [“Understanding EX Series Virtual Chassis Port Link Aggregation” on page 21](#).



**NOTE:** You cannot modify the mastership priority when you are using a preprovisioned configuration. The mastership priority values are generated automatically and controlled by the role that is assigned to the member switch in the configuration file. The two Routing Engines are assigned the same mastership priority value. However, the member that was powered on first has higher prioritization according to the master election algorithm. See [“Understanding How the Master in a Virtual Chassis Is Elected” on page 16](#).

**Related  
Documentation**

- [Configuring Mastership of a Virtual Chassis \(CLI Procedure\) on page 41](#)
- [Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 57](#)

---

## Adding an EX4600 Switch to a Mixed or Non-mixed Virtual Chassis (CLI Procedure)

You can use this procedure to add an EX4600 switch to a Virtual Chassis.

Before you begin, be sure you have:

- Mounted the new switch in a rack.
- Confirmed that the new switch is powered off.
- If you are expanding a preprovisioned configuration, made a note of the serial number (the number is on the back of the switch). You will need to edit the Virtual Chassis configuration to include the serial number of the new member switch.
- If you are expanding a preprovisioned configuration, edited the existing Virtual Chassis configuration to include the serial number of the new member switch. The parameters specified in the master Virtual Chassis configuration file are applied to the new switch after it has been interconnected to an existing member switch.



**NOTE:** If you are expanding a preprovisioned Virtual Chassis configuration, you can use the autoprovisioning feature to add member switches to that configuration.

- (Optional) Configured Ethernet interfaces on different member switches into the same LAG. See *Example: Configuring Aggregated Ethernet High-Speed Uplinks Between an EX4200 Virtual Chassis Access Switch and an EX4200 Virtual Chassis Distribution Switch*.

An active member switch might temporarily go down before coming back up as part of this procedure. Having traffic load-balanced across member switches using a LAG helps alleviate traffic loss during this procedure.

To add a new member switch to an existing Virtual Chassis configuration:

1. If the new member switch has been previously configured, revert that switch's configuration to the factory defaults before interconnecting it into the Virtual Chassis. See *Reverting to the Default Factory Configuration for the EX Series Switch*.
2. (Required for a mixed Virtual Chassis) Set the master switch into mixed mode, and reboot the switch to complete the configuration:

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

If adding this switch to the Virtual Chassis converts a non-mixed Virtual Chassis into a mixed Virtual Chassis, log into the Virtual Chassis and set the switches into mixed mode. Reboot the Virtual Chassis to complete the configuration:

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

3. If you are rebooting the switch or the Virtual Chassis to complete a mixed mode setting change, wait for the reboot to complete before performing this step. Interconnect the unpowered new switch to one member of the existing Virtual Chassis configuration using a QSFP+ or SFP+ port.

Connect only one VCP on the unpowered new switch to a VCP on a member switch in the existing Virtual Chassis at this point of the procedure.

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

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

5. Confirm that the new member switch is now included within the Virtual Chassis configuration by entering the **show virtual-chassis** command. The new member switch should be listed in the output and the **Status** is **Prsnt**.
6. Cable the next port into the Virtual Chassis, using Steps 2 through 5.



**CAUTION:** If you immediately cable both VCPs on the new switch into the existing Virtual Chassis at the same time, a member switch that was already part of the Virtual Chassis might become nonoperational for several seconds. Network traffic to this switch is dropped during the downtime.

The member switch will return to the normal operational state with no user intervention, and normal operation of the Virtual Chassis will resume after this downtime.

---

**Related  
Documentation**

- [Configuring EX4600 Switches in a Mixed or Non-Mixed Virtual Chassis \(CLI Procedure\) on page 31](#)

## Replacing a Member Switch of a Virtual Chassis Configuration (CLI Procedure)

---



**NOTE:** This topic does not apply to Virtual Chassis Fabric (VCF) or EX8200 Virtual Chassis. See *Removing a Device From a Virtual Chassis Fabric* or *Adding or Replacing a Member Switch or an External Routing Engine in an EX8200 Virtual Chassis (CLI Procedure)*.

---

You can replace a member switch in a Virtual Chassis without disrupting network service on the other members. You can retain the existing configuration of the member switch and apply it to a new member switch, or you can free up the member ID and make it available for assignment to a new member switch.

If you want to replace a member switch of a *mixed* Virtual Chassis that contains EX4200, EX4500, or EX4550 switches, see *Removing an EX4200, EX4500, or EX4550 Switch From a Mixed Virtual Chassis (CLI Procedure)*.

To replace a member switch, use the procedure that matches what you need to accomplish:

- [Remove, Repair, and Reinstall the Same Switch on page 38](#)
- [Remove a Member Switch, Replace It with a Different Switch, and Reapply the Old Configuration on page 39](#)
- [Remove a Member Switch and Make Its Member ID Available for Reassignment to a Different Switch on page 40](#)

### Remove, Repair, and Reinstall the Same Switch

If you need to repair a member switch, you can remove it from the Virtual Chassis configuration without disrupting network service for the other members. The master stores the configuration for the member ID so that it can be reapplied when the member switch (with the same base MAC address) is reconnected.

To remove, repair, and reinstall the member switch:

1. Power off and disconnect the member switch to be repaired.
2. Repair, as necessary.
3. Reconnect the switch and power it on.

## Remove a Member Switch, Replace It with a Different Switch, and Reapply the Old Configuration

If you are unable to repair a member switch, you can replace it with a different member switch while retaining the previous configuration. The master stores the configuration of the member that was removed. When you connect a different member switch, the master assigns a new member ID. But the old configuration is still stored under the previous member ID of the previous member switch.



**NOTE:** If you have used a preprovisioned configuration, you can use the **replace** command to change the serial number in the Virtual Chassis configuration file. Substitute the serial number of the replacement member switch (on the back of the switch) for the serial number of the member switch that was removed.

To remove and replace a switch and reapply the old configuration:

1. Power off and disconnect the member switch to be replaced.
2. If the replacement member switch has been previously configured, revert that switch's configuration to the factory defaults. See *Reverting to the Default Factory Configuration for the EX Series Switch* for information about reverting to the factory default configuration on an EX Series switch or *Reverting to the Default Factory Configuration* for information about reverting to the factory default configuration on a QFX Series switch.
3. (Required when automatic software update is not enabled on the Virtual Chassis and the new member switch is running a version of Junos OS that is different than the version of Junos OS running on the Virtual Chassis) Ensure that the correct version of Junos OS is or will be installed on the new member switch by performing one of the following tasks:
  - Enable automatic software update on the Virtual Chassis. See [“Configuring Automatic Software Update on Virtual Chassis Member Switches \(CLI Procedure\)”](#) on page 48.
  - Install the version of Junos OS running on the Virtual Chassis onto the new member switch before interconnecting it into the Virtual Chassis. See *Installing Software on an EX Series Switch with a Single Routing Engine (CLI Procedure)*.
4. If you are interconnecting a switch using a dedicated VCP, connect one VCP on the replacement member switch to a VCP of another Virtual Chassis member switch.  
  
If you are interconnecting a switch using an optical port configured as a VCP, cable the optical ports together then configure the port on the Virtual Chassis as a VCP:  
  
`user@switch> request virtual-chassis vc-port set pic-slot 1 port port-number`
5. Power on the new member switch.
6. Confirm that the new member switch is now included in the Virtual Chassis configuration by checking the front-panel LCD or the for the member ID. It should display a member ID in the range from 0 through 9.

If you are using a switch that does not have an LCD interface, confirm the switch is part of the Virtual Chassis configuration by entering the **show virtual-chassis** and reviewing the output.

7. Cable the other VCP on the new member switch into the Virtual Chassis. Use the instruction in Step 3 to complete this step.



**CAUTION:** If you immediately cable both VCPs on the new switch into the existing Virtual Chassis at the same time, a member switch that was already part of the Virtual Chassis might become nonoperational for several seconds. Network traffic to this switch is dropped during the downtime.

The member switch will return to the normal operational state with no user intervention, and normal operation of the Virtual Chassis will resume after this downtime.

8. On the master switch, Issue the **request virtual-chassis renumber** command from the Virtual Chassis master to change the member switch's current member ID to the member ID of the member switch that was removed from the Virtual Chassis configuration.

## Remove a Member Switch and Make Its Member ID Available for Reassignment to a Different Switch

When you remove a member switch from the Virtual Chassis configuration, the master keeps that member switch's member ID in reserve. To make that member switch's member ID available for reassignment, issue the **request virtual-chassis recycle** command from the Virtual Chassis master.



**NOTE:** When you add or delete members in a Virtual Chassis configuration, internal routing changes might cause temporary traffic loss for a few seconds.

### Related Documentation

- *Adding or Replacing a Member Switch or an External Routing Engine in an EX8200 Virtual Chassis (CLI Procedure)*
- *Adding a New Switch to an Existing QFX Series Virtual Chassis (CLI Procedure)*
- [Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 57](#)
- *Adding a New EX4200 Switch to an Existing EX4200 Virtual Chassis (CLI Procedure)*
- *Adding an EX4200 Switch to a Preprovisioned EX4500 Virtual Chassis or a Preprovisioned Mixed EX4200 and EX4500 Virtual Chassis (CLI Procedure)*
- *Adding an EX4500 Switch to a Preprovisioned EX4200 Virtual Chassis (CLI Procedure)*
- *Adding an EX4500 Switch to a Nonprovisioned EX4200 Virtual Chassis (CLI Procedure)*



## Configuring Mastership of a Virtual Chassis (CLI Procedure)

---



**NOTE:** This topic applies to all EX Series Virtual Chassis except EX8200 Virtual Chassis. See *Configuring an EX8200 Virtual Chassis (CLI Procedure)* for information about EX8200 Virtual Chassis.

You can designate the role (master, backup, or linecard) that a member switch performs within any Virtual Chassis, whether or not you are using a preprovisioned configuration.



**NOTE:** A Virtual Chassis configuration has two Routing Engines—one is the switch in the master role and the other is the switch in the backup role. Therefore, we recommend that you always use `commit synchronize` rather than `commit` to save configuration changes made for a Virtual Chassis. This ensures that the configuration changes are saved in both Routing Engines.

This topic describes:

- [Configuring Mastership Using a Preprovisioned Configuration File on page 41](#)
- [Configuring Mastership Using a Configuration File That Is Not Preprovisioned on page 42](#)

### Configuring Mastership Using a Preprovisioned Configuration File

To configure mastership using a preprovisioned configuration:

1. Note the serial numbers of the switches that you want to function in the master role and backup role.
2. Power on only the switch that you want to function in the master role.
3. Edit the configuration to specify the preprovisioned configuration mode:

```
[edit virtual-chassis]
user@switch# set preprovisioned
```
4. Specify the serial numbers of the member switches that you want to function as master and backup, specifying their role as **routing-engine**:

```
[edit]
user@switch# set virtual-chassis member 0 serial-number abc123 role routing-engine
user@switch# set virtual-chassis member 1 serial-number def456 role routing-engine
```



**NOTE:** You cannot directly modify the mastership priority value when you are using a preprovisioned configuration. The mastership priority values are generated automatically and controlled by the role that is assigned to the member switch in the configuration file. The two members assigned the **routing-engine** role are assigned the same mastership priority value (128). However, the member that was powered on first has higher priority for the master role election according to the master election algorithm. See [“Understanding How the Master in a Virtual Chassis Is Elected” on page 16](#). Only two members can be configured with the **routing-engine** role.

5. Specify the serial numbers of any other member switches that you are including in the Virtual Chassis configuration. You can also explicitly configure their role as **line-card**.

## Configuring Mastership Using a Configuration File That Is Not Preprovisioned

To configure mastership of the Virtual Chassis through a configuration that is not preprovisioned:

1. Power on only the switch that you want to function in the master role.
2. Configure the highest possible mastership priority value (**255**) for the member that you want to function in the master role:
3. Configure the same mastership priority value (continue to edit the Virtual Chassis configuration on the master) for the member that you want to be in the backup role:

```
[edit virtual-chassis]
user@switch# set member 0 mastership-priority 255

[edit virtual-chassis]
user@switch# set member 1 mastership-priority 255
```



**NOTE:** We recommend that the master and backup have the same mastership priority value to prevent the master and backup status from switching back and forth between master and backup members in failover conditions.

4. Use the default mastership priority value (**128**) for the remaining member switches or configure the mastership priority to a value that is lower than the value specified for members functioning in the master and backup roles.

### Related Documentation

- [Configuring an EX8200 Virtual Chassis \(CLI Procedure\)](#)
- [Configuring a QFX Series Virtual Chassis \(CLI Procedure\)](#)
- [Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 57](#)
- [Adding a New EX4200 Switch to an Existing EX4200 Virtual Chassis \(CLI Procedure\)](#)

- *Adding an EX4200 Switch to a Preprovisioned EX4500 Virtual Chassis or a Preprovisioned Mixed EX4200 and EX4500 Virtual Chassis (CLI Procedure)*
- *Adding an EX4500 Switch to a Preprovisioned EX4200 Virtual Chassis (CLI Procedure)*
- *Adding an EX4500 Switch to a Nonprovisioned EX4200 Virtual Chassis (CLI Procedure)*

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

---

When a backup member takes control of a Virtual Chassis because of a reset or other temporary failure, the backup member uses the MAC address of the old master switch as the system MAC base address. This process helps ensure a smooth transition of mastership with no disruption to network connectivity.

The MAC persistence timer is used in situations in which the master switch is no longer a member of the Virtual Chassis because it has been physically disconnected or removed. If the old master switch does not rejoin the Virtual Chassis before the timer elapses, the new master switch starts using its own MAC address as the system's MAC base address. For information regarding how the system MAC base address is used to assign MAC addresses to ports in a Virtual Chassis, see ["Understanding MAC Address Assignment on a Virtual Chassis" on page 29](#).

The default timer value is 10 minutes. The maximum timer value is 60 minutes.

You can disable the MAC persistence timer starting in Junos OS Release 12.1R3. When the MAC persistence timer is disabled, the MAC address of the old master switch is used as the system MAC base address; no MAC address changes occur within the Virtual Chassis even when the old master switch is no longer a member of the Virtual Chassis because it has been physically disconnected or removed.

To configure or modify the MAC persistence timer:

```
[edit virtual-chassis]
user@switch# set mac-persistence-timer minutes
```

To disable the MAC persistence timer:

```
[edit virtual-chassis]
user@switch# set mac-persistence-timer disable
```

### Related Documentation

- *Configuring an EX3300 Virtual Chassis (CLI Procedure)*
- *Configuring an EX4200, EX4500, or EX4550 Virtual Chassis (CLI Procedure)*
- *Configuring a QFX Series Virtual Chassis (CLI Procedure)*
- [Understanding EX Series Virtual Chassis Components on page 5](#)

## Setting an Uplink Port on an EX Series Switch as a Virtual Chassis Port (CLI Procedure)

---

The procedure described in this topic can be used to connect two EX series switches together within the same Virtual Chassis.

You typically configure an uplink port as a Virtual Chassis Port (VCP) for one of the following reasons:

- You want to interconnect two EX series switches into a Virtual Chassis that are located in different wiring closets or sites, and the switches are farther apart than the maximum length of the dedicated VCP cable.
- You are configuring an EX Series Virtual Chassis composed of switches that support Virtual Chassis but do not have dedicated VCPs. EX2200, EX3300, and EX4300 switches support Virtual Chassis but do not have dedicated VCPs.

You must manually configure VCPs to connect EX2200 switches together to form an EX2200 Virtual Chassis. See *Setting a Port on an EX2200 Switch as a Virtual Chassis Port (CLI Procedure)*.

This procedure is usually not needed to configure an EX3300 Virtual Chassis. Uplink ports 2 and 3 on an EX3300 switch are configured as VCPs by default and, therefore, do not require user configuration to be set as VCPs. We recommend that you use this procedure to configure an uplink port on an EX3300 switch as a VCP only if you configured ports 2 and 3 as network uplink ports and the ports need to be reconfigured as VCPs, or when ports 2 and 3 cannot be used as VCPs for an unexpected reason. You can use this procedure to configure any uplink port on an EX3300 switch as a VCP.

QSFP+ ports on EX4300 switches are configured as VCPs, by default. You must use this procedure if you want to configure an SFP+ port on an EX4300 switch as a VCP, or if you want to configure a QSFP+ port that had been configured into a network port back into a VCP.

You can interconnect EX4200, EX4500, and EX4550 switches that are beyond the reach of the dedicated Virtual Chassis cables as members of a Virtual Chassis by using the uplink ports—including the ports on the SFP uplink module, SFP+ uplink module, or XFP uplink module—and connecting the uplink ports. To use the uplink ports or SFP network ports for interconnecting member switches, you must explicitly set the uplink ports as VCPs.



**NOTE:** You cannot set a 1000BASE-T copper SFP transceiver (EX-SFP-1GE-T) connection as a VCP on EX4200, EX4500, and EX4550 switches.

---



**NOTE:** When an uplink port is set as a VCP, it cannot be used for any other purpose. You can set one port as a VCP and configure the other port in trunk mode as an uplink to another switch.

---

Before you set an uplink port as a VCP:

1. Install the uplink module in the member switches that you want to interconnect, if you are configuring an uplink module port as a VCP.
2. Power on and connect to the switch that you plan to designate as the master of the Virtual Chassis.



**NOTE:** Do not power on the other switches at this point.

3. Run EZSetup on the switch that you are configuring to be the master. Follow the prompts to specify the hostname and other identification, time zone, and network properties. See *Connecting and Configuring an EX Series Switch (CLI Procedure)* for details. The properties that you specify for the master apply to the entire Virtual Chassis, including all the member switches that you later interconnect with the master.
4. If you want to configure and manage the Virtual Chassis remotely, specify the VME global management interface. You can configure the VME global management interface when you are setting up the master or you can do it after completing the other configuration steps for the Virtual Chassis. See *Configuring the Virtual Management Ethernet Interface for Global Management of an EX Series Virtual Chassis (CLI Procedure)*.
5. Configure mastership of the Virtual Chassis by using either the nonprovisioned or preprovisioned configuration. See “[Configuring Mastership of a Virtual Chassis \(CLI Procedure\)](#)” on page 41 for details.



**NOTE:** A Virtual Chassis has two Routing Engines, one in the master and the other in the backup. Therefore, we recommend that you always use **commit synchronize** rather than simply **commit** to save configuration changes made for a Virtual Chassis. This ensures that the configuration changes are saved in both Routing Engines.

Before you begin to interconnect a Virtual Chassis across long distances, such as between wiring closets:

- Prepare the existing Virtual Chassis for interconnecting with a potential member switch that is beyond the reach of a dedicated Virtual Chassis cable by setting at least one uplink VCP on an existing member of the Virtual Chassis.
- Prepare the potential member switch for interconnecting with the existing Virtual Chassis by setting at least one uplink VCP on the standalone switch.



**NOTE:** We recommend that you set two uplink VCPs within each wiring closet for redundancy.

This topic describes:

1. [Setting an Uplink VCP Between the Member Switches on page 45](#)
2. [Setting an Uplink VCP on a Standalone Switch on page 46](#)

## Setting an Uplink VCP Between the Member Switches

You can set an uplink port as a VCP.



**NOTE:** If you use the SFP+ uplink module, you must configure all member switches to support either 1-gigabit SFP transceivers or 10-gigabit SFP+ transceivers on EX4200 switches. See *Setting the Mode on an SFP+ or SFP+ MACSec Uplink Module (CLI Procedure)*.

To set the uplink ports for the local member switch (for example, member 0) and for a different member switch (for example, member 1) to function as VCPs:

1. Set one uplink port of member 0 as a VCP. You do not need to specify the **member member-id** option, because the command applies by default on the member where it is executed.

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

2. Set one uplink port of member 1 as a VCP.

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

This step includes the member *member-id* option, because it is executed on a different member switch than the local member switch.

## Setting an Uplink VCP on a Standalone Switch

You can set an uplink VCP on a standalone switch. You must set an uplink port on the standalone switch as a VCP prior to physically interconnecting the switch with the existing Virtual Chassis. Otherwise, the master cannot detect that the switch is a member of the Virtual Chassis.

To set one uplink VCP on the potential member, which is currently operating as a standalone switch:

1. Power on the standalone switch.
2. Set one uplink port as a VCP. You do not need to specify the **member member-id** option, because the command applies by default on the member where it is executed.

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



**NOTE:** If you do specify the member *member-id* option, use member ID 0. Because the switch is not yet interconnected with the other members of the Virtual Chassis, its current member ID is 0. Its member ID will change when it is interconnected with the Virtual Chassis. It does not impact the functioning of the uplink VCP that its VCP is set with 0 as the member ID. The VCP has significance only on the local switch.

3. After you have set the uplink VCP on the standalone switch, physically interconnect its uplink port with the VCP uplink ports of the members in the existing Virtual Chassis.

The new member switch reboots and joins the now expanded Virtual Chassis with a different member ID.



**NOTE:** The setting for the new member switch's uplink VCP remains intact and is not affected by the change of member ID.

4. If you have additional members in the second wiring closet, set a redundant VCP uplink on another member switch by issuing the **request virtual-chassis vc-port** command.

#### Related Documentation

- [Configuring an EX3300 Virtual Chassis \(CLI Procedure\)](#)
- [Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\)](#)
- [Configuring a Virtual Chassis on an EX Series Switch \(J-Web Procedure\)](#)
- [Example: Configuring an EX4200 Virtual Chassis Interconnected Across Multiple Wiring Closets](#)
- [Example: Configuring an EX4200 Virtual Chassis Using a Preprovisioned Configuration File](#)
- [Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 57](#)

## Disabling Split and Merge in a Virtual Chassis (CLI Procedure)

The split and merge feature is enabled by default on all EX Series switches and QFX Series devices in a Virtual Chassis. You can disable the split and merge feature. If you disable the split and merge feature and the Virtual Chassis splits, both parts of the split Virtual Chassis configuration remain active.

In a preprovisioned Virtual Chassis, if both of the Routing Engines end up in the same Virtual Chassis configuration after a split, the other part of the split Virtual Chassis configuration remains inactive. If the Routing Engines end up in different parts of the split Virtual Chassis configuration and the rest of the member switches are configured as having linecard roles, then a backup Routing Engine might not be selected for either part.

We recommend disabling split and merge on a Virtual Chassis with two member switches. A two-member switch Virtual Chassis that has disabled split and merge can reform more quickly and with less complications as a result of the feature being disabled.

To disable the split and merge feature in a Virtual Chassis:

```
[edit]
```

```
user@switch# set virtual-chassis no-split-detection
```

To remove this configuration item to enable the split and merge feature again (recommended when a two-member Virtual Chassis is expanded to have more members):

```
[edit]
```

```
user@switch# delete virtual-chassis no-split-detection
```

#### Related Documentation

- [Example: Assigning the Virtual Chassis ID to Determine Precedence During an EX4200 Virtual Chassis Merge](#)

- [Understanding Split and Merge in a Virtual Chassis on page 23](#)

## Configuring Automatic Software Update on Virtual Chassis Member Switches (CLI Procedure)

---

The automatic software update feature allows you to automatically update the software version on prospective member switches as they are added so that they can join the Virtual Chassis.



**NOTE:** The version of Junos OS running on the Virtual Chassis must be compatible with the software running on the prospective member switch for an automatic software update to occur. For information on Junos OS compatibility and other automatic software update restrictions, see [“Understanding Automatic Software Update on Virtual Chassis Member Switches” on page 26](#).

Before you begin, ensure that you know the name or the URL of the software package to be used by the automatic software update feature.

To configure the automatic software update feature for an EX Series or QFX Series Virtual Chassis with the exception of a mixed Virtual Chassis containing at least one EX4200 switch and at least one EX4500 or EX4550 switch:

```
[edit]
user@switch# set virtual-chassis auto-sw-update package-name package-name
```

To configure the automatic software update feature on a mixed Virtual Chassis containing at least one EX4200 switch and at least one EX4500 or EX4550 switch:

```
[edit]
user@switch# set virtual-chassis auto-sw-update ex-4200 package-name package-name
user@switch# set virtual-chassis auto-sw-update ex-4500 package-name package-name
```

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

**/pathname/package-name**

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

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

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

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

If you are configuring a mixed Virtual Chassis containing at least one EX4200 switch and at least one EX4500 or EX4550 switch, use the **ex-4200** keyword when you are specifying a path to a package for the EX4200 switches and the **ex-4500** when you are specifying a path to a package for the EX4500 or EX4550 switches. You do not need to specify the **ex4500** keyword when configuring automatic software update for a mixed EX4500 and



EX4550 Virtual Chassis, however, because the Junos OS package for an EX4500 switch updates the software for both EX4500 and EX4550 switches.

**Related  
Documentation**

- *Example: Configuring Automatic Software Update on EX4200 Virtual Chassis Member Switches*
- [Understanding Automatic Software Update on Virtual Chassis Member Switches on page 26](#)

---

## Assigning the Virtual Chassis ID to Determine Precedence During a Virtual Chassis Merge (CLI Procedure)

---

Every Virtual Chassis has a unique ID that is automatically assigned when the Virtual Chassis configuration is formed. You can also explicitly assign a Virtual Chassis ID using the **set virtual-chassis id** command. When two Virtual Chassis configurations attempt to merge, the Virtual Chassis ID that you assigned takes precedence over the automatically assigned Virtual Chassis IDs and becomes the ID for the newly merged Virtual Chassis configuration.

To configure the Virtual Chassis ID:

```
[edit]
user@switch# set virtual-chassis id id
```

**Related  
Documentation**

- *Example: Assigning the Virtual Chassis ID to Determine Precedence During an EX4200 Virtual Chassis Merge*
- [Understanding Split and Merge in a Virtual Chassis on page 23](#)

---

## Command Forwarding Usage with an EX Series Virtual Chassis

---

Some CLI commands can be run either on all members or on a specific member of a Virtual Chassis configuration. This functionality is referred to as command forwarding.

You can always specify that these commands be applied to all member switches in the Virtual Chassis by using the **all-members** option, or to a specific member switch by using the **member-member-id** option. If neither option is specified, the default command forwarding behavior, which varies by command, is used. See the **Default** row in [Table 9 on page 50](#) to learn the command forwarding behavior for a specific command.

For example, to collect information about a particular member switch prior to contacting Juniper Networks Technical Assistance Center (JTAC), use the **request support information member member-id** command to gather data for the specified member switch. If you want to gather this data for all member switches in the Virtual Chassis, you can enter the **request support information** command, which by default uses the **all-members** option, or the **request support information all-members** command.

[Table 9 on page 50](#) provides a list of commands that can be run either on all members of the Virtual Chassis configuration or on a specific member switch.

Table 9: Commands That Can be Run on All or Specific Members of the Virtual Chassis Configuration

Commands Available for Command Forwarding	Purpose	all-members	member-member-id	Default
<b>request support information</b>	<p>Use this command when you contact JTAC about your component problem. This command is the equivalent of using the following CLI commands:</p> <ul style="list-style-type: none"> <li>• <b>show version</b></li> <li>• <b>show chassis firmware</b></li> <li>• <b>show chassis hardware</b></li> <li>• <b>show chassis environment</b></li> <li>• <b>show interfaces extensive</b> (for each configured interface)</li> <li>• <b>show configuration</b> (excluding any SECRET-DATA)</li> <li>• <b>show system virtual-memory</b></li> </ul>	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	<b>all-members</b>
<b>request system partition hard-disk</b>	Set up the hard disk for partitioning. After this command is issued, the hard disk is partitioned the next time the system is rebooted. When the hard disk is partitioned, the contents of <b>/altroot</b> and <b>/altconfig</b> are saved and restored. All other data on the hard disk is at risk of being lost.	Partitions the hard disk on all members of the Virtual Chassis configuration.	Partitions the hard disk on the specified member switch.	<b>all-members</b>
<b>request system reboot</b>	Reboot Junos OS for EX Series switches after a software upgrade and occasionally to recover from an error condition.	Reboots all members of the Virtual Chassis configuration.	Reboots the specified member switch.	<b>all-members</b>

Table 9: Commands That Can be Run on All or Specific Members of the Virtual Chassis Configuration (*continued*)

Commands Available for Command Forwarding	Purpose	all-members	member-member-id	Default
<b>request system snapshot</b>	Back up the currently running and active file system.	Backs up the file systems on all members of the Virtual Chassis configuration.	Backs up the file system on the specified member switch.	<b>all-members</b>
<b>request system storage cleanup</b>	Free storage space on the switch by rotating log files and proposing a list of files for deletion. User input is required for file deletion.	Runs cleanup on all members of the Virtual Chassis configuration.	Runs cleanup on the specified member switch.	<b>all-members</b>
<b>show log user</b>	Display users who are viewing the system log.	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	master switch only
<b>show system alarms</b>	Display active system alarms.	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	<b>all-members</b>
<b>show system audit</b>	Display the state and checksum values for file systems.	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	<b>all-members</b>
<b>show system boot-messages</b>	Display initial messages generated by the system kernel upon startup. These messages are the contents of <code>/var/run/dmesg.boot</code> .	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	<b>all-members</b>

Table 9: Commands That Can be Run on All or Specific Members of the Virtual Chassis Configuration (*continued*)

Commands Available for Command Forwarding	Purpose	all-members	member-member-id	Default
<b>show system buffers</b>	Display information about the buffer pool that the Routing Engine uses for local traffic. Local traffic is the routing and management traffic that is exchanged between the Routing Engine and the Packet Forwarding Engine within the switch, as well as the routing and management traffic from IP (that is, from OSPF, BGP, SNMP, ping operations, and so on).	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	<b>all-members</b>
<b>show system connections</b>	Display information about the active IP sockets on the Routing Engine. Use this command to verify which servers are active on a system and which connections are currently in progress.	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	<b>all-members</b>
<b>show system core-dumps</b>	Display a core file generated by an internal Junos OS process.	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	<b>all-members</b>
<b>show system directory-usage</b>	Display directory usage information.	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	master switch only
<b>show system processes</b>	Display information about software processes that are running on the switch and that have controlling terminals.	Displays information for all members of the Virtual Chassis configuration.		<b>all-members</b>

Table 9: Commands That Can be Run on All or Specific Members of the Virtual Chassis Configuration (*continued*)

Commands Available for Command Forwarding	Purpose	all-members	member-member-id	Default
<b>show system reboot</b>	Display pending system reboots or halts.	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	<b>all-members</b>
<b>show system snapshot</b>	Display information about the backup software that is located in the <code>/altroot</code> and <code>/altconfig</code> file systems. To back up software, use the <code>request system snapshot</code> command.	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	<b>all-members</b>
<b>show system software</b>	Display the Junos OS extensions loaded on your switch.	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	<b>all-members</b>
<b>show system statistics</b>	Display systemwide protocol-related statistics.	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	<b>all-members</b>
<b>show system storage</b>	Display statistics about the amount of free disk space in the switch's file systems.	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	<b>all-members</b>
<b>show system uptime</b>	Display the current time and information about how long the switch, the switch software, and any existing protocols have been running	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	<b>all-members</b>
<b>show system users</b>	Show all users who are currently logged in.	Shows all users who are currently logged in to any members of the Virtual Chassis configuration.	Shows all users who are currently logged in to the specified member switch.	<b>all-members</b>

Table 9: Commands That Can be Run on All or Specific Members of the Virtual Chassis Configuration (*continued*)

Commands Available for Command Forwarding	Purpose	all-members	member-member-id	Default
<b>show system virtual-memory</b>	Display the usage of Junos OS kernel memory, listed first by size of allocation and then by type of usage. Use <b>show system virtual-memory</b> for troubleshooting with JTAC.	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	<b>all-members</b>

**Related Documentation**

- [Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 57](#)
- [Understanding EX Series Virtual Chassis Components on page 5](#)
- [Junos OS System Basics and Services Command Reference](#)

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

**Purpose** You can designate the role that a member performs within a Virtual Chassis or you can allow the role to be assigned by default. You can designate the member ID that is assigned to a specific switch by creating a permanent association between the switch's serial number and a member ID, using a preprovisioned configuration. Or you can let the member ID be assigned by the master, based on the sequence in which the member switch is powered on and on which member IDs are currently available.

The role and member ID of the member switch are displayed on the front-panel LCD.

Each member switch can be cabled to one or two other member switches, using either the dedicated Virtual Chassis ports (VCPs) on the rear panel, an uplink port that has been configured as a VCP, or an optical port that has been configured as a VCP. The members that are cabled together are considered neighbor members.

**Action** To display the role and member ID assignments using the CLI:

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

```
Virtual Chassis ID: 0000.e255.00e0
```

Member ID	Status	Serial No	Model	Mastership Priority	Role	Neighbor List ID, Interface
0 (FPC 0)	Prsnt	abc123	ex4200-48p	255	Master*	1 vcp-0 2 vcp-1
1 (FPC 1)	Prsnt	def456	ex4200-24t	255	Backup	2 vcp-0

```

0 vcp-1
2 (FPC 2) Prsnt abd231 ex4200-24p 128 Linecard 0 vcp-0
1 vcp-1

```

**Meaning** This output verifies that three EX4200 switches have been interconnected as a Virtual Chassis configuration through their dedicated VCPs to create an EX4200 Virtual Chassis. The display shows which of the VCPs is connected to which neighbor. The first port (**vcp-0**) of member **0** is connected to member **1** and the second port of member **0** (**vcp-1**) is connected to member **2**. The FPC slots for the switches are the same as the member IDs.

The **Mastership Priority** values indicate that the master and backup members have been explicitly configured, because they are not using the default value (128).



**NOTE:** This example uses output from an EX4200 Virtual Chassis. The output, with the exception of the Model column, would be identical on all other Virtual Chassis.

#### Related Documentation

- [Configuring Mastership of a Virtual Chassis \(CLI Procedure\) on page 41](#)
- [Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\)](#)
- [Configuring a Virtual Chassis on an EX Series Switch \(J-Web Procedure\)](#)
- [Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\)](#)
- [Configuring a QFX Series Virtual Chassis \(CLI Procedure\)](#)
- [Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 57](#)

## Verifying That Virtual Chassis Ports Are Operational

**Purpose** Display the status of Virtual Chassis ports (VCPs) in a Virtual Chassis or Virtual Chassis Fabric (VCF).



**NOTE:** VCPs are not displayed when you issue the `show interfaces` command.

**Action** Display the VCPs:

```
user@switch> show virtual-chassis vc-port all-members
```

```
fpc0:
```

Interface or	Type	Trunk ID	Status	Speed (mbps)	Neighbor ID Interface
-----------------	------	-------------	--------	-----------------	--------------------------

PIC / Port						
vcp-0	Dedicated	1	Up	32000	1	vcp-0
vcp-1	Dedicated	2	Up	32000	1	vcp-1
1/0	Configured	3	Up	1000	2	vcp-255/1/0
1/1	Configured	3	Up	1000	2	vcp-255/1/1
1/2	Configured	4	Up	1000	4	vcp-255/0/20
1/3	Configured	4	Up	1000	4	vcp-255/0/21

## fpc1:

Interface or PIC / Port	Type	Trunk ID	Status	Speed (mbps)	Neighbor ID	Interface
vcp-0	Dedicated	1	Up	32000	0	vcp-0
vcp-1	Dedicated	2	Up	32000	0	vcp-1
1/0	Configured	3	Up	10000	3	vcp-255/1/0
1/1	Configured	3	Up	10000	3	vcp-255/1/1

## fpc2:

Interface or PIC / Port	Type	Trunk ID	Status	Speed (mbps)	Neighbor ID	Interface
vcp-0	Dedicated	1	Up	32000	3	vcp-0
vcp-1	Dedicated	2	Up	32000	3	vcp-1
1/0	Configured	3	Up	1000	0	vcp-255/1/0
1/1	Configured	3	Up	1000	0	vcp-255/1/1
1/2		-1	Down	1000		
1/3		-1	Down	1000		

## fpc3:

Interface or PIC / Port	Type	Trunk ID	Status	Speed (mbps)	Neighbor ID	Interface
vcp-0	Dedicated	1	Up	32000	2	vcp-0
vcp-1	Dedicated	2	Up	32000	2	vcp-1
1/0	Configured	3	Up	10000	1	vcp-255/1/0
1/1	Configured	3	Up	10000	1	vcp-255/1/1

## fpc4:

Interface or PIC / Port	Type	Trunk ID	Status	Speed (mbps)	Neighbor ID	Interface
vcp-0	Dedicated	1	Down	32000		
vcp-1	Dedicated	2	Down	32000		
0/20	Configured	3	Up	1000	0	vcp-255/1/2
0/21	Configured	3	Up	1000	0	vcp-255/1/3

**Meaning** The dedicated VCPs are displayed as **vcp-0** and **vcp-1**. The uplink interfaces that have been set as uplink VCPs are displayed as **1/0**, **1/1**, **1/2**, and **1/3**. The network interfaces that have been set as VCPs are displayed as **0/20** and **0/21**. The neighbor interface names of uplink and network VCPs are of the form **vcp-255/pic/port**—for example, **vcp-255/1/0**. In that name, **vcp-255** indicates that the interface is a VCP, 1 is the uplink PIC number, and 0 is the port number. The **fpc** number is the same as the member ID. The trunk ID is a positive number ID assigned to the link aggregation group (LAG) formed by the Virtual Chassis. If no LAG is formed, the value is -1.





**NOTE:** This example uses output from an EX4200 Virtual Chassis. The output is similar on all other types of Virtual Chassis or for a VCF.

**Related Documentation**

- [Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 57](#)
- *Configuring an EX3300 Virtual Chassis (CLI Procedure)*
- *Configuring an EX4200, EX4500, or EX4550 Virtual Chassis (CLI Procedure)*
- *Configuring a Virtual Chassis on an EX Series Switch (J-Web Procedure)*
- *Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches (CLI Procedure)*

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

**Purpose**



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

Use the monitoring functionality to view the following information about the switches and the ports on EX2200, EX3300, EX4200, EX4300, EX4500, EX4550, and EX8200 switches that are members of a Virtual Chassis:

- Member details and how members are connected with each other
- Traffic statistics for Virtual Chassis ports (VCPs) of the selected members
- Details of the VCP packet counters

**Action**

To view Virtual Chassis monitoring details in the J-Web interface for a Virtual Chassis, select **Monitor > Virtual Chassis**.



**NOTE:** Virtual Chassis monitoring is supported on J-Web, on all Virtual Chassis platforms except EX3200 and EX6210.

To view member details for all members in the CLI, enter the following command:

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

To view VCP traffic statistics for a specific member in the CLI, enter the following command:

```
user@switch> show virtual-chassis vc-port statistics member member-id
```

To view the path a packet takes when going from a source interface to a destination interface in a Virtual Chassis configuration using the CLI, enter the following command:

```
user@switch> show virtual-chassis vc-path
```

**Meaning** In the J-Web interface, the top half of the screen displays details of the Virtual Chassis configuration, such as:

- Member
- Role
- Status
- Interface
- Type
- Speed
- Neighboring Member ID
- Link Status
- Error count



**NOTE:** If the member switch in the Virtual Chassis is not provisioned, the member ID will be displayed as -.

---

Click the **Stop** button to stop fetching values from the switch, and click the **Start** button to start plotting data again from the point where it was stopped.

To view a graph of the statistics for the selected VCP of the member, click **Show Graph**.

**Refresh Interval (sec)**—Displays the time interval you have set for page refresh.

Click **Clear Statistics** to clear the monitoring statistics for the selected member switch. You can specify the interval at which the member details and statistics must be refreshed.

The bottom half of the screen displays a chart of the Virtual Chassis statistics and the port packet counters.

For details about the output from CLI commands, see the [show virtual-chassis](#) and [show virtual-chassis vc-port statistics](#) command summaries.

**Related Documentation**

- [Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\)](#)
- [Configuring an EX8200 Virtual Chassis \(CLI Procedure\)](#)
- [Configuring a Virtual Chassis on an EX Series Switch \(J-Web Procedure\)](#)
- [Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\)](#)
- [Verifying the Member ID, Role, and Neighbor Member Connections of a Virtual Chassis Member on page 54](#)

## Verifying That Graceful Routing Engine Switchover Is Working in the Virtual Chassis

**Purpose** Verify that a Graceful Routing Engine switchover (GRES) between two member switches acting as the master and backup routing engines in a Virtual Chassis has occurred.

**Action** On the master switch, verify the member ID of the backup Routing Engine:

```
{master:0}
user@switch> show virtual-chassis
Virtual Chassis ID: 5efa.4b7a.aae6
```

Member ID	Status	Serial No	Model	Mastership priority	Role	Neighbor List ID	Interface
0 (FPC 0)	Prsnt	BM0208105281	ex4200-24t	255	Master*	1	vcp-0
1 (FPC 1)	Prsnt	BP0208192350	ex4200-48t	255	Backup	0	vcp-0

Member ID for next new member: 2 (FPC 2)

1. Connect to the backup Routing Engine:

```
{master:0}
user@switch> request session member 1

{backup:1}
user@switch>
```

2. Verify that the backup Routing Engine is ready for switchover on member ID 1:

```
{backup:1}
user@switch> show system switchover

Graceful switchover: On
Configuration database: Ready
Kernel database: Ready
Peer state: Steady State
```

3. Switch the current backup Routing Engine to master Routing Engine:



**NOTE:** You must wait a minimum of two minutes between Routing Engine failovers for the Routing Engines to synchronize.

```
{backup:1}
user@switch> request chassis routing-engine master acquire
```

4. Verify that the master and backup Routing Engines have switched roles:



**NOTE:** Member ID 1 is now the master and member ID 0 is now the backup.

```
{master:1}
user@switch> show virtual-chassis

Virtual Chassis ID: 5efa.4b7a.aae6
```

				Mastership		Neighbor List	
Member ID	Status	Serial No	Model	priority	Role	ID	Interface
0 (FPC 0)	Prsnt	BM0208105281	ex4200-24t	255	Backup	1	vcp-0
1 (FPC 1)	Prsnt	BP0208192350	ex4200-48t	255	Master*	0	vcp-0

Member ID for next new member: 2 (FPC 2)

**Meaning** With graceful Routing Engine switchover enabled, when you initiated a switchover from the backup Routing Engine, the backup Routing Engine became the master and the master Routing Engine became the backup.

**Related Documentation** • [Configuring Graceful Routing Engine Switchover in a Virtual Chassis \(CLI Procedure\)](#)

## Troubleshooting an EX Series Virtual Chassis

This topic describes the following troubleshooting issues for a Virtual Chassis:

- [A Disconnected Member Switch's ID Is Not Available for Reassignment on page 60](#)
- [Load Factory Default Does Not Commit on a Multimember Virtual Chassis on page 60](#)
- [The Member ID Persists When a Member Switch Is Disconnected From a Virtual Chassis on page 61](#)
- [A Member Switch Is Not Participating in a Mixed Virtual Chassis on page 61](#)

### A Disconnected Member Switch's ID Is Not Available for Reassignment

**Problem Description:** You disconnected a switch from the Virtual Chassis, but the disconnected switch's member ID is still displayed in the status output. You cannot reassign that member ID to another switch.

**Solution** When you disconnect a member of a Virtual Chassis configuration, the master retains the member ID and member configuration in its configuration database. Output from the [show virtual-chassis](#) command continues to display the member ID of the disconnected member with a status of **NotPrsnt**.

If want to permanently disconnect the member switch, you can free up the member ID by using the [request virtual-chassis recycle](#) command. This will also clear the status of that member.

### Load Factory Default Does Not Commit on a Multimember Virtual Chassis

**Problem Description:** The **load factory-default** command fails on a multimember Virtual Chassis.

**Solution** The **load factory-default** command is not supported on a multimember Virtual Chassis configuration. For information on how to revert the switches in the Virtual Chassis to

factory default settings, see *Reverting to the Default Factory Configuration for the EX Series Switch*.

## The Member ID Persists When a Member Switch Is Disconnected From a Virtual Chassis

**Problem** **Description:** Gigabit Ethernet interfaces retain their previous slot numbers when a member switch is disconnected from the Virtual Chassis.

**Solution** If a switch had been previously connected as a member of a Virtual Chassis configuration, it retains the member ID that it was assigned as a member of that configuration even after it is disconnected and operating as a standalone switch. The interfaces that were configured while the switch was a member of the Virtual Chassis configuration retain the old member ID as the first digit of the interface name.

For example, if the switch was previously member 1, its interfaces are named **ge-1/0/0** and so on.

To change the switch's member ID, so that its member ID is **0**, and to rename the switch's interfaces accordingly:

1. To change the member ID to 0:

```
user@switch> request virtual-chassis renumber member-id 1 new-member-id 0
```

2. To rename the interfaces to match the new member ID:

```
[edit virtual-chassis]
user@switch# replace pattern ge-1/ with ge-0/
```

## A Member Switch Is Not Participating in a Mixed Virtual Chassis

**Problem** **Description:** A member switch in a mixed Virtual Chassis is not participating in the Virtual Chassis. The **show virtual-chassis** output indicates the member switch status is **Inactive** or **NotPrsnt**.

This issue is most likely to occur immediately after you have cabled a mixed Virtual Chassis.

**Solution** The Virtual Chassis mode on the switch might not be set to **mixed** mode. If the member switch is an EX4500 switch and is cabled into the Virtual Chassis through the dedicated Virtual Chassis port (VCP), the PIC mode might also be set to **Intraconnect** instead of **virtual-chassis**.

To verify the Virtual Chassis mode:

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

```
-----
Mixed Mode: Enabled
fpc1:
```

```
-----
Mixed Mode: Enabled
fpc2:
```

```
-----
Mixed Mode: Enabled
```

fpc3:

-----  
Mixed Mode: Enabled

fpc4:

-----  
Mixed Mode: Disabled

fpc5:

-----  
Mixed Mode: Enabled

To change the Virtual Chassis mode on a member switch (in this case, member ID 4) to **mixed** mode:

```
user@switch> request virtual-chassis mode mixed member 4
```

(EX4500 switch only) To verify the PIC mode:

```
user@switch> show chassis pic-mode
```

fpc0:

-----  
Pic Mode: Not-Applicable

fpc1:

-----  
Pic Mode: Not-Applicable

fpc2:

-----  
Pic Mode: Not-Applicable

fpc3:

-----  
Pic Mode: Not-Applicable

fpc4:

-----  
Pic Mode: PIC 3: Intraconnect

fpc5:

-----  
Pic Mode: PIC 3: virtual-chassis

To change the PIC mode on an EX4500 switch to **virtual-chassis** mode (in this case, member ID 4):

```
user@switch> request chassis pic-mode virtual-chassis member 4
```

The member switch must be rebooted for the Virtual Chassis mode or PIC mode setting change to take effect. To reboot the member switch (in this case, member ID 4):

```
user@switch> request system reboot member 4
```

#### Related Documentation

- [Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 57](#)
- [Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\)](#)
- [Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\)](#)
- [Configuring a Virtual Chassis on an EX Series Switch \(J-Web Procedure\)](#)

## CHAPTER 2

# Configuration Statements for Virtual Chassis

- [\[edit virtual-chassis\] Configuration Statement Hierarchy on page 63](#)
- [aliases \(Virtual Chassis\) on page 66](#)
- [alias-name \(Virtual Chassis aliases\) on page 67](#)
- [auto-sw-update on page 68](#)
- [id on page 70](#)
- [location \(Virtual Chassis\) on page 71](#)
- [mac-persistence-timer on page 72](#)
- [mastership-priority on page 73](#)
- [member on page 75](#)
- [no-management-vlan on page 76](#)
- [no-split-detection on page 77](#)
- [package-name on page 78](#)
- [preprovisioned on page 79](#)
- [role on page 80](#)
- [serial-number on page 83](#)
- [serial-number \(Virtual Chassis aliases\) on page 84](#)
- [traceoptions \(Virtual Chassis\) on page 85](#)
- [vcp-no-hold-time on page 88](#)
- [vcp-snmp-statistics on page 89](#)
- [virtual-chassis on page 90](#)

### [\[edit virtual-chassis\] Configuration Statement Hierarchy](#)

---

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

- *Supported* statements are those that you can use to configure some aspect of a software feature on the switch.

- *Unsupported* statements are those that appear in the command-line interface (CLI) on the switch, but that have no effect on switch operation if you configure them.
- Not all features are supported on all switch platforms.

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

This topic lists:

- [Supported Statements in the \[edit virtual-chassis\] Hierarchy Level on page 64](#)
- [Unsupported Statements in the \[edit virtual-chassis\] Hierarchy Level on page 65](#)

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

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

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



}

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

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

### Related Documentation

- *Preprovisioning a Virtual Chassis Fabric*
- *Autoprovisioning a Virtual Chassis Fabric*
- *Adding a Device to a Virtual Chassis Fabric*
- *Configuring a QFX Series Virtual Chassis (CLI Procedure)*
- *Configuring an EX4300 Virtual Chassis (CLI Procedure)*
- *Configuring an EX2200 Virtual Chassis (CLI Procedure)*
- *Configuring an EX3300 Virtual Chassis (CLI Procedure)*
- *Configuring an EX4200, EX4500, or EX4550 Virtual Chassis (CLI Procedure)*
- *Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches (CLI Procedure)*
- *Configuring an EX8200 Virtual Chassis (CLI Procedure)*

## aliases (Virtual Chassis)

---

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

## alias-name (Virtual Chassis aliases)

**Syntax** `alias-name alias-name;`

**Hierarchy Level** `[edit virtual-chassis aliases serial-number serial-number]`

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

**Description** Create an alias for a member switch in a Virtual Chassis or Virtual Chassis Fabric (VCF). An alias allows you to more clearly identify the member switches in your Virtual Chassis or VCF by assigning a text label to a member switch's serial number.

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

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

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

### set serial-number

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

### show virtual-chassis

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

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

**Options** *alias-name*—The text label, or alias, assigned to the member switch by the user.

**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*
- *Preprovisioning a Virtual Chassis Fabric*
- *Configuring a QFX Series Virtual Chassis (CLI Procedure)*
- *Understanding Virtual Chassis Fabric Components*
- *Understanding QFX Series Virtual Chassis Components*

## auto-sw-update

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

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

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

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

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

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

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

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

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

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

<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <i>Example: Configuring Automatic Software Update on EX4200 Virtual Chassis Member Switches</i></li><li>• <a href="#">Configuring Automatic Software Update on Virtual Chassis Member Switches (CLI Procedure) on page 48</a></li><li>• <i>Understanding Software Upgrades in a Virtual Chassis Fabric</i></li></ul>
------------------------------	--

## id

---

<b>Syntax</b>	<code>id id;</code>
<b>Hierarchy Level</b>	[edit <a href="#">virtual-chassis</a> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 9.3 for EX Series switches. Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
<b>Description</b>	Configure the alphanumeric string that identifies a Virtual Chassis or Virtual Chassis Fabric (VCF) configuration.
<b>Options</b>	<i>id</i> —Virtual Chassis ID (VCID), which uses the ISO family address format—for example, <b>9622.6ac8.5345</b> .
<b>Required Privilege Level</b>	system—To view this statement in the configuration. system-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <i>Example: Assigning the Virtual Chassis ID to Determine Precedence During an EX4200 Virtual Chassis Merge</i></li><li>• <a href="#">Assigning the Virtual Chassis ID to Determine Precedence During a Virtual Chassis Merge (CLI Procedure) on page 49</a></li><li>• <i>Configuring a QFX Series Virtual Chassis (CLI Procedure)</i></li><li>• <i>Autoprovisioning a Virtual Chassis Fabric</i></li><li>• <i>Preprovisioning a Virtual Chassis Fabric</i></li><li>• <i>Configuring an EX8200 Virtual Chassis (CLI Procedure)</i></li><li>• <i>Understanding Virtual Chassis Member ID Numbering in an EX8200 Virtual Chassis</i></li></ul>

## location (Virtual Chassis)

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

## mac-persistence-timer

---

<b>Syntax</b>	<code>mac-persistence-timer [<i>minutes</i>   <b>disable</b>];</code>
<b>Hierarchy Level</b>	[edit <a href="#">virtual-chassis</a> ]
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Option <b>disable</b> introduced in Junos OS Release 12.2 for EX Series switches.</p> <p>The maximum timer limit changed from no maximum timer limit to 60 minutes in Junos OS Release 12.2 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).</p>
<b>Description</b>	<p>Specify how long the Virtual Chassis or VCF continues to use the MAC address of the switch that was originally configured in the master role as the system MAC base address after the original master switch is removed from the Virtual Chassis or VCF. The system MAC base address does not change in the event of a switchover provided the switch originally configured in the master role remains a member of the Virtual Chassis or VCF.</p> <p>The maximum timer limit is 60 minutes starting in Junos OS Release 12.2. There are no minimum or maximum timer limits in prior Junos OS releases.</p>
<b>Default</b>	The MAC persistence timer is set to 10 minutes by default.
<b>Options</b>	<p><b>minutes</b>—Time in minutes that the member switch in the backup role continues to use the system MAC base address of the old master before using its own system MAC base address after the switch in the master role is physically disconnected or removed from the Virtual Chassis or VCF.</p> <p><b>disable</b>—Disable the MAC persistence timer. The system MAC base address never changes when the MAC persistence timer is disabled, even when the switch in the master role is physically disconnected or removed from the Virtual Chassis or VCF.</p>
<b>Required Privilege Level</b>	<p>system—To view this statement in the configuration.</p> <p>system-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Configuring the Timer for the Backup Member to Start Using Its Own MAC Address, as Master of a Virtual Chassis (CLI Procedure) on page 43</a></li><li>• <a href="#">Autoprovisioning a Virtual Chassis Fabric</a></li><li>• <a href="#">Preprovisioning a Virtual Chassis Fabric</a></li></ul>



## mastership-priority

<b>Syntax</b>	<code>mastership-priority <i>number</i>;</code>
<b>Hierarchy Level</b>	[edit <code>virtual-chassis member <i>member-id</i></code> ]
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Mastership priority option <b>0</b> introduced in Junos OS Release 11.1 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).</p>
<b>Description</b>	<p>The mastership priority value is the most important factor in determining the role of the member switch within a nonprovisioned Virtual Chassis or VCF configuration. Other factors (see <a href="#">“Understanding How the Master in a Virtual Chassis Is Elected” on page 16</a>) also affect the election of the master.</p> <p>The mastership priority value takes the highest precedence in the master election algorithm. The member switch with highest mastership priority assumes the master Routing Engine role of the Virtual Chassis or VCF. Toggling back and forth between master and backup status in failover conditions is undesirable, so we recommend that you assign the same mastership priority value to both the master and the backup. Secondary factors in the master election algorithm determine which of these two members (that is, the two members that are assigned the highest mastership priority value) functions as the master of the Virtual Chassis or VCF.</p> <p>This statement is not used for the EX8200 Virtual Chassis, which determines mastership by external Routing Engine uptime. See <i>Understanding Virtual Chassis Roles in an EX8200 Virtual Chassis</i>.</p> <p>A switch with a mastership priority of <b>0</b> never takes the master or backup role.</p>
<b>Default</b>	128
<b>Options</b>	<p><i>number</i>—Mastership priority value.</p> <p><b>Range:</b> 0 through 255</p>
<b>Required Privilege Level</b>	<p>system—To view this statement in the configuration.</p> <p>system-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>Autoprovisioning a Virtual Chassis Fabric</i></li> <li>• <i>Preprovisioning a Virtual Chassis Fabric</i></li> <li>• <i>Configuring a QFX Series Virtual Chassis (CLI Procedure)</i></li> <li>• <i>Configuring an EX4300 Virtual Chassis (CLI Procedure)</i></li> <li>• <i>Example: Configuring an EX3300 Virtual Chassis with a Master and Backup</i></li> <li>• <i>Example: Configuring an EX4200 Virtual Chassis with a Master and Backup in a Single Wiring Closet</i></li> </ul>

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

## member

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

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

---

## no-management-vlan

---

<b>Syntax</b>	no-management-vlan;
<b>Hierarchy Level</b>	[edit <a href="#">virtual-chassis member member-id</a> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
<b>Description</b>	<p>Remove the specified member's out-of-band management port from the virtual management Ethernet (VME) global management VLAN of the Virtual Chassis or VCF configuration.</p> <p>For a member that is functioning in a linecard role, you can use this configuration to reserve the member's management Ethernet port for local troubleshooting:</p> <pre>virtual-chassis {   member 2 {     no-management-vlan;   } }</pre> <p>You cannot configure the IP address for a local management Ethernet port using the CLI or the J-Web interface. To do this, you need to use the shell <b>ifconfig</b> command.</p>
<b>Required Privilege Level</b>	system—To view this statement in the configuration. system-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <i>Example: Setting Up a Multimember EX4200 Virtual Chassis Access Switch with a Default Configuration</i></li><li>• <i>Configuring the Virtual Management Ethernet Interface for Global Management of an EX Series Virtual Chassis (CLI Procedure)</i></li><li>• <a href="#">Understanding Global Management of a Virtual Chassis on page 18</a></li><li>• <i>Understanding Virtual Chassis Fabric Configuration</i></li></ul>

## no-split-detection

<b>Syntax</b>	no-split-detection;
<b>Hierarchy Level</b>	[edit <a href="#">virtual-chassis</a> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 9.3 for EX Series switches. Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
<b>Description</b>	<p>Disable the split and merge feature in a Virtual Chassis or VCF configuration.</p> <p>We recommend using this statement to disable the split and merge feature when configuring a two-member Virtual Chassis. Enabling this statement on a two-member Virtual Chassis ensures that both switches remain in the correct Virtual Chassis roles in the event of a Virtual Chassis split.</p> <p>The split and merge feature is enabled by default when a Virtual Chassis is initially set up. If you set this option in a two-member Virtual Chassis and then expand the Virtual Chassis to add more members, we recommend deleting this configuration item to re-enable the split and merge feature again.</p>
<b>Default</b>	The split and merge feature is enabled.
<b>Required Privilege Level</b>	system—To view this statement in the configuration. system-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>Example: Assigning the Virtual Chassis ID to Determine Precedence During an EX4200 Virtual Chassis Merge</i></li> <li>• <a href="#">Disabling Split and Merge in a Virtual Chassis (CLI Procedure) on page 47</a></li> <li>• <a href="#">Assigning the Virtual Chassis ID to Determine Precedence During a Virtual Chassis Merge (CLI Procedure) on page 49</a></li> <li>• <i>Disabling Split Detection in a Virtual Chassis Configuration</i></li> <li>• <a href="#">Understanding Split and Merge in a Virtual Chassis on page 23</a></li> </ul>

## package-name

---

<b>Syntax</b>	<code>package-name <i>package-name</i>;</code>
<b>Hierarchy Level</b>	[edit virtual-chassis <a href="#">auto-sw-update</a> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 10.0 for EX Series switches. Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
<b>Description</b>	Specify the software package name or location of the software package to be used by the automatic software update feature for Virtual Chassis or VCF.
<b>Default</b>	No package name is specified.
<b>Options</b>	<p><b><i>package-name</i></b>—Name of the software package or the URL to the software package to be used.</p> <ul style="list-style-type: none"><li>• If the software package is located on a local directory on the switch, use the following format for <b><i>package-name</i></b>:  <b><i>/pathname/package-name</i></b></li><li>• If the software package is to be downloaded and installed from a remote location, use one of the following formats:  <b><i>ftp://hostname/pathname/package-name</i></b> <b><i>ftp://username:prompt@ftp.hostname.net/package-name</i></b> <b><i>http://hostname/pathname/package-name</i></b></li></ul>
<b>Required Privilege Level</b>	system—To view this statement in the configuration. system-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <i>Example: Configuring Automatic Software Update on EX4200 Virtual Chassis Member Switches</i></li><li>• <a href="#">Configuring Automatic Software Update on Virtual Chassis Member Switches (CLI Procedure)</a> on page 48</li><li>• <i>Understanding Software Upgrades in a Virtual Chassis Fabric</i></li></ul>

## preprovisioned

---

<b>Syntax</b>	preprovisioned;
<b>Hierarchy Level</b>	[edit <a href="#">virtual-chassis</a> ]
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).</p>
<b>Description</b>	<p>Enable the preprovisioned configuration mode for a Virtual Chassis or Virtual Chassis Fabric (VCF) configuration.</p> <p>When the preprovisioned configuration mode is enabled, you cannot use the CLI or the J-Web interface to change the mastership priority or member ID of member switches.</p> <p>You must use this statement to configure an EX8200 Virtual Chassis. Nonprovisioned configuration of an EX8200 Virtual Chassis is not supported.</p>
<b>Required Privilege Level</b>	<p>system—To view this statement in the configuration.</p> <p>system-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>Preprovisioning a Virtual Chassis Fabric</i></li> <li>• <i>Example: Configuring an EX4200 Virtual Chassis Using a Preprovisioned Configuration File</i></li> <li>• <i>Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines</i></li> <li>• <i>Configuring an EX4200, EX4500, or EX4550 Virtual Chassis (CLI Procedure)</i></li> <li>• <i>Configuring an EX8200 Virtual Chassis (CLI Procedure)</i></li> <li>• <i>Configuring an EX9200 Virtual Chassis</i></li> <li>• <i>Configuring a QFX Series Virtual Chassis (CLI Procedure)</i></li> <li>• <a href="#">Replacing a Member Switch of a Virtual Chassis Configuration (CLI Procedure) on page 38</a></li> </ul>

## role

---

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

### Virtual Chassis Fabric

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

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

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

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

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

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

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

### EX8200 Virtual Chassis

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



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

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

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

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

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

(All Virtual Chassis composed of EX Series switches, except EX8200 switches, or QFX Series devices) Specify two and only two members as **routing-engine**. The software determines which of the two members assigned the **routing-engine** role functions as master, based on the master election algorithm. See [“Understanding How the Master in a Virtual Chassis Is Elected” on page 16](#). In these Virtual Chassis, the **routing-engine** role is associated with a switch.

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

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

**Related  
Documentation**

- *Autoprovisioning a Virtual Chassis Fabric*
- *Preprovisioning a Virtual Chassis Fabric*
- *Example: Configuring an EX4200 Virtual Chassis Using a Preprovisioned Configuration File*
- *Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines*
- *Configuring an EX3300 Virtual Chassis (CLI Procedure)*
- *Configuring an EX4200, EX4500, or EX4550 Virtual Chassis (CLI Procedure)*
- *Configuring an EX8200 Virtual Chassis (CLI Procedure)*
- *Configuring an EX9200 Virtual Chassis*
- *Configuring a QFX Series Virtual Chassis (CLI Procedure)*
- *Configuring a Virtual Chassis on an EX Series Switch (J-Web Procedure)*
- *Adding a New EX4200 Switch to an Existing EX4200 Virtual Chassis (CLI Procedure)*
- [Replacing a Member Switch of a Virtual Chassis Configuration \(CLI Procedure\) on page 38](#)

## serial-number

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

## serial-number (Virtual Chassis aliases)

---

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

## traceoptions (Virtual Chassis)

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

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

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

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

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



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

**Default** Tracing operations are disabled.

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



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

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

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

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

**Range:** 2 through 1000

**Default:** 3 files

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

- **all**—All tracing operations.



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

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

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

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

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

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

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

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

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

**Range:** 10 KB through 1 GB

**Default:** 128 KB


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

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

<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 57</a></li><li>• <a href="#">Verifying the Member ID, Role, and Neighbor Member Connections of a Virtual Chassis Member on page 54</a></li><li>• <a href="#">Verifying That Virtual Chassis Ports Are Operational on page 55</a></li><li>• <a href="#">Verifying Virtual Chassis Ports in an EX8200 Virtual Chassis</a></li><li>• <a href="#">Troubleshooting an EX Series Virtual Chassis on page 60</a></li><li>• <a href="#">Troubleshooting Virtual Chassis Fabric</a></li></ul>
------------------------------	--

## vcp-no-hold-time

---

<b>Syntax</b>	vcp-no-hold-time;
<b>Hierarchy Level</b>	[edit <a href="#">virtual-chassis</a> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 13.2X50-D10 for EX Series switches. Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series.
<b>Description</b>	<p>Disable the Virtual Chassis port (VCP) holddown timer for all VCPs in the Virtual Chassis or Virtual Chassis Fabric (VCF).</p> <p>The VCP holddown timer is an internal mechanism that delays a Virtual Chassis reconvergence for several seconds when a VCP becomes inactive. The purpose of this delay is to provide the VCP time to return online without having to reconverge the Virtual Chassis to adjust to the inactive VCP. All traffic to the VCP is dropped while the VCP is inactive. If the VCP remains down for a time that exceeds the VCP holddown timer, a Virtual Chassis reconvergence occurs.</p> <p>When this statement is enabled, the VCP holddown timer is disabled and the Virtual Chassis reconvergence occurs when a VCP becomes inactive. The period of time where traffic is dropped waiting for the VCP to return online is avoided.</p> <p>We recommend enabling this statement after a Virtual Chassis is operational. We recommend disabling this statement when you are adding or removing member switches from your Virtual Chassis.</p> <p>The VCP holddown timer cannot be viewed and is not user-configurable. You can only control whether the VCP holddown timer is enabled or disabled by configuring this statement.</p> <div> <b>NOTE:</b> For the EX4300 Virtual Chassis, you should enable the <code>vcp-no-hold-time</code> statement before performing a software upgrade using NSSU. If you do not enable the <code>vcp-no-hold-time</code> statement, the Virtual Chassis may split during the upgrade. A split Virtual Chassis can cause disruptions to your network, and you may have to manually reconfigure your Virtual Chassis after the NSSU if the split and merge feature was disabled. For more information about a split Virtual Chassis, see <a href="#">“Understanding Split and Merge in a Virtual Chassis”</a> on page 23</div>
<b>Default</b>	The VCP holddown timer is enabled by default on all devices that support this statement.
<b>Required Privilege Level</b>	system—To view this statement in the configuration. system-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <i>Understanding EX4300 Virtual Chassis</i></li><li>• <i>Understanding QFX Series Virtual Chassis</i></li></ul>



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

## vcp-snmp-statistics

<b>Syntax</b>	vcp-snmp-statistics;
<b>Hierarchy Level</b>	[edit <a href="#">virtual-chassis</a> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 14.1X53-D15 for EX Series switches. Statement introduced in Junos OS Release 14.1X53-D15 for the QFX Series.
<b>Description</b>	<p>Enable SNMP monitoring of the Virtual Chassis ports (VCPs) for all VCPs in the Virtual Chassis or Virtual Chassis Fabric (VCF).</p> <p>When this statement is enabled, SNMP gathers statistics on the Junos VCP MIBs. You can retrieve the statistics gathered by SNMP for these MIBs by using the <a href="#">show snmp mib</a> command with the <b>walk</b> and <b>ascii</b> options and specifying <b>jnxVirtualChassisPortInPkts</b>.</p>
<b>Default</b>	SNMP is disabled by default on devices running Junos OS.
<b>Required Privilege Level</b>	<p>system—To view this statement in the configuration.</p> <p>system-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>Understanding EX4300 Virtual Chassis</i></li> <li>• <i>Understanding QFX Series Virtual Chassis</i></li> <li>• <a href="#">Understanding EX Series Virtual Chassis Components on page 5</a></li> <li>• <i>Understanding QFX Series Virtual Chassis Components</i></li> </ul>

## virtual-chassis

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

**Hierarchy Level** [edit]

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

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

The remaining statements are explained separately.

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

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

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

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

<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <i>Autoprovisioning a Virtual Chassis Fabric</i></li><li>• <i>Preprovisioning a Virtual Chassis Fabric</i></li><li>• <i>Adding a Device to a Virtual Chassis Fabric</i></li><li>• <i>Configuring a QFX Series Virtual Chassis (CLI Procedure)</i></li><li>• <i>Example: Configuring an EX3300 Virtual Chassis with a Master and Backup</i></li><li>• <i>Example: Configuring an EX4200 Virtual Chassis with a Master and Backup in a Single Wiring Closet</i></li><li>• <i>Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines</i></li><li>• <i>Configuring an EX3300 Virtual Chassis (CLI Procedure)</i></li><li>• <i>Configuring an EX4200, EX4500, or EX4550 Virtual Chassis (CLI Procedure)</i></li><li>• <i>Configuring an EX8200 Virtual Chassis (CLI Procedure)</i></li><li>• <i>Configuring an EX9200 Virtual Chassis</i></li></ul>
------------------------------	---



## CHAPTER 3

# Operational Commands for Virtual Chassis

- `clear virtual-chassis vc-port statistics`
- `request session member`
- `request virtual-chassis recycle`
- `request virtual-chassis renumber`
- `request virtual-chassis vc-port`
- `show snmp mib`
- `show virtual-chassis active-topology`
- `show virtual-chassis device-topology`
- `show virtual-chassis protocol adjacency`
- `show virtual-chassis protocol database`
- `show virtual-chassis protocol interface`
- `show virtual-chassis protocol route`
- `show virtual-chassis protocol statistics`
- `show virtual-chassis login`
- `show virtual-chassis`
- `show virtual-chassis vc-path`
- `show virtual-chassis vc-port`
- `show virtual-chassis vc-port statistics`

## clear virtual-chassis vc-port statistics

---

<b>Syntax</b>	<code>clear virtual-chassis vc-port statistics</code> <code>&lt;all-members&gt;</code> <code>&lt;interface-name&gt;</code> <code>&lt;local&gt;</code> <code>&lt;member member-id&gt;</code>
<b>Release Information</b>	Command introduced in Junos OS Release 9.0 for EX Series switches. The options <b>all-members</b> and <b>local</b> were added in Junos OS Release 9.3 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric.
<b>Description</b>	Clear—reset to zero (0)—the traffic statistics counters on Virtual Chassis ports (VCPs).
<b>Options</b>	<b>none</b> —Clear traffic statistics for VCPs of all members of a Virtual Chassis or VCF.  <b>all-members</b> —(Optional) Clear traffic statistics for VCPs of all members of a Virtual Chassis or VCF.  <b>interface-name</b> —(Optional) Clear traffic statistics for the specified VCP.  <b>local</b> —(Optional) Clear traffic statistics for VCPs from the switch or external Routing Engine on which this command is entered.  <b>member member-id</b> —(Optional) Clear traffic statistics for VCPs from the specified member of a Virtual Chassis or VCF.
<b>Required Privilege Level</b>	clear
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">show virtual-chassis vc-port statistics on page 146</a></li><li>• <a href="#">show virtual-chassis vc-port on page 142</a></li><li>• <a href="#">Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 57</a></li></ul>
<b>List of Sample Output</b>	<a href="#">clear virtual-chassis vc-port statistics (EX4200 Virtual Chassis) on page 94</a> <a href="#">clear virtual-chassis vc-port statistics (EX8200 Virtual Chassis) on page 95</a> <a href="#">clear virtual-chassis vc-port statistics member 3 on page 95</a>

### Sample Output

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

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

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

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

```
member0:
```

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

```
member1:
```

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

```
member8:
```

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

```
member9:
```

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

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

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

```
Cleared statistics on member 3
```


## request session member

---

<b>Syntax</b>	<code>request session member <i>member-id</i></code>
<b>Release Information</b>	Command introduced in Junos OS Release 9.0 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
<b>Description</b>	Start a session with the specified member of a Virtual Chassis or a VCF.
<b>Options</b>	<i>member-id</i> —Member ID for the specific member of the Virtual Chassis or VCF.
<b>Required Privilege Level</b>	maintenance
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">member on page 75</a></li><li>• <a href="#">Understanding EX Series Virtual Chassis Components on page 5</a></li><li>• <a href="#">Understanding QFX Series Virtual Chassis Components</a></li></ul>



## request virtual-chassis recycle

<b>Syntax</b>	<code>request virtual-chassis recycle member-id <i>member-id</i></code>
<b>Release Information</b>	Command introduced in Junos OS Release 9.0 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series.
<b>Description</b>	<p>Make a previously used member ID available for reassignment.</p> <p>When you remove a member switch from the Virtual Chassis configuration, the master reserves that member ID. To make the member ID available for reassignment, you must use this command.</p>
	<div>  <p><b>NOTE:</b> You must run this command from the Virtual Chassis member in the master role.</p> </div>
<b>Options</b>	<code>member-id <i>member-id</i></code> —Specify the member ID that you want to make available for reassignment to a different member.
<b>Required Privilege Level</b>	system-control
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">request virtual-chassis renumber on page 98</a></li> <li>• <a href="#">Replacing a Member Switch of a Virtual Chassis Configuration (CLI Procedure) on page 38</a></li> <li>• <a href="#">Adding or Replacing a Member Switch or an External Routing Engine in an EX8200 Virtual Chassis (CLI Procedure)</a></li> </ul>
<b>List of Sample Output</b>	<a href="#">request virtual-chassis recycle member-id 3 on page 97</a> <a href="#">request virtual-chassis recycle member-id 1 on page 97</a>

### Sample Output

`request virtual-chassis recycle member-id 3`

```
user@switch> request virtual-chassis recycle member-id 3
```


### Sample Output

`request virtual-chassis recycle member-id 1`

```
user@external-routing-engine> request virtual-chassis recycle member-id 1
```

## request virtual-chassis renumber

---

Syntax	<code>request virtual-chassis renumber member-id <i>old-member-id</i> new-member-id <i>new-member-id</i></code>
Release Information	Command introduced in Junos OS Release 9.0 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series.
Description	Renumber a member of a Virtual Chassis configuration.
<div> <b>NOTE:</b> You must run this command from the Virtual Chassis member in the master role.</div>	
Options	<code>member-id <i>old-member-id</i></code> —Specify the ID of the member that you wish to renumber. <code>new-member-id <i>new-member-id</i></code> —Specify an unassigned member ID.
Required Privilege Level	system-control
Related Documentation	<ul style="list-style-type: none"><li>• <a href="#">request virtual-chassis recycle on page 97</a></li><li>• <a href="#">Replacing a Member Switch of a Virtual Chassis Configuration (CLI Procedure) on page 38</a></li><li>• <a href="#">Adding or Replacing a Member Switch or an External Routing Engine in an EX8200 Virtual Chassis (CLI Procedure)</a></li></ul>
List of Sample Output	<a href="#">request virtual-chassis renumber member-id 5 new-member-id 4 on page 98</a> <a href="#">request virtual-chassis renumber member-id 1 new-member-id 0 on page 98</a>

### Sample Output

`request virtual-chassis renumber member-id 5 new-member-id 4`

```
user@switch> request virtual-chassis renumber member-id 5 new-member-id 4
```

`request virtual-chassis renumber member-id 1 new-member-id 0`

```
user@external-routing-engine> request virtual-chassis renumber member-id 1 new-member-id 0
```

## request virtual-chassis vc-port

<b>Syntax</b>	<code>request virtual-chassis vc-port set   delete &lt;fpc-slot <i>fpc-slot</i>&gt; pic-slot <i>pic-slot</i> port <i>port-number</i> &lt;member <i>member-id</i>&gt;</code>
<b>Release Information</b>	<p>Command introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Option <b>fpc-slot</b> introduced in Junos OS Release 10.4 for EX Series switches.</p> <p>Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series.</p> <p>Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).</p>
<b>Description</b>	<p>Enable or disable an optical port as a Virtual Chassis port (VCP).</p> <p>If you omit <b>member <i>member-id</i></b>, this command defaults to enabling or disabling the uplink VCP or SFP network port configured as a VCP on the switch where the command is issued.</p> <p>On an EX3300 switch, uplink ports 2 and 3 are configured as VCPs by default. No other uplink ports on any other EX Series switches are configured as VCPs by default.</p> <p>You might experience a temporary traffic disruption immediately after creating or deleting a user-configured VCP in an EX8200 Virtual Chassis.</p>
<b>Options</b>	<p><b>pic-slot <i>pic-slot</i></b>—Number of the PIC slot for the port on the switch.</p> <p><b>port <i>port-number</i></b>—Number of the port that is to be enabled or disabled as a VCP.</p> <p><b>member <i>member-id</i></b>—(Optional) Enable or disable the specified VCP on the specified member of the Virtual Chassis or VCF.</p>
<b>Required Privilege Level</b>	system-control
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">request virtual-chassis vc-port</a> (dedicated port)</li> <li>• <a href="#">show virtual-chassis vc-port on page 142</a></li> <li>• <a href="#">show virtual-chassis vc-port statistics on page 146</a></li> <li>• <a href="#">clear virtual-chassis vc-port statistics on page 94</a></li> <li>• <a href="#">Virtual Chassis Port (VCP) Interface Names in an EX8200 Virtual Chassis</a></li> <li>• <a href="#">Understanding EX Series Virtual Chassis Components on page 5</a></li> <li>• <a href="#">Understanding QFX Series Virtual Chassis Components</a></li> </ul>
<b>List of Sample Output</b>	<p><a href="#">request virtual-chassis vc-port set pic-slot 1 port 0 on page 100</a></p> <p><a href="#">request virtual-chassis vc-port set pic-slot 1 port 1 member 3 on page 100</a></p> <p><a href="#">request virtual-chassis vc-port delete pic-slot 1 port 1 member 3 on page 100</a></p>

## Sample Output

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

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

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

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

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

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

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

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

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

## show snmp mib

<b>Syntax</b>	<code>show snmp mib (get   get-next   walk) (ascii   decimal) <i>object-id</i></code>
<b>Release Information</b>	<p>Command introduced before Junos OS Release 7.4.</p> <p>Command introduced in Junos OS Release 9.0 for EX Series switches.</p> <p><b>ascii</b> and <b>decimal</b> options introduced in Junos OS Release 9.6.</p> <p><b>ascii</b> and <b>decimal</b> options introduced in Junos OS Release 9.6 for EX Series switches.</p> <p>Command introduced in Junos OS Release 11.1 for the QFX Series.</p> <p>Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.</p>
<b>Description</b>	Display local Simple Network Management Protocol (SNMP) Management Information Base (MIB) object values.
<b>Options</b>	<p><b>get</b>—Retrieve and display one or more SNMP object values.</p> <p><b>get-next</b>—Retrieve and display the next SNMP object values.</p> <p><b>walk</b>—Retrieve and display the SNMP object values that are associated with the requested object identifier (OID). When you use this option, the Junos OS displays the objects below the subtree that you specify.</p> <p><b>ascii</b>—Display the SNMP object's string indices as an ASCII-key representation.</p> <p><b>decimal</b>—Display the SNMP object values in the decimal (default) format. The <b>decimal</b> option is the default option for this command. Therefore, issuing the <b>show snmp mib (get   get-next   walk) decimal object-id</b> and the <b>show snmp mib (get   get-next   walk) object-id</b> commands display the same output.</p> <p><b>object-id</b>—The object can be represented by a sequence of dotted integers (such as 1.3.6.1.2.1.2) or by its subtree name (such as <b>interfaces</b>). When entering multiple objects, enclose the objects in quotation marks.</p>
<b>Required Privilege Level</b>	snmp—To view this statement in the configuration.
<b>List of Sample Output</b>	<p><a href="#">show snmp mib get on page 102</a></p> <p><a href="#">show snmp mib get (Multiple Objects) on page 102</a></p> <p><a href="#">show snmp mib get (Layer 2 Policer) on page 102</a></p> <p><a href="#">show snmp mib get-next on page 102</a></p> <p><a href="#">show snmp mib get-next (Specify an OID) on page 102</a></p> <p><a href="#">show snmp mib walk on page 102</a></p> <p><a href="#">show snmp mib walk (QFX Series) on page 102</a></p> <p><a href="#">show snmp mib walk decimal on page 103</a></p> <p><a href="#">show snmp mib walk (ASCII) on page 103</a></p> <p><a href="#">show snmp mib walk (Multiple Indices) on page 103</a></p> <p><a href="#">show snmp mib walk decimal (Multiple Indices) on page 103</a></p>
<b>Output Fields</b>	<a href="#">Table 10 on page 102</a> describes the output fields for the <b>show snmp mib</b> command. Output fields are listed in the approximate order in which they appear.

Table 10: show snmp mib Output Fields

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

## Sample Output

### show snmp mib get

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

### show snmp mib get (Multiple Objects)

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

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

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

### show snmp mib get-next

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

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

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

### show snmp mib walk

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

### show snmp mib walk (QFX Series)

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

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

#### show snmp mib walk decimal

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

#### show snmp mib walk (ASCII)

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

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

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

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

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

## show virtual-chassis active-topology

<b>Syntax</b>	show virtual-chassis active-topology <all-members> <local> <member <i>member-id</i> >
<b>Release Information</b>	Command introduced in Junos OS Release 9.0 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
<b>Description</b>	Display the active topology of the Virtual Chassis or VCF with next-hop reachability information.
<b>Options</b>	<p><b>none</b>—Display the active topology of the member switch where the command is issued.</p> <p><b>all-members</b>—(Optional) Display the active topology of all members of the Virtual Chassis or VCF.</p> <p><b>local</b>—(Optional) Display the active topology of the switch or external Routing Engine on which this command is entered.</p> <p><b>member <i>member-id</i></b>—(Optional) Display the active topology of the specified member of the Virtual Chassis or VCF.</p>
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 57</a></li> <li>• <a href="#">Understanding EX Series Virtual Chassis Configuration</a></li> </ul>
<b>List of Sample Output</b>	<a href="#">show virtual-chassis active-topology (EX4200 Virtual Chassis) on page 105</a> <a href="#">show virtual-chassis active-topology (EX8200 Virtual Chassis) on page 105</a> <a href="#">show virtual-chassis active-topology (Virtual Chassis Fabric) on page 106</a>
<b>Output Fields</b>	Table 11 on page 104 lists the output fields for the <b>show virtual-chassis active-topology</b> command. Output fields are listed in the approximate order in which they appear.

Table 11: show virtual-chassis active-topology Output Fields

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



## Sample Output

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

```

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

 2                      1(vcp-1)

 3                      1(vcp-1)

 4                      1(vcp-1)

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

 6                      8(vcp-0)

 7                      8(vcp-0)

 8                      8(vcp-0)

```

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

```

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

```

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

```
member1:
```

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

```
member8:
```

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

member9:

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

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

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

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

fpc1:

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

fpc2:

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

fpc3:

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

fpc4:

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

fpc5:

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

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

fpc6:

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

## show virtual-chassis device-topology

<b>Syntax</b>	show virtual-chassis device-topology <all-members> <local> <member <i>member-id</i> >
<b>Release Information</b>	Command introduced in Junos OS Release 10.4 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
<b>Description</b>	Display the device topology—the member and system IDs, the VCP numbers, and device status—for all hardware devices in the Virtual Chassis or VCF.
<b>Options</b>	<p><b>none</b>—Display the device topology for all members of the Virtual Chassis or VCF.</p> <p><b>all-members</b>—(Optional) Display the device topology for all members of the Virtual Chassis or VCF.</p> <p><b>local</b>—(Optional) Display the device topology for the switch or external Routing Engine on which this command is entered.</p> <p><b>member <i>member-id</i></b>—(Optional) Display the device topology for the specified member of the Virtual Chassis or VCF.</p>
<b>Required Privilege Level</b>	clear
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Understanding EX Series Virtual Chassis Port Link Aggregation on page 21</a></li> <li>• <a href="#">Understanding EX8200 Virtual Chassis Topologies</a></li> </ul>
<b>Output Fields</b>	<a href="#">Table 12 on page 109</a> lists the output fields for the <b>show virtual-chassis device-topology</b> command. Output fields are listed in the approximate order in which they appear.

**Table 12: show virtual-chassis device-topology Output Fields**

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

Table 12: show virtual-chassis device-topology Output Fields (*continued*)

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

## Sample Output

### show virtual-chassis device-topology

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

```
member0:
```

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

```
member1:
```

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

```
member8:
```

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

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

member9:

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

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

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

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

fpc1:

Neighbor List

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

fpc2:

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

fpc3:

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



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

fpc4:

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

fpc5:

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

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

fpc6:

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

## show virtual-chassis protocol adjacency

<b>Syntax</b>	<pre>show virtual-chassis protocol adjacency &lt;brief   detail   extensive&gt; &lt;all-members&gt; &lt;local&gt; &lt;member member-id&gt; &lt;system-id&gt;</pre>
<b>Release Information</b>	<p>Command introduced in Junos OS Release 10.4 for EX Series switches.</p> <p>Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series.</p> <p>Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).</p>
<b>Description</b>	Display the Virtual Chassis Control Protocol (VCCP) adjacency statistics in the Virtual Chassis or VCF for all hardware devices.
<b>Options</b>	<p><b>none</b>—Display VCCP adjacency statistics in brief form for all members of the Virtual Chassis or VCF.</p> <p><b>brief   detail   extensive</b>—(Optional) Display the specified level of output. Using the <b>brief</b> option is equivalent to entering the command with no options (the default). The <b>detail</b> and <b>extensive</b> options provide identical displays.</p> <p><b>all-members</b>—(Optional) Display VCCP adjacency statistics in brief form for all members of the Virtual Chassis or VCF.</p> <p><b>local</b>—(Optional) Display VCCP adjacency statistics for the switch or external Routing Engine on which this command is entered.</p> <p><b>member member-id</b>—(Optional) Display VCCP adjacency statistics for the specified member of the Virtual Chassis or VCF.</p> <p><b>system-id</b>—(Optional) Display VCCP adjacency statistics for the specified member of the Virtual Chassis or VCF.</p>
<b>Required Privilege Level</b>	clear
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Understanding EX Series Virtual Chassis Port Link Aggregation on page 21</a></li> <li>• <a href="#">Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis</a></li> </ul>
<b>List of Sample Output</b>	<p><a href="#">show virtual-chassis protocol adjacency on page 116</a></p> <p><a href="#">show virtual-chassis protocol adjacency detail on page 117</a></p>
<b>Output Fields</b>	Table 13 on page 116 lists the output fields for the <b>show virtual-chassis protocol adjacency</b> command. Output fields are listed in the approximate order in which they appear.

Table 13: show virtual-chassis protocol adjacency Output Fields

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

## Sample Output

### show virtual-chassis protocol adjacency

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

```
member0:
```

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

```
member1:
```

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

```
member8:
```

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

```
member9:
```

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

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

### show virtual-chassis protocol adjacency detail

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

```
member0:
```

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

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

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

```
member1:
```

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

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

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

```
member8:
```

```
-----
0000.73e9.9a57
  interface-name: vcp-1/0.32768, State: Up, Expires in 58 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 19:26:38 ago
```

```
0021.59f7.d000
  interface-name: vcp-1/1.32768, State: Up, Expires in 59 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 19:26:38 ago
```

```
0026.888d.6800
  interface-name: vcp-1/2.32768, State: Up, Expires in 59 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 19:26:38 ago
```

```
0021.59f7.d000
  interface-name: vcp-2/0.32768, State: Up, Expires in 57 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 19:26:38 ago
```

```
member9:
```

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

```
0026.888d.6800
  interface-name: vcp-1/1.32768, State: Up, Expires in 58 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 22:17:36 ago
```

## show virtual-chassis protocol database

<b>Syntax</b>	show virtual-chassis protocol database <brief   detail   extensive> <all-members> <local> <member <i>member-id</i> >
<b>Release Information</b>	Command introduced in Junos OS Release 10.4 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
<b>Description</b>	Display the Virtual Chassis Control Protocol (VCCP) database statistics for all hardware devices within the Virtual Chassis or VCF.
<b>Options</b>	<p><b>none</b>—Display VCCP database statistics in brief form for all members of the Virtual Chassis or VCF.</p> <p><b>brief   detail   extensive</b>—(Optional) Display the specified level of output. Using the <b>brief</b> option is equivalent to entering the command with no options (the default). The <b>detail</b> option provides more output than the <b>brief</b> option. The <b>extensive</b> option provides all output and is most useful for customer support personnel.</p> <p><b>all-members</b>—(Optional) Display VCCP database statistics in brief form for all members of the Virtual Chassis or VCF.</p> <p><b>local</b>—(Optional) Display VCCP database statistics for the switch or external Routing Engine on which this command is entered.</p> <p><b>member <i>member-id</i></b>—(Optional) Display VCCP database statistics for the specified member of the Virtual Chassis or VCF.</p>
<b>Required Privilege Level</b>	clear
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis</i></li> <li>• <a href="#">Understanding EX Series Virtual Chassis Components on page 5</a></li> <li>• <i>Understanding QFX Series Virtual Chassis Components</i></li> </ul>
<b>List of Sample Output</b>	<a href="#">show virtual-chassis protocol database on page 120</a> <a href="#">show virtual-chassis protocol database detail on page 121</a>
<b>Output Fields</b>	<a href="#">Table 14 on page 119</a> lists the output fields for the <b>show virtual-chassis protocol database</b> command. Output fields are listed in the approximate order in which they appear.

Table 14: show virtual-chassis protocol database Output Fields

Field Name	Field Description	Level of Output
LSP ID	Link-state protocol (LSP) data unit identifier.	All levels

Table 14: show virtual-chassis protocol database Output Fields (*continued*)

Field Name	Field Description	Level of Output
<b>Sequence</b>	Sequence number of the LSP.	All levels
<b>Checksum</b>	Checksum value of the LSP.	All levels
<b>Lifetime</b>	Remaining lifetime of the LSP, in seconds.	All levels
<b>Neighbor</b>	MAC address of the neighbor on the advertising system.	detail
<b>Interface</b>	Virtual Chassis port (VCP) interface name.	detail
<b>Metric</b>	Metric of the prefix or neighbor.	detail

The **extensive** output was omitted from this list. The **extensive** output is useful for customer support personnel only.

## Sample Output

### show virtual-chassis protocol database

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

```
member0:
```

```
-----
LSP ID          Sequence Checksum Lifetime
0000.4a75.9b7c.00-00  0x1dd80  0xc2e3   116
0000.73e9.9a57.00-00  0xf361   0x27e8   113
0021.59f7.d000.00-00  0x16882  0x3993   118
0026.888d.6800.00-00  0x1691f  0x82b7   116
  4 LSPs
```

```
member1:
```

```
-----
LSP ID          Sequence Checksum Lifetime
0000.4a75.9b7c.00-00  0x1dd80  0xc2e3   116
0000.73e9.9a57.00-00  0xf361   0x27e8   114
0021.59f7.d000.00-00  0x16883  0x289    116
0026.888d.6800.00-00  0x1691f  0x82b7   118
  4 LSPs
```

```
member8:
```

```
-----
LSP ID          Sequence Checksum Lifetime
0000.4a75.9b7c.00-00  0x1dd80  0xc2e3   118
0000.73e9.9a57.00-00  0xf361   0x27e8   114
0021.59f7.d000.00-00  0x16883  0x289    116
0026.888d.6800.00-00  0x16920  0xa335   116
  4 LSPs
```

```
member9:
```

```
-----
LSP ID          Sequence Checksum Lifetime
0000.4a75.9b7c.00-00  0x1dd80  0xc2e3   116
0000.73e9.9a57.00-00  0xf361   0x27e8   116
0021.59f7.d000.00-00  0x16883  0x289    114
```



```
0026.888d.6800.00-00      0x16920   0xa335      116
4 LSPs
```

### show virtual-chassis protocol database detail

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

```
member0:
```

```
-----
0000.4a75.9b7c.00-00 Sequence: 0x1ddbc, Checksum: 0x3111, Lifetime: 115 secs
Neighbor: 0000.73e9.9a57.00 Interface: vcp-1/0.32768 Metric: 150
Neighbor: 0021.59f7.d000.00 Interface: vcp-1/1.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-1/2.32768 Metric: 150
```

```
0000.73e9.9a57.00-00 Sequence: 0xf381, Checksum: 0xe065, Lifetime: 114 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-1/0.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-1/1.32768 Metric: 150
```

```
0021.59f7.d000.00-00 Sequence: 0x168af, Checksum: 0x8b0b, Lifetime: 118 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-0/0.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-4/0/1.32768 Metric: 15
```

```
0026.888d.6800.00-00 Sequence: 0x1694e, Checksum: 0xca97, Lifetime: 115 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-0/0.32768 Metric: 150
Neighbor: 0000.73e9.9a57.00 Interface: vcp-0/1.32768 Metric: 150
Neighbor: 0021.59f7.d000.00 Interface: vcp-3/0/4.32768 Metric: 15
```

```
member1:
```

```
-----
0000.4a75.9b7c.00-00 Sequence: 0x1ddbc, Checksum: 0x3111, Lifetime: 115 secs
Neighbor: 0000.73e9.9a57.00 Interface: vcp-1/0.32768 Metric: 150
Neighbor: 0021.59f7.d000.00 Interface: vcp-1/1.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-1/2.32768 Metric: 150
```

```
0000.73e9.9a57.00-00 Sequence: 0xf381, Checksum: 0xe065, Lifetime: 116 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-1/0.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-1/1.32768 Metric: 150
```

```
0021.59f7.d000.00-00 Sequence: 0x168af, Checksum: 0x8b0b, Lifetime: 116 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-0/0.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-4/0/1.32768 Metric: 15
```

```
0026.888d.6800.00-00 Sequence: 0x1694e, Checksum: 0xca97, Lifetime: 117 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-0/0.32768 Metric: 150
Neighbor: 0000.73e9.9a57.00 Interface: vcp-0/1.32768 Metric: 150
Neighbor: 0021.59f7.d000.00 Interface: vcp-3/0/4.32768 Metric: 15
```

```
member8:
```

```
-----
0000.4a75.9b7c.00-00 Sequence: 0x1ddbd, Checksum: 0xfd83, Lifetime: 118 secs
Neighbor: 0000.73e9.9a57.00 Interface: vcp-1/0.32768 Metric: 150
Neighbor: 0021.59f7.d000.00 Interface: vcp-1/1.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-1/2.32768 Metric: 150
```

```
0000.73e9.9a57.00-00 Sequence: 0xf381, Checksum: 0xe065, Lifetime: 115 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-1/0.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-1/1.32768 Metric: 150
```

```
0021.59f7.d000.00-00 Sequence: 0x168af, Checksum: 0x8b0b, Lifetime: 116 secs
```

```
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-0/0.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-4/0/1.32768 Metric: 15

0026.888d.6800.00-00 Sequence: 0x1694e, Checksum: 0xca97, Lifetime: 115 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-0/0.32768 Metric: 150
Neighbor: 0000.73e9.9a57.00 Interface: vcp-0/1.32768 Metric: 150
Neighbor: 0021.59f7.d000.00 Interface: vcp-3/0/4.32768 Metric: 15

member9:
-----

0000.4a75.9b7c.00-00 Sequence: 0x1ddbd, Checksum: 0xfd83, Lifetime: 116 secs
Neighbor: 0000.73e9.9a57.00 Interface: vcp-1/0.32768 Metric: 150
Neighbor: 0021.59f7.d000.00 Interface: vcp-1/1.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-1/2.32768 Metric: 150

0000.73e9.9a57.00-00 Sequence: 0xf381, Checksum: 0xe065, Lifetime: 117 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-1/0.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-1/1.32768 Metric: 150

0021.59f7.d000.00-00 Sequence: 0x168af, Checksum: 0x8b0b, Lifetime: 113 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-0/0.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-4/0/1.32768 Metric: 15

0026.888d.6800.00-00 Sequence: 0x1694f, Checksum: 0xa61a, Lifetime: 116 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-0/0.32768 Metric: 150
Neighbor: 0000.73e9.9a57.00 Interface: vcp-0/1.32768 Metric: 150
Neighbor: 0021.59f7.d000.00 Interface: vcp-3/0/4.32768 Metric: 15
```

## show virtual-chassis protocol interface

<b>Syntax</b>	<pre>show virtual-chassis protocol interface &lt;brief   detail&gt; &lt;all-members&gt; &lt;interface-name&gt; &lt;local&gt; &lt;member member-id&gt;</pre>
<b>Release Information</b>	<p>Command introduced in Junos OS Release 10.4 for EX Series switches.</p> <p>Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series.</p> <p>Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).</p>
<b>Description</b>	Display information about Virtual Chassis Control Protocol (VCCP) statistics for VCCP-enabled interfaces within the Virtual Chassis or VCF.
<b>Options</b>	<p><b>none</b>—Display the VCCP interface statistics in brief form for all members of the Virtual Chassis or VCF.</p> <p><b>brief   detail</b> —(Optional) Display the specified level of output. Using the <b>brief</b> option is equivalent to entering the command with no options (the default). The <b>detail</b> option provides more output than the <b>brief</b> option.</p> <p><b>all-members</b>—(Optional) Display VCCP interface statistics for all members of the Virtual Chassis or VCF.</p> <p><b>interface-name</b>—(Optional) Display VCCP interface statistics for the specified interface.</p> <p><b>local</b>—(Optional) Display VCCP interface statistics for the switch or external Routing Engine on which this command is entered.</p> <p><b>member member-id</b>—(Optional) Display VCCP interface statistics for the specified member of the Virtual Chassis or VCF.</p>
<b>Required Privilege Level</b>	clear
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>EX Series Virtual Chassis Overview</i></li> <li>• <i>Understanding QFX Series Virtual Chassis</i></li> <li>• <i>Understanding Virtual Chassis Ports in an EX8200 Virtual Chassis</i></li> <li>• <i>Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis</i></li> </ul>
<b>List of Sample Output</b>	<a href="#">show virtual-chassis protocol interface on page 124</a>
<b>Output Fields</b>	<a href="#">Table 15 on page 124</a> lists the output fields for the <b>show virtual-chassis protocol interface</b> command. Output fields are listed in the approximate order in which they appear.

Table 15: show virtual-chassis protocol interface Output Fields

Field Name	Field Description	Level of Output
<b>Interface</b>	Name of the VCP.	All levels
<b>State</b>	State of the link. Outputs include: <ul style="list-style-type: none"> <li>• <b>Up</b>—The link is up.</li> <li>• <b>Down</b>—The link is down.</li> </ul>	All levels
<b>Metric</b>	Metric of the prefix or neighbor.	All levels

## Sample Output

### show virtual-chassis protocol interface

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

```
member0:
```

```
-----
```

```
IS-IS interface database:
```

Interface	State	Metric
vcp-0/0.32768	Up	150
vcp-0/1.32768	Up	150
vcp-4/0/1.32768	Up	15
vcp-4/0/7.32768	Down	15

```
member1:
```

```
-----
```

```
IS-IS interface database:
```

Interface	State	Metric
vcp-0/0.32768	Up	150
vcp-0/1.32768	Up	150
vcp-3/0/4.32768	Up	15

```
member8:
```

```
-----
```

```
IS-IS interface database:
```

Interface	State	Metric
vcp-0/0.32768	Down	150
vcp-1/0.32768	Up	150
vcp-1/1.32768	Up	150
vcp-1/2.32768	Up	150
vcp-1/3.32768	Down	150
vcp-2/0.32768	Up	150
vcp-2/1.32768	Down	150
vcp-2/2.32768	Down	150
vcp-2/3.32768	Down	150

```
member9:
```

```
-----
```

```
IS-IS interface database:
```

Interface	State	Metric
vcp-0/0.32768	Down	150
vcp-1/0.32768	Up	150
vcp-1/1.32768	Up	150
vcp-1/2.32768	Down	150
vcp-1/3.32768	Down	150



## show virtual-chassis protocol route

<b>Syntax</b>	show virtual-chassis protocol route <all-members> <destination-id> <local> <member member-id>
<b>Release Information</b>	Command introduced in Junos OS Release 10.4 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
<b>Description</b>	Display the unicast and multicast Virtual Chassis Control Protocol (VCCP) routing tables within the Virtual Chassis or VCF.
<b>Options</b>	<p><b>none</b>—Display the unicast and multicast routing tables for all members of the Virtual Chassis.</p> <p><b>all-members</b>—(Optional) Display the unicast and multicast routing tables for all members of the Virtual Chassis or VCF.</p> <p><b>destination-id</b>—(Optional) Display the unicast and multicast routing tables to the specified destination member ID for each member of the Virtual Chassis or VCF.</p> <p><b>local</b>—(Optional) Display the unicast and multicast routing tables on the device where this command is entered.</p> <p><b>member member-id</b>—(Optional) Display the unicast and multicast routing tables for the specified member of the Virtual Chassis or VCF.</p>
<b>Required Privilege Level</b>	clear
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>EX Series Virtual Chassis Overview</i></li> <li>• <i>Understanding QFX Series Virtual Chassis</i></li> <li>• <i>Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis</i></li> </ul>
<b>List of Sample Output</b>	<a href="#">show virtual-chassis protocol route on page 127</a>
<b>Output Fields</b>	<a href="#">Table 16 on page 126</a> lists the output fields for the <b>show virtual-chassis protocol route</b> command. Output fields are listed in the approximate order in which they appear.

**Table 16: show virtual-chassis protocol route Output Fields**

Field Name	Field Description
<b>Dev</b>	MAC address of the member storing the VCCP routing table.
<b>Version</b>	Version of the shortest-path-first algorithm that generated the routing table.

Table 16: show virtual-chassis protocol route Output Fields (*continued*)

Field Name	Field Description
<b>System ID</b>	MAC address of the device.
<b>Version</b>	Version of the shortest-path-first (SPF) algorithm that generated the route.
<b>Metric</b>	The metric number to get to that device.
<b>Interface</b>	Name of the Virtual Chassis port (VCP) interface connecting the devices.
<b>Via</b>	MAC address of the next-hop device, if applicable.

## Sample Output

### show virtual-chassis protocol route

```

user@switch> show virtual-chassis protocol route
member0:
-----
Dev 0021.59f7.d000 ucast routing table          Current version: 21
-----
System ID      Version  Metric Interface  Via
0000.4a75.9b7c    21      150 vcp-0/1.32768 0000.4a75.9b7c
0000.73e9.9a57    21      165 vcp-4/0/1.32768 0026.888d.6800
0021.59f7.d000    21        0
0026.888d.6800    21      15 vcp-4/0/1.32768 0026.888d.6800

Dev 0021.59f7.d000 mcast routing table          Current version: 21
-----
System ID      Version  Metric Interface  Via
0000.4a75.9b7c    21
0000.73e9.9a57    21
0021.59f7.d000    21          vcp-4/0/1.32768
                   vcp-0/1.32768
0026.888d.6800    21

member1:
-----
Dev 0026.888d.6800 ucast routing table          Current version: 25
-----
System ID      Version  Metric Interface  Via
0000.4a75.9b7c    25      150 vcp-0/0.32768 0000.4a75.9b7c
0000.73e9.9a57    25      150 vcp-0/1.32768 0000.73e9.9a57
0021.59f7.d000    25      15 vcp-3/0/4.32768 0021.59f7.d000
0026.888d.6800    25        0

Dev 0026.888d.6800 mcast routing table          Current version: 25
-----
System ID      Version  Metric Interface  Via
0000.4a75.9b7c    25
0000.73e9.9a57    25          vcp-3/0/4.32768
0021.59f7.d000    25          vcp-0/1.32768
0026.888d.6800    25          vcp-3/0/4.32768
                   vcp-0/0.32768

```

vcp-0/1.32768

member8:

-----

Dev 0000.4a75.9b7c ucast routing table                      Current version: 39

-----

System ID	Version	Metric	Interface	Via
0000.4a75.9b7c	39	0		
0000.73e9.9a57	39	150	vcp-1/0.32768	0000.73e9.9a57
0021.59f7.d000	39	150	vcp-2/0.32768	0021.59f7.d000
0026.888d.6800	39	150	vcp-1/2.32768	0026.888d.6800

Dev 0000.4a75.9b7c mcast routing table                      Current version: 39

-----

System ID	Version	Metric	Interface	Via
0000.4a75.9b7c	39		vcp-1/0.32768	
			vcp-2/0.32768	
			vcp-1/2.32768	
0000.73e9.9a57	39			
0021.59f7.d000	39			
0026.888d.6800	39			

member9:

-----

Dev 0000.73e9.9a57 ucast routing table                      Current version: 31

-----

System ID	Version	Metric	Interface	Via
0000.4a75.9b7c	31	150	vcp-1/0.32768	0000.4a75.9b7c
0000.73e9.9a57	31	0		
0021.59f7.d000	31	165	vcp-1/1.32768	0026.888d.6800
0026.888d.6800	31	150	vcp-1/1.32768	0026.888d.6800

Dev 0000.73e9.9a57 mcast routing table                      Current version: 31

-----

System ID	Version	Metric	Interface	Via
0000.4a75.9b7c	31			
0000.73e9.9a57	31		vcp-1/0.32768	
			vcp-1/1.32768	
0021.59f7.d000	31			
0026.888d.6800	31			



## show virtual-chassis protocol statistics

<b>Syntax</b>	show virtual-chassis protocol statistics <all-members> <interface-name> <local> <member member-id>
<b>Release Information</b>	Command introduced in Junos OS Release 10.4 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
<b>Description</b>	Display the Virtual Chassis Control Protocol (VCCP) statistics for all hardware devices within the Virtual Chassis or VCF.
<b>Options</b>	<p><b>none</b>—Display VCCP statistics for all members of the Virtual Chassis or VCF.</p> <p><b>all-members</b>—(Optional) Display VCCP statistics for all members of the Virtual Chassis or VCF.</p> <p><b>interface-name</b>—(Optional) Display VCCP statistics for the specified interface.</p> <p><b>local</b>—(Optional) Display VCCP statistics for the switch or external Routing Engine on which this command is entered.</p> <p><b>member member-id</b>—(Optional) Display VCCP statistics for the specified member of the Virtual Chassis or VCF.</p>
<b>Required Privilege Level</b>	clear
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>EX Series Virtual Chassis Overview</i></li> <li>• <i>Understanding QFX Series Virtual Chassis</i></li> <li>• <i>Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis</i></li> </ul>
<b>List of Sample Output</b>	<a href="#">show virtual-chassis protocol statistics on page 130</a>
<b>Output Fields</b>	<a href="#">Table 17 on page 129</a> lists the output fields for the <b>show virtual-chassis protocol interface</b> command. Output fields are listed in the approximate order in which they appear.

**Table 17: show virtual-chassis protocol statistics Output Fields**

Field Name	Field Description
<b>PDU type</b>	Protocol data unit type.
<b>Received</b>	Number of PDUs received since VCCP started or since the statistics were set to zero.
<b>Processed</b>	Number of PDUs received minus the number of PDUs dropped.

Table 17: show virtual-chassis protocol statistics Output Fields (*continued*)

Field Name	Field Description
<b>Drops</b>	Number of PDUs dropped.
<b>Sent</b>	Number of PDUs transmitted since VCCP started or since the statistics were set to zero.
<b>Rexmit</b>	Number of PDUs retransmitted since VCCP started or since the statistics were set to zero.
<b>Total Packets Received</b>	Number of PDUs received since VCCP started or since the statistics were set to zero.
<b>Total Packets Sent</b>	Number of PDUs sent since VCCP started or since the statistics were set to zero.
<b>LSP queue length</b>	Number of link-state PDUs waiting in the queue for processing. This value is almost always 0.
<b>SPF runs</b>	Number of shortest-path-first (SPF) calculations that have been performed.
<b>Fragments Rebuilt</b>	Number of link-state PDU fragments that the local system has computed.
<b>LSP Regenerations</b>	Number of link-state PDUs that have been regenerated. A link-state PDU is regenerated when it is nearing the end of its lifetime and it has not changed.
<b>Purges initiated</b>	Number of purges that the system initiated. A purge is initiated if the software determines that a link-state PDU must be removed from the network.

## Sample Output

### show virtual-chassis protocol statistics

```

user@switch> show virtual-chassis protocol statistics
member0:
-----
IS-IS statistics for 0021.59f7.d000:
PDU type      Received    Processed      Drops      Sent      Rexmit
LSP            8166        8166           0         4551         0
HELLO          1659        1659           0         1693         0
CSNP             2            2             0            3         0
PSNP           1909        1909           0         2293         0
Unknown         0            0             0            0         0
Totals        11736       11736           0         8540         0

Total packets received: 11736 Sent: 8540

LSP queue length: 0 Drops: 0
SPF runs: 9
Fragments rebuilt: 1640
LSP regenerations: 1
Purges initiated: 0

member1:
-----
IS-IS statistics for 0026.888d.6800:

```

PDU type	Received	Processed	Drops	Sent	Rexmit
LSP	10909	10909	0	12088	0
HELLO	1877	1877	0	2251	0
CSNP	3	3	0	3	0
PSNP	3846	3846	0	3732	0
Unknown	0	0	0	0	0
Totals	16635	16635	0	18074	0

Total packets received: 16635 Sent: 18074

LSP queue length: 0 Drops: 0  
 SPF runs: 13  
 Fragments rebuilt: 1871  
 LSP regenerations: 2  
 Purges initiated: 0

member8:

IS-IS statistics for 0000.4a75.9b7c:

PDU type	Received	Processed	Drops	Sent	Rexmit
LSP	7935	7935	0	14865	0
HELLO	2695	2695	0	7124	0
CSNP	4	4	0	4	0
PSNP	4398	4398	0	3666	0
Unknown	0	0	0	0	0
Totals	15032	15032	0	25659	0

Total packets received: 15032 Sent: 25659

LSP queue length: 0 Drops: 0  
 SPF runs: 26  
 Fragments rebuilt: 2666  
 LSP regenerations: 4  
 Purges initiated: 0

member9:

IS-IS statistics for 0000.73e9.9a57:

PDU type	Received	Processed	Drops	Sent	Rexmit
LSP	10800	10800	0	6327	0
HELLO	1492	1492	0	2356	0
CSNP	2	2	0	2	0
PSNP	2683	2683	0	3149	0
Unknown	0	0	0	0	0
Totals	14977	14977	0	11834	0

Total packets received: 14977 Sent: 11834

LSP queue length: 0 Drops: 0  
 SPF runs: 19  
 Fragments rebuilt: 1510  
 LSP regenerations: 6  
 Purges initiated: 0

## show virtual-chassis login

---

<b>Syntax</b>	<b>show virtual-chassis login</b>
<b>Release Information</b>	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).
<b>Description</b>	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.
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">request session member on page 96</a></li><li>• <a href="#">Understanding Global Management of a Virtual Chassis on page 18</a></li></ul>
<b>List of Sample Output</b>	<a href="#">show virtual-chassis login (Direct Login to the Master Console Port) on page 132</a> <a href="#">show virtual-chassis login (Backup Console Session Redirected to the Master Console Port) on page 132</a>

### 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 248.1.2.3
```

#### 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

<b>Syntax</b>	<b>show virtual-chassis</b> <b>&lt;status&gt;</b>
<b>Release Information</b>	<p>Command introduced in Junos OS Release 9.2 for EX Series switches.</p> <p>Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series.</p> <p>Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF). <b>Fabric ID</b>, <b>Fabric Mode</b>, and <b>Route Mode</b> output fields introduced in Junos OS Release 13.2X51-D20.</p> <p><b>Alias-Name</b> output field introduced in Junos OS Release 14.1X53-D10.</p>
<b>Description</b>	Display information about all members of the Virtual Chassis or VCF.
<b>Options</b>	<p><b>none</b>—Display information about all Virtual Chassis or VCF member devices.</p> <p><b>status</b>—Same output as for <b>show virtual-chassis</b>.</p>
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">show virtual-chassis active-topology on page 104</a></li> <li>• <a href="#">show virtual-chassis protocol adjacency on page 115</a></li> <li>• <a href="#">show virtual-chassis vc-path on page 138</a></li> <li>• <a href="#">Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 57</a></li> </ul>
<b>List of Sample Output</b>	<p><a href="#">show virtual-chassis (EX4200 Virtual Chassis) on page 135</a></p> <p><a href="#">show virtual-chassis (EX8200 Virtual Chassis) on page 136</a></p> <p><a href="#">show virtual-chassis (Virtual Chassis Fabric) on page 136</a></p>
<b>Output Fields</b>	<p><a href="#">Table 18 on page 133</a> lists the output fields for the <b>show virtual-chassis</b> command. Output fields are listed in the approximate order in which they appear.</p>

**Table 18: show virtual-chassis Output Fields**

Field Name	Field Description
<b>Fabric ID</b>	Assigned ID used to identify the VCF.
<b>Fabric Mode</b>	Mode of the VCF: Enabled, Disabled, or Mixed.
<b>Virtual Chassis ID</b>	Assigned ID that applies to the entire Virtual Chassis or VCF.

Table 18: show virtual-chassis Output Fields (*continued*)

Field Name	Field Description
<b>Virtual Chassis Mode</b>	<p>Mode of the Virtual Chassis or VCF. This field indicates support for the Virtual Chassis feature and, if a Virtual Chassis is configured, if it is a mixed or homogenous Virtual Chassis. Values can be:</p> <ul style="list-style-type: none"> <li>• <b>Enabled</b>—The platform supports the Virtual Chassis feature. If a Virtual Chassis is currently configured, this is a homogenous Virtual Chassis (all members are the same type of switch).</li> <li>• <b>Disabled</b>—The switch does not support the Virtual Chassis feature.</li> </ul> <p><b>NOTE:</b> Switches that support the Virtual Chassis feature do not display this value. Even if a Virtual Chassis is not currently configured, those switches display <b>Enabled</b> in this field.</p> <ul style="list-style-type: none"> <li>• <b>Mixed</b>—The platform supports the Virtual Chassis feature, and is configured as a mixed mode Virtual Chassis (members consist of more than one type of switch).</li> </ul>
<b>Member ID</b>	<p>Assigned member ID and FPC:</p> <ul style="list-style-type: none"> <li>• On all EX Series Virtual Chassis except EX8200 Virtual Chassis, and on a VCF, the FPC number refers to the member ID assigned to the switch.</li> <li>• On EX8200 Virtual Chassis, member IDs are numbered 0 through 9. The FPC number indicates the slot number of the line card within the Virtual Chassis. The FPC number on member 0 is always 0 through 15. The FPC number on member 1 is always 16 through 31. The FPC number on member 2 is always 32 through 47; and so on for the members.</li> </ul>
<b>Status</b>	<p>For a nonprovisioned configuration:</p> <ul style="list-style-type: none"> <li>• <b>Prsnt</b> for a member that is currently connected to the Virtual Chassis or VCF configuration.</li> <li>• <b>NotPrsnt</b> for a member ID that has been assigned but is not currently connected.</li> </ul> <p>For a preprovisioned configuration:</p> <ul style="list-style-type: none"> <li>• <b>Prsnt</b> for a member that is specified in the preprovisioned configuration file and is currently connected to the Virtual Chassis or VCF.</li> <li>• <b>Unprvsnd</b> for a member that is interconnected with the Virtual Chassis or VCF configuration but is not specified in the preprovisioned configuration file.</li> </ul>
<b>Serial No</b>	Serial number of the member device.
<b>Alias-Name</b>	<p>The user-configured alias of the member device.</p> <p>The <b>Alias-Name</b> field appears only if an alias has been configured for at least one device in the Virtual Chassis or VCF. Aliases are configured using the <b>alias-name</b> statement in the <code>[edit virtual-chassis aliases serial-number serial-number]</code> hierarchy.</p>
<b>Model</b>	Model number of the member device.
<b>Mastership Priority</b>	Mastership priority value of the member device.
<b>Role</b>	Role of the member device: master, backup, or linecard.

Table 18: show virtual-chassis Output Fields (*continued*)

Field Name	Field Description
<b>Mixed Mode</b>	Mixed mode configuration status: <ul style="list-style-type: none"> <li>• <b>Y</b> for a member device configured in mixed mode.</li> <li>• <b>N</b> for a member device not configured in mixed mode.</li> <li>• <b>NA</b> for a member device that cannot be configured in mixed mode.</li> </ul>
<b>Route Mode</b>	The route mode of the member device: fabric (F) or Virtual Chassis (V).
<b>Location</b>	Location of the member device.  If this field is empty, the location field was not set for the device.
<b>Neighbor List</b>	Member ID of the neighbor member to which this member's Virtual Chassis port (VCP) is connected.

## Sample Output

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

```

user@switch> show virtual-chassis
Virtual Chassis ID: 0019.e250.47a0
Virtual Chassis Mode: Enabled

```

Member ID	Status	Serial No	Model	Mastership priority	Role	Mixed Mode	Neighbor List ID	Interface
0 (FPC 0)	Prsnt	AK0207360276	ex4200-24t	249	Master*	N	8	vcp-0
							1	vcp-1
1 (FPC 1)	Prsnt	AK0207360281	ex4200-24t	248	Backup	N	0	vcp-0
							2	vcp-1
2 (FPC 2)	Prsnt	AJ0207391130	ex4200-48p	247	Linecard	N	1	vcp-0
							3	vcp-1
3 (FPC 3)	Prsnt	AK0207360280	ex4200-24t	246	Linecard	N	2	vcp-0
							4	vcp-1
4 (FPC 4)	Prsnt	AJ0207391113	ex4200-48p	245	Linecard	N	3	vcp-0
							5	vcp-1
5 (FPC 5)	Prsnt	BP0207452204	ex4200-48t	244	Linecard	N	4	vcp-0
							6	vcp-1
6 (FPC 6)	Prsnt	BP0207452222	ex4200-48t	243	Linecard	N	5	vcp-0
							7	vcp-1
7 (FPC 7)	Prsnt	BR0207432028	ex4200-24f	242	Linecard	N	6	vcp-0

```

8 vcp-1
8 (FPC 8) Prsnt BR0207431996 ex4200-24f 241 Linecard N 7 vcp-0
0 vcp-1

```

Member ID for next new member: 9 (FPC 9)

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

```

user@external-routing-engine> show virtual-chassis
Virtual Chassis ID: c806.0842.de51
Virtual Chassis Mode: Enabled

```

Member ID	Status	Serial No	Model	Mastership priority	Role	Neighbor List ID Interface
0 (FPC 0-15)	Prsnt	BA0908380001	ex8216	0	Linecard	8 vcp-0/0 8 vcp-0/1 1 vcp-4/0/4
1 (FPC 16-31)	Prsnt	BT0909411634	ex8208	0	Linecard	8 vcp-0/0 0 vcp-3/0/4
8 (FPC 128-143)	Prsnt	062009000021	ex-xre	128	Master	9 vcp-1/0 1 vcp-1/2 9 vcp-1/3 0 vcp-2/0 9 vcp-2/1 0 vcp-1/1
9 (FPC 144-159)	Prsnt	062009000022	ex-xre	128	Backup*	8 vcp-1/0 8 vcp-1/2 8 vcp-1/3 8 vcp-1/3

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

```

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

```

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



3 (FPC 3) Prsnt	AB3112460011 qfx5100-48s	0	Linecard	N	F	1
vcp-255/1/0						0
vcp-255/1/1						
4 (FPC 4) Prsnt	AB3112430011 qfx5100-48s	0	Linecard	N	F	1
vcp-255/1/0						0
vcp-255/1/1						

## show virtual-chassis vc-path

---

Syntax	<b>show virtual-chassis vc-path source-interface <i>interface-name</i> destination-interface <i>interface-name</i></b>
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	<p>Show the forwarding path a packet takes when going from a source interface to a destination interface in a Virtual Chassis or VCF configuration.</p> <p>Starting in Junos OS Releases 14.1X53-D40, 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.</p>
Options	<p><b>source-interface <i>interface-name</i></b>—Name of the interface from which the packet originates in the Virtual Chassis or VCF</p> <p><b>destination-interface <i>interface-name</i></b>—Name of the interface to which the packet is being delivered in the Virtual Chassis or VCF</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"><li>• <a href="#">Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 57</a></li><li>• <i>Understanding EX Series Virtual Chassis Configuration</i></li><li>• <i>EX8200 Virtual Chassis Overview</i></li></ul>
List of Sample Output	<p><a href="#">show virtual-chassis vc-path source-interface destination-interface (Virtual Chassis) on page 139</a></p> <p><a href="#">show virtual-chassis vc-path source-interface destination-interface (Virtual Chassis Fabric) on page 140</a></p>
Output Fields	<p><a href="#">Table 19 on page 139</a> lists the output fields for the <b>show virtual-chassis vc-path</b> command. Output fields are listed in the approximate order in which they appear.</p> <p>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 <a href="#">Table 19 on page 139</a> and shown in sample output for each mode.</p>

Table 19: show virtual-chassis vc-path Output Fields

Field Name	Field Description
<b>Hop</b>	<p>The hop number along the path between the source and destination interfaces. The first hop entry (<b>Hop 0</b>) is the packet's source, intermediate hop information represents transitions through the members within the Virtual Chassis or VCF, and the last hop entry represents arrival at the packet's destination.</p> <p>For VCF output, each <b>Hop</b> entry also shows information about multiple possible next hops towards the destination. See <b>Next-hop PFE</b>, <b>Interface</b>, and <b>Bandwidth</b> output field descriptions for details.</p>
<b>Member</b> (Virtual Chassis)  <b>Member-ID</b> (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>
<b>PFE-Device</b> (Virtual Chassis)  <b>PFE</b> (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 <b>Next-hop PFE</b> output fields with interface name and bandwidth information for each possible next hop.</p>
<b>Next-hop PFE</b> (VCF only)	<p>One or more possible next-hop Packet Forwarding Engine numbers for VCF member <b>Member-ID</b>. Each <b>Next-hop</b> entry includes the next-hops's interface name and bandwidth.</p>
<b>Interface</b>	<p>The name of the interface through which the Packet Forwarding Engines are connected. The interface for the first hop (<b>Hop 0</b>) is always the source interface.</p> <p>VCF output provides more information about multiple possible next hops for each hop entry, listing the <b>Interface</b> and the interface <b>Bandwidth</b> for each possible <b>Next-hop PFE</b>.</p>
<b>Bandwidth</b> (VCF only)	<p>The bandwidth (in Gbps) of the next-hop interface for the associated <b>Next-hop PFE</b> entry.</p>

## Sample Output

### show virtual-chassis vc-path source-interface destination-interface (Virtual Chassis)

```
user@switch> show virtual-chassis vc-path source-interface ge-0/0/0 destination-interface ge-1/0/1
```

```
vc-path from ge-0/0/0 to ge-1/0/1
Hop      Member  PFE-Device  Interface
0        0        1           ge-0/0/0
1        0        0           internal-1/24
2        1        3           vcp-0
3        1        4           ge-1/0/1
```

**show virtual-chassis vc-path source-interface destination-interface (Virtual Chassis Fabric)**

This example shows **vc-path** command output for two source-to-destination paths in the following Virtual Chassis Fabric displayed by the **show virtual-chassis** command:

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

```
Preprovisioned Virtual Chassis Fabric
Fabric ID: ec56.0915.8595
Fabric Mode: Enabled
```

Member ID	Status	Serial No	Model	Mstr prio	Role	Mixed Mode	Route Mode	Neighbor List ID	Interface
1 (FPC 1)	Prsnt	VX1234560001	qfx5100-24q-aa	129	Backup	N	F	7	vcp-255/0/7
								8	vcp-255/0/8
								12	vcp-255/0/12
								3	vcp-255/0/3
								4	vcp-255/0/4
								5	vcp-255/0/5
								6	vcp-255/0/6
								10	vcp-255/0/10
								11	vcp-255/0/11
								9	vcp-255/0/9
2 (FPC 2)	Prsnt	VX1234560002	qfx5100-24q-aa	129	Master*	N	F	7	vcp-255/0/7
								8	vcp-255/0/8
								12	vcp-255/0/12
								3	vcp-255/0/3
								4	vcp-255/0/4
								5	vcp-255/0/5
								6	vcp-255/0/6
								10	vcp-255/0/10
								11	vcp-255/0/11
								9	vcp-255/0/9
3 (FPC 3)	Prsnt	VX1234560003	qfx5100-24q-aa	0	Linecard	N	F	1	vcp-255/0/1
								2	vcp-255/0/2
4 (FPC 4)	Prsnt	VX1234560004	qfx5100-24q-aa	0	Linecard	N	F	1	vcp-255/0/1
								2	vcp-255/0/2
5 (FPC 5)	Prsnt	VX1234560005	qfx5100-24q-aa	0	Linecard	N	F	1	vcp-255/0/1
								2	vcp-255/0/2
6 (FPC 6)	Prsnt	VX1234560006	qfx5100-24q-aa	0	Linecard	N	F	1	vcp-255/0/1
								2	vcp-255/0/2
7 (FPC 7)	Prsnt	VX1234560007	qfx5100-24q-aa	0	Linecard	N	F	1	vcp-255/0/1
								2	vcp-255/0/2
8 (FPC 8)	Prsnt	VX1234560008	qfx5100-24q-aa	0	Linecard	N	F	1	vcp-255/0/1
								2	vcp-255/0/2
9 (FPC 9)	Prsnt	VX1234560009	qfx5100-24q-aa	0	Linecard	N	F	1	vcp-255/0/1
								2	vcp-255/0/2
10 (FPC 10)	Prsnt	VX1234560010	qfx5100-24q-aa	0	Linecard	N	F	1	vcp-255/0/1
								2	vcp-255/0/2
11 (FPC 11)	Prsnt	VX1234560011	qfx5100-24q-aa	0	Linecard	N	F	1	vcp-255/0/1
								2	vcp-255/0/2
12 (FPC 12)	Prsnt	VX1234560012	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

<b>Syntax</b>	show virtual-chassis vc-port <all-members> <local> <member <i>member-id</i> >
<b>Release Information</b>	Command introduced in Junos OS Release 9.0 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
<b>Description</b>	Display the status of the Virtual Chassis ports (VCPs), including both the dedicated VCPs and the uplink ports configured as VCPs.
<b>Options</b>	<p><b>none</b>—Display the operational status of all VCPs of the member switch where the command is issued.</p> <p><b>all-members</b>—(Optional) Display the operational status of all VCPs on all members of the Virtual Chassis or VCF.</p> <p><b>local</b>—(Optional) Display the operational status of the switch or external Routing Engine on which this command is entered.</p> <p><b>member <i>member-id</i></b>—(Optional) Display the operational status of all VCPs for the specified member of the Virtual Chassis or VCF.</p>
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">show virtual-chassis vc-port statistics on page 146</a></li> <li>• <a href="#">Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 57</a></li> <li>• <a href="#">Verifying Virtual Chassis Ports in an EX8200 Virtual Chassis</a></li> </ul>
<b>List of Sample Output</b>	<a href="#">show virtual-chassis vc-port (EX4200 Virtual Chassis) on page 144</a> <a href="#">show virtual-chassis vc-port (EX8200 Virtual Chassis) on page 144</a> <a href="#">show virtual-chassis vc-port all-members on page 145</a>
<b>Output Fields</b>	<a href="#">Table 20 on page 142</a> lists the output fields for the <b>show virtual-chassis vc-port</b> command. Output fields are listed in the approximate order in which they appear.

Table 20: show virtual-chassis vc-port Output Fields

Field Name	Field Description
<i>fpcnumber</i>	The FPC number is the same as the member ID.

Table 20: show virtual-chassis vc-port Output Fields (*continued*)

Field Name	Field Description
Interface or PIC/Port	<p>VCP name.</p> <ul style="list-style-type: none"> <li>The dedicated VCPs in an EX4200 or EX4500 Virtual Chassis are <b>vcp-0</b> and <b>vcp-1</b>. The dedicated VCPs in an EX4550 Virtual Chassis are <b>VCP-1/0</b>, <b>VCP-1/1</b>, <b>VCP-2/0</b>, and <b>VCP-2/1</b>.</li> <li>Optical ports set as VCPs are named <b>1/0</b> and <b>1/1</b>, representing the PIC number and the port number.</li> <li>The native VCP (port 0) on an XRE200 External Routing Engine in an EX8200 Virtual Chassis is named <b>vcp-0</b>.</li> <li>The VCPs on each Virtual Chassis Control Interface (VCCI) module in an XRE200 External Routing Engine are named using the <b>vcp-slot-number/port-number</b> convention; for instance, <b>vcp-1/0</b>.</li> <li>The VCPs on EX8200 member switches are named using the <b>vcp-slot-number/pic-number/interface-number</b> convention; for instance, <b>vcp-3/0/2</b>.</li> <li>A <b>255</b> as the first number in your port number indicates that your VCP is part of a Link Aggregation group (LAG) bundle. For instance, a display of <b>vcp-255/1/0</b> indicates that the dedicated VCP named <b>vcp-1/0</b> is part of a LAG bundle. A display of <b>vcp-255/1/0</b> indicates that an uplink port that was previously named <b>xe-0/1/0</b> is now part of a VCP LAG bundle.</li> </ul>
Type	<p>Type of VCP:</p> <ul style="list-style-type: none"> <li><b>Dedicated</b>—The rear panel VCP on an EX4200, EX4500, or EX4550 switch, or any VCP link connected to an XRE200 External Routing Engine in an EX8200 Virtual Chassis.</li> <li><b>Configured</b>—Optical port configured as a VCP.</li> <li><b>Auto-Configured</b>—Optical port autoconfigured as a VCP.</li> </ul> <p>See <a href="#">“Setting an Uplink Port on an EX Series Switch as a Virtual Chassis Port (CLI Procedure)” on page 43</a> or <a href="#">Setting a 10-Gigabit Ethernet Port as a Virtual Chassis Port in an EX8200 Virtual Chassis (CLI Procedure)</a> for information about configuring VCPs.</p>
Trunk ID	<p>A positive-number ID assigned to a link aggregation group (LAG) formed by the Virtual Chassis. The trunk ID value is –1 if no trunk is formed. A LAG between uplink VCPs requires that the link speed be the same on connected interfaces and that at least two VCPs on one member be connected to at least two VCPs on the other member in an EX4200 or EX4500 Virtual Chassis.</p> <p>Dedicated VCP LAGs are assigned trunk IDs 1 and 2. Trunk IDs for LAGs formed with uplink VCPs therefore have values of 3 or greater.</p> <p>The trunk ID value changes if the link-adjacency state between LAG members changes; trunk membership is then allocated or deallocated.</p>
Status	<p>Interface status:</p> <ul style="list-style-type: none"> <li><b>absent</b>—Interface is not a VCP link.</li> <li><b>down</b>—VCP link is down.</li> <li><b>up</b>—VCP link is up.</li> </ul>
Speed (mbps)	Speed of the interface in megabits per second.
Neighbor ID/Interface	The Virtual Chassis member ID and interface of a VCP on a member that is connected to the interface or PIC/Port field in the same row as this interface.

## Sample Output

### show virtual-chassis vc-port (EX4200 Virtual Chassis)

```
user@switch> show virtual-chassis vc-port
```

```
fpc0:
```

Interface or PIC / Port	Type	Trunk ID	Status	Speed (mbps)	Neighbor ID	Interface
vcp-0	Dedicated	1	Up	32000	1	vcp-1
vcp-1	Dedicated	2	Up	32000	0	vcp-0
1/0	Auto-Configured	3	Up	1000	2	vcp-255/1/0
1/0	Auto-Configured	3	Up	1000	2	vcp-255/1/1

### show virtual-chassis vc-port (EX8200 Virtual Chassis)

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

```
member0:
```

Interface or Slot/PIC/Port	Type	Trunk ID	Status	Speed (mbps)	Neighbor ID	Interface
vcp-0/0	Dedicated	-1	Up	1000	8	vcp-1/1
vcp-0/1	Dedicated	-1	Up	1000	8	vcp-2/0
4/0/4	Configured	-1	Up	10000	1	vcp-3/0/4
4/0/7	Configured	-1	Down	10000		
4/0/3	Configured		Absent			
4/0/2	Configured		Absent			
4/0/5	Configured		Absent			
4/0/6	Configured		Absent			
4/0/1	Configured		Absent			
4/0/0	Configured		Absent			

```
member1:
```

Interface or Slot/PIC/Port	Type	Trunk ID	Status	Speed (mbps)	Neighbor ID	Interface
vcp-0/0	Dedicated	-1	Up	1000	8	vcp-1/2
3/0/0	Configured	-1	Down	10000		
3/0/1	Configured	-1	Down	10000		
3/0/4	Configured	-1	Up	10000	0	vcp-4/0/4
3/0/5	Configured		Absent			
4/0/5	Configured		Absent			
4/0/4	Configured		Absent			

```
member8:
```

Interface or Slot/PIC/Port	Type	Trunk ID	Status	Speed (mbps)	Neighbor ID	Interface
vcp-0/0	Dedicated	-1	Down	1000		
vcp-1/0	Dedicated	-1	Up	1000	9	vcp-1/0
vcp-1/1	Dedicated	-1	Up	1000	0	vcp-0/0
vcp-1/2	Dedicated	-1	Up	1000	1	vcp-0/0
vcp-1/3	Dedicated	-1	Up	1000	9	vcp-1/3
vcp-2/0	Dedicated	-1	Up	1000	0	vcp-0/1
vcp-2/1	Dedicated	-1	Up	1000	9	vcp-1/2
vcp-2/2	Dedicated	-1	Down	1000		



```
vcp-2/3      Dedicated      -1   Down      1000
```

```
member9:
```

```
-----
Interface    Type           Trunk  Status    Speed    Neighbor
or           or              ID      (mbps)    ID  Interface
Slot/PIC/Port
vcp-0/0      Dedicated      -1     Disabled  1000
vcp-1/0      Dedicated      -1     Up        1000      8   vcp-1/0
vcp-1/1      Dedicated      -1     Down      1000
vcp-1/2      Dedicated      -1     Up        1000      8   vcp-2/1
vcp-1/3      Dedicated      -1     Up        1000      8   vcp-1/3
```

### show virtual-chassis vc-port all-members

```
user@switch> show virtual-chassis vc-port all-members
```

```
fpc0:
```

```
-----
Interface    Type           Trunk  Status    Speed    Neighbor
or           or              ID      (mbps)    ID  Interface
PIC / Port
vcp-0        Dedicated      1      Up        32000    1   vcp-1
vcp-1        Dedicated      2      Up        32000    0   vcp-0
1/0          Auto-Configured 3      Up        1000     2   vcp-255/1/0
1/1          Auto-Configured 3      Up        1000     2   vcp-255/1/1
```

```
fpc1:
```

```
-----
Interface    Type           Trunk  Status    Speed    Neighbor
or           or              ID      (mbps)    ID  Interface
PIC / Port
vcp-0        Dedicated      1      Up        32000    0   vcp-1
vcp-1        Dedicated      2      Up        32000    0   vcp-0
1/0          Auto-Configured -1     Up        1000     3   vcp-255/1/0
```

```
fpc2:
```

```
-----
Interface    Type           Trunk  Status    Speed    Neighbor
or           or              ID      (mbps)    ID  Interface
PIC / Port
vcp-0        Dedicated      1      Up        32000    3   vcp-1
vcp-1        Dedicated      2      Up        32000    3   vcp-0
1/0          Auto-Configured 3      Up        1000     0   vcp-255/1/0
1/1          Auto-Configured 3      Up        1000     0   vcp-255/1/1
```

```
fpc3:
```

```
-----
Interface    Type           Trunk  Status    Speed    Neighbor
or           or              ID      (mbps)    ID  Interface
PIC / Port
vcp-0        Dedicated      1      Up        32000    2   vcp-0
vcp-1        Dedicated      2      Up        32000    2   vcp-1
1/0          Auto-Configured -1     Up        1000     1   vcp-255/1/0
```

## show virtual-chassis vc-port statistics

---

<b>Syntax</b>	<pre>show virtual-chassis vc-port statistics &lt;all-members&gt; &lt;brief   detail   extensive &gt; &lt;interface-name&gt; &lt;local&gt; &lt;member member-id&gt;</pre>
<b>Release Information</b>	<p>Command introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>The options <b>all-members</b>, <b>brief</b>, <b>detail</b>, <b>extensive</b>, and <b>local</b> were added in Junos OS Release 9.3 for EX Series switches.</p> <p>Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series.</p> <p>Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).</p>
<b>Description</b>	Display the traffic statistics collected on Virtual Chassis ports (VCPs).
<b>Options</b>	<p><b>none</b>—Display traffic statistics for VCPs of all members of a Virtual Chassis or VCF.</p> <p><b>brief   detail   extensive</b>—(Optional) Display the specified level of output. Using the <b>brief</b> option is equivalent to entering the command with no options (the default). The <b>detail</b> and <b>extensive</b> options provide identical displays.</p> <p><b>all-members</b>—(Optional) Display traffic statistics for VCPs of all members of a Virtual Chassis or VCF.</p> <p><b>interface-name</b>—(Optional) Display traffic statistics for the specified VCP.</p> <p><b>local</b>—(Optional) Display traffic statistics for VCPs on the switch or external Routing Engine on which this command is entered.</p> <p><b>member member-id</b>—(Optional) Display traffic statistics for VCPs on the specified member of a Virtual Chassis or VCF.</p>
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">clear virtual-chassis vc-port statistics on page 94</a></li><li>• <a href="#">show virtual-chassis vc-port on page 142</a></li><li>• <a href="#">Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 57</a></li><li>• <a href="#">Verifying Virtual Chassis Ports in an EX8200 Virtual Chassis</a></li></ul>
<b>List of Sample Output</b>	<p><a href="#">show virtual-chassis vc-port statistics on page 149</a></p> <p><a href="#">show virtual-chassis vc-port statistics (EX8200 Virtual Chassis) on page 150</a></p> <p><a href="#">show virtual-chassis vc-port statistics brief on page 150</a></p> <p><a href="#">show virtual-chassis vc-port statistics extensive on page 150</a></p> <p><a href="#">show virtual-chassis vc-port statistics member 0 on page 152</a></p>

**Output Fields** Table 21 on page 147 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 21: show virtual-chassis vc-port statistics Output Fields**

Field Name	Field Description	Level of Output
<b>fpcnumber</b>	(All Virtual Chassis except EX8200 Virtual Chassis. VCF) ID of the Virtual Chassis member. The FPC number is the same as the member ID.	All levels
<b>member number</b>	(EX8200 Virtual Chassis only) Member ID of the Virtual Chassis member.	All levels
<b>Interface</b>	VCP name.	<b>brief</b>
<b>Input Octets/Packets</b>	Number of octets and packets received on the VCP.	<b>brief, member, none</b>
<b>Output Octets/Packets</b>	Number of octets and packets transmitted on the VCP.	<b>brief, member, none</b>
<b>master: number</b>	Member ID of the master Routing Engine.	All levels
<b>Port</b>	VCP for which RX (Receive) statistics, TX (Transmit) statistics, or both are reported by the VCP subsystem during a sampling interval—since the statistics counter was last cleared.	<b>detail, extensive</b>
<b>Total octets</b>	Total number of octets received and transmitted on the VCP.	<b>detail, extensive</b>
<b>Total packets</b>	Total number of packets received and transmitted on the VCP.	<b>detail, extensive</b>
<b>Unicast packets</b>	Number of unicast packets received and transmitted on the VCP.	<b>detail, extensive</b>
<b>Broadcast packets</b>	Number of broadcast packets received and transmitted on the VCP.	<b>detail, extensive</b>
<b>Multicast packets</b>	Number of multicast packets received and transmitted on the VCP.	<b>detail, extensive</b>
<b>MAC control frames</b>	Number of media access control (MAC) control frames received and transmitted on the VCP.	<b>detail, extensive</b>

Table 21: show virtual-chassis vc-port statistics Output Fields (*continued*)

Field Name	Field Description	Level of Output
<b>CRC alignment errors</b>	<p>Number of packets received on the VCP that had a length—excluding framing bits, but including frame check sequence (FCS) octets—of between 64 and 1518 octets, inclusive, and had one of the following errors:</p> <ul style="list-style-type: none"> <li>Invalid FCS with an integral number of octets (FCS error)</li> <li>Invalid FCS with a nonintegral number of octets (alignment error)</li> </ul>	<b>detail, extensive</b>
<b>Oversize packets</b>	Number of packets received on the VCP that were longer than 1518 octets (excluding framing bits, but including FCS octets) but were otherwise well formed.	<b>detail, extensive</b>
<b>Undersize packets</b>	Number of packets received on the VCP that were shorter than 64 octets (excluding framing bits but including FCS octets) and were otherwise well formed..	<b>detail, extensive</b>
<b>Jabber packets</b>	<p>Number of packets received on the VCP that were longer than 1518 octets—excluding framing bits, but including FCS octets—and that had either an FCS error or an alignment error.</p> <p><b>NOTE:</b> This definition of <i>jabber</i> is different from the definition in IEEE-802.3 section 8.2.1.5 (10Base5) and section 10.3.1.4 (10Base2). These documents define <i>jabber</i> as the condition in which any packet exceeds 20 ms. The allowed range to detect jabber is between 20 ms and 150 ms.</p>	<b>detail, extensive</b>
<b>Fragments received</b>	<p>Number of packets received on the VCP that were shorter than 64 octets (excluding framing bits, but including FCS octets), and had either an FCS error or an alignment error.</p> <p>Fragment frames normally increment because both runs (which are normal occurrences caused by collisions) and noise hits are counted.</p>	<b>detail, extensive</b>
<b>Ifout errors</b>	Number of outbound packets received on the VCP that could not be transmitted because of errors.	<b>detail, extensive</b>
<b>Packet drop events</b>	Number of outbound packets received on the VCP that were dropped, rather than being encapsulated and sent out of the switch as fragments. The packet drop counter is incremented if a temporary shortage of packet memory causes packet fragmentation to fail.	<b>detail, extensive</b>
<b>64 octets frames</b>	Number of packets received on the VCP (including invalid packets) that were 64 octets in length (excluding framing bits, but including FCS octets).	<b>detail, extensive</b>

Table 21: show virtual-chassis vc-port statistics Output Fields (*continued*)

Field Name	Field Description	Level of Output
65–127 octets frames	Number of packets received on the VCP (including invalid packets) that were between 65 and 127 octets in length, inclusive (excluding framing bits, but including FCS octets).	detail, extensive
128–255 octets frames	Number of packets received on the VCP (including invalid packets) that were between 128 and 255 octets in length, inclusive (excluding framing bits, but including FCS octets).	detail, extensive
256–511 octets frames	Number of packets received on the VCP (including invalid packets) that were between 256 and 511 octets in length, inclusive (excluding framing bits, but including FCS octets).	detail, extensive
512–1023 octets frames	Number of packets received on the VCP (including invalid packets) that were between 512 and 1023 octets in length, inclusive (excluding framing bits, but including FCS octets).	detail, extensive
1024–1518 octets frames	Number of packets received on the VCP (including invalid packets) that were between 1024 and 1518 octets in length, inclusive (excluding framing bits, but including FCS octets).	detail, extensive
Rate packets per second	Number of packets per second received and transmitted on the VCP.	detail, extensive
Rate bytes per second	Number of bytes per second received and transmitted on the VCP.	detail, extensive

## Sample Output

### show virtual-chassis vc-port statistics

```
user@switch> show virtual-chassis vc-port statistics
fpc0:
```

```
-----
Interface          Input  Octets/Packets      Output  Octets/Packets
internal-0/24       0      / 0                0      / 0
internal-0/25       0      / 0                0      / 0
internal-1/26       0      / 0                0      / 0
internal-1/27       0      / 0                0      / 0
vcp-0               0      / 0                0      / 0
vcp-1               0      / 0                0      / 0
internal-0/26       0      / 0                0      / 0
internal-0/27       0      / 0                0      / 0
internal-1/24       0      / 0                0      / 0
internal-1/25       0      / 0                0      / 0
```

```
{master:0}
```

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

```

user@external-routing-engine> show virtual-chassis vc-port statistics
member0:
-----
Interface          Input Octets/Packets      Output Octets/Packets
vcp-4/0/4          43171238 / 48152          47687133 / 51891
vcp-4/0/7          0 / 0                     0 / 0

member1:
-----
Interface          Input Octets/Packets      Output Octets/Packets
vcp-3/0/0          0 / 0                     0 / 0
vcp-3/0/1          0 / 0                     0 / 0
vcp-3/0/4          47695376 / 51899          43180556 / 48160

member8:
-----

member9:
-----

```

**show virtual-chassis vc-port statistics brief**

```

user@switch> show virtual-chassis vc-port statistics brief
fpc0:
-----
Interface          Input Octets/Packets      Output Octets/Packets
internal-0/24       0 / 0                     0 / 0
internal-0/25       0 / 0                     0 / 0
internal-1/26       0 / 0                     0 / 0
internal-1/27       0 / 0                     0 / 0
vcp-0               0 / 0                     0 / 0
vcp-1               0 / 0                     0 / 0
internal-0/26       0 / 0                     0 / 0
internal-0/27       0 / 0                     0 / 0
internal-1/24       0 / 0                     0 / 0
internal-1/25       0 / 0                     0 / 0

{master:0}

```

**show virtual-chassis vc-port statistics extensive**

```

user@switch> show virtual-chassis vc-port statistics extensive
fpc0:
-----

```

	RX	TX
Port: internal-0/24		
Total octets:	0	0
Total packets:	0	0
Unicast packets:	0	0
Broadcast packets:	0	0
Multicast packets:	0	0
MAC control frames:	0	0
CRC alignment errors:	0	
Oversize packets:	0	
Undersize packets:	0	
Jabber packets:	0	
Fragments received:	0	

```

Ifout errors:          0
Packet drop events:    0
64      octets frames: 0
65-127   octets frames: 0
128-255  octets frames: 0
256-511  octets frames: 0
512-1023 octets frames: 0
1024-1518 octets frames: 0
Rate packets per second: 0          0
Rate bytes per second:   0          0

...

Port: vcp-0
Total octets:          0          0
Total packets:         0          0
Unicast packets:       0          0
Broadcast packets:     0          0
Multicast packets:     0          0
MAC control frames:    0          0
CRC alignment errors:  0
Oversize packets:     0
Undersize packets:     0
Jabber packets:        0
Fragments received:    0
Ifout errors:          0
Packet drop events:    0
64      octets frames: 0
65-127   octets frames: 0
128-255  octets frames: 0
256-511  octets frames: 0
512-1023 octets frames: 0
1024-1518 octets frames: 0
Rate packets per second: 0          0
Rate bytes per second:   0          0

Port: vcp-1
Total octets:          0          0
Total packets:         0          0
Unicast packets:       0          0
Broadcast packets:     0          0
Multicast packets:     0          0
MAC control frames:    0          0
CRC alignment errors:  0
Oversize packets:     0
Undersize packets:     0
Jabber packets:        0
Fragments received:    0
Ifout errors:          0
Packet drop events:    0
64      octets frames: 0
65-127   octets frames: 0
128-255  octets frames: 0
256-511  octets frames: 0
512-1023 octets frames: 0
1024-1518 octets frames: 0
Rate packets per second: 0          0
Rate bytes per second:   0          0

...

```

```
{master:0}
```

#### show virtual-chassis vc-port statistics member 0

```
user@switch>show virtual-chassis vc-port statistics member 0  
fpc0:
```

```
-----  
Interface          Input  Octets/Packets      Output  Octets/Packets  
internal-0/24       0      / 0              0      / 0  
internal-0/25       0      / 0              0      / 0  
internal-1/26       0      / 0              0      / 0  
internal-1/27       0      / 0              0      / 0  
vcp-0               0      / 0              0      / 0  
vcp-1               0      / 0              0      / 0  
internal-0/26       0      / 0              0      / 0  
internal-0/27       0      / 0              0      / 0  
internal-1/24       0      / 0              0      / 0  
internal-1/25       0      / 0              0      / 0
```

```
{master:0}
```