



Virtual Chassis Feature Guide for EX9200 Switches



Modified: 2019-02-05

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

Juniper Networks, the Juniper Networks logo, Juniper, and Junos are registered trademarks of Juniper Networks, Inc. in the United States and other countries. All other trademarks, service marks, registered marks, or registered service marks are the property of their respective owners.

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

Virtual Chassis Feature Guide for EX9200 Switches
Copyright © 2019 Juniper Networks, Inc. All rights reserved.

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

Table of Contents

	About the Documentation	ix
	Documentation and Release Notes	ix
	Using the Examples in This Manual	ix
	Merging a Full Example	x
	Merging a Snippet	x
	Documentation Conventions	xi
	Documentation Feedback	xiii
	Requesting Technical Support	xiii
	Self-Help Online Tools and Resources	xiv
	Creating a Service Request with JTAC	xiv
Chapter 1	Virtual Chassis Overview	15
	Understanding EX9200 Virtual Chassis	15
	Understanding Virtual Chassis Components	16
	Maximum Switch Support	17
	Maximum Number of Switches in an EX Series Virtual Chassis	17
	Maximum Switch Support in a QFX Series Virtual Chassis (Including Mixed Virtual Chassis with EX Series Switches)	19
	Virtual Chassis Ports (VCPs)	20
	Virtual Chassis Port Options	20
	Automatic Virtual Chassis Port (VCP) Conversion	22
	Virtual Chassis Port Link Aggregation Groups	23
	Master Routing Engine Role	23
	Backup Routing Engine Role	24
	Linecard Role	25
	Member Switch and Member ID	26
	Mastership Priority	26
	Virtual Chassis Identifier (VCID)	27
	Nonvolatile Storage in a Virtual Chassis	27
Part 1	Configuring a Virtual Chassis	
Chapter 2	Accessing and Configuring a Virtual Chassis and Managing Files	33
	Accessing the Virtual Chassis Through the Management Interface	34
	Managing Files on Virtual Chassis Member Routers or Switches	35
	Virtual Chassis Slot Number Mapping for Use with SNMP	36
	Configuring an EX9200 Virtual Chassis	38
	Creating Configuration Groups for an EX9200 Virtual Chassis	38
	Configuring the EX9200 Virtual Chassis	39
	Creating and Applying Configuration Groups for a Virtual Chassis	43

Chapter 3	Virtual Chassis Ports	47
	Configuring Virtual Chassis Ports to Interconnect Member Routers or Switches	48
	Deleting Virtual Chassis Ports in a Virtual Chassis Configuration	51
	Verifying the Operation of Virtual Chassis Ports	52
Chapter 4	Upgrading Junos OS in a Virtual Chassis	55
	Upgrading Junos OS Software in an EX9200 Virtual Chassis	55
Chapter 5	Deleting a Member ID	59
	Deleting Member IDs in a Virtual Chassis Configuration	59
Chapter 6	Module Redundancy and GRES	61
	Targeted Traffic Distribution on Aggregated Ethernet Interfaces in a Virtual Chassis	61
	Targeted Distribution in a Virtual Chassis	61
	Benefits of Targeted Distribution	62
	Configuring Module Redundancy for a Virtual Chassis	62
	Determining GRES Readiness in a Virtual Chassis Configuration	64
Chapter 7	Global Roles and Local Roles that Determine Mastership and Switchover Behavior	67
	Mastership Election in a Virtual Chassis	67
	Global Roles and Local Roles in a Virtual Chassis	69
	Role Name Format	69
	Global Role and Local Role Descriptions	70
	Switching the Global Master and Backup Roles in a Virtual Chassis Configuration	71
Chapter 8	Minimizing Network Disruption Using Split Detection	73
	Split Detection Behavior in a Virtual Chassis	73
	How Split Detection Works in a Virtual Chassis	73
	Effect of Split Detection on Virtual Chassis Failure Scenarios	74
	Disabling Split Detection in a Virtual Chassis Configuration	78
Part 2	Troubleshooting	
Chapter 9	Acquiring Troubleshooting Information	81
	Configuring the Name of the Virtual Chassis Trace Log File	81
	Configuring Characteristics of the Virtual Chassis Trace Log File	82
	Configuring Access to the Virtual Chassis Trace Log File	83
	Using Regular Expressions to Refine the Output of the Virtual Chassis Trace Log File	84
	Configuring the Virtual Chassis Operations to Trace	85
	traceoptions (Virtual Chassis)	87

Part 3	Routine Monitoring	
Chapter 10	Monitoring a Virtual Chassis	93
	Verifying the Status of Virtual Chassis Member Routers or Switches	93
	Verifying Neighbor Reachability for Member Routers or Switches in a Virtual Chassis	93
	Verifying Neighbor Reachability for Hardware Devices in a Virtual Chassis	94
	Viewing Information in the Virtual Chassis Control Protocol Adjacency Database	95
	Viewing Information in the Virtual Chassis Control Protocol Link-State Database	95
	Viewing Information About Virtual Chassis Port Interfaces in the Virtual Chassis Control Protocol Database	96
	Viewing Virtual Chassis Control Protocol Statistics for Member Devices and Virtual Chassis Ports	97
Part 4	Configuration Statements and Operational Commands	
Chapter 11	Configuration Statements	101
	aggregated-ether-options	102
	logical-interface-fpc-redundancy (Aggregated Ethernet Subscriber Interfaces)	105
	member	106
	no-split-detection	107
	preprovisioned	108
	role	109
	serial-number	112
	targeted-distribution (Static Interfaces over Aggregated Ethernet)	113
	traceoptions (Virtual Chassis)	114
	virtual-chassis	117
Chapter 12	Operational Commands: Administrative	119
	request virtual-chassis member-id delete (MX Series Virtual Chassis)	120
	request virtual-chassis member-id set	121
	request virtual-chassis routing-engine master switch	123
	request virtual-chassis vc-port	126
Chapter 13	Operational Commands: Monitoring	129
	show virtual-chassis active-topology	130
	show virtual-chassis device-topology	135
	show virtual-chassis protocol adjacency	142
	show virtual-chassis protocol database	146
	show virtual-chassis protocol interface	150
	show virtual-chassis protocol route	153
	show virtual-chassis protocol statistics	156
	show virtual-chassis	160
	show virtual-chassis vc-port	166

List of Tables

	About the Documentation	ix
	Table 1: Notice Icons	xi
	Table 2: Text and Syntax Conventions	xii
Chapter 1	Virtual Chassis Overview	15
	Table 3: Maximum Member Switch Support for Virtual Chassis by Junos OS Release	17
	Table 4: VCP Options by Switch Type	20
Part 1	Configuring a Virtual Chassis	
Chapter 2	Accessing and Configuring a Virtual Chassis and Managing Files	33
	Table 5: jnxFruSlot Numbers and Corresponding Slot Numbers in an MX Series or EX9200 Virtual Chassis	37
Chapter 7	Global Roles and Local Roles that Determine Mastership and Switchover Behavior	67
	Table 6: Global Roles and Local Roles	70
	Table 7: Virtual Chassis Global Role Transitions Before and After Mastership Switchover	72
Chapter 8	Minimizing Network Disruption Using Split Detection	73
	Table 8: Effect of Split Detection on Common Virtual Chassis Failure Scenarios	74
Part 2	Troubleshooting	
Chapter 9	Acquiring Troubleshooting Information	81
	Table 9: Tracing Flags for Virtual Chassis	85
Part 4	Configuration Statements and Operational Commands	
Chapter 13	Operational Commands: Monitoring	129
	Table 10: show virtual-chassis active-topology Output Fields	130
	Table 11: show virtual-chassis device-topology Output Fields	135
	Table 12: show virtual-chassis protocol adjacency Output Fields	143
	Table 13: show virtual-chassis protocol database Output Fields	147
	Table 14: show virtual-chassis protocol interface Output Fields	151
	Table 15: show virtual-chassis protocol route Output Fields	153
	Table 16: show virtual-chassis protocol statistics Output Fields	156
	Table 17: show virtual-chassis Output Fields	161
	Table 18: show virtual-chassis vc-port Output Fields	166

About the Documentation

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

Documentation and Release Notes

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

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

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

Using the Examples in This Manual

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

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

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

Merging a Full Example

To merge a full example, follow these steps:

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

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

```
system {
  scripts {
    commit {
      file ex-script.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 xi defines notice icons used in this guide.

Table 1: Notice Icons







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

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

Table 2: Text and Syntax Conventions

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

GUI Conventions

Table 2: Text and Syntax Conventions (continued)

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

Documentation Feedback

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

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



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

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active 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 <https://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf>.
- Product warranties—For product warranty information, visit <https://www.juniper.net/support/warranty/>.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

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

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

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

Creating a Service Request with JTAC

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

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

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

CHAPTER 1

Virtual Chassis Overview

- Understanding EX9200 Virtual Chassis on page 15
- Understanding Virtual Chassis Components on page 16

Understanding EX9200 Virtual Chassis



CAUTION: We do not recommend using EX9200 switches in a Virtual Chassis, and support for that architecture was phased out as of Junos OS Release 17.1R1. For deployments with EX9200 switches, we recommend planning or moving to MC-LAG or Junos Fusion Enterprise architectures instead of using a Virtual Chassis.

EX9200 Virtual Chassis brings the Virtual Chassis flexible, scaling switch solution to Juniper Networks EX9200 Ethernet Switches. You can connect two EX9200 switches into an EX9200 Virtual Chassis and manage the interconnected switches 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 devices can be managed as a single device, 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 flatter, simplified Layer 2 network topology that minimizes or eliminates the need for loop prevention protocols such as Spanning Tree Protocol (STP).



NOTE: Starting with Junos OS Release 17.1R1, EX9200 switches support EX9200-RE2 module. You cannot form a Virtual Chassis using an EX9200 switch with an EX9200-RE2 module installed in it. If inadvertently configured as a Virtual Chassis, the device will not start up properly; use the [request virtual-chassis member-id delete <force>](#) command to remove the Virtual Chassis setting.

You configure an EX9200 Virtual Chassis by configuring optical interfaces 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.

You can increase VCP bandwidth between member switches by configuring multiple interfaces between the same two switches into VCPs. When multiple VCPs are interconnecting the same two member switches, a link aggregation group (LAG) bundle is automatically formed when the VCPs are on interfaces supporting identical speeds. For instance, if you have two 10-Gigabit SFP+ interfaces configured as VCPs between member switches, a LAG with two member links with 20 Gbps of total bandwidth is formed.

An EX9200 Virtual Chassis can be composed of any two EX9200 switches. Both EX9200 switches must have dual Routing Engines installed, and all Routing Engines must be running the same version of Junos OS. All combinations of EX9204, EX9208, and EX9214 switches can be interconnected to form a Virtual Chassis.

Release History Table

Release	Description
17.1R1	We do not recommend using EX9200 switches in a Virtual Chassis, and support for that architecture was phased out as of Junos OS Release 17.1R1.

Related Documentation

- [Configuring an EX9200 Virtual Chassis on page 38](#)

Understanding Virtual Chassis Components

This topic describes the components of an EX series or a QFX Series Virtual Chassis.

- An EX Series Virtual Chassis is a supported combination of standalone EX Series switches interconnected and managed as a single chassis. 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.



NOTE: We do not recommend using EX9200 switches in a Virtual Chassis, and support for that architecture was phased out as of Junos OS Release 17.1R1. For deployments with EX9200 switches, we recommend planning or moving to MC-LAG or Junos Fusion Enterprise architectures instead of using a Virtual Chassis.

- A QFX Series Virtual Chassis is a supported combination of standalone QFX3500, QFX3600, QFX5100, QFX5110, or QFX5200 switches interconnected and managed as a single chassis. EX4300 switches (excluding multigigabit models (EX4300-48MP)) can also be interconnected into a mixed Virtual Chassis with QFX3500, QFX3600, and QFX5100 switches.

This topic does not discuss Virtual Chassis Fabric components. For information on Virtual Chassis Fabric components, see *Understanding Virtual Chassis Fabric Components*.

- [Maximum Switch Support on page 17](#)
- [Virtual Chassis Ports \(VCPs\) on page 20](#)
- [Master Routing Engine Role on page 23](#)
- [Backup Routing Engine Role on page 24](#)
- [Linecard Role on page 25](#)
- [Member Switch and Member ID on page 26](#)
- [Mastership Priority on page 26](#)
- [Virtual Chassis Identifier \(VCID\) on page 27](#)
- [Nonvolatile Storage in a Virtual Chassis on page 27](#)

Maximum Switch Support

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

- [Maximum Number of Switches in an EX Series Virtual Chassis on page 17](#)
- [Maximum Switch Support in a QFX Series Virtual Chassis \(Including Mixed Virtual Chassis with EX Series Switches\) on page 19](#)

Maximum Number of Switches in an EX Series Virtual Chassis

[Table 3 on page 17](#) lists the maximum member switch support by EX Series 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.
EX2300 Virtual Chassis	<p>15.1X53-D50—Initial release. Support for up to four EX2300 member switches.</p> <p>18.1R2—Support for up to four multigigabit EX2300 (EX2300-24MP and EX2300-48MP) member switches.</p> <p>18.4R1—Starting in Junos OS Release 18.4R1, up to four of any model EX2300 member switches (including multigigabit models and any other EX2300 switches) can be combined in the same Virtual Chassis.</p>

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

Maximum Member Switch Support	Initial Junos OS Release
EX3300 Virtual Chassis	<p>11.3R1—Initial release. Support for up to six EX3300 member switches</p> <p>12.2R1—Starting in Junos OS Release 12.2R1, an EX3300 Virtual Chassis can support up to ten EX3300 member switches.</p>
EX3400 Virtual Chassis	15.1X53-D50—Initial release. Support for up to ten EX3400 member switches
EX4200 Virtual Chassis	9.0R1—Initial release. Support for up to ten EX4200 member switches
EX4300 Virtual Chassis	<p>13.2X50-D10—Initial release. Support for up to ten EX4300 member switches</p> <p>13.2X50-D20—Starting in Junos OS Release 13.2X50-D20, EX4300 switch support was added in a mixed QFX Series Virtual Chassis or in a VCF.</p> <p>18.2R1—Starting in Junos OS Release 18.2R1 with the introduction of EX4300 multigigabit model switches (EX4300-48MP), an EX4300 Virtual Chassis can contain up to ten EX4300 multigigabit model switches as a non-mixed Virtual Chassis or a combination of EX4300 multigigabit model switches with other EX4300 switches as a mixed EX4300 Virtual Chassis.</p>
EX4500 Virtual Chassis	<p>11.1R1—Initial release. Support for up to two EX4500 switches</p> <p>11.4R1—Support for up to ten EX4500 member switches</p>
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	<p>11.1R1—Initial release. Support for up to two EX4500 switches and up to eight EX4200 switches</p> <p>11.2R1—Support for up to nine EX4200 switches</p> <p>11.4R1—Support for up to nine EX4500 switches</p>
Mixed EX4200 and EX4550 Virtual Chassis	12.2R1—Initial release. Support for up to ten total EX4200 and EX4550 switches

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

Maximum Member Switch Support	Initial Junos OS Release
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. NOTE: EX4300 multigigabit model (EX4300-48MP) switches are not supported in a mixed Virtual Chassis with EX4600 switches.
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. NOTE: Support for EX9200 switches in a Virtual Chassis was phased out as of Junos OS Release 17.1R1. For deployments with EX9200 switches, we recommend planning or moving to MC-LAG or Junos Fusion Enterprise architectures instead of using a Virtual Chassis.

Maximum Switch Support in a QFX Series Virtual Chassis (Including Mixed Virtual Chassis with EX Series Switches)

In a QFX5200 Virtual Chassis, you can interconnect up to a maximum of 3 standalone QFX5200 switches as a non-mixed Virtual Chassis. (QFX5200 switches cannot be mixed with other types of switches in a Virtual Chassis.)

For all other QFX Series Virtual Chassis, you can interconnect up to 10 standalone switches in the following supported combinations:

- QFX5110 switches or a combination of QFX5110 switches and QFX5100 switches (a non-mixed Virtual Chassis)
- QFX5100 switches (a non-mixed Virtual Chassis)
- QFX5100 switches with any combination of QFX3500 and QFX3600 switches, and EX4300 switches excluding the multigigabit models (a mixed mode Virtual Chassis)
- QFX3500 switches, or QFX3600 switches, or any combination of QFX3500 switches and QFX3600 switches (a non-mixed Virtual Chassis)
- Any combination of QFX3500 and QFX3600 switches with EX4300 switches excluding the multigigabit models (a mixed mode Virtual Chassis)



NOTE: In Junos OS release 13.2X51-D20, you can interconnect only up to four QFX5100-96S switches in a non-mixed QFX5100 Virtual Chassis. Starting in Junos OS release 13.2X51-D25, you can configure up to ten QFX5100-96S switches into a mixed or non-mixed QFX5100 Virtual Chassis.

Virtual Chassis Ports (VCPs)

You set up a Virtual Chassis by configuring Virtual Chassis ports (VCPs) on the member switches, and interconnecting the switches using the VCPs. VCPs are responsible for passing all data and control traffic between member switches in the Virtual Chassis.

- [Virtual Chassis Port Options on page 20](#)
- [Automatic Virtual Chassis Port \(VCP\) Conversion on page 22](#)
- [Virtual Chassis Port Link Aggregation Groups on page 23](#)

Virtual Chassis Port Options

Some switches have dedicated VCPs; these ports can only be used as VCPs and cannot be reconfigured as network ports. Dedicated VCPs allow you to interconnect switches without requiring any additional interface configuration.

Some switches have ports that are configured as VCPs by default. You do not need to explicitly configure those as VCPs to use them to interconnect those switches into a Virtual Chassis.

Most switches have optical or uplink ports that can also be configured as VCPs.

To interconnect switches that do not have dedicated or default-configured VCPs, or to interconnect switches across greater distances than allowed by a dedicated VCP connection, you must configure the VCPs. Also, when adding switches to an existing Virtual Chassis, or adding new redundant links between existing members, if the automatic VCP conversion feature is enabled, under the right conditions the ports on both sides of the connection will convert into VCPs automatically (see [“Automatic Virtual Chassis Port \(VCP\) Conversion” on page 22](#)).

[Table 4 on page 20](#) summarizes the available VCP options on switches in an EX Series or QFX Series Virtual Chassis. For complete details on where dedicated VCPs, default-configured VCPs, or ports that can be configured as VCPs are located on a switch, and what transceivers and cables are supported to use for VCP connections on the switch, see the hardware documentation for that type of switch.

Table 4: VCP Options by Switch Type

Switch	Dedicated VCPs	Default VCPs	Ports that can be configured as VCPs
EX2200	None	None	Any uplink ports All RJ-45 interfaces, including built-in network ports with 10/100/1000BASE-T Gigabit Ethernet connectors and 1000BASE-T RJ-45 transceivers

Table 4: VCP Options by Switch Type (continued)

Switch	Dedicated VCPs	Default VCPs	Ports that can be configured as VCPs
EX2300 (including multigigabit EX2300 models)	None	None	Uplink ports with SFP+ transceivers NOTE: You cannot use ports with SFP transceivers as VCPs on EX2300 switches to form a Virtual Chassis.
EX3300	None	Uplink ports 2 and 3	Any uplink ports
EX3400	None	All QSFP+ uplink ports	Any SFP+ uplink ports NOTE: You cannot use ports with SFP transceivers as VCPs on EX3400 switches to form a Virtual Chassis.
EX4200	2 ports on rear panel	None	Any uplink module ports (SFP, SFP+, or XFP) or through an SFP+ port on the EX4200-24F switch NOTE: You cannot set a 1000BASE-T copper SFP transceiver (EX-SFP-1GE-T) connection as a VCP on EX4200 switches.
EX4300	None	All QSFP+ ports	Any uplink ports installed with SFP+ or QSPF+ transceivers
EX4300 Multigigabit Models (EX4300-48MP)	4 40-Gbps QSFP+ ports on rear panel	None	None
EX4500 and EX4550	Two ports on the Virtual Chassis module	None	Any SFP+ port NOTE: You cannot use SFP+ uplink ports installed with 1000BASE-T copper SFP transceivers (EX-SFP-1GE-T) as VCP connections on EX4500 and EX4550 switches.
EX4600	None	None	Any SFP+ and QSFP+ ports
QFX3500 and QFX3600	None	None	Any non-channelized 40-Gbps QSFP+ interfaces
QFX5100	None	None	Any non-channelized 40-Gbps QSFP+ interfaces
QFX5110	None	None	Any 100-Gbps or 40-Gbps QSFP28 ports Any non-channelized 40-Gbps QSFP+ interfaces
QFX5200	None	None	Any 40-Gbps QSFP+ ports In Junos OS Release 17.3R2-S4, 100-Gbps QSFP28 ports are also supported as VCPs on QFX5200 switches.

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

QSFP+ interfaces that have been channelized into SFP+ interfaces using a breakout cable cannot be configured into VCPs.

Automatic Virtual Chassis Port (VCP) Conversion

When the automatic VCP conversion feature is enabled and you cable a new link from a new switch being added into an existing Virtual Chassis, or add a redundant link between two members of a Virtual Chassis, ports that can be VCPs are automatically converted into VCPs under the following conditions:

- Link Layer Discovery Protocol (LLDP) or LLDP-Media Endpoint Discovery (LLDP-MED) is enabled on the interfaces for the members on both ends of the new link. The two sides exchange LLDP packets to accomplish the port conversion.
- The Virtual Chassis must be preprovisioned with the switches on both sides of the link already configured in the members list of the Virtual Chassis using the **set virtual-chassis member** command.
- The interfaces for the ports on both ends of the link are not already configured as VCPs. Both sides of the link must be in the same state to handshake and establish the VCP link.

Using automatic VCP conversion when adding a switch to a preprovisioned Virtual Chassis is also referred to as *autoprovisioning* the new member.

Ports that are configured as VCPs by default on a switch or that were previously configured into VCPs must be converted back into network ports using the **request virtual-chassis vc-port delete** command for the port to be eligible for automatic VCP conversion. A port that has been automatically converted into a VCP is not automatically converted back into a network port when you remove a switch from a Virtual Chassis and disconnect the link.

Automatic VCP conversion is enabled by default on all Virtual Chassis, except in the following cases:

- Starting in Junos OS Releases 15.1R7 and 14.1X53-D47, in EX2200, EX3300, EX4200, EX4500, and EX4550 Virtual Chassis, automatic VCP conversion is disabled by default. If desired, you can enable the feature by configuring the **auto-conversion** statement at the **[edit virtual-chassis]** hierarchy level on the Virtual Chassis.



CAUTION: When automatic VCP conversion is enabled in a Virtual Chassis with switches that have dedicated VCPs (EX4200, EX4500, or EX4550 Virtual Chassis), if network or uplink ports are automatically converted into VCPs to create a redundant link with a dedicated VCP connection between the same two Virtual Chassis members, you must reboot the Virtual Chassis to avoid creating a traffic loop within the Virtual Chassis. (The same issue can occur even if the ports are manually converted into VCPs to create the redundant VCP link with a dedicated VCP link, so the reboot is required to avoid traffic looping in that case as well.)

- Starting in Junos OS Releases 14.1X53-D47, 17.4R2, 18.1R3, 18.2R2, and 18.3R1 for EX4300, EX4600, and QFX Series Virtual Chassis (which have the automatic VCP conversion feature enabled by default), you can choose to disable the feature by configuring the **no-auto-conversion** statement at the **[edit virtual-chassis]** hierarchy level on the Virtual Chassis. To return to the default behavior to re-enable automatic VCP conversion, delete the **no-auto-conversion** statement from the configuration.

Virtual Chassis Port Link Aggregation Groups

You can increase VCP bandwidth between member switches by configuring multiple interfaces between the same two switches into VCPs. When multiple VCPs interconnect the same two member switches, a Link Aggregation Group (LAG) or bundle is automatically formed when the VCPs are on interfaces supporting identical speeds. For example, if you have two 40-Gbps QSFP+ interfaces configured as VCPs between member switches, a LAG with two member links with 80-Gbps of total bandwidth is formed. However, 10-Gigabit SFP+ and 40-Gbps QSFP+ interfaces configured as VCPs will not become members of the same LAG.

Within a Virtual Chassis, you can also configure network interfaces located on different Virtual Chassis member switches to form a LAG, which provides load-balancing and redundancy for network traffic forwarded by the Virtual Chassis. See *Understanding Virtual Chassis Port Link Aggregation* for details on the difference between VCP LAGs and network interface LAGs within a Virtual Chassis.

Master Routing Engine Role

In a Virtual Chassis, each member switch is assigned one of two roles: Routing Engine role or linecard role, and for Routing Engine role, receives a further designation as the master or backup Routing Engine.

The member that functions as the master in the Routing Engine role in the Virtual Chassis:

- Manages the member switches.
- Runs Junos OS for the switches as a master Routing Engine.
- 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.)

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*.

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.

All switches that are not assigned the master or backup Routing Engine role function in the linecard role.

Use the following guidelines for assigning Routing Engine roles to the switches in a mixed Virtual Chassis:

- 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.
- In a mixed EX4300 Virtual Chassis composed of EX4300 multigigabit model (EX4300-48MP) and other EX4300 model switches, the switches in the Routing Engine role should always be EX4300 multigigabit model switches.
- In a mixed EX4600 Virtual Chassis with EX4300 switches, an EX4600 switch must assume the master role.
- In a mixed QFX Series Virtual Chassis made up of QFX5100 switches with QFX3500, QFX3600, or EX4300 switches, we recommend configuring QFX5100 switches into the Routing Engine role. If the mixed Virtual Chassis does not contain QFX5100 switches, we recommend configuring QFX3500 or QFX3600 switches into the Routing Engine role.
- In a QFX5110 Virtual Chassis with QFX5110 and QFX5100 switches, we recommend configuring only QFX5110 switches into the Routing Engine role.

Backup Routing Engine Role

The member that functions in the backup Routing Engine role in a Virtual Chassis:

- Maintains a state of readiness to take over the master Routing Engine role if the master fails.
- Runs Junos OS for the switches as a backup Routing Engine.
- Synchronizes with the master in terms of protocol states, forwarding tables, and other information, 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 Routing Engine 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*.

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.

Use the following guidelines for assigning Routing Engine roles to the switches in a mixed Virtual Chassis:

- 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.
- In a mixed EX4300 Virtual Chassis composed of EX4300 multigigabit model (EX4300-48MP) and other EX4300 model switches, the switches in the master and backup Routing Engine roles should always be EX4300 multigigabit model switches.
- In a mixed EX4600 Virtual Chassis with EX4300 switches, we strongly recommend configuring an EX4600 switch into the backup role. A mixed EX4600 and EX4300 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 a mixed QFX Series Virtual Chassis made up of QFX5100 switches with QFX3500, QFX3600, or EX4300 switches, we recommend configuring the QFX5100 switches into the Routing Engine role. If the mixed Virtual Chassis does not contain QFX5100 switches, we recommend configuring QFX3500 or QFX3600 switches into the Routing Engine role.
- In a QFX5110 Virtual Chassis with QFX5110 and QFX5100 switches, we recommend configuring only QFX5110 switches into the Routing Engine role.

Linecard Role

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

- Runs only a subset of Junos OS.
- 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 Routing Engine.

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.

Any switch can function in the linecard role in a mixed or non-mixed Virtual Chassis.

Use the following guidelines for assigning Routing Engine and linecard roles to the switches in a QFX Series Virtual Chassis:

- In a mixed QFX Series Virtual Chassis made up of QFX5100 switches with QFX3500, QFX3600, or EX4300 switches, we recommend configuring the QFX5100 switches into the Routing Engine role. If the mixed Virtual Chassis does not contain QFX5100 switches, we recommend configuring QFX3500 or QFX3600 switches into the Routing Engine role.
- In a QFX5110 Virtual Chassis made up of QFX5110 and QFX5100 switches, we recommend configuring only QFX5110 switches into the Routing Engine role.

Member Switch and Member ID

Each standalone 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*.

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 or backup Routing Engine role, or linecard role) 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 Routing Engine role.

The default value for mastership priority is **128**. 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.

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.

Nonvolatile Storage in a Virtual Chassis

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

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

- 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.



NOTE: File storage management differs in an EX8200 Virtual Chassis; see *Understanding File Storage in an EX8200 Virtual Chassis* for details.

Release History Table

Release	Description
18.4R1	Starting in Junos OS Release 18.4R1, up to four of any model EX2300 member switches (including multigigabit models and any other EX2300 switches) can be combined in the same Virtual Chassis.
18.2R1	Starting in Junos OS Release 18.2R1 with the introduction of EX4300 multigigabit model switches (EX4300-48MP), an EX4300 Virtual Chassis can contain up to ten EX4300 multigigabit model switches as a non-mixed Virtual Chassis or a combination of EX4300 multigigabit model switches with other EX4300 switches as a mixed EX4300 Virtual Chassis.
17.3R2-S4	In Junos OS Release 17.3R2-S4, 100-Gbps QSFP28 ports are also supported as VCPs on QFX5200 switches.
15.1R7	Starting in Junos OS Releases 15.1R7 and 14.1X53-D47, in EX2200, EX3300, EX4200, EX4500, and EX4550 Virtual Chassis, automatic VCP conversion is disabled by default.
14.1X53-D47	Starting in Junos OS Releases 14.1X53-D47, 17.4R2, 18.1R3, 18.2R2, and 18.3R1 for EX4300, EX4600, and QFX Series Virtual Chassis (which have the automatic VCP conversion feature enabled by default), you can choose to disable the feature by configuring the no-auto-conversion statement at the [edit virtual-chassis] hierarchy level on the Virtual Chassis.
13.2X53-D25	Starting in Junos OS release 13.2X51-D25, you can configure up to ten QFX5100-96S switches into a mixed or non-mixed QFX5100 Virtual Chassis.
13.2X51-D20	In Junos OS release 13.2X51-D20, you can interconnect only up to four QFX5100-96S switches in a non-mixed QFX5100 Virtual Chassis.
13.2X50-D20	Starting in Junos OS Release 13.2X50-D20, EX4300 switch support was added in a mixed QFX Series Virtual Chassis or in a VCF.
12.2R1	Starting in Junos OS Release 12.2R1, an EX3300 Virtual Chassis can support up to ten EX3300 member switches.

Related Documentation

- *Virtual Chassis Overview for Switches*
- *Understanding EX8200 Virtual Chassis Components*
- *Understanding EX Series Virtual Chassis*
- *Understanding QFX Series Virtual Chassis*
- *Understanding Mixed EX Series and QFX Series Virtual Chassis*
- *Configuring a QFX Series Virtual Chassis*
- *Setting an Uplink Port on an EX Series or QFX Series Switch as a Virtual Chassis Port*
- *Command Forwarding Usage with EX Series and QFX Series 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*

PART 1

Configuring a Virtual Chassis

- [Accessing and Configuring a Virtual Chassis and Managing Files on page 33](#)
- [Virtual Chassis Ports on page 47](#)
- [Upgrading Junos OS in a Virtual Chassis on page 55](#)
- [Deleting a Member ID on page 59](#)
- [Module Redundancy and GRES on page 61](#)
- [Global Roles and Local Roles that Determine Mastership and Switchover Behavior on page 67](#)
- [Minimizing Network Disruption Using Split Detection on page 73](#)

CHAPTER 2

Accessing and Configuring a Virtual Chassis and Managing Files

- [Accessing the Virtual Chassis Through the Management Interface on page 34](#)
- [Managing Files on Virtual Chassis Member Routers or Switches on page 35](#)
- [Virtual Chassis Slot Number Mapping for Use with SNMP on page 36](#)
- [Configuring an EX9200 Virtual Chassis on page 38](#)
- [Creating and Applying Configuration Groups for a Virtual Chassis on page 43](#)

Accessing the Virtual Chassis Through the Management Interface

The management Ethernet interface (**fxp0**) on an MX Series router or EX9200 switch is an out-of-band management interface, also referred to as a management port, that enables you to use Telnet or SSH to access and manage the device remotely. You typically configure the management interface with an IP address and prefix length when you first install Junos OS.

You can configure a management Ethernet interface in one of two ways to access a Virtual Chassis:

- To access the Virtual Chassis as a whole, configure a consistent IP address for the management interface using the **master-only** option. You can use this management IP address to consistently access the master (primary) Routing Engine in the master router or switch (protocol master) for the Virtual Chassis.
- To access a specific Routing Engine in an individual member router or switch of the Virtual Chassis, configure an IP address for one of the following configuration groups:
 - **member0-re0**
 - **member0-re1**
 - **member1-re0**
 - **member1-re1**



BEST PRACTICE: For most management tasks, we recommend that you access the Virtual Chassis as a whole through a consistent management IP address. For troubleshooting purposes, however, accessing a specific Routing Engine in an individual member router or switch may be useful.

To access a Virtual Chassis through the management Ethernet interface, do one of the following:

- Configure a consistent management IP address that accesses the entire Virtual Chassis through the master Routing Engine in the Virtual Chassis master router or switch.

```
{master:member0-re0}[edit]
user@host# set interfaces fxp0 unit 0 family inet address ip-address/prefix-length
master-only
```

For example, to access the entire Virtual Chassis via management IP address 10.4.5.33/16:

```
{master:member0-re0}[edit]
user@host# set interfaces fxp0 unit 0 family inet address 10.4.5.33/16 master-only
```

- Configure a management IP address that accesses a specified Routing Engine in an individual member router or switch in the Virtual Chassis.

```
{master:member0-re0}[edit groups]
user@host# set membern-ren interfaces fxp0 unit 0 family inet address
ip-address/prefix-length
```

For example, to access the Routing Engine installed in slot 1 of member router 1 (**member1-re1**) in the Virtual Chassis:

```
{master:member0-re0}[edit groups]
user@host# set member1-re1 interfaces fxp0 unit 0 family inet address 10.4.3.145/32
```

Related Documentation

- [Configuring a Consistent Management IP Address](#)

Managing Files on Virtual Chassis Member Routers or Switches

In a Virtual Chassis configuration for MX Series 5G Universal Routing Platforms or EX9200 switches, you can manage files on local and remote member routers or switches by including a member specification in the following **file** operational commands:

file archive	file copy
file checksum md5	file delete
file checksum sha1	file list
file checksum sha-256	file rename
file compare	file show

The member specification identifies the specific Virtual Chassis member router or switch and Routing Engine on which you want to manage files, and includes both of the following elements:

- The Virtual Chassis member ID (**0** or **1**)
- The Routing Engine slot number (**re0** or **re1**)

To manage files on a specific member router or switch and a specific Routing Engine:

- From operational mode, issue the **file** command and Virtual Chassis member specification:

```
{master:member0-re0}
user@host> file option member(0 | 1)-re(0 | 1):command-option
```

For example, the following **file list** command uses the **member1-re0** specification to display a list of the files in the **/config** directory on the Routing Engine in slot 0 (**re0**) in Virtual Chassis **member 1**. The router or switch forwards the command from **member**

0, where it is issued, to **member 1**, and displays the results as if the command were processed on the local device.

```
{master:member0-re0}

user@host> file list member1-re0:/config
member1-re0:
-----

/config:
.snap/
juniper.conf.1.gz
juniper.conf.2.gz
juniper.conf.3.gz
juniper.conf.gz
juniper.conf.md5
license/
license.old/
usage.db
vchassis/
```

- Related Documentation**
- *Interchassis Redundancy and Virtual Chassis Overview*
 - *Virtual Chassis Components Overview*
 - *Format for Specifying Filenames and URLs in Junos OS CLI Commands*

Virtual Chassis Slot Number Mapping for Use with SNMP

Junos OS supports the use of SNMP to monitor the routers, switches, and other devices in your network. For example, the Juniper Networks jnxBoxAnatomy enterprise-specific Chassis MIB contains the jnxFruTable object, which shows the status of field-replaceable units (FRUs) in the chassis. Within the jnxFruTable object, the jnxFruSlot object displays the slot number where the FRU is installed.

If you are using the jnxFruSlot object in jnxFruTable to display the slot numbers of line cards installed in a member router of an MX Series Virtual Chassis or a member switch of an EX9200 Virtual Chassis, keep in mind that the offset used for slot numbering in the Virtual Chassis affects the value that appears for the jnxFruSlot object.

[Table 5 on page 37](#) lists the jnxFruSlot number that appears in the jnxFruTable of the jnxBoxAnatomy MIB, and the corresponding line card physical slot number in each member router of a two-member EX9200 or MX Series Virtual Chassis. For example, a jnxFruSlot value of 15 corresponds to physical slot 3 in member 0 of the Virtual Chassis. A jnxFruSlot value of 30 corresponds to physical slot 6 in member 1 of the Virtual Chassis.

Table 5: jnxFruSlot Numbers and Corresponding Slot Numbers in an MX Series or EX9200 Virtual Chassis

jnxFruSlot Number	Line Card Slot Number	MX Series or EX9200 Virtual Chassis Member ID
Line Cards in MX Series Virtual Chassis Member ID 0 (offset = 12):		
12	0	0
13	1	0
14	2	0
15	3	0
16	4	0
17	5	0
18	6	0
19	7	0
20	8	0
21	9	0
22	10	0
23	11	0
Line Cards in MX Series Virtual Chassis Member ID 1 (offset = 24):		
24	0	1
25	1	1
26	2	1
27	3	1
28	4	1
29	5	1
30	6	1
31	7	1
32	8	1

Table 5: jnxFruSlot Numbers and Corresponding Slot Numbers in an MX Series or EX9200 Virtual Chassis (continued)

jnxFruSlot Number	Line Card Slot Number	MX Series or EX9200 Virtual Chassis Member ID
33	9	1
34	10	1
35	11	1

- Related Documentation**
- [Virtual Chassis Components Overview](#)
 - [SNMP MIB Explorer](#)

Configuring an EX9200 Virtual Chassis



CAUTION: We do not recommend using EX9200 switches in a Virtual Chassis, and support for that architecture was phased out as of Junos OS Release 17.1R1. For deployments with EX9200 switches, we recommend planning or moving to MC-LAG or Junos Fusion Enterprise architectures instead of using a Virtual Chassis.

You configure an EX9200 Virtual Chassis by configuring optical interfaces 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.



NOTE: Starting with Junos OS Release 17.1R1, EX9200 switches support the EX9200-RE2 module. You cannot form a Virtual Chassis using an EX9200 switch with an EX9200-RE2 module installed in it. If inadvertently configured as a Virtual Chassis, the device will not start up properly; use the `request virtual-chassis member-id delete <force>` command to remove the Virtual Chassis setting.

This topic includes:

- [Creating Configuration Groups for an EX9200 Virtual Chassis on page 38](#)
- [Configuring the EX9200 Virtual Chassis on page 39](#)

Creating Configuration Groups for an EX9200 Virtual Chassis

A standalone EX9200 switch uses the re0 and re1 configuration groups to apply groups on both Routing Engines. Because a Virtual Chassis with two member switches has four Routing Engines that often have a need to share configuration amongst one another, you

should create four groups—one group for each Routing Engine in the Virtual Chassis—instead of using the standard re0 and re1 configuration groups.

For a Virtual Chassis configuration consisting of two EX9200 switches, we strongly recommend that you create and apply on the switch in the master role of the Virtual Chassis the following configuration groups, instead of using the standard re0 and re1 configuration groups:

- member0-re0
- member0-re1
- member1-re0
- member1-re1

We recommend that you configure these groups before you configure your Virtual Chassis, to ensure that your configuration is always identical on all Routing Engines in the Virtual Chassis.

For information on creating and applying configuration groups for your EX9200 Virtual Chassis, see [“Creating and Applying Configuration Groups for a Virtual Chassis” on page 43](#).

Configuring the EX9200 Virtual Chassis

To configure an EX9200 Virtual Chassis:

Before you perform this procedure:

- Ensure that both EX9200 member switches in the Virtual Chassis have dual Routing Engines installed.
- Ensure all Routing Engines on both member switches are running the same version of Junos OS Release 13.2R2 or later.
- Cable the Virtual Chassis member switches together. See *Connecting a Fiber-Optic Cable*, *Installing and Removing EX9204 Switch Hardware Components*, *Installing and Removing EX9208 Switch Hardware Components*, or *Installing and Removing EX9214 Switch Hardware Components*.
- Create and configure the configuration groups, as described in [“Creating Configuration Groups for an EX9200 Virtual Chassis” on page 38](#).

1. Log onto the switch that you want to assign as member 0 in your Virtual Chassis.
2. Specify the preprovisioned configuration mode:

```
[edit virtual-chassis]
user@switch-0# set preprovisioned
```

You must use preprovisioned configuration mode to configure an EX9200 Virtual Chassis.

3. Configure the Virtual Chassis by including both member switches in the Virtual Chassis configuration:

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

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

```
user@switch-0# set member 1 serial-number serial-number role routing-engine
```

where *serial-number* is the chassis serial number of the member switch. You can retrieve the chassis serial number in the *show chassis hardware* command output or by physically viewing the serial number label on the switch. See *Locating the Serial Number on an EX9204 Switch or Component*, *Locating the Serial Number on an EX9208 Switch or Component*, or *Locating the Serial Number on an EX9214 Switch or Component* for additional information

An EX9200 Virtual Chassis supports two member switches. Both switches should be assigned the **routing-engine** role.

For instance, if you wanted to configure the switch with chassis serial number *JN1234567ABC* as member 0 and the switch with chassis serial number *JN9876543ZYX* as member 1 in your EX9200 Virtual Chassis:

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

```
user@switch-0# set member 0 serial-number JN1234567ABC role routing-engine
```

```
user@switch-0# set member 1 serial-number JN9876543ZYX role routing-engine
```

4. Disable the split and merge feature:

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

```
user@switch-0# set no-split-detection
```

Disabling split and merge ensures that all interfaces on the member switch in the master Routing Engine role remain up if the member switch in the backup Routing Engine role fails.

Split and merge is enabled by default. If the member switch in the backup Routing Engine role fails when split and merge is enabled, all interfaces on all line cards that do not contain at least one Virtual Chassis port (VCP) on the member switch in the master Routing Engine role also fail.

5. Commit the configuration:

```
[edit]
```

```
user@switch-0# commit
```

6. Enable Virtual Chassis mode and set the member ID of the switch:

```
user@switch-0>request virtual-chassis member-id set member 0
```

This command will enable virtual-chassis mode and reboot the system.

```
Continue? [yes, no] (no) yes
```

You must reboot both Routing Engines on the switch to complete this step. We recommend rebooting the switch by answering **yes** to the prompt that appears on the screen after entering the **request virtual-chassis member-id set member**

command, but you can also enter the **request system reboot** command to reboot one Routing Engine on the switch or the **request system reboot both-routing-engines** command to reboot both Routing Engines simultaneously.

7. Log onto the switch that you want to assign as member 1 in your Virtual Chassis.
8. Enable Virtual Chassis mode and set the member ID of the switch:

```
user@switch-1>request virtual-chassis member-id set member 1
This command will enable virtual-chassis mode and reboot the system.
Continue? [yes, no] (no) yes
```

You must reboot both Routing Engines on the switch to complete this step. We recommend rebooting the switch by answering **yes** to the prompt that appears on the screen after entering the **request virtual-chassis member-id set member** command, but you can also enter the **request system reboot** command to reboot one Routing Engine on the switch or the **request system reboot both-routing-engines** command to reboot both Routing Engines simultaneously.

9. Log back onto member 0 after the reboot is complete. Configure the interfaces that you want to configure as VCPs as VCPs:

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



NOTE: A VCP is not created until the **request virtual-chassis vc-port set** command is enabled on the interfaces on the member switches at both ends of the link.

The **request virtual-chassis vc-port set** is enabled on the interface on the other end of the link in step 10 to complete the VCP configuration process.

For instance, you can configure port 0 on PIC slot 1 in FPC slot 1 as a VCP using the following command:

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

10. Log back onto member 1 after the reboot is complete. Configure the interfaces that you want to configure as VCPs:

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



NOTE: You can configure multiple links into VCPs. If you configure interfaces of the same speed between the member switches into the VCPs, the VCPs automatically form a Link Aggregation group (LAG) bundle.

You can use the [show virtual-chassis vc-port](#) command to verify that a VCP is created.

You can use the [show virtual-chassis](#) command to verify Virtual Chassis status.

Release History Table

Release	Description
17.1R1	We do not recommend using EX9200 switches in a Virtual Chassis, and support for that architecture was phased out as of Junos OS Release 17.1R1.

Related Documentation

- [Deleting Member IDs in a Virtual Chassis Configuration on page 59](#)
- [Deleting Virtual Chassis Ports in a Virtual Chassis Configuration on page 51](#)
- [Configuring Virtual Chassis Ports to Interconnect Member Routers or Switches on page 48](#)
- [Creating and Applying Configuration Groups for a Virtual Chassis on page 43](#)

Creating and Applying Configuration Groups for a Virtual Chassis

For a Virtual Chassis configuration consisting of two MX Series routers or two EX9200 switches, each of which supports dual Routing Engines, you must create and apply on the master device of the Virtual Chassis the following configuration groups, instead of using the standard **re0** and **re1** configuration groups:

- **member0-re0**
- **member0-re1**
- **member1-re0**
- **member1-re1**



NOTE: The *membern-ren* naming format for configuration groups is reserved for exclusive use by member routers or switches in EX9200 or MX Series Virtual Chassis configurations.

Using configuration group names of the form *membern-ren* in an existing non-Virtual Chassis configuration or configuration script could interfere with Virtual Chassis operation. This misconfiguration could cause the router or switch to assign no IP address or an incorrect IP address to the **fxp0** management Ethernet interface, and could result in a display of the Amnesiac prompt during login.

To create and apply configuration group information from the router or switch to be configured as the master of the Virtual Chassis:

1. In the console window on the master router or switch (**member 0** in this procedure), create and apply the **member0-re0** configuration group.

```
[edit]
user@host# copy groups re0 to member0-re0
user@host# set apply-groups member0-re0
```

2. Delete the standard **re0** configuration group from the global configuration on **member 0**.

```
[edit]
user@host# delete apply-groups re0
user@host# delete groups re0
```

3. Create and apply the **member0-re1** configuration group.

```
[edit]
user@host# copy groups re1 to member0-re1
user@host# set apply-groups member0-re1
```

4. Delete the standard **re1** configuration group from the global configuration on **member 0**.

```
[edit]
user@host# delete apply-groups re1
user@host# delete groups re1
```

5. Create and apply the **member1-re0** configuration information.

```
[edit]
user@host# set groups member1-re0 system host-name host-name
user@host# set groups member1-re0 system backup-router address
user@host# set groups member1-re0 system backup-router destination
destination-address
user@host# set groups member1-re0 system backup-router destination
destination-address
...
user@host# set groups member1-re0 interfaces fxp0 unit unit-number family inet
address address
user@host# set apply-groups member1-re0
```

The commands in Steps 5 and 6 set the IP address for the **fxp0** management interface and add an IP route for it in the event that routing becomes inactive.

6. Create and apply the **member1-re1** configuration information.

```
[edit]
user@host# set groups member1-re1 system host-name host-name
user@host# set groups member1-re1 system backup-router address
user@host# set groups member1-re1 system backup-router destination
destination-address
user@host# set groups member1-re1 system backup-router destination
destination-address
...
user@host# set groups member1-re1 interfaces fxp0 unit unit-number family inet
address address
user@host# set apply-groups member1-re1
```

7. Commit the configuration.



BEST PRACTICE: We recommend that you use the **commit synchronize** command to save any configuration changes to the Virtual Chassis.

For an EX9200 or MX Series Virtual Chassis, the **force** option is the default and only behavior when you issue the **commit synchronize** command. Issuing the **commit synchronize** command for a Virtual Chassis configuration has the same effect as issuing the **commit synchronize force** command.

**Related
Documentation**

- *Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis*
- [Configuring an EX9200 Virtual Chassis on page 38](#)
- *Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis*
- For more information about creating and managing configuration groups, see the *Junos OS CLI User Guide*

CHAPTER 3

Virtual Chassis Ports

- [Configuring Virtual Chassis Ports to Interconnect Member Routers or Switches on page 48](#)
- [Deleting Virtual Chassis Ports in a Virtual Chassis Configuration on page 51](#)
- [Verifying the Operation of Virtual Chassis Ports on page 52](#)

Configuring Virtual Chassis Ports to Interconnect Member Routers or Switches

To interconnect the member routers in an MX Series Virtual Chassis, you must use the **request virtual-chassis vc-port set** command to configure network ports into Virtual Chassis ports on Modular Port Concentrator/Modular Interface Card (MPC/MIC) interfaces. To interconnect the member switches into an EX9200 Virtual Chassis, you must use the **request virtual-chassis vc-port set** command to configure network ports into Virtual Chassis ports on line card interfaces. After the **request virtual-chassis vc-port set** is configured on both ends of the link, a Virtual Chassis port that is dedicated to the task of interconnecting member devices is created and the link can no longer be used as a standard network port.



NOTE: If you issue the **request virtual-chassis vc-port set** command without first installing an MX Virtual Chassis Redundancy Feature Pack license on both member routers in an MX Series Virtual Chassis, the software displays a warning message that you are operating without a valid Virtual Chassis software license.

A software license is not needed to create an EX9200 Virtual Chassis.

To configure Virtual Chassis ports:

1. Configure the Virtual Chassis ports on the router or switch configured as member 0.
 - a. Configure the first Virtual Chassis port that connects to member 1.

```
{local:member0-re0}
```

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

After the Virtual Chassis port is created, it is renamed **vcp-slot/pic/port**, and the line card associated with that port comes online. The line cards in the other member devices remain offline until the Virtual Chassis forms.

For example, the following command configures Virtual Chassis port **vcp-2/2/0** on member 0:

```
{local:member0-re0}
```

```
user@hostA> request virtual-chassis vc-port set fpc-slot 2 pic-slot 2 port 0  
vc-port successfully set
```

- b. When the first Virtual Chassis port is up on member 0, repeat Step 1a to configure the second Virtual Chassis port that connects to member 1.

```
{local:member0-re0}
```

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

2. Configure the Virtual Chassis ports on the device configured as member 1.
 - a. Configure the first Virtual Chassis port that connects to member 0.

```
{master:member1-re0}
```

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

- b. When the first Virtual Chassis port is up on member 1, repeat Step 2a to configure the second Virtual Chassis port that connects to member 0.

```
{master:member1-re0}
```

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

When all of the line cards in all of the member routers or switches are online, and the Virtual Chassis has formed, you can issue Virtual Chassis commands from the terminal window of the master router or switch.



NOTE: When the Virtual Chassis forms, the FPC slots are renumbered to reflect the slot numbering and offsets used in the Virtual Chassis instead of the physical slot numbers where the FPC is actually installed. Member 0 in the Virtual Chassis uses FPC slot numbers 0 through 11 with no offset, and member 1 uses FPC slot numbers 12 through 23, with an offset of 12.

For example, a 10-Gigabit Ethernet interface that appears as xe-14/2/2 (FPC slot 14, PIC slot 2, port 2) in the `show interfaces` command output is actually interface xe-2/2/2 (FPC slot 2, PIC slot 2, port 2) on member 1 after deducting the FPC slot numbering offset of 12 for member 1.

3. (Optional) Verify that the Virtual Chassis is properly configured and that the Virtual Chassis ports are operational.

```
{master:member0-re0}
```

```
user@hostA> show virtual-chassis status
```

```
{master:member0-re0}
```

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

4. Commit the configuration on the master router or switch.

The commit step is required to ensure that the configuration groups and Virtual Chassis configuration are propagated to both members of the Virtual Chassis.



BEST PRACTICE: We recommend that you use the `commit synchronize` command to save any configuration changes to the Virtual Chassis.

For an MX Series or Virtual Chassis, the `force` option is the default and only behavior when you issue the `commit synchronize` command. Issuing the `commit synchronize` command for an MX Series Virtual Chassis configuration has the same effect as issuing the `commit synchronize force` command.

Related Documentation

- [Guidelines for Configuring Virtual Chassis Ports](#)
- [Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis](#)
- [Configuring an EX9200 Virtual Chassis on page 38](#)
- [Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis](#)

Deleting Virtual Chassis Ports in a Virtual Chassis Configuration

You can delete a Virtual Chassis port (**vcp-slot/pic/port**) as part of the procedure for deleting a Virtual Chassis configuration. You can also delete a Virtual Chassis port when you want to replace it with a Virtual Chassis port configured on a different FPC slot, PIC slot, or port number in the router or switch. After you delete a Virtual Chassis port by using the **request virtual-chassis vc-port delete** command, the port becomes available to the global configuration and can again function as a standard network port.



NOTE: If you issue the **request virtual-chassis vc-port delete** command without first installing an MX Virtual Chassis Redundancy Feature Pack license on both member routers, the software displays a warning message that you are operating without a valid Virtual Chassis software license.

A software license is not needed to create an EX9200 Virtual Chassis.

To remove the Virtual Chassis ports from both member devices in a Virtual Chassis:

1. In the console window on the router or switch configured as **member 0**, remove one or more Virtual Chassis ports.

```
{master:member0-re0}
```

```
user@host1> request virtual-chassis vc-port delete fpc-slot fpc-slot-number pic-slot  
pic-slot-number port port-number
```

For example, the following command deletes **vcp-2/2/0** (the Virtual Chassis port on FPC slot 2, PIC slot 2, and port 0) from **member 0** in the Virtual Chassis.

```
{master:member0-re0}
```

```
user@host1> request virtual-chassis vc-port delete fpc-slot 2 pic-slot 2 port 0  
vc-port successfully deleted
```

2. In the console window on the router or switch configured as **member 1**, remove one or more Virtual Chassis ports.

```
{master:member1-re0}
```

```
user@host2> request virtual-chassis vc-port delete fpc-slot fpc-slot-number pic-slot  
pic-slot-number port port-number
```

3. (Optional) Confirm that the Virtual Chassis ports have been deleted from each of the two member routers or switches.

When you delete a Virtual Chassis port, its name (**vcp-slot/pic/port**) no longer appears in the output of the **show virtual-chassis vc-port** command. For example, the following

output for the **show virtual-chassis vc-port** command on each member router or switch confirms that all Virtual Chassis ports have been deleted from both member devices.

For member 0 (**host1**):

```
{master:member0-re0}
```

```
user@host1> show virtual-chassis vc-port all-members
```

```
member0:
```

For member 1 (**host2**):

```
{backup:member1-re0}
```

```
user@host2> show virtual-chassis vc-port all-members
```

```
member1:
```



TIP: Deleting and then re-creating a Virtual Chassis port configuration may cause the Virtual Chassis port to appear as Absent in the Status column of the **show virtual-chassis vc-port** command display. To resolve this issue, reboot the FPC that hosts the re-created Virtual Chassis port.

Related Documentation

- [Deleting a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers](#)
- [Configuring an EX9200 Virtual Chassis on page 38](#)
- [Example: Deleting a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers](#)
- [Guidelines for Configuring Virtual Chassis Ports](#)

Verifying the Operation of Virtual Chassis Ports

Purpose Verify that the Virtual Chassis ports in an MX Series or EX9200 Virtual Chassis are properly configured and operational.

Action • To display the status of the Virtual Chassis ports for both member routers or switches in the Virtual Chassis:

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

- To display the status of the Virtual Chassis ports for a specified member router or switch in the Virtual Chassis:

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

- To display the status of the Virtual Chassis ports for the member router or switch on which you are issuing the command:

```
user@host> show virtual-chassis vc-port local
```

**Related
Documentation**

- *Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis*
- [Configuring an EX9200 Virtual Chassis on page 38](#)
- *Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis*

Upgrading Junos OS in a Virtual Chassis

- Upgrading Junos OS Software in an EX9200 Virtual Chassis on page 55

Upgrading Junos OS Software in an EX9200 Virtual Chassis

You can upgrade the Junos OS in an EX9200 Virtual Chassis by installing Junos OS software onto each Routing Engine in the Virtual Chassis, and rebooting each Routing Engine individually to complete the installation. Both member switches in the EX9200 Virtual Chassis must have dual Routing Engines installed.



NOTE: Make sure all four Routing Engines in the Virtual Chassis (both Routing Engines in the master switch and both Routing Engines in the backup switch) are running the same version of Junos OS software.



CAUTION: Support for EX9200 Virtual Chassis was phased out as of Junos OS Release 17.1R1, so do not upgrade an EX9200 Virtual Chassis to Junos OS Release 17.1R1 and beyond. For deployments with EX9200 switches, we recommend planning or moving to MC-LAG or Junos Fusion Enterprise architectures instead of using a Virtual Chassis.

To upgrade Junos OS software in an EX9200 Virtual Chassis:

1. Download the Junos OS software package to the master Routing Engine on the master switch in the Virtual Chassis (VC-M).
See Downloading Software.
2. Disable nonstop active routing (NSR) from the master Routing Engine on the master switch in the Virtual Chassis (VC-M):

```
{master:member0-re0}  
[edit routing-options]
```

```
user@switch# deactivate nonstop-routing
```

3. Disable graceful Routing Engine switchover (GRES) from the master Routing Engine on the master switch in the Virtual Chassis (VC-M):

```
{master:member0-re0}  
[edit chassis redundancy]  
  
user@switch# deactivate graceful-switchover
```

4. Commit the configuration.
5. Exit CLI configuration mode.

```
{master:member0-re0}  
[edit]  
  
user@switch# exit
```

6. Install the Junos OS software package onto each Routing Engine in the Virtual Chassis:

```
{master:member0-re0}  
  
user@switch> request system software add package-name
```

On a properly formed Virtual Chassis, entering the **request system software add *package-name*** from the master Routing Engine on the master switch in the Virtual Chassis (VC-M) propagates the Junos OS software package to all four Routing Engines in the Virtual Chassis.

7. Enter CLI configuration mode, and re-enable GRES:

```
{master:member0-re0}  
  
user@switch> configure  
[edit]  
  
user@switch# edit chassis redundancy  
[edit chassis redundancy]  
  
user@switch# activate graceful-switchover
```

8. Re-enable NSR:

```
{master:member0-re0}  
[edit routing-options]  
  
user@switch# activate nonstop-routing
```


9. Commit the configuration on the master Routing Engine on the master switch in the Virtual Chassis (VC-M).
10. Reboot all Routing Engines in the Virtual Chassis by entering the **request system reboot** command with no options from the master Routing Engine on the master switch in the Virtual Chassis (VC-M):

```
{master:member0-re0}
```

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

This command reboots all Routing Engines and line cards in both member switches. The Routing Engines run the updated version of Junos OS software after the reboot is complete.

A traffic disruption occurs until all line cards return online and the Virtual Chassis reforms.

Related Documentation • [Configuring an EX9200 Virtual Chassis on page 38](#)

Deleting a Member ID

- [Deleting Member IDs in a Virtual Chassis Configuration on page 59](#)

Deleting Member IDs in a Virtual Chassis Configuration

In most cases, you delete the member ID from a member router or switch as part of the procedure for deleting a Virtual Chassis configuration. When you delete the member ID by using the **request virtual-chassis member-id delete** command, the router or switch reboots and the software disables Virtual Chassis mode on that device. After the reboot, the router or switch is no longer part of the Virtual Chassis and functions as an independent device.



NOTE: If you issue the **request virtual-chassis member-id delete** command without first installing an MX Virtual Chassis Redundancy Feature Pack license on both member routers, the software displays a warning message that you are operating without a valid Virtual Chassis software license.

A software license is not needed to create an EX9200 Virtual Chassis.

To delete the Virtual Chassis member IDs from both member routers or switches and disable Virtual Chassis mode:

1. In the console window on the router or switch configured as **member 0**, delete member ID 0.

```
{master:member0-re0}
user@host1> request virtual-chassis member-id delete

This command will disable virtual-chassis mode and reboot the system.
Continue? [yes,no] (no) yes

Updating VC configuration and rebooting system, please wait...

{master:member0-re0}
user@host1>

*** FINAL System shutdown message from root@host1 ***
System going down IMMEDIATELY
```

2. In the console window on the router or switch configured as **member 1**, delete member ID 1.

```
{master:member1-re0}

user@host2> request virtual-chassis member-id delete
This command will disable virtual-chassis mode and reboot the system.
Continue? [yes,no] (no) yes

Updating VC configuration and rebooting system, please wait...

{master:member1-re0}
user@host2>

*** FINAL System shutdown message from root@host2 ***
System going down IMMEDIATELY
```

3. (Optional) Confirm that Virtual Chassis mode has been disabled on both member routers or switches.

For example:

```
user@host1> show virtual-chassis status
error: the virtual-chassis-control subsystem is not running
```

Related Documentation

- [Deleting a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers](#)
- [Configuring an EX9200 Virtual Chassis on page 38](#)
- [Example: Deleting a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers](#)

CHAPTER 6

Module Redundancy and GRES

- [Targeted Traffic Distribution on Aggregated Ethernet Interfaces in a Virtual Chassis on page 61](#)
- [Configuring Module Redundancy for a Virtual Chassis on page 62](#)
- [Determining GRES Readiness in a Virtual Chassis Configuration on page 64](#)

Targeted Traffic Distribution on Aggregated Ethernet Interfaces in a Virtual Chassis

By default, member routers or switches in an EX9200 or MX Series Virtual Chassis use hash-based traffic distribution for subscriber interfaces in aggregated Ethernet bundles configured without link protection. The hash-based model distributes subscriber interface traffic over multiple links in the bundle, enabling you to load balance multiple traffic flows through the logical subscriber interface.

Starting in Junos OS Release 13.2, as an alternative to using hash-based distribution in an EX9200 or MX Series Virtual Chassis, you can configure targeted traffic distribution for IP demultiplexing (demux) or VLAN demux subscriber interfaces in an aggregated Ethernet bundle that is configured without link protection.

- [Targeted Distribution in a Virtual Chassis on page 61](#)
- [Benefits of Targeted Distribution on page 62](#)

Targeted Distribution in a Virtual Chassis

Targeted distribution enables you to configure the Virtual Chassis to send (target) all egress data traffic for a logical subscriber interface across a single member link in an *aggregated Ethernet bundle*, also referred to as an IEEE 802.3ad link aggregation group (LAG) bundle. You configure targeted distribution for a demux subscriber interface on the Virtual Chassis master router or switch.

With targeted distribution, the router or switch in a Virtual Chassis assigns the primary member link and backup member link for the aggregated Ethernet bundle across *all* Virtual Chassis port links that belong to the aggregated Ethernet bundle. To accomplish load balancing, the router or switch evenly distributes the demux subscriber interfaces over these member links.

Benefits of Targeted Distribution

Targeted distribution is especially useful in a Virtual Chassis configuration in which subscriber traffic enters through a Virtual Chassis port on one member router or switch and exits through a Virtual Chassis port on a different member router or switch. By combining Virtual Chassis ports from different member router or switches as member links of the aggregated Ethernet bundle, targeted distribution provides increased redundancy in the event of a chassis or link failure.

Release History Table

Release	Description
13.2	Starting in Junos OS Release 13.2, as an alternative to using hash-based distribution in an EX9200 or MX Series Virtual Chassis, you can configure targeted traffic distribution for IP demultiplexing (demux) or VLAN demux subscriber interfaces in an aggregated Ethernet bundle that is configured without link protection.

Related Documentation

- *Redundancy Mechanisms on Aggregated Ethernet Interfaces in a Virtual Chassis*
- [Configuring Module Redundancy for a Virtual Chassis on page 62](#)
- *Configuring Chassis Redundancy for a Virtual Chassis*

Configuring Module Redundancy for a Virtual Chassis

By default, a router or switch uses link redundancy for aggregated Ethernet interfaces (bundles) configured with targeted traffic distribution. Starting in Junos OS Release 13.2, as an alternative to using link redundancy, you can configure module redundancy, also known as FPC redundancy, for a Virtual Chassis configured with targeted traffic distribution for IP demux or VLAN demux subscribers on aggregated Ethernet interfaces.

In a Virtual Chassis, module redundancy assigns the primary link and backup link to *different* MPC/MIC modules or line cards, regardless of the Virtual Chassis role (master or backup) of the member device in which the module is installed. Module redundancy provides redundancy protection if a module or a link in the Virtual Chassis fails.

Before you begin:

- Configure a Virtual Chassis consisting of two MX Series routers or two EX9200 switches.
See Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis
- Ensure that the aggregated Ethernet bundle is configured *without* link protection.
See Configuring Aggregated Ethernet Link Protection

To configure module redundancy:

1. Log in to the console on the master device of the Virtual Chassis.

2. Specify that you want to configure the demux logical interface.

```
{master:member0-re0} [edit]
user@host# edit interfaces demux0 unit logical-unit-number
```

3. Enable targeted distribution for the interface.

```
{master:member0-re0} [edit interfaces demux0 unit logical-unit-number]
user@host# set targeted-distribution
```

4. Specify the aggregated Ethernet bundle for which you want to configure module redundancy.

```
{master:member0-re0} [edit]
user@host# edit interfaces aenumber aggregated-ether-options
```

5. Enable module (FPC) redundancy for the specified aggregated Ethernet bundle.

```
{master:member0-re0} [edit interfaces aenumber aggregated-ether-options]
user@host# set logical-interface-fpc-redundancy
```



BEST PRACTICE: We recommend that you do not configure both module (FPC) redundancy and chassis redundancy for the same aggregated Ethernet interface in the Virtual Chassis. If you do, module redundancy takes precedence over chassis redundancy.

Release History Table

Release	Description
13.2	Starting in Junos OS Release 13.2, as an alternative to using link redundancy, you can configure module redundancy, also known as FPC redundancy, for a Virtual Chassis configured with targeted traffic distribution for IP demux or VLAN demux subscribers on aggregated Ethernet interfaces.

Related Documentation

- [Targeted Traffic Distribution on Aggregated Ethernet Interfaces in a Virtual Chassis on page 61](#)
- [Redundancy Mechanisms on Aggregated Ethernet Interfaces in a Virtual Chassis](#)
- [Configuring Chassis Redundancy for a Virtual Chassis](#)

Determining GRES Readiness in a Virtual Chassis Configuration

Depending on the configuration, a variable amount of time is required before a router or switch is ready to perform a graceful Routing Engine switchover (GRES). Attempting a GRES operation before the device is ready can cause system errors and unexpected behavior.

To determine whether the member routers or switches in a Virtual Chassis configuration are ready for a GRES operation from a database synchronization perspective, you can issue the **request virtual-chassis routing-engine master switch check** command from the Virtual Chassis master router or switch (VC-Mm) before you initiate the GRES operation. Using the **request virtual-chassis routing-engine master switch check** command before you initiate the GRES operation ensures that the subscriber management and kernel databases on both member routers or switches are synchronized and ready for the GRES operation.

To determine whether the member routers or switches are ready for GRES from a database synchronization perspective:

1. Issue the **request virtual-chassis routing-engine master switch check** command from the Virtual Chassis master router or switch (VC-Mm).

```
{master:member0-re0}
```

```
user@host> request virtual-chassis routing-engine master switch check
```

The **request virtual-chassis routing-engine master switch check** command checks various system and database components on the member routers or switches to determine whether they are ready for GRES, but does not initiate the global GRES operation itself. The readiness check includes ensuring that a system timer, which expires after 300 seconds, completes before the global GRES operation begins.

2. Review the results of the **request virtual-chassis routing-engine master switch check** command to determine whether the member routers or switches are ready for a GRES operation from a database synchronization perspective.
 - If the member routers or switches are ready for GRES, the **request virtual-chassis routing-engine master switch check** command displays a message confirming GRES readiness. For example:

```
{master:member0-re0}
```

```
user@host> request virtual-chassis routing-engine master switch check
Switchover Ready
```

- If the member routers or switches are not ready for GRES, the **request virtual-chassis routing-engine master switch check** command displays information about the readiness of the system. For example:


```
{master:member0-re0}
```

```
user@host> request virtual-chassis routing-engine master switch check
```

```
error: chassisd Not ready for mastership switch, try after 217 secs.  
mastership switch request NOT honored, backup not ready
```

The specific command output differs depending on the GRES readiness state of the member routers or switches.

**Related
Documentation**

- *Switchover Behavior in an MX Series Virtual Chassis*
- *Virtual Chassis Components Overview*
- [Global Roles and Local Roles in a Virtual Chassis on page 69](#)
- *Understanding Graceful Routing Engine Switchover*

CHAPTER 7

Global Roles and Local Roles that Determine Mastership and Switchover Behavior

- [Mastership Election in a Virtual Chassis on page 67](#)
- [Global Roles and Local Roles in a Virtual Chassis on page 69](#)
- [Switching the Global Master and Backup Roles in a Virtual Chassis Configuration on page 71](#)

Mastership Election in a Virtual Chassis

In a two-member MX Series or EX9200 Virtual Chassis, either member device can be elected as the master device (also known as the protocol master, or VC-M) of the Virtual Chassis. The first member device to join the Virtual Chassis becomes the initial master device by default. After the Virtual Chassis is formed with both member devices, the Virtual Chassis Control Protocol (VCCP) software runs a mastership election algorithm to elect the master device for the Virtual Chassis configuration.

If the master device in a Virtual Chassis fails, the backup device (also known as the protocol backup, or VC-B) takes over mastership of the Virtual Chassis. You can also switch the global roles of the master device and backup device in a Virtual Chassis by issuing the [request virtual-chassis routing-engine master switch](#) command.



NOTE: You cannot configure mastership election for an MX Series or EX9200 Virtual Chassis in the current release.

The VCCP software uses the following algorithm to elect the master device for an EX9200 or MX Series Virtual Chassis:

1. Choose the member device that has the highest value for the internal mastership election flag.

The mastership election algorithm uses an internal flag that keeps track of the member state for the purpose of electing the Virtual Chassis master device. In most cases, VCCP elects the member device with the higher flag value over the member device with the lower flag value as the protocol master.

To display the mastership election flag value, issue the **show virtual-chassis protocol database extensive** command. The flag value used for mastership election appears in the **TLVs** field of the command output, as shown in the following example:

```
{master:member1-re0}

user@host> show virtual-chassis protocol database member 0 extensive
...
TLVs:
  Node Info: Member ID: 1, VC ID: 5a6a.e747.8511, Flags: 3, Priority: 129
            System ID: 001d.b510.0800, Device ID: 1
...
```

2. Choose the member device with the highest mastership priority value.

The mastership priority value is assigned to the member device by the VCCP software, and is not configurable in the current release. The mastership priority value can be one of the following:

- **129**—The **routing-engine** role is assigned to the member device.
- **128**—No role is assigned to the member device.
- **0**—The **line-card** role is assigned to the member device (not supported in the current release).

To display the mastership priority value for the member devices in the Virtual Chassis, issue the **show virtual-chassis status** command.

3. Choose the member device that is active in the Virtual Chassis.
4. Choose the member device that belongs to the Virtual Chassis with the largest number of members.



NOTE: This criterion is not used in the current release because all EX9200 and MX Series Virtual Chassis configurations have two member devices.

5. Choose the member device that is the accepted (elected) protocol master of the Virtual Chassis.
6. Choose the member device that is the current protocol master (VC-M) of the same Virtual Chassis.
7. Choose the member device that is the current protocol backup (VC-B) of the same Virtual Chassis.
8. Choose the member device that has been part of the Virtual Chassis configuration for the longest period of time.
9. Choose the member device that was the previous protocol master of the same Virtual Chassis.
10. Choose the member device with the lowest media access control (MAC) address.

- Related Documentation**
- [Virtual Chassis Components Overview](#)
 - [Global Roles and Local Roles in a Virtual Chassis on page 69](#)
 - [Switching the Global Master and Backup Roles in a Virtual Chassis Configuration on page 71](#)

Global Roles and Local Roles in a Virtual Chassis

In a Virtual Chassis configuration for MX Series 5G Universal Routing Platforms or EX9200 switches, each of the two member devices and each of the two Routing Engines in each member device has a distinct role. A *global role* defines the function of each member device in the Virtual Chassis, and applies globally across the entire Virtual Chassis. A *local role* defines the function of each Routing Engine in the member device, and applies locally only to that member device.

Global roles change when you switch the Virtual Chassis mastership, and both global roles and local roles change when you switch the Routing Engine mastership in one of the member devices. In addition, the **line-card** global role, though not supported in a preprovisioned configuration for a two-member MX Series or EX9200 Virtual Chassis, applies in the context of split detection behavior.

This topic describes the global roles and local roles in a MX Series or EX9200 Virtual Chassis so you can better understand how the Virtual Chassis behaves during a global mastership switch, a local Routing Engine switchover, or when split detection is enabled.

- [Role Name Format on page 69](#)
- [Global Role and Local Role Descriptions on page 70](#)

Role Name Format

The global and local role names in an MX Series or EX9200 Virtual Chassis use the following format:

VC-*GlobalRole*<*LocalRole*>

where:

- ***GlobalRole*** applies to the global function of the member device for the entire Virtual Chassis, and can be one of the following:
 - **M**—Virtual Chassis master device, also referred to as the protocol master.
 - **B**—Virtual Chassis backup device, also referred to as the protocol backup.
 - **L**—Virtual Chassis line-card device. The **line-card** role is not supported in the preprovisioned configuration for a two-member Virtual Chassis. The **line-card** role applies only in the context of split detection behavior.
- ***LocalRole*** (optional) applies to the function of the Routing Engine in the local member device, and can be one of the following:
 - **m**—Master Routing Engine

- **s**—Standby Routing Engine

Global Role and Local Role Descriptions

Table 6 on page 70 describes the global roles and local roles in an MX Series or EX9200 Virtual Chassis.

Table 6: Global Roles and Local Roles

Virtual Chassis Role	Type of Role	Description
VC-M	Global	Master device for the Virtual Chassis
VC-B	Global	Backup device for the Virtual Chassis
VC-L	Global	Line-card device for the Virtual Chassis NOTE: The line-card role is not supported in the preprovisioned configuration for a two-member MX Series or EX9200 Virtual Chassis. The line-card role applies only in the context of split detection behavior.
VC-Mm	Local	Master Routing Engine in the Virtual Chassis master device
VC-Ms	Local	Standby Routing Engine in the Virtual Chassis master device
VC-Bm	Local	Master Routing Engine in the Virtual Chassis backup device
VC-Bs	Local	Standby Routing Engine in the Virtual Chassis backup device
VC-Lm	Local	Master Routing Engine in the Virtual Chassis line-card device NOTE: The line-card role is not supported in the preprovisioned configuration for a two-member MX Series or EX9200 Virtual Chassis. The line-card role applies only in the context of split detection behavior.
VC-Ls	Local	Standby Routing Engine in the Virtual Chassis line-card device NOTE: The line-card role is not supported in the preprovisioned configuration for a two-member MX Series or EX9200 Virtual Chassis. The line-card role applies only in the context of split detection behavior.

- Related Documentation**
- [Virtual Chassis Components Overview](#)
 - [Mastership Election in a Virtual Chassis on page 67](#)
 - [Switching the Global Master and Backup Roles in a Virtual Chassis Configuration on page 71](#)
 - [Disabling Split Detection in a Virtual Chassis Configuration on page 78](#)

Switching the Global Master and Backup Roles in a Virtual Chassis Configuration

You can change the mastership in an MX Series Virtual Chassis by switching the global roles of the master router and backup router in the Virtual Chassis configuration. When you change the mastership by issuing the **request virtual-chassis routing-engine master switch** administrative command, the current master router in the Virtual Chassis (also known as the Virtual Chassis protocol master, or VC-M) becomes the backup router, and the current backup router (also known as the Virtual Chassis protocol backup, or VC-B) becomes the master router.

Before you begin:

- Make sure the system configuration is synchronized between the master router and the backup router.

If the configuration between the member routers is not synchronized when you issue the **request virtual-chassis routing-engine master switch** command, the router displays the following error message and rejects the command.

```
Error: mastership switch request NOT honored, backup not ready
```

- Make sure the Virtual Chassis is not in a transition state (for example, the backup router is in the process of disconnecting from the Virtual Chassis) when you issue the **request virtual-chassis routing-engine master switch** command.

If you attempt to issue the **request virtual-chassis routing-engine master switch** command during a transition state, the router does not process the command.

To switch the global master and backup roles:

- Issue the **request virtual-chassis routing-engine master switch** command from the Virtual Chassis master Routing Engine in the Virtual Chassis master router (VC-Mm):

```
{master:member0-re0}
```

```
user@host> request virtual-chassis routing-engine master switch
```

```
Do you want to continue ? [yes,no] (no) yes
```

If you attempt to issue the **request virtual-chassis routing-engine master switch** command from the backup router, the router displays the following error message and rejects the command.

```
error: Virtual Chassis member is not the protocol master
```

Issuing the **request virtual-chassis routing-engine master switch** command from the VC-Mm causes the global role transitions listed in [Table 7 on page 72](#).

Table 7: Virtual Chassis Global Role Transitions Before and After Mastership Switchover

Virtual Chassis Role Before Switching Mastership	Virtual Chassis Role After Switching Mastership
Master Routing Engine in Virtual Chassis master router (VC-Mm)	Standby Routing Engine in Virtual Chassis backup router (VC-Bs)
Standby Routing Engine in Virtual Chassis master router (VC-Ms)	Master Routing Engine in Virtual Chassis backup router (VC-Bm)
Master Routing Engine in Virtual Chassis backup router (VC-Bm)	Master Routing Engine in Virtual Chassis master router (VC-Mm)
Standby Routing Engine in Virtual Chassis backup router (VC-Bs)	Standby Routing Engine in Virtual Chassis master router (VC-Ms)

Related Documentation

- *Switchover Behavior in an MX Series Virtual Chassis*
- *Virtual Chassis Components Overview*
- [Global Roles and Local Roles in a Virtual Chassis on page 69](#)
- [Mastership Election in a Virtual Chassis on page 67](#)

CHAPTER 8

Minimizing Network Disruption Using Split Detection

- [Split Detection Behavior in a Virtual Chassis on page 73](#)
- [Disabling Split Detection in a Virtual Chassis Configuration on page 78](#)

Split Detection Behavior in a Virtual Chassis

If there is a disruption to a Virtual Chassis configuration for MX Series 5G Universal Routing Platforms or EX9200 Switches due to the failure of a member router or switch or one or more Virtual Chassis port interfaces, the resulting connectivity loss can cause a split in the Virtual Chassis configuration. *Split detection* identifies the split and can minimize further network disruption.



BEST PRACTICE: We recommend that you use the heartbeat connection instead of the split detection feature in an MX Series Virtual Chassis to avoid unnecessary mastership role changes during an adjacency disruption or split, and to provide additional member health information for the mastership election process.

This topic covers:

- [How Split Detection Works in a Virtual Chassis on page 73](#)
- [Effect of Split Detection on Virtual Chassis Failure Scenarios on page 74](#)

How Split Detection Works in a Virtual Chassis

Split detection is enabled by default in an EX9200 or MX Series Virtual Chassis. Optionally, you can disable split detection by including the **no-split-detection** statement at the **[edit virtual-chassis]** hierarchy level. Disabling split detection can be useful in certain Virtual Chassis configurations.



NOTE: Using the `no-split-detection` statement is prohibited when you configure a heartbeat connection, and the software prevents you from configuring both the `no-split-detection` and `heartbeat-address` statements at the same time. If you attempt to do so, the software displays an error message and causes the commit operation to fail. We recommend configuring the `heartbeat-address` statement rather than the `no-split-detection` statement.

For example, if the backup router or switch fails in a two-member Virtual Chassis configuration and split detection is enabled (the default behavior), the master router or switch takes a **line-card** role, and the line cards (FPCs) that do not host Virtual Chassis ports go offline. This state effectively halts routing and disables the Virtual Chassis configuration. By contrast, if the backup router or switch fails in a two-member Virtual Chassis configuration and split detection is disabled, the master router or switch retains mastership and maintains all of the Virtual Chassis ports, effectively resulting in a single-member Virtual Chassis consisting of only the master router or switch.



BEST PRACTICE:

Effect of Split Detection on Virtual Chassis Failure Scenarios

The behavior of a Virtual Chassis during certain failure scenarios depends on whether split detection is enabled or disabled. [Table 8 on page 74](#) describes the effect of the split detection setting on common failure scenarios in a two-member MX Series Virtual Chassis.

Table 8: Effect of Split Detection on Common Virtual Chassis Failure Scenarios

Type of Failure	Split Detection Setting	Results
Virtual Chassis port interfaces go down	Enabled	<ul style="list-style-type: none"> VC-B takes VC-M role. Previous VC-M takes line-card (VC-L) role. The line-card role isolates the router or switch and removes it from the Virtual Chassis until connectivity is restored. Result is a single-member Virtual Chassis consisting of only a single VC-M. The VC-M continues to maintain subscriber state information and route traffic. <p>When Virtual Chassis port interfaces are reconnected:</p> <ul style="list-style-type: none"> VC-M retains VC-M role. VC-L takes VC-B role. Subscribers are not affected.

Table 8: Effect of Split Detection on Common Virtual Chassis Failure Scenarios (continued)

Type of Failure	Split Detection Setting	Results
Virtual Chassis port interfaces go down	Disabled	<p>When Virtual Chassis port interfaces are disconnected:</p> <ul style="list-style-type: none"> VC-M retains VC-M role, and VC-B also takes VC-M role. The result is a Virtual Chassis with two VC-M routers or switches, each of which maintains subscriber state information. Initially, both VC-M routers or switches have a complete list of subscribers. Because the two routers or switches have the same configuration, the effect on subscribers, traffic patterns, behavior of external applications, and subscriber login and logout operations is unpredictable while the Virtual Chassis port interfaces are disconnected. <p>When Virtual Chassis port interfaces are reconnected:</p> <ul style="list-style-type: none"> Original VC-M before the disconnection resumes VC-M role, and original VC-B before the disconnection resumes VC-B role. Subscribers on the VC-M are preserved. Subscribers on the VC-B are purged. The subscribers preserved on the VC-M are unaffected, and all remaining subscribers are able to log back in to the router or switch.

Table 8: Effect of Split Detection on Common Virtual Chassis Failure Scenarios (continued)

Type of Failure	Split Detection Setting	Results
Virtual Chassis backup router or switch (VC-B) goes down	Enabled	<ul style="list-style-type: none"> VC-M takes line-card (VC-L) role, which causes all line cards (FPCs) that do not host Virtual Chassis ports to go offline. Previous VC-B is out of service. The line-card role isolates the master router or switch and removes it from the Virtual Chassis until connectivity is restored. As a result, the Virtual Chassis is left without a master router or switch, which halts interchassis routing and effectively disables the Virtual Chassis configuration. <p>When the failed router or switch is brought back into service:</p> <ul style="list-style-type: none"> The mastership election algorithm is run to determine whether the router or switch takes a VC-M or VC-B role. The Virtual Chassis then becomes operational. All subscribers can log back in to the router or switch. Previous subscriber state information is not preserved.
Virtual Chassis backup router or switch (VC-B) goes down	Disabled	<ul style="list-style-type: none"> VC-M retains VC-M role and maintains all Virtual Chassis ports. Previous VC-B is out of service. Result is a single-member Virtual Chassis consisting of only a single VC-M. The VC-M continues to maintain subscriber state information and route traffic.
Virtual Chassis master router or switch (VC-M) goes down	Split detection setting has no effect on behavior	<ul style="list-style-type: none"> VC-B takes over VC-M role regardless of whether split detection is enabled or disabled. Previous VC-M is out of service. Result is a single-member Virtual Chassis consisting of only a single VC-M. The new VC-M continues to maintain subscriber state information and route traffic. <p>When the original VC-M is brought back into service, or when the original VC-M is replaced with a new router or switch:</p> <ul style="list-style-type: none"> Original VC-M or its replacement takes VC-B role. Subscribers are not affected.

Table 8: Effect of Split Detection on Common Virtual Chassis Failure Scenarios (continued)

Type of Failure	Split Detection Setting	Results
Active access link between the VC-M and the access node, such as a digital subscriber line access multiplexer (DSLAM), goes down	Split detection setting has no effect on behavior	<ul style="list-style-type: none"> • Previous standby access link becomes the active access link between the VC-B and the access node. • Traffic is routed through the new active access link. • The VC-M continues to maintain subscriber state information and route traffic.

Related Documentation

- [Virtual Chassis Components Overview](#)
- [Global Roles and Local Roles in a Virtual Chassis on page 69](#)
- [Mastership Election in a Virtual Chassis on page 67](#)
- [Switchover Behavior in an MX Series Virtual Chassis](#)
- [Disabling Split Detection in a Virtual Chassis Configuration on page 78](#)

Disabling Split Detection in a Virtual Chassis Configuration

If there is a disruption to a Virtual Chassis due to failure of a member device or one or more Virtual Chassis port links, the resulting connectivity loss can cause a split in the Virtual Chassis configuration. Split detection, which is enabled by default in an MX Series and EX9200 Virtual Chassis, identifies the split and minimizes further network disruption.

You can disable split detection by including the **no-split-detection** statement at the **[edit virtual-chassis]** hierarchy level. Disabling split detection can be useful in certain Virtual Chassis configurations.

For example, if the backup device fails in a two-member Virtual Chassis configuration and split detection is enabled (the default behavior), the master device takes a **line-card** role, and the line cards (FPCs) that do not host Virtual Chassis ports go offline. This state effectively isolates the master router or switch and removes it from the Virtual Chassis until connectivity is restored. As a result, routing or switching is halted and the Virtual Chassis configuration is disabled. By contrast, if the backup router or switch fails in a two-member Virtual Chassis configuration and split detection is disabled, the master router or switch retains mastership and maintains all of the Virtual Chassis ports, effectively resulting in a single-member Virtual Chassis consisting of only the master device.



BEST PRACTICE: We recommend that you disable split detection for a two-member Virtual Chassis configuration if you think the backup router or switch is more likely to fail than the Virtual Chassis port interfaces to the backup router or switch. Configuring redundant Virtual Chassis ports on different line cards in each member router or switch reduces the likelihood that all Virtual Chassis port interfaces to the backup router or switch can fail.

To disable split detection:

1. Specify that you want to disable the default detection of splits in the Virtual Chassis.

```
[edit virtual-chassis]
user@host# set no-split-detection
```

2. Commit the configuration.

Disabling split detection causes different results for different types of Virtual Chassis failures. For information, see [“Split Detection Behavior in a Virtual Chassis” on page 73](#).

Related Documentation

- [Split Detection Behavior in a Virtual Chassis on page 73](#)
- [Global Roles and Local Roles in a Virtual Chassis on page 69](#)
- [Switchover Behavior in an MX Series Virtual Chassis](#)
- [Virtual Chassis Components Overview](#)

PART 2

Troubleshooting

- [Acquiring Troubleshooting Information on page 81](#)

CHAPTER 9

Acquiring Troubleshooting Information

- [Configuring the Name of the Virtual Chassis Trace Log File on page 81](#)
- [Configuring Characteristics of the Virtual Chassis Trace Log File on page 82](#)
- [Configuring Access to the Virtual Chassis Trace Log File on page 83](#)
- [Using Regular Expressions to Refine the Output of the Virtual Chassis Trace Log File on page 84](#)
- [Configuring the Virtual Chassis Operations to Trace on page 85](#)
- [traceoptions \(Virtual Chassis\) on page 87](#)

Configuring the Name of the Virtual Chassis Trace Log File

To trace operations for a Virtual Chassis, you must configure the name of the trace log file that the software saves in the `/var/log` directory.

To configure the filename for tracing Virtual Chassis operations:

- On the device to be designated as the master of the Virtual Chassis, specify the name of the trace log file.

```
[edit virtual-chassis]
user@host# set traceoptions file filename
```

Related Documentation

- [Tracing Virtual Chassis Operations for MX Series 3D Universal Edge Routers](#)
- [Configuring Preprovisioned Member Information for a Virtual Chassis](#)
- [Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis](#)

Configuring Characteristics of the Virtual Chassis Trace Log File

You can optionally configure the following characteristics of the trace log file for a Virtual Chassis:

- **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. You can optionally specify the maximum number of trace files to be from 2 through 1000. If you specify a maximum number of files with the **files** option, you must also specify a maximum file size with the **size** option.
- **Maximum trace file size**—You can configure the maximum trace file size to be from 10 KB through 1 gigabyte (GB). If you specify a maximum file size with the **size** option, you must also specify a maximum number of files with the **files** option.
- **Timestamp**—By default, timestamp information is placed at the beginning of each line of trace output. You can optionally prevent placement of a timestamp on any trace log file.
- **Appending or replacing the trace file**—By default, the router or switch appends new information to an existing trace file. You can optionally specify that the router or switch replace an existing trace file instead of appending information to it.

To configure the maximum number and maximum size of trace files:

- On the router or switch to be designated as the master of the Virtual Chassis, specify the maximum number and maximum size of the trace file.

```
[edit virtual-chassis]
user@host# set traceoptions file filename files number size maximum-file-size
```

For example, to set the maximum number of files to 20 and the maximum file size to 2 MB for a trace file named **vccp**:

```
[edit virtual-chassis]
user@host# set traceoptions file vccp files 20 size 2097152
```

When the **vccp** trace file for this example reaches 2 MB, **vccp** is renamed **vccp.0**, and a new file named **vccp** is created. When the new **vccp** file reaches 2 MB, **vccp.0** is renamed **vccp.1** and **vccp** is renamed **vccp.0**. This process repeats until there are 20 trace files. Then the oldest file (**vccp.19**) is overwritten by the newest file (**vccp.0**).

To prevent the router or switch from placing a timestamp on the trace log file:

- On the router or switch to be designated as the master of the Virtual Chassis, specify that a timestamp not appear on the trace log file:

```
[edit virtual-chassis]
user@host# set traceoptions file filename no-stamp
```

To replace an existing trace file instead of appending information to it:

- On the router or switch to be designated as the master of the Virtual Chassis, specify that the router or switch replaces an existing trace file:

```
[edit virtual-chassis]
user@host# set traceoptions file filename replace
```

Related Documentation

- *Tracing Virtual Chassis Operations for MX Series 3D Universal Edge Routers*
- *Configuring Preprovisioned Member Information for a Virtual Chassis*
- *Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis*

Configuring Access to the Virtual Chassis Trace Log File

By default, only the user who configures the tracing operation can access the log files. You can enable all users to read the log file, and you can explicitly set the default behavior of the log file.

To configure access to the trace log file for all users:

- On the router or switch to be designated as the master of the Virtual Chassis, specify that all users can read the trace log file.

```
[edit virtual-chassis]
user@host# set traceoptions file filename world-readable
```

To explicitly set the default behavior to enable access to the trace log file only for the user who configured tracing:

- On the router or switch to be designated as the master of the Virtual Chassis, specify that only the user who configured tracing can read the trace log file.

```
[edit virtual-chassis]
user@host# set traceoptions file filename no-world-readable
```

Related Documentation

- *Tracing Virtual Chassis Operations for MX Series 3D Universal Edge Routers*
- *Configuring Preprovisioned Member Information for a Virtual Chassis*
- *Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis*

Using Regular Expressions to Refine the Output of the Virtual Chassis Trace Log File

By default, the trace operation output includes all lines relevant to the logged events. You can refine the output of the trace log file for a Virtual Chassis by including regular expressions to be matched.

To refine the output of the trace log file:

- On the router or switch to be designated as the master of the Virtual Chassis, configure a regular expression to be matched.

```
[edit virtual-chassis]
user@host# set traceoptions file filename match regular-expression
```

Related Documentation

- *Tracing Virtual Chassis Operations for MX Series 3D Universal Edge Routers*
- *Configuring Preprovisioned Member Information for a Virtual Chassis*
- *Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis*

Configuring the Virtual Chassis Operations to Trace

By default, the router or switch logs only important events. You can specify which operations to trace for a Virtual Chassis by including specific tracing flags when you configure tracing. [Table 9 on page 85](#) describes the flags that you can include.

Table 9: Tracing Flags for Virtual Chassis

Flag	Description
all	Trace all operations.
auto-configuration	Trace Virtual Chassis ports that have been automatically configured.
csn	Trace Virtual Chassis complete sequence number (CSN) packets.
error	Trace Virtual Chassis errored packets.
graceful-restart	Trace Virtual Chassis graceful restart events.
hello	Trace Virtual Chassis hello packets.
krt	Trace Virtual Chassis kernel routing table (KRT) events.
lsp	Trace Virtual Chassis link-state packets.
lsp-generation	Trace Virtual Chassis link-state packet generation.
me	Trace Virtual Chassis mastership election (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 shortest-path-first (SPF) events.
state	Trace Virtual Chassis state transitions.
task	Trace Virtual Chassis task operations.

To configure the flags for the Virtual Chassis operations to be logged:

1. Specify the tracing flag that represents the operation you want to trace.

```
[edit virtual-chassis]
user@host# set traceoptions flag flag
```

2. (Optional) Specify one or more of the following additional tracing options for the specified flag:

- To generate detailed trace output, use the **detail** option.
- To disable a particular flag, use the **disable** option.
- To trace received packets, use the **receive** option.
- To trace transmitted packets, use the **send** option.

For example, to generate detailed trace output for Virtual Chassis mastership election events in received packets:

```
[edit virtual-chassis]
user@host# set traceoptions flag me detail receive
```

**Related
Documentation**

- *Tracing Virtual Chassis Operations for MX Series 3D Universal Edge Routers*
- *Configuring Preprovisioned Member Information for a Virtual Chassis*
- *Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis*

traceoptions (Virtual Chassis)

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

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

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

Description Define tracing operations for the Virtual Chassis or VCF.

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



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

Default Tracing operations are disabled.

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



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

disable—(Optional) Disable a flag.

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

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

is overwritten. If you specify a maximum number of files, you also must specify a maximum file size with the **size** option.

Range: 2 through 1000

Default: 3 files

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

- **all**—All tracing operations.



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

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

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

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

receive—(Optional) Trace received packets.

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

send—(Optional) Trace transmitted packets.

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

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

Range: 10 KB through 1 GB

Default: 128 KB

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

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

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

PART 3

Routine Monitoring

- [Monitoring a Virtual Chassis on page 93](#)

CHAPTER 10

Monitoring a Virtual Chassis

- [Verifying the Status of Virtual Chassis Member Routers or Switches on page 93](#)
- [Verifying Neighbor Reachability for Member Routers or Switches in a Virtual Chassis on page 93](#)
- [Verifying Neighbor Reachability for Hardware Devices in a Virtual Chassis on page 94](#)
- [Viewing Information in the Virtual Chassis Control Protocol Adjacency Database on page 95](#)
- [Viewing Information in the Virtual Chassis Control Protocol Link-State Database on page 95](#)
- [Viewing Information About Virtual Chassis Port Interfaces in the Virtual Chassis Control Protocol Database on page 96](#)
- [Viewing Virtual Chassis Control Protocol Statistics for Member Devices and Virtual Chassis Ports on page 97](#)

Verifying the Status of Virtual Chassis Member Routers or Switches

Purpose Verify that the member routers or switches in an MX Series or EX9200 Virtual Chassis are properly configured.

Action Display the status of the members of the Virtual Chassis configuration:

```
user@host> show virtual-chassis status
```

Related Documentation

- [Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis](#)
- [Configuring an EX9200 Virtual Chassis on page 38](#)
- [Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis](#)

Verifying Neighbor Reachability for Member Routers or Switches in a Virtual Chassis

Purpose Verify that each member router or switch in an MX Series or EX9200 Virtual Chassis has a path to reach the neighbor devices to which it is connected.

- Action**
- To display neighbor reachability information for both member devices in the Virtual Chassis:

```
user@host> show virtual-chassis active-topology all-members
```

- To display neighbor reachability information for a specified member device in the Virtual Chassis:

```
user@host> show virtual-chassis active-topology member member-id
```

- To display neighbor reachability information for the member device on which you are issuing the command:

```
user@host> show virtual-chassis active-topology local
```

- Related Documentation**
- *Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis*
 - [Configuring an EX9200 Virtual Chassis on page 38](#)
 - *Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis*

Verifying Neighbor Reachability for Hardware Devices in a Virtual Chassis

- Purpose**
- Verify that each hardware device in an MX Series Virtual Chassis or an EX9200 Virtual Chassis can reach the neighbor routers and devices to which it is connected. On the MX Series routing platform, there is only one active device for each member router.

- Action**
- To display neighbor reachability information for the devices in both member routers in the Virtual Chassis:

```
user@host> show virtual-chassis device-topology all-members
```

- To display neighbor reachability information for the device in a specified member router in the Virtual Chassis:

```
user@host> show virtual-chassis device-topology member member-id
```

- To display neighbor reachability information for the device in the member router on which you are issuing the command:

```
user@host> show virtual-chassis device-topology local
```

- Related Documentation**
- *Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis*
 - [Configuring an EX9200 Virtual Chassis on page 38](#)

- *Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis*

Viewing Information in the Virtual Chassis Control Protocol Adjacency Database

Purpose	View information about neighbors in the Virtual Chassis Control Protocol (VCCP) adjacency database for a Virtual Chassis configuration.
Action	<ul style="list-style-type: none">• To display VCCP neighbor adjacency information for both member devices in the Virtual Chassis: user@host> show virtual-chassis protocol adjacency all-members• To display VCCP neighbor adjacency information for a specified member device in the Virtual Chassis: user@host> show virtual-chassis protocol adjacency member <i>member-id</i>• To display VCCP neighbor adjacency information for the device with a specified system ID: user@host> show virtual-chassis protocol adjacency system-id• To display VCCP neighbor adjacency information for the device with a specified system ID on the specified member router or switch: user@host> show virtual-chassis protocol adjacency member <i>member-id</i> system-id• To display VCCP neighbor adjacency information for the member device on which you are issuing the command: user@host> show virtual-chassis protocol adjacency local
Related Documentation	<ul style="list-style-type: none">• <i>Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis</i>• Configuring an EX9200 Virtual Chassis on page 38• <i>Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis</i>

Viewing Information in the Virtual Chassis Control Protocol Link-State Database

Purpose	View information about protocol data unit (PDU) packets in the Virtual Chassis Control Protocol (VCCP) link-state database for a Virtual Chassis configuration.
Action	<ul style="list-style-type: none">• To display VCCP PDU information for both member routers or switches in the Virtual Chassis:

```
user@host> show virtual-chassis protocol database all-members
```

- To display VCCP PDU information for a specified member router or switch in the Virtual Chassis:

```
user@host> show virtual-chassis protocol database member member-id
```

- To display VCCP PDU information for the device with a specified system ID:

```
user@host> show virtual-chassis protocol database system-id
```

- To display VCCP PDU information for the device with a specified system ID on the specified member router or switch:

```
user@host> show virtual-chassis protocol database member member-id system-id
```

- To display VCCP PDU information for the member router or switch on which you are issuing the command:

```
user@host> show virtual-chassis protocol database local
```

Related Documentation

- *Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis*
- [Configuring an EX9200 Virtual Chassis on page 38](#)
- *Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis*

Viewing Information About Virtual Chassis Port Interfaces in the Virtual Chassis Control Protocol Database

Purpose View information in the Virtual Chassis Control Protocol (VCCP) database about Virtual Chassis port interfaces in the Virtual Chassis.

Action • To display VCCP information about Virtual Chassis port interfaces for both member routers or switches:

```
user@host> show virtual-chassis protocol interface all-members
```

- To display VCCP information about Virtual Chassis port interfaces for a specified member router or switch:

```
user@host> show virtual-chassis protocol interface member member-id
```

- To display VCCP information about a specified Virtual Chassis port interface:

```
user@host> show virtual-chassis protocol interface vcp-slot/pic/port.logical-unit-number
```


- To display VCCP information about Virtual Chassis port interfaces for the member router or switch on which you are issuing the command:

```
user@host> show virtual-chassis protocol interface local
```

**Related
Documentation**

- [Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis](#)
- [Configuring an EX9200 Virtual Chassis on page 38](#)
- [Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis](#)

Viewing Virtual Chassis Control Protocol Statistics for Member Devices and Virtual Chassis Ports

Purpose View Virtual Chassis Control Protocol (VCCP) statistics for one or both member routers or switches, or for a specified Virtual Chassis port interface, in a Virtual Chassis configuration.

Action • To display VCCP statistics for both member routers or switches in the Virtual Chassis:

```
user@host> show virtual-chassis protocol statistics all-members
```

- To display VCCP statistics for a specified member router or switch in the Virtual Chassis:

```
user@host> show virtual-chassis protocol statistics member member-id
```

- To display VCCP statistics for a specified Virtual Chassis port interface:

```
user@host> show virtual-chassis protocol statistics vcp-slot/pic/port.logical-unit-number
```

- To display VCCP statistics for the member router or switch on which you are issuing the command:

```
user@host> show virtual-chassis protocol statistics local
```

**Related
Documentation**

- [Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis](#)
- [Configuring an EX9200 Virtual Chassis on page 38](#)
- [Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis](#)

PART 4

Configuration Statements and Operational Commands

- [Configuration Statements on page 101](#)
- [Operational Commands: Administrative on page 119](#)
- [Operational Commands: Monitoring on page 129](#)

CHAPTER 11

Configuration Statements

- [aggregated-ether-options](#) on page 102
- [logical-interface-fpc-redundancy](#) (Aggregated Ethernet Subscriber Interfaces) on page 105
- [member](#) on page 106
- [no-split-detection](#) on page 107
- [preprovisioned](#) on page 108
- [role](#) on page 109
- [serial-number](#) on page 112
- [targeted-distribution](#) (Static Interfaces over Aggregated Ethernet) on page 113
- [traceoptions](#) (Virtual Chassis) on page 114
- [virtual-chassis](#) on page 117

aggregated-ether-options

List of Syntax [Syntax \(EX, MX Series\) on page 102](#)
 [Syntax \(NFX, QFX Series, EX4600, OCX1100, QFabric\) on page 102](#)

Syntax (EX, MX Series)

```

aggregated-ether-options {
    ethernet-switch-profile {
        tag-protocol-id;
    }
    (flow-control | no-flow-control);
    lacp {
        (active | passive);
        admin-key key;
        periodic interval;
        system-id mac-address;
    }
    (link-protection | no-link-protection);
    link-speed speed;
    local-bias;
    logical-interface-fpc-redundancy;
    (loopback | no-loopback);
    mc-ae {
        chassis-id chassis-id;
        events {
            iccp-peer-down {
                force-icl-down;
                prefer-status-control-active;
            }
        }
        init-delay-time seconds;
        mc-ae-id mc-ae-id;
        mode (active-active | active-standby);
        redundancy-group group-id;
        revert-time revert-time;
        status-control (active | standby);
        switchover-mode (non-revertive | revertive);
    }
    minimum-links number;
    system-priority
}

```

Syntax (NFX, QFX Series, EX4600, OCX1100, QFabric) The **fcoe-lag** and **mc-ae** statements are not supported on OCX Series switches.

```

aggregated-ether-options {
    configured-flow-control {
        rx-buffers (on | off);
        tx-buffers (on | off);
    }
    ethernet-switch-profile {
        tag-protocol-id;
        (fcoe-lag | no-fcoe-lag);
        (flow-control | no-flow-control);
    }
}

```

```

lacp mode {
    admin-key key;
    periodic interval;
    system-id mac-address;
    force-up;
}
(link-protection | no-link-protection);
link-speed speed;
local-bias;
local-minimum-links-threshold threshold-value;
(loopback | no-loopback);
mc-ae {
    chassis-id chassis-id;
    mc-ae-id mc-ae-id;
    mode (active-active);
    status-control (active | standby);
}
minimum-links number;
rebalance-periodic;
resilient-hash;
source-address-filter filter;
(source-filtering | no-source-filtering);
}

```

Hierarchy Level (EX Series, QFX Series)

[edit interfaces *aex*]

Release Information

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

Statement introduced in Junos OS Release 11.1 for the QFX Series.

Statement introduced in Junos OS Release 12.3R2.

Statements **fcoe-lag** and **no-fcoe-lag** introduced in Junos OS Release 13.2X52-D10 for the QFX Series.

Statements **force-up**, **lacp**, and **resilient-hash** introduced in Junos OS Release 14.1X53-D10 for the QFX Series.

Statement **local-minimum-links-threshold** introduced in Junos OS Release 14.1X53-D40 for the QFX Series.

Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Configure the aggregated Ethernet properties of a specific aggregated Ethernet interface.



NOTE:

- The `fcoe-lag` and `mc-ae` statements are not supported on OCX Series switches.
- The `force-up` statement is not supported on QFX10002 switches.
- The `resilient-hash` statement is not supported on QFX10002 switches.

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

Default Options are not enabled.

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

Related Documentation

- *Understanding Aggregated Ethernet Interfaces and LACP for Switches*
- *Configuring Aggregated Ethernet LACP (CLI Procedure)*
- *Example: Configuring Link Aggregation with LACP Between a QFX Series Product and an Aggregation Switch*
- *Junos OS Network Interfaces Library for Routing Devices*
- *Example: Configuring Aggregated Ethernet High-Speed Uplinks Between an EX4200 Virtual Chassis Access Switch and an EX4200 Virtual Chassis Distribution Switch*
- *Example: Configuring Aggregated Ethernet High-Speed Uplinks with LACP Between an EX4200 Virtual Chassis Access Switch and an EX4200 Virtual Chassis Distribution Switch*
- *Configuring Aggregated Ethernet Links (CLI Procedure)*
- *Configuring Aggregated Ethernet LACP (CLI Procedure)*
- *Configuring LACP Link Protection of Aggregated Ethernet Interfaces for Switches*
- *Configuring Q-in-Q Tunneling on EX Series Switches with ELS Support*
- [Junos OS Ethernet Interfaces Configuration Guide](#)

logical-interface-fpc-redundancy (Aggregated Ethernet Subscriber Interfaces)

Syntax	logical-interface-fpc-redundancy;
Hierarchy Level	[edit interfaces <i>aenumber</i> aggregated-ether-options]
Release Information	Statement introduced in Junos OS Release 11.2. Statement introduced in Junos OS Release 13.2R2 for EX Series switches.
Description	<p>Provide module redundancy for demux subscribers on aggregated Ethernet bundles configured with targeted distribution. Backup links for a subscriber are chosen on a different EQ DPC or MPC from the primary link, based on the link with the fewest number of subscribers among the links on different modules. If all links are on a single module when this is configured, backup links are not provisioned.</p> <p>By default, link redundancy is provided for the aggregated Ethernet bundle.</p>
Required Privilege Level	<p>interface—To view this statement in the configuration.</p> <p>interface-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Configuring Link and Module Redundancy for Demux Subscribers in an Aggregated Ethernet Interface • Configuring Module Redundancy for a Virtual Chassis on page 62

member

Syntax	<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>
Hierarchy Level	[edit virtual-chassis]
Release Information	<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>
Description	Configure a switch or an XRE200 External Routing Engine as a member of a Virtual Chassis or a Virtual Chassis Fabric (VCF) with characteristics specified by the available options.
Default	<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>
Options	<p><i>member-id</i>—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>
Required Privilege Level	<p>system—To view this statement in the configuration.</p> <p>system-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> <i>Autoprovisioning a Virtual Chassis Fabric</i> <i>Preprovisioning a Virtual Chassis Fabric</i>

- *Adding a Device to a Virtual Chassis Fabric*
- *Configuring a QFX Series Virtual Chassis*
- *Configuring an EX2300, EX3400, or EX4300 Virtual Chassis*
- *Configuring EX4600 Switches in a Mixed or Non-Mixed Virtual Chassis*
- [Configuring an EX9200 Virtual Chassis on page 38](#)
- *Configuring a QFX Series Virtual Chassis*

no-split-detection

Syntax	no-split-detection;
Hierarchy Level	[edit virtual-chassis]
Release Information	Statement introduced in Junos OS Release 9.3 for EX Series switches. Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
Description	<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>
Default	The split and merge feature is enabled.
Required Privilege Level	system—To view this statement in the configuration. system-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> • <i>Example: Assigning the Virtual Chassis ID to Determine Precedence During an EX4200 Virtual Chassis Merge</i> • <i>Disabling Split and Merge in a Virtual Chassis</i> • <i>Assigning the Virtual Chassis ID to Determine Precedence During a Virtual Chassis Merge</i> • Disabling Split Detection in a Virtual Chassis Configuration on page 78 • <i>Understanding Split and Merge in a Virtual Chassis</i>

preprovisioned

Syntax	preprovisioned;
Hierarchy Level	[edit virtual-chassis]
Release Information	<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>
Description	<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>
Required Privilege Level	<p>system—To view this statement in the configuration.</p> <p>system-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • <i>Preprovisioning a Virtual Chassis Fabric</i> • <i>Example: Configuring an EX4200 Virtual Chassis Using a Preprovisioned Configuration File</i> • <i>Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines</i> • <i>Configuring an EX4200, EX4500, or EX4550 Virtual Chassis (CLI Procedure)</i> • <i>Configuring an EX8200 Virtual Chassis (CLI Procedure)</i> • Configuring an EX9200 Virtual Chassis on page 38 • <i>Configuring a QFX Series Virtual Chassis</i> • <i>Removing or Replacing a Member Switch of a Virtual Chassis Configuration</i>

role

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

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

Virtual Chassis Fabric

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

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

Specify the role to be performed by each member switch. Associate the role with the member's serial number (see [serial-number](#)).

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

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

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

Explicitly configure two members as **routing-engine** and configure additional switches as members of the preprovisioned Virtual Chassis by specifying only their serial numbers. If you do not explicitly configure the role of the additional members, they function in a **line-card** role by default. In that case, a member that is functioning in a **line-card** role can

take over mastership if the members functioning as master and backup (**routing-engine** role) both fail.

EX8200 Virtual Chassis

Specify the role to be performed by each XRE200 External Routing Engine and each EX8200 member switch. Associate the role with the member's serial number (see [serial-number](#)). An EX8200 Virtual Chassis cannot function when both external Routing Engines, which must be configured in the **routing-engine** role, have failed.

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

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

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

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

In a Virtual Chassis composed of EX Series switches (except EX8200 switches) or QFX Series switches, specify two and only two members in the **routing-engine** role. The software determines which of the two members assigned the **routing-engine** role functions as master, based on the master election algorithm. See *Understanding How the Master in a Virtual Chassis Is Elected*. In these Virtual Chassis, the **routing-engine** role is associated with a switch. For a mixed Virtual Chassis or VCF, see *Understanding Mixed EX Series and QFX Series Virtual Chassis* or *Understanding Mixed Virtual Chassis Fabric* for specific recommendations or requirements for assigning the Routing Engine role based on the types of switches comprising the Virtual Chassis or VCF. The remaining switches are configured into the linecard role.

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

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

**Related
Documentation**

- *Autoprovisioning a Virtual Chassis Fabric*
- *Preprovisioning a Virtual Chassis Fabric*
- *Configuring an EX2300, EX3400, or EX4300 Virtual Chassis*
- *Configuring EX4600 Switches in a Mixed or Non-Mixed Virtual Chassis*
- [Configuring an EX9200 Virtual Chassis on page 38](#)
- *Configuring a QFX Series Virtual Chassis*
- *Removing or Replacing a Member Switch of a Virtual Chassis Configuration*

serial-number

Syntax	<code>serial-number serial-number;</code>
Hierarchy Level	[edit virtual-chassis member member-id]
Release Information	<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>
Description	<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. Serial number values are case-sensitive.</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>
Options	serial-number —Permanent serial number for the external Routing Engine or for the member switch.
Required Privilege Level	<p>system—To view this statement in the configuration.</p> <p>system-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • <i>Autoprovisioning a Virtual Chassis Fabric</i> • <i>Preprovisioning a Virtual Chassis Fabric</i> • <i>Configuring an EX2300, EX3400, or EX4300 Virtual Chassis</i> • <i>Configuring EX4600 Switches in a Mixed or Non-Mixed Virtual Chassis</i> • Configuring an EX9200 Virtual Chassis on page 38 • <i>Configuring a QFX Series Virtual Chassis</i>

targeted-distribution (Static Interfaces over Aggregated Ethernet)

Syntax	targeted-distribution;
Hierarchy Level	[edit interfaces demux0 unit <i>logical-unit-number</i>], [edit interfaces pp0 unit <i>logical-unit-number</i>]
Release Information	Statement introduced in Junos OS Release 11.2. Statement introduced in Junos OS Release 13.2R2 for EX Series switches.
Description	Configure egress data for a logical interface to be sent across a single member link in an aggregated Ethernet bundle. A backup link is provisioned and CoS scheduling resources are switched to the backup link in the event that the primary assigned link goes down. The aggregated Ethernet interface must be configured without link protection.
Required Privilege Level	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> • <i>CoS for PPPoE Subscriber Interfaces Overview</i> • <i>Configuring the Distribution Type for PPPoE Subscribers on Aggregated Ethernet Interfaces</i> • <i>Verifying the Distribution of PPPoE Subscribers in an Aggregated Ethernet Interface</i> • Targeted Traffic Distribution on Aggregated Ethernet Interfaces in a Virtual Chassis on page 61 • Configuring Module Redundancy for a Virtual Chassis on page 62 • <i>Configuring Chassis Redundancy for a Virtual Chassis</i>

traceoptions (Virtual Chassis)

Syntax

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

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

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

Description Define tracing operations for the Virtual Chassis or VCF.

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



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

Default Tracing operations are disabled.

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



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

disable—(Optional) Disable a flag.

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

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

is overwritten. If you specify a maximum number of files, you also must specify a maximum file size with the **size** option.

Range: 2 through 1000

Default: 3 files

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

- **all**—All tracing operations.



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

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

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

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

receive—(Optional) Trace received packets.

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

send—(Optional) Trace transmitted packets.

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

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

Range: 10 KB through 1 GB

Default: 128 KB

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

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

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

virtual-chassis

```

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

```

Hierarchy Level [\[edit\]](#)

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

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

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

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

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

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

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

Related Documentation

- *Autoprovisioning a Virtual Chassis Fabric*
- *Preprovisioning a Virtual Chassis Fabric*
- *Adding a Device to a Virtual Chassis Fabric*
- *Configuring a QFX Series Virtual Chassis*
- *Configuring an EX2300, EX3400, or EX4300 Virtual Chassis*
- *Configuring EX4600 Switches in a Mixed or Non-Mixed Virtual Chassis*
- [Configuring an EX9200 Virtual Chassis on page 38](#)

CHAPTER 12

Operational Commands: Administrative

- request virtual-chassis member-id delete (MX Series Virtual Chassis)
- request virtual-chassis member-id set
- request virtual-chassis routing-engine master switch
- request virtual-chassis vc-port

request virtual-chassis member-id delete (MX Series Virtual Chassis)

Syntax request virtual-chassis member-id delete
<force>

Release Information Command introduced in Junos OS Release 11.2.
Command introduced in Junos OS Release 13.2R2 for EX Series switches.

Description Remove (**delete**) the member ID from a router or switch that you want to remove from a Virtual Chassis configuration.



NOTE: Issuing the command to remove the member ID causes the device to reboot, and requires you to confirm that you want to proceed with this operation. If you do not confirm the operation, the software cancels the command. You can alternatively use the force option to reboot without requiring confirmation.

Required Privilege Level system-control

Related Documentation

- [Deleting Member IDs in a Virtual Chassis Configuration on page 59](#)
- *Example: Deleting a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers*

List of Sample Output [request virtual-chassis member-id delete on page 120](#)

Sample Output

[request virtual-chassis member-id delete](#)

```
user@host> request virtual-chassis member-id delete
```

```
This command will disable virtual-chassis mode and reboot the system.
```

```
Continue? [yes,no] (no)
```


request virtual-chassis member-id set

Syntax	<code>request virtual-chassis member-id set member <i>member-id</i></code>
Syntax (MX960, MX2010, and MX2020 Routers)	<code>request virtual-chassis member-id set member <i>member-id</i> <slots-per-chassis <i>slot-count</i>></code>
Release Information	<p>Command introduced in Junos OS Release 11.2.</p> <p>Command introduced in Junos OS Release 13.2R2 for EX Series switches.</p> <p>slots-per-chassis option added in Junos OS Release 15.1 for MX960 routers, MX2010 routers, and MX2020 routers.</p>
Description	Assign (set) a member ID and, optionally, a slot count to a router or switch that you want to add as a member of a Virtual Chassis configuration.



NOTE: Issuing the `request virtual-chassis member-id set` command causes the device to reboot, and requires you to confirm that you want to proceed with this operation. If you do not confirm the operation, the software cancels the command. After the reboot all MPCs remain powered off until the Virtual Chassis port connection is configured.

Options **member *member-id***—Assign the numeric value that identifies a member router or switch in a Virtual Chassis configuration. When you assign a member ID to a router or switch, assign the same member ID defined for this device in the preprovisioned configuration. Replace ***member-id*** with the value 0 or 1.

slots-per-chassis *slot-count*—(MX960, MX2010, and MX2020 routers) (Optional)
Identify the number of chassis slots in the Virtual Chassis member router. To ensure that a Virtual Chassis consisting of an MX2020 member router and either an MX960 or MX2010 member router forms properly, you must explicitly set the *slot-count* value for the MX960 router or MX2010 router to 20 to match the slot count of the MX2020 router.

Values:

The valid values for ***slot-count*** are as follows:

- MX960 router: 12 or 20
- MX2010 router: 12 or 20
- MX2020 router: 20

Default:

The default values for ***slot-count*** are as follows:

- MX960 router: 12

- MX2010 router: 12
- MX2020 router: 20

Required Privilege Level system-control

Related Documentation

- [Configuring Member IDs for a Virtual Chassis](#)
- [Configuring an EX9200 Virtual Chassis on page 38](#)
- [Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis](#)

List of Sample Output [request virtual-chassis member-id set \(Assigning a Member ID\) on page 122](#)
[request virtual-chassis member-id set \(Assigning a Member ID and Slot Count\) on page 122](#)

Sample Output

[request virtual-chassis member-id set \(Assigning a Member ID\)](#)

```
user@host> request virtual-chassis member-id set member 0
```

```
This command will enable virtual-chassis mode and reboot the system.
```

```
Continue? [yes,no] (no)
```

[request virtual-chassis member-id set \(Assigning a Member ID and Slot Count\)](#)

```
user@host> request virtual-chassis member-id set member 1 slots-per-chassis 20
```

```
This command will enable virtual-chassis mode and reboot the system.
```

```
Continue? [yes,no] (no)
```

request virtual-chassis routing-engine master switch

Syntax request virtual-chassis routing-engine master switch
<check>

Release Information Command introduced in Junos OS Release 11.2.
Option **check** introduced in Junos OS Release 12.2.
Command introduced in Junos OS Release 13.2R2 for EX Series switches.

Description Change the mastership in an MX Series Virtual Chassis or EX9200 Virtual Chassis by switching the global roles of the master router or switch and backup router or switch in the Virtual Chassis configuration. The **request virtual-chassis routing-engine master switch** command must be issued from the master router or switch (VC-Mm).

(MX Series routers only) The local roles (**master** and **standby**, or **m** and **s**) of the Routing Engines in the Virtual Chassis master router change after a global switchover, but the local roles of the Routing Engines in the Virtual Chassis backup router do not change. For example, the master Routing Engine in the Virtual Chassis master router (VC-Mm) becomes the standby Routing Engine in the Virtual Chassis backup router (VC-Bs) after the global switchover. By contrast, the master Routing Engine in the Virtual Chassis backup router (VC-Bm) remains the master Routing Engine in the Virtual Chassis master router (VC-Mm) after the global switchover.



NOTE: Before you issue the **request virtual-chassis routing-engine master switch** command from the master router or switch in the Virtual Chassis, make sure that the system configuration is synchronized between the master and backup router or switch. If the configuration is not synchronized, or if you attempt to issue the **request virtual-chassis routing-engine master switch** command from the backup router or switch instead of from the master router or switch, the device displays an error message and rejects the command.

If you issue the **request virtual-chassis routing-engine master switch** command when the Virtual Chassis is in a transition state (for example, the backup router or switch is disconnecting from the Virtual Chassis), the device does not process the command.

Options **check**—(Optional) Perform a check from the master router or switch to determine whether the member routers or switches are ready for GRES from a database synchronization perspective, without initiating the GRES operation itself.

Required Privilege Level system-control

Related Documentation

- [Switching the Global Master and Backup Roles in a Virtual Chassis Configuration on page 71](#)
- [Determining GRES Readiness in a Virtual Chassis Configuration on page 64](#)
- [Switchover Behavior in an MX Series Virtual Chassis](#)
- [Mastership Election in a Virtual Chassis on page 67](#)

List of Sample Output

[request virtual-chassis routing-engine master switch \(From Master Router\) on page 124](#)
[request virtual-chassis routing-engine master switch \(Error When Configuration Not Synchronized\) on page 124](#)
[request virtual-chassis routing-engine master switch \(Error When Run from Backup Router\) on page 124](#)
[request virtual-chassis routing-engine master switch check \(Ready for GRES\) on page 124](#)
[request virtual-chassis routing-engine master switch check \(Not Ready for GRES\) on page 125](#)

Sample Output

[request virtual-chassis routing-engine master switch \(From Master Router\)](#)

```
{master:member0-re0}
```

```
user@host> request virtual-chassis routing-engine master switch
Do you want to continue ? [yes,no] (no)
```

[request virtual-chassis routing-engine master switch \(Error When Configuration Not Synchronized\)](#)

```
{master:member0-re0}
```

```
user@host> request virtual-chassis routing-engine master switch
Error: mastership switch request NOT honored, backup not ready
```

[request virtual-chassis routing-engine master switch \(Error When Run from Backup Router\)](#)

```
{backup:member1-re0}
```

```
user@host> request virtual-chassis routing-engine master switch
error: Virtual Chassis member is not the protocol master
```

[request virtual-chassis routing-engine master switch check \(Ready for GRES\)](#)

```
{master:member0-re0}
```

```
user@host> request virtual-chassis routing-engine master switch check
Switchover Ready
```

request virtual-chassis routing-engine master switch check (Not Ready for GRES)

```
{master:member0-re0}
```

```
user@host> request virtual-chassis routing-engine master switch check
```

```
error: chassisd Not ready for mastership switch, try after 217 secs.  
mastership switch request NOT honored, backup not ready
```

request virtual-chassis vc-port

Syntax request virtual-chassis vc-port [set | delete]
 <fpc-slot *fpc-slot*>
 pic-slot *pic-slot*
 port *port-number*
 <member *member-id*>

Release Information Command introduced in Junos OS Release 9.0 for EX Series switches.
 Option **fpc-slot** introduced in Junos OS Release 10.4 for EX Series switches.
 Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series.
 Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).

Description Enable or disable an optical port as a Virtual Chassis port (VCP).

If you omit **member *member-id***, 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.

You might experience a temporary traffic disruption immediately after creating or deleting a user-configured VCP in an EX8200 Virtual Chassis.



NOTE: Some EX Series switches have ports that are configured as VCPs by default. You do not need to explicitly configure those ports as VCPs to use them to interconnect the switch into a Virtual Chassis. See [“Understanding Virtual Chassis Components” on page 16](#) for details.

Options **set**—Set a network port as a VCP to convert a network port into a VCP.

delete—Delete the VCP setting on a port to convert a VCP into a network port.

pic-slot *pic-slot*—Number of the PIC slot for the port on the switch.

port *port-number*—Number of the port that is to be enabled or disabled as a VCP.

member *member-id*—(Optional) Enable or disable the specified VCP on the specified member of the Virtual Chassis or VCF.

Required Privilege Level system-control

Related Documentation

- [request virtual-chassis vc-port \(Dedicated VCP\)](#)
- [show virtual-chassis vc-port on page 166](#)
- [show virtual-chassis vc-port statistics](#)

- *clear virtual-chassis vc-port statistics*
- *Virtual Chassis Port (VCP) Interface Names in an EX8200 Virtual Chassis*
- [Understanding Virtual Chassis Components on page 16](#)

List of Sample Output [request virtual-chassis vc-port set pic-slot 1 port 0 on page 127](#)
[request virtual-chassis vc-port set pic-slot 1 port 1 member 3 on page 127](#)
[request virtual-chassis vc-port delete pic-slot 1 port 1 member 3 on page 127](#)

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.

CHAPTER 13

Operational Commands: Monitoring

- `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`
- `show virtual-chassis vc-port`

show virtual-chassis active-topology

Syntax	<pre>show virtual-chassis active-topology <all-members> <local> <member <i>member-id</i>></pre>
Release Information	<p>Command introduced in Junos OS Release 9.0 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>
Description	Display the active topology of the Virtual Chassis or VCF with next-hop reachability information.
Options	<p>none—Display the active topology of the member switch where the command is issued.</p> <p>all-members—(Optional) Display the active topology of all members of the Virtual Chassis or VCF.</p> <p>local—(Optional) Display the active topology of the switch or external Routing Engine on which this command is entered.</p> <p>member <i>member-id</i>—(Optional) Display the active topology of the specified member of the Virtual Chassis or VCF.</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> <i>Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis</i>
List of Sample Output	<p>show virtual-chassis active-topology (EX4200 Virtual Chassis) on page 131</p> <p>show virtual-chassis active-topology (EX8200 Virtual Chassis) on page 131</p> <p>show virtual-chassis active-topology (Virtual Chassis Fabric) on page 132</p>
Output Fields	Table 10 on page 130 lists the output fields for the show virtual-chassis active-topology command. Output fields are listed in the approximate order in which they appear.

Table 10: show virtual-chassis active-topology Output Fields

Field Name	Field Description
Destination ID	Specifies the member ID of the destination.
Next-hop	<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                4(vcp-255/0/2.32768)  5(vcp-255/0/3.32768)
6(vcp-255/0/1.32768)
1                4(vcp-255/0/2.32768)  5(vcp-255/0/3.32768)
6(vcp-255/0/1.32768)
3                4(vcp-255/0/2.32768)  5(vcp-255/0/3.32768)
6(vcp-255/0/1.32768)
4                4(vcp-255/0/2.32768)
5                5(vcp-255/0/3.32768)
6                6(vcp-255/0/1.32768)
fpc3:
-----
Destination ID    Next-hop
0                4(vcp-255/0/2.32768)  5(vcp-255/0/3.32768)
6(vcp-255/0/1.32768)
1                4(vcp-255/0/2.32768)  5(vcp-255/0/3.32768)
6(vcp-255/0/1.32768)
2                4(vcp-255/0/2.32768)  5(vcp-255/0/3.32768)
6(vcp-255/0/1.32768)
4                4(vcp-255/0/2.32768)
5                5(vcp-255/0/3.32768)
6                6(vcp-255/0/1.32768)
fpc4:
-----
Destination ID    Next-hop
0                0(vcp-255/0/48.32768)
1                1(vcp-255/0/49.32768)
2                2(vcp-255/0/50.32768)
3                3(vcp-255/0/51.32768)
5                3(vcp-255/0/51.32768)  2(vcp-255/0/50.32768)
0(vcp-255/0/48.32768)  1(vcp-255/0/49.32768)
6                3(vcp-255/0/51.32768)  2(vcp-255/0/50.32768)
0(vcp-255/0/48.32768)  1(vcp-255/0/49.32768)

```

fpc5:

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

fpc6:

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

show virtual-chassis device-topology

Syntax show virtual-chassis device-topology
 <all-members>
 <local>
 <member *member-id*>

Release Information Command introduced in Junos OS Release 10.4 for EX Series switches.
 Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series.
 Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).

Description Display the device topology—the member and system IDs, the VCP numbers, and device status—for all hardware devices in the Virtual Chassis or VCF.

Options **none**—Display the device topology for all members of the Virtual Chassis or VCF.

all-members—(Optional) Display the device topology for all members of the Virtual Chassis or VCF.

local—(Optional) Display the device topology for the switch or external Routing Engine on which this command is entered.

member *member-id*—(Optional) Display the device topology for the specified member of the Virtual Chassis or VCF.

Required Privilege Level clear

Related Documentation

- *Understanding Virtual Chassis Port Link Aggregation*
- *Understanding EX8200 Virtual Chassis Topologies*

Output Fields [Table 11 on page 135](#) lists the output fields for the **show virtual-chassis device-topology** command. Output fields are listed in the approximate order in which they appear.

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

Field Name	Field Description
Member	Assigned member ID.
Device	Assigned device ID. For an EX8200 Virtual Chassis, the member ID and the device ID are always identical.

Table 11: show virtual-chassis device-topology Output Fields (continued)

Field Name	Field Description
Status	The status of the device within the Virtual Chassis or VCF. Outputs include: <ul style="list-style-type: none"> Prsnt—Device is currently connected to and participating in the Virtual Chassis or VCF. NotPrsnt—Device is assigned but is not currently connected.
System ID	System ID of the device. The system ID of the device is the device's MAC address.
Member (Neighbor List)	Assigned member ID of the neighbor device.
Device (Neighbor List)	Assigned device ID of the neighbor device. For an EX8200 Virtual Chassis, the member ID and the device ID are always identical.
Interface (Neighbor List)	The interface connecting the device to the neighbor.

Sample Output

show virtual-chassis device-topology

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

```
member0:
```

```
-----
                        Neighbor List
Member  Device  Status  System ID      Member  Device  Interface
  0      0      Prsnt   0021.59f7.d000    8      8      vcp-0/0
                        1      1      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:
```

```
-----
                        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
```


9	9	Prsnt	0000.73e9.9a57	1	1	vcp-1/2
				8	8	vcp-1/0
				1	1	vcp-1/1
member8:						

Neighbor List						
Member	Device	Status	System ID	Member	Device	Interface
0	0	Prsnt	0021.59f7.d000	8	8	vcp-0/0
				1	1	vcp-4/0/1
1	1	Prsnt	0026.888d.6800	8	8	vcp-0/0
				9	9	vcp-0/1
				0	0	vcp-3/0/4
8	8	Prsnt	0000.4a75.9b7c	9	9	vcp-1/0
				0	0	vcp-1/1
				1	1	vcp-1/2
9	9	Prsnt	0000.73e9.9a57	8	8	vcp-1/0
				1	1	vcp-1/1
member9:						

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

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

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

fpc0:						

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

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
fpc1:						

Member	Device	Status	System ID	Neighbor List		
				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:						

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

6	6	Prsnt	100e.7eb6.3b00	3	3	vcp-255/0/3
				2	2	vcp-255/0/2
				0	0	vcp-255/0/0
				1	1	vcp-255/0/1

fpc3:

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

fpc4:

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

```

0      0      vcp-255/0/0
1      1      vcp-255/0/1

```

fpc5:

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

fpc6:

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

show virtual-chassis protocol adjacency

Syntax	<pre>show virtual-chassis protocol adjacency <brief detail extensive> <all-members> <local> <member <i>member-id</i>> <system-id></pre>
Release Information	<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>
Description	Display the Virtual Chassis Control Protocol (VCCP) adjacency statistics in the Virtual Chassis or VCF for all hardware devices.
Options	<p>none—Display VCCP adjacency statistics in brief form for all members of the Virtual Chassis or VCF.</p> <p>brief detail extensive—(Optional) Display the specified level of output. Using the brief option is equivalent to entering the command with no options (the default). The detail and extensive options provide identical displays.</p> <p>all-members—(Optional) Display VCCP adjacency statistics in brief form for all members of the Virtual Chassis or VCF.</p> <p>local—(Optional) Display VCCP adjacency statistics for the switch or external Routing Engine on which this command is entered.</p> <p>member <i>member-id</i>—(Optional) Display VCCP adjacency statistics for the specified member of the Virtual Chassis or VCF.</p> <p>system-id—(Optional) Display VCCP adjacency statistics for the specified member of the Virtual Chassis or VCF.</p>
Required Privilege Level	clear
Related Documentation	<ul style="list-style-type: none"> • <i>Understanding Virtual Chassis Port Link Aggregation</i> • <i>Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis</i>
List of Sample Output	<p>show virtual-chassis protocol adjacency on page 143</p> <p>show virtual-chassis protocol adjacency detail on page 144</p>
Output Fields	Table 12 on page 143 lists the output fields for the show virtual-chassis protocol adjacency command. Output fields are listed in the approximate order in which they appear.

Table 12: show virtual-chassis protocol adjacency Output Fields

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

Sample Output

show virtual-chassis protocol adjacency

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

```
member0:
```

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

```
member1:
```

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

```
member8:
```

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

```
member9:
```

```

-----
Interface          System          State          Hold (secs)
vcp-1/0.32768      0000.4a75.9b7c Up              58
vcp-1/1.32768      0026.888d.6800 Up              59

```

show virtual-chassis protocol adjacency detail

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

```
member0:
```

```

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

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

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

```

```
member1:
```

```

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

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

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

```

```
member8:
```

```

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

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

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

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

```

```
member9:
```



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

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

show virtual-chassis protocol database

Syntax	<pre>show virtual-chassis protocol database <brief detail extensive> <all-members> <local> <member <i>member-id</i>></pre>
Release Information	<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>
Description	Display the Virtual Chassis Control Protocol (VCCP) database statistics for all hardware devices within the Virtual Chassis or VCF.
Options	<p>none—Display VCCP database statistics in brief form for all members of the Virtual Chassis or VCF.</p> <p>brief detail extensive—(Optional) Display the specified level of output. Using the brief option is equivalent to entering the command with no options (the default). The detail option provides more output than the brief option. The extensive option provides all output and is most useful for customer support personnel.</p> <p>all-members—(Optional) Display VCCP database statistics in brief form for all members of the Virtual Chassis or VCF.</p> <p>local—(Optional) Display VCCP database statistics for the switch or external Routing Engine on which this command is entered.</p> <p>member <i>member-id</i>—(Optional) Display VCCP database statistics for the specified member of the Virtual Chassis or VCF.</p>
Required Privilege Level	clear
Related Documentation	<ul style="list-style-type: none"> • Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis • Understanding Virtual Chassis Components on page 16
List of Sample Output	<p>show virtual-chassis protocol database on page 147</p> <p>show virtual-chassis protocol database detail on page 148</p>
Output Fields	Table 13 on page 147 lists the output fields for the show virtual-chassis protocol database command. Output fields are listed in the approximate order in which they appear.

Table 13: show virtual-chassis protocol database Output Fields

Field Name	Field Description	Level of Output
LSP ID	Link-state protocol (LSP) data unit identifier.	All levels
Sequence	Sequence number of the LSP.	All levels
Checksum	Checksum value of the LSP.	All levels
Lifetime	Remaining lifetime of the LSP, in seconds.	All levels
Neighbor	MAC address of the neighbor on the advertising system.	detail
Interface	Virtual Chassis port (VCP) interface name.	detail
Metric	Metric of the prefix or neighbor.	detail

The **extensive** output was omitted from this list. The **extensive** output is useful for customer support personnel only.

Sample Output

show virtual-chassis protocol database

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

```
member0:
```

```
-----
LSP ID          Sequence Checksum Lifetime
0000.4a75.9b7c.00-00  0x1dd80  0xc2e3   116
0000.73e9.9a57.00-00  0xf361   0x27e8   113
0021.59f7.d000.00-00  0x16882  0x3993   118
0026.888d.6800.00-00  0x1691f  0x82b7   116
  4 LSPs
```

```
member1:
```

```
-----
LSP ID          Sequence Checksum Lifetime
0000.4a75.9b7c.00-00  0x1dd80  0xc2e3   116
0000.73e9.9a57.00-00  0xf361   0x27e8   114
0021.59f7.d000.00-00  0x16883  0x289    116
0026.888d.6800.00-00  0x1691f  0x82b7   118
  4 LSPs
```

```
member8:
```

```
-----
LSP ID          Sequence Checksum Lifetime
0000.4a75.9b7c.00-00  0x1dd80  0xc2e3   118
0000.73e9.9a57.00-00  0xf361   0x27e8   114
0021.59f7.d000.00-00  0x16883  0x289    116
0026.888d.6800.00-00  0x16920  0xa335   116
  4 LSPs
```

```
member9:
```

LSP ID	Sequence	Checksum	Lifetime
0000.4a75.9b7c.00-00	0x1dd80	0xc2e3	116
0000.73e9.9a57.00-00	0xf361	0x27e8	116
0021.59f7.d000.00-00	0x16883	0x289	114
0026.888d.6800.00-00	0x16920	0xa335	116
4 LSPs			

show virtual-chassis protocol database detail

user@switch> show virtual-chassis protocol database detail

member0:

```
-----
0000.4a75.9b7c.00-00 Sequence: 0x1ddbc, Checksum: 0x3111, Lifetime: 115 secs
Neighbor: 0000.73e9.9a57.00 Interface: vcp-1/0.32768 Metric: 150
Neighbor: 0021.59f7.d000.00 Interface: vcp-1/1.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-1/2.32768 Metric: 150

0000.73e9.9a57.00-00 Sequence: 0xf381, Checksum: 0xe065, Lifetime: 114 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-1/0.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-1/1.32768 Metric: 150

0021.59f7.d000.00-00 Sequence: 0x168af, Checksum: 0x8b0b, Lifetime: 118 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-0/0.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-4/0/1.32768 Metric: 15

0026.888d.6800.00-00 Sequence: 0x1694e, Checksum: 0xca97, Lifetime: 115 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-0/0.32768 Metric: 150
Neighbor: 0000.73e9.9a57.00 Interface: vcp-0/1.32768 Metric: 150
Neighbor: 0021.59f7.d000.00 Interface: vcp-3/0/4.32768 Metric: 15
```

member1:

```
-----
0000.4a75.9b7c.00-00 Sequence: 0x1ddbc, Checksum: 0x3111, Lifetime: 115 secs
Neighbor: 0000.73e9.9a57.00 Interface: vcp-1/0.32768 Metric: 150
Neighbor: 0021.59f7.d000.00 Interface: vcp-1/1.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-1/2.32768 Metric: 150

0000.73e9.9a57.00-00 Sequence: 0xf381, Checksum: 0xe065, Lifetime: 116 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-1/0.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-1/1.32768 Metric: 150

0021.59f7.d000.00-00 Sequence: 0x168af, Checksum: 0x8b0b, Lifetime: 116 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-0/0.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-4/0/1.32768 Metric: 15

0026.888d.6800.00-00 Sequence: 0x1694e, Checksum: 0xca97, Lifetime: 117 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-0/0.32768 Metric: 150
Neighbor: 0000.73e9.9a57.00 Interface: vcp-0/1.32768 Metric: 150
Neighbor: 0021.59f7.d000.00 Interface: vcp-3/0/4.32768 Metric: 15
```

member8:

```
-----
0000.4a75.9b7c.00-00 Sequence: 0x1ddbd, Checksum: 0xfd83, Lifetime: 118 secs
Neighbor: 0000.73e9.9a57.00 Interface: vcp-1/0.32768 Metric: 150
Neighbor: 0021.59f7.d000.00 Interface: vcp-1/1.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-1/2.32768 Metric: 150
```

```

0000.73e9.9a57.00-00 Sequence: 0xf381, Checksum: 0xe065, Lifetime: 115 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-1/0.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-1/1.32768 Metric: 150

0021.59f7.d000.00-00 Sequence: 0x168af, Checksum: 0x8b0b, Lifetime: 116 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-0/0.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-4/0/1.32768 Metric: 15

0026.888d.6800.00-00 Sequence: 0x1694e, Checksum: 0xca97, Lifetime: 115 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-0/0.32768 Metric: 150
Neighbor: 0000.73e9.9a57.00 Interface: vcp-0/1.32768 Metric: 150
Neighbor: 0021.59f7.d000.00 Interface: vcp-3/0/4.32768 Metric: 15

member9:
-----

0000.4a75.9b7c.00-00 Sequence: 0x1ddb, Checksum: 0xfd83, Lifetime: 116 secs
Neighbor: 0000.73e9.9a57.00 Interface: vcp-1/0.32768 Metric: 150
Neighbor: 0021.59f7.d000.00 Interface: vcp-1/1.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-1/2.32768 Metric: 150

0000.73e9.9a57.00-00 Sequence: 0xf381, Checksum: 0xe065, Lifetime: 117 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-1/0.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-1/1.32768 Metric: 150

0021.59f7.d000.00-00 Sequence: 0x168af, Checksum: 0x8b0b, Lifetime: 113 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-0/0.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-4/0/1.32768 Metric: 15

0026.888d.6800.00-00 Sequence: 0x1694f, Checksum: 0xa61a, Lifetime: 116 secs
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-0/0.32768 Metric: 150
Neighbor: 0000.73e9.9a57.00 Interface: vcp-0/1.32768 Metric: 150
Neighbor: 0021.59f7.d000.00 Interface: vcp-3/0/4.32768 Metric: 15

```

show virtual-chassis protocol interface

Syntax	<pre>show virtual-chassis protocol interface <brief detail> <all-members> <interface-name> <local> <member member-id></pre>
Release Information	<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>
Description	Display information about Virtual Chassis Control Protocol (VCCP) statistics for VCCP-enabled interfaces within the Virtual Chassis or VCF.
Options	<p>none—Display the VCCP interface statistics in brief form for all members of the Virtual Chassis or VCF.</p> <p>brief detail —(Optional) Display the specified level of output. Using the brief option is equivalent to entering the command with no options (the default). The detail option provides more output than the brief option.</p> <p>all-members—(Optional) Display VCCP interface statistics for all members of the Virtual Chassis or VCF.</p> <p>interface-name—(Optional) Display VCCP interface statistics for the specified interface.</p> <p>local—(Optional) Display VCCP interface statistics for the switch or external Routing Engine on which this command is entered.</p> <p>member member-id—(Optional) Display VCCP interface statistics for the specified member of the Virtual Chassis or VCF.</p>
Required Privilege Level	clear
Related Documentation	<ul style="list-style-type: none"> • <i>Understanding EX Series Virtual Chassis</i> • <i>Understanding QFX Series Virtual Chassis</i> • <i>Understanding Virtual Chassis Ports in an EX8200 Virtual Chassis</i> • <i>Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis</i>
List of Sample Output	show virtual-chassis protocol interface on page 151
Output Fields	Table 14 on page 151 lists the output fields for the show virtual-chassis protocol interface command. Output fields are listed in the approximate order in which they appear.

Table 14: show virtual-chassis protocol interface Output Fields

Field Name	Field Description	Level of Output
Interface	Name of the VCP.	All levels
State	State of the link. Outputs include: <ul style="list-style-type: none"> • Up—The link is up. • Down—The link is down. 	All levels
Metric	Metric of the prefix or neighbor.	All levels

Sample Output

show virtual-chassis protocol interface

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

```
member0:
```

```
-----
IS-IS interface database:
```

Interface	State	Metric
vcp-0/0.32768	Up	150
vcp-0/1.32768	Up	150
vcp-4/0/1.32768	Up	15
vcp-4/0/7.32768	Down	15

```
member1:
```

```
-----
IS-IS interface database:
```

Interface	State	Metric
vcp-0/0.32768	Up	150
vcp-0/1.32768	Up	150
vcp-3/0/4.32768	Up	15

```
member8:
```

```
-----
IS-IS interface database:
```

Interface	State	Metric
vcp-0/0.32768	Down	150
vcp-1/0.32768	Up	150
vcp-1/1.32768	Up	150
vcp-1/2.32768	Up	150
vcp-1/3.32768	Down	150
vcp-2/0.32768	Up	150
vcp-2/1.32768	Down	150
vcp-2/2.32768	Down	150
vcp-2/3.32768	Down	150

```
member9:
```

```
-----
IS-IS interface database:
```

Interface	State	Metric
vcp-0/0.32768	Down	150
vcp-1/0.32768	Up	150
vcp-1/1.32768	Up	150

vcp-1/2.32768	Down	150
vcp-1/3.32768	Down	150

show virtual-chassis protocol route

Syntax	<pre>show virtual-chassis protocol route <all-members> <destination-id> <local> <member member-id></pre>
Release Information	<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>
Description	Display the unicast and multicast Virtual Chassis Control Protocol (VCCP) routing tables within the Virtual Chassis or VCF.
Options	<p>none—Display the unicast and multicast routing tables for all members of the Virtual Chassis.</p> <p>all-members—(Optional) Display the unicast and multicast routing tables for all members of the Virtual Chassis or VCF.</p> <p>destination-id—(Optional) Display the unicast and multicast routing tables to the specified destination member ID for each member of the Virtual Chassis or VCF.</p> <p>local—(Optional) Display the unicast and multicast routing tables on the device where this command is entered.</p> <p>member member-id—(Optional) Display the unicast and multicast routing tables for the specified member of the Virtual Chassis or VCF.</p>
Required Privilege Level	clear
Related Documentation	<ul style="list-style-type: none"> • <i>Understanding EX Series Virtual Chassis</i> • <i>Understanding QFX Series Virtual Chassis</i> • <i>Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis</i>
List of Sample Output	show virtual-chassis protocol route on page 154
Output Fields	Table 15 on page 153 lists the output fields for the show virtual-chassis protocol route command. Output fields are listed in the approximate order in which they appear.

Table 15: show virtual-chassis protocol route Output Fields

Field Name	Field Description
Dev	MAC address of the member storing the VCCP routing table.

Table 15: show virtual-chassis protocol route Output Fields (continued)

Field Name	Field Description
Version	Version of the shortest-path-first algorithm that generated the routing table.
System ID	MAC address of the device.
Version	Version of the shortest-path-first (SPF) algorithm that generated the route.
Metric	The metric number to get to that device.
Interface	Name of the Virtual Chassis port (VCP) interface connecting the devices.
Via	MAC address of the next-hop device, if applicable.

Sample Output

show virtual-chassis protocol route

```

user@switch> show virtual-chassis protocol route
member0:
-----
Dev 0021.59f7.d000 ucast routing table          Current version: 21
-----
System ID      Version  Metric Interface  Via
0000.4a75.9b7c    21      150 vcp-0/1.32768 0000.4a75.9b7c
0000.73e9.9a57    21      165 vcp-4/0/1.32768 0026.888d.6800
0021.59f7.d000    21        0
0026.888d.6800    21      15 vcp-4/0/1.32768 0026.888d.6800

Dev 0021.59f7.d000 mcast routing table          Current version: 21
-----
System ID      Version  Metric Interface  Via
0000.4a75.9b7c    21
0000.73e9.9a57    21
0021.59f7.d000    21      vcp-4/0/1.32768
                                vcp-0/1.32768
0026.888d.6800    21

member1:
-----
Dev 0026.888d.6800 ucast routing table          Current version: 25
-----
System ID      Version  Metric Interface  Via
0000.4a75.9b7c    25      150 vcp-0/0.32768 0000.4a75.9b7c
0000.73e9.9a57    25      150 vcp-0/1.32768 0000.73e9.9a57
0021.59f7.d000    25      15 vcp-3/0/4.32768 0021.59f7.d000
0026.888d.6800    25        0

Dev 0026.888d.6800 mcast routing table          Current version: 25
-----
System ID      Version  Metric Interface  Via
0000.4a75.9b7c    25

```

```

0000.73e9.9a57      25      vcp-3/0/4.32768
0021.59f7.d000      25      vcp-0/1.32768
0026.888d.6800      25      vcp-3/0/4.32768
                        vcp-0/0.32768
                        vcp-0/1.32768

```

```
member8:
```

```
-----
Dev 0000.4a75.9b7c ucast routing table          Current version: 39
-----
```

System ID	Version	Metric	Interface	Via
0000.4a75.9b7c	39	0		
0000.73e9.9a57	39	150	vcp-1/0.32768	0000.73e9.9a57
0021.59f7.d000	39	150	vcp-2/0.32768	0021.59f7.d000
0026.888d.6800	39	150	vcp-1/2.32768	0026.888d.6800

```
Dev 0000.4a75.9b7c mcast routing table          Current version: 39
-----
```

System ID	Version	Metric	Interface	Via
0000.4a75.9b7c	39		vcp-1/0.32768 vcp-2/0.32768 vcp-1/2.32768	
0000.73e9.9a57	39			
0021.59f7.d000	39			
0026.888d.6800	39			

```
member9:
```

```
-----
Dev 0000.73e9.9a57 ucast routing table          Current version: 31
-----
```

System ID	Version	Metric	Interface	Via
0000.4a75.9b7c	31	150	vcp-1/0.32768	0000.4a75.9b7c
0000.73e9.9a57	31	0		
0021.59f7.d000	31	165	vcp-1/1.32768	0026.888d.6800
0026.888d.6800	31	150	vcp-1/1.32768	0026.888d.6800

```
Dev 0000.73e9.9a57 mcast routing table          Current version: 31
-----
```

System ID	Version	Metric	Interface	Via
0000.4a75.9b7c	31			
0000.73e9.9a57	31		vcp-1/0.32768 vcp-1/1.32768	
0021.59f7.d000	31			
0026.888d.6800	31			

show virtual-chassis protocol statistics

Syntax	<pre>show virtual-chassis protocol statistics <all-members> <interface-name> <local> <member member-id></pre>
Release Information	<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>
Description	Display the Virtual Chassis Control Protocol (VCCP) statistics for all hardware devices within the Virtual Chassis or VCF.
Options	<p>none—Display VCCP statistics for all members of the Virtual Chassis or VCF.</p> <p>all-members—(Optional) Display VCCP statistics for all members of the Virtual Chassis or VCF.</p> <p>interface-name—(Optional) Display VCCP statistics for the specified interface.</p> <p>local—(Optional) Display VCCP statistics for the switch or external Routing Engine on which this command is entered.</p> <p>member member-id—(Optional) Display VCCP statistics for the specified member of the Virtual Chassis or VCF.</p>
Required Privilege Level	clear
Related Documentation	<ul style="list-style-type: none"> • <i>Understanding EX Series Virtual Chassis</i> • <i>Understanding QFX Series Virtual Chassis</i> • <i>Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis</i>
List of Sample Output	show virtual-chassis protocol statistics on page 157
Output Fields	Table 16 on page 156 lists the output fields for the show virtual-chassis protocol interface command. Output fields are listed in the approximate order in which they appear.

Table 16: show virtual-chassis protocol statistics Output Fields

Field Name	Field Description
PDU type	Protocol data unit type.
Received	Number of PDUs received since VCCP started or since the statistics were set to zero.

Table 16: show virtual-chassis protocol statistics Output Fields (continued)

Field Name	Field Description
Processed	Number of PDUs received minus the number of PDUs dropped.
Drops	Number of PDUs dropped.
Sent	Number of PDUs transmitted since VCCP started or since the statistics were set to zero.
Rexmit	Number of PDUs retransmitted since VCCP started or since the statistics were set to zero.
Total Packets Received	Number of PDUs received since VCCP started or since the statistics were set to zero.
Total Packets Sent	Number of PDUs sent since VCCP started or since the statistics were set to zero.
LSP queue length	Number of link-state PDUs waiting in the queue for processing. This value is almost always 0.
SPF runs	Number of shortest-path-first (SPF) calculations that have been performed.
Fragments Rebuilt	Number of link-state PDU fragments that the local system has computed.
LSP Regenerations	Number of link-state PDUs that have been regenerated. A link-state PDU is regenerated when it is nearing the end of its lifetime and it has not changed.
Purges initiated	Number of purges that the system initiated. A purge is initiated if the software determines that a link-state PDU must be removed from the network.

Sample Output

show virtual-chassis protocol statistics

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

```
member0:
```

```
-----
```

```
IS-IS statistics for 0021.59f7.d000:
```

PDU type	Received	Processed	Drops	Sent	Rexmit
LSP	8166	8166	0	4551	0
HELLO	1659	1659	0	1693	0
CSNP	2	2	0	3	0
PSNP	1909	1909	0	2293	0
Unknown	0	0	0	0	0
Totals	11736	11736	0	8540	0

```
Total packets received: 11736 Sent: 8540
```

```
LSP queue length: 0 Drops: 0
SPF runs: 9
Fragments rebuilt: 1640
LSP regenerations: 1
Purges initiated: 0
```

member1:

IS-IS statistics for 0026.888d.6800:

PDU type	Received	Processed	Drops	Sent	Rexmit
LSP	10909	10909	0	12088	0
HELLO	1877	1877	0	2251	0
CSNP	3	3	0	3	0
PSNP	3846	3846	0	3732	0
Unknown	0	0	0	0	0
Totals	16635	16635	0	18074	0

Total packets received: 16635 Sent: 18074

LSP queue length: 0 Drops: 0

SPF runs: 13

Fragments rebuilt: 1871

LSP regenerations: 2

Purges initiated: 0

member8:

IS-IS statistics for 0000.4a75.9b7c:

PDU type	Received	Processed	Drops	Sent	Rexmit
LSP	7935	7935	0	14865	0
HELLO	2695	2695	0	7124	0
CSNP	4	4	0	4	0
PSNP	4398	4398	0	3666	0
Unknown	0	0	0	0	0
Totals	15032	15032	0	25659	0

Total packets received: 15032 Sent: 25659

LSP queue length: 0 Drops: 0

SPF runs: 26

Fragments rebuilt: 2666

LSP regenerations: 4

Purges initiated: 0

member9:

IS-IS statistics for 0000.73e9.9a57:

PDU type	Received	Processed	Drops	Sent	Rexmit
LSP	10800	10800	0	6327	0
HELLO	1492	1492	0	2356	0
CSNP	2	2	0	2	0
PSNP	2683	2683	0	3149	0
Unknown	0	0	0	0	0
Totals	14977	14977	0	11834	0

Total packets received: 14977 Sent: 11834

LSP queue length: 0 Drops: 0

SPF runs: 19

Fragments rebuilt: 1510

LSP regenerations: 6

Purges initiated: 0

show virtual-chassis

Syntax	show virtual-chassis <status>
Release Information	<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). Fabric ID, Fabric Mode, and Route Mode output fields introduced in Junos OS Release 13.2X51-D20.</p> <p>Alias-Name output field introduced in Junos OS Release 14.1X53-D10.</p>
Description	Display information about all members of the Virtual Chassis or VCF.
Options	<p>none—Display information about all Virtual Chassis or VCF member devices.</p> <p>status—Same output as for show virtual-chassis without any options.</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • show virtual-chassis active-topology on page 130 • show virtual-chassis protocol adjacency on page 142 • <i>show virtual-chassis vc-path</i> • <i>Understanding Mixed EX Series and QFX Series Virtual Chassis</i> • <i>Understanding Mixed Virtual Chassis Fabric</i> • <i>Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis</i>
List of Sample Output	<p>show virtual-chassis (EX2300 multigigabit model—EX2300-24MP and EX2300-48MP—Virtual Chassis) on page 162</p> <p>show virtual-chassis (EX2300 and EX2300 multigigabit model (EX2300-24MP, EX2300-48MP) Virtual Chassis) on page 163</p> <p>show virtual-chassis (EX4200 Virtual Chassis) on page 163</p> <p>show virtual-chassis (Mixed EX4300 multigigabit model—EX4300-48MP—Virtual Chassis) on page 163</p> <p>show virtual-chassis (EX8200 Virtual Chassis) on page 164</p> <p>show virtual-chassis (QFX5110 Virtual Chassis) on page 164</p> <p>show virtual-chassis (QFX5200 Virtual Chassis) on page 165</p> <p>show virtual-chassis (QFX5100 Virtual Chassis Fabric) on page 165</p>
Output Fields	Table 17 on page 161 lists the output fields for the show virtual-chassis command. Output fields are listed in the approximate order in which they appear.

Table 17: show virtual-chassis Output Fields

Field Name	Field Description
Fabric ID	(VCF only) Assigned ID used to identify the VCF.
Fabric Mode	(VCF only) Mode of the VCF: Enabled, Disabled, or Mixed.
Preprovisioned Virtual Chassis or Preprovisioned Virtual Chassis Fabric	Virtual Chassis or VCF is configured using preprovisioning.
Virtual Chassis ID	Assigned ID that applies to the entire Virtual Chassis or VCF.
Virtual Chassis Mode	<p>Mode of the Virtual Chassis or VCF. This field indicates support for the Virtual Chassis feature and, if a Virtual Chassis is configured, if it is a mixed or homogenous Virtual Chassis. Values can be:</p> <ul style="list-style-type: none"> • Enabled—The platform supports the Virtual Chassis feature. If a Virtual Chassis is currently configured, this is a homogenous Virtual Chassis (all members are the same type of switch). • Disabled—The switch does not support the Virtual Chassis feature. <p>NOTE: Switches that support the Virtual Chassis feature do not display this value. Even if a Virtual Chassis is not currently configured, those switches display Enabled in this field.</p> <ul style="list-style-type: none"> • Mixed—The platform supports the Virtual Chassis feature, and is configured as a mixed mode Virtual Chassis (members consist of more than one type of switch).
Member ID	<p>Assigned member ID and FPC:</p> <ul style="list-style-type: none"> • On all EX Series Virtual Chassis except EX8200 Virtual Chassis, and on a VCF, the FPC number refers to the member ID assigned to the switch. • On EX8200 Virtual Chassis, member IDs are numbered 0 through 9. The FPC number indicates the slot number of the line card within the Virtual Chassis. The FPC number on member 0 is always 0 through 15. The FPC number on member 1 is always 16 through 31. The FPC number on member 2 is always 32 through 47; and so on for the members.
Status	<p>For a nonprovisioned configuration:</p> <ul style="list-style-type: none"> • Prsnt for a member that is currently connected to the Virtual Chassis or VCF configuration. • NotPrsnt for a member ID that has been assigned but is not currently connected. <p>For a preprovisioned configuration:</p> <ul style="list-style-type: none"> • Prsnt for a member that is specified in the preprovisioned configuration file and is currently connected to the Virtual Chassis or VCF. • Unprvsnd for a member that is interconnected with the Virtual Chassis or VCF configuration but is not specified in the preprovisioned configuration file.
Serial No	Serial number of the member device.

Table 17: show virtual-chassis Output Fields (continued)

Field Name	Field Description
Alias-Name	<p>The user-configured alias of the member device.</p> <p>The Alias-Name field appears only if an alias has been configured for at least one device in the Virtual Chassis or VCF. Aliases are configured using the alias-name statement in the <code>[edit virtual-chassis aliases serial-number serial-number]</code> hierarchy.</p>
Model	Model number of the member device.
Mastership Priority or Mstr prio	Mastership priority value of the member device.
Role	<p>Role of the member device: master, backup, or linecard.</p> <p>An asterisk (*) following the Role denotes the member device on which the show virtual-chassis <status> command was issued.</p>
Mixed Mode	<p>Mixed mode configuration status:</p> <ul style="list-style-type: none"> • Y for a member device configured in mixed mode. • N for a member device not configured in mixed mode. • NA for a member device that cannot be configured in mixed mode.
Route Mode	The route mode of the member device: fabric (F) or Virtual Chassis (V).
Location	<p>Location of the member device.</p> <p>If this field is empty, the location field was not set for the device.</p>
Neighbor List ID and Interface	Member ID of the neighbor member to which this member's Virtual Chassis port (VCP) is connected, and the VCP interface name.

Sample Output

show virtual-chassis (EX2300 multigigabit model—EX2300-24MP and EX2300-48MP—Virtual Chassis)

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

```
Virtual Chassis ID: cdef.789a.bc12
Virtual Chassis Mode: Enabled
```

Member ID	Status	Serial No	Model	Mstr prio	Role	Mixed Mode	Route Mode	Neighbor List ID	Interface
0 (FPC 0)	Prsnt	XY1222340011	ex2300-24mp	128	Backup	N	VC	3	vcp-255/1/0
								1	vcp-255/1/2
1 (FPC 1)	Prsnt	XY1222340021	ex2300-48mp	128	Master*	N	VC	2	vcp-255/1/4
								0	vcp-255/1/1
2 (FPC 2)	Prsnt	XY1222340031	ex2300-24mp	128	Linecard	N	VC	3	vcp-255/1/0
								1	vcp-255/1/1
3 (FPC 3)	Prsnt	XY1222340041	ex2300-48mp	128	Linecard	N	VC	0	vcp-255/1/4
								2	vcp-255/1/5

show virtual-chassis (EX2300 and EX2300 multigigabit model (EX2300-24MP, EX2300-48MP) Virtual Chassis)

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

Preprovisioned Virtual Chassis

Virtual Chassis ID: 9876.5432.abcd

Virtual Chassis Mode: Enabled

Member ID	Status	Serial No	Model	Mstr prio	Role	Mixed Mode	Route Mode	Neighbor ID	List Interface
0 (FPC 0)	Prsnt	XP0123450001	ex2300-24mp	129	Backup	N	VC	1	vcp-255/1/3
								3	vcp-255/1/0
1 (FPC 1)	Prsnt	JJ0123450001	ex2300-24p	129	Master*	N	VC	2	vcp-255/1/0
								0	vcp-255/1/1
2 (FPC 2)	Prsnt	JJ0123450002	ex2300-24p	0	Linecard	N	VC	3	vcp-255/1/2
								1	vcp-255/1/3
3 (FPC 3)	Prsnt	XP0123450002	ex2300-48mp	0	Linecard	N	VC	2	vcp-255/1/5
								0	vcp-255/1/0

show virtual-chassis (EX4200 Virtual Chassis)

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

Virtual Chassis ID: 00ab.cdef.1234

Virtual Chassis Mode: Enabled

Member ID	Status	Serial No	Model	Mastership priority	Role	Mixed Mode	Neighbor ID	List 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 (Mixed EX4300 multigigabit model—EX4300-48MP—Virtual Chassis)

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

Preprovisioned Virtual Chassis

Virtual Chassis ID: abcd.ef00.1234

Virtual Chassis Mode: Mixed

Member ID	Status	Serial No	Model	Mstr prio	Role	Mixed Mode	Route Mode	Neighbor ID	List Interface
0 (FPC 0)	Prsnt	XR3617480028	ex4300-48mp	129	Master*	Y	VC	1	vcp-255/1/0

1 (FPC 1)	Prsnt	XR3617480029	ex4300-48mp	129	Backup	Y	VC	3	vcp-255/1/3
								0	vcp-255/1/1
2 (FPC 2)	Prsnt	XR3617480001	ex4300-48mp	0	Linecard	Y	VC	2	vcp-255/1/3
								3	vcp-255/1/1
3 (FPC 3)	Prsnt	PE3715471074	ex4300-48t	0	Linecard	Y	VC	1	vcp-255/1/3
								0	vcp-255/1/0
								2	vcp-255/1/2

show virtual-chassis (EX8200 Virtual Chassis)

user@external-routing-engine> show virtual-chassis

Virtual Chassis ID: cdc1.1212.efef

Virtual Chassis Mode: Enabled

Member ID	Status	Serial No	Model	Mastership 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 (QFX5110 Virtual Chassis)

user@switch> show virtual-chassis

Preprovisioned Virtual Chassis

Virtual Chassis ID: abab.1212.cdc1

Virtual Chassis Mode: Enabled

Member ID	Status	Serial No	Model	Mstr prio	Role	Mixed Route Mode Mode	Neighbor List ID Interface
0 (FPC 0)	Prsnt	WT3116360038	qfx5110-32q	129	Backup	Y VC	9 vcp-255/0/12 1 vcp-255/0/16
1 (FPC 1)	Prsnt	WT3116360041	qfx5110-32q	129	Master*	Y VC	2 vcp-255/0/16 0 vcp-255/0/17
2 (FPC 2)	Prsnt	WT3116360051	qfx5110-32q	0	Linecard	Y VC	3 vcp-255/0/11 1 vcp-255/0/14
3 (FPC 3)	Prsnt	WT3116360055	qfx5110-32q	0	Linecard	Y VC	4 vcp-255/0/14 2 vcp-255/0/31
4 (FPC 4)	Prsnt	WT3116360056	qfx5110-32q	0	Linecard	Y VC	5 vcp-255/0/10 3 vcp-255/0/5
5 (FPC 5)	Prsnt	TB3716340058	qfx5100e-24q-2p	0	Linecard	Y VC	6 vcp-255/0/12 4 vcp-255/0/20
6 (FPC 6)	Prsnt	WS3715500018	qfx5110-48s-4c	0	Linecard	Y VC	7 vcp-255/0/10 5 vcp-255/0/49
7 (FPC 7)	Prsnt	TA3714110007	qfx5100e-48s-6q	0	Linecard	Y VC	6 vcp-255/0/18

```

8 (FPC 8) Prsnt WS3715500022 qfx5110-48s-4c 0 Linecard Y VC 8 vcp-255/0/31
7 vcp-255/0/21
9 vcp-255/0/49
9 (FPC 9) Prsnt WT3116360061 qfx5110-32q 0 Linecard Y VC 8 vcp-255/0/13
0 vcp-255/0/17

```

show virtual-chassis (QFX5200 Virtual Chassis)

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

```
Virtual Chassis ID: abab.1212.cdcd
```

```
Virtual Chassis Mode: Enabled
```

Member ID	Status	Serial No	Model	prio	Role	Mixed Mode	Route Mode	Neighbor List ID	Interface
0 (FPC 0)	Prsnt	XY0123456789	qfx5200-32c-r	128	Master*	N	VC	1	vcp-255/0/19
								2	vcp-255/0/26
1 (FPC 1)	Prsnt	XY0123456780	qfx5200-32c-r	128	Linecard	N	VC	0	vcp-255/0/20
								2	vcp-255/0/16
2 (FPC 2)	Prsnt	YZ0123456789	qfx5200-32c-32q	128	Backup	N	VC	0	vcp-255/0/15
								1	vcp-255/0/18

show virtual-chassis (QFX5100 Virtual Chassis Fabric)

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

```
Preprovisioned Virtual Chassis Fabric
```

```
Fabric ID: 0123.abcd.4567
```

```
Fabric Mode: Enabled
```

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

Syntax	<pre>show virtual-chassis vc-port <all-members> <local> <member <i>member-id</i>></pre>
Release Information	<p>Command introduced in Junos OS Release 9.0 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>
Description	Display the status of the Virtual Chassis ports (VCPs), including both dedicated VCPs and uplink ports configured as VCPs, if present.
Options	<p>none—Display the operational status of all VCPs of the member switch where the command is issued.</p> <p>all-members—(Optional) Display the operational status of all VCPs on all members of the Virtual Chassis or VCF.</p> <p>local—(Optional) Display the operational status of the switch or external Routing Engine on which this command is entered.</p> <p>member <i>member-id</i>—(Optional) Display the operational status of all VCPs for the specified member of the Virtual Chassis or VCF.</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> show virtual-chassis vc-port statistics Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis
List of Sample Output	<p>show virtual-chassis vc-port (EX4200 Virtual Chassis) on page 168</p> <p>show virtual-chassis vc-port (EX8200 Virtual Chassis) on page 168</p> <p>show virtual-chassis vc-port all-members on page 169</p>
Output Fields	<p>Table 18 on page 166 lists the output fields for the show virtual-chassis vc-port command. Output fields are listed in the approximate order in which they appear.</p>

Table 18: show virtual-chassis vc-port Output Fields

Field Name	Field Description
fpcnumber	The FPC number is the same as the member ID.

Table 18: 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"> The dedicated VCPs in an EX4200 or EX4500 Virtual Chassis are vcp-0 and vcp-1. The dedicated VCPs in an EX4550 Virtual Chassis are VCP-1/0, VCP-1/1, VCP-2/0, and VCP-2/1. Optical ports set as VCPs are named 1/0 and 1/1, representing the PIC number and the port number. The native VCP (port 0) on an XRE200 External Routing Engine in an EX8200 Virtual Chassis is named vcp-0. The VCPs on each Virtual Chassis Control Interface (VCCI) module in an XRE200 External Routing Engine are named using the vcp-slot-number/port-number convention; for instance, vcp-1/0. The VCPs on EX8200 member switches are named using the vcp-slot-number/pic-number/interface-number convention; for instance, vcp-3/0/2. A 255 as the first number in your port number indicates that your VCP is part of a Link Aggregation group (LAG) bundle. For instance, a display of vcp-255/1/0 indicates that the dedicated VCP named vcp-1/0 is part of a LAG bundle. A display of vcp-255/1/0 indicates that an uplink port that was previously named xe-0/1/0 is now part of a VCP LAG bundle.
Type	<p>Type of VCP:</p> <ul style="list-style-type: none"> Dedicated—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. Configured—Optical port configured as a VCP. Auto-Configured—Optical port autoconfigured as a VCP. <p>See <i>Setting an Uplink Port on an EX Series or QFX Series Switch as a Virtual Chassis Port</i> or <i>Configuring a QFX Series Virtual Chassis</i> for information about configuring VCPs.</p>
Trunk ID	<p>A positive-number ID assigned to a link aggregation group (LAG) formed by the Virtual Chassis. The trunk ID value is –1 if no trunk is formed. A LAG between uplink VCPs requires that the link speed be the same on connected interfaces and that at least two VCPs on one member be connected to at least two VCPs on the other member in an EX4200 or EX4500 Virtual Chassis.</p> <p>Dedicated VCP LAGs are assigned trunk IDs 1 and 2. Trunk IDs for LAGs formed with uplink VCPs therefore have values of 3 or greater.</p> <p>The trunk ID value changes if the link-adjacency state between LAG members changes; trunk membership is then allocated or deallocated.</p>
Status	<p>Interface status:</p> <ul style="list-style-type: none"> absent—Interface is not a VCP link. down—VCP link is down. up—VCP link is up.
Speed (mbps)	Speed of the interface in megabits per second.
Neighbor ID/Interface	The Virtual Chassis member ID and interface of a VCP on a member that is connected to the interface or PIC/Port field in the same row as this interface.

Sample Output

show virtual-chassis vc-port (EX4200 Virtual Chassis)

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

```
fpc0:
```

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

show virtual-chassis vc-port (EX8200 Virtual Chassis)

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

```
member0:
```

Interface or Slot/PIC/Port	Type	Trunk ID	Status	Speed (mbps)	Neighbor ID	Interface
vcp-0/0	Dedicated	-1	Up	1000	8	vcp-1/1
vcp-0/1	Dedicated	-1	Up	1000	8	vcp-2/0
4/0/4	Configured	-1	Up	10000	1	vcp-3/0/4
4/0/7	Configured	-1	Down	10000		
4/0/3	Configured		Absent			
4/0/2	Configured		Absent			
4/0/5	Configured		Absent			
4/0/6	Configured		Absent			
4/0/1	Configured		Absent			
4/0/0	Configured		Absent			

```
member1:
```

Interface or Slot/PIC/Port	Type	Trunk ID	Status	Speed (mbps)	Neighbor ID	Interface
vcp-0/0	Dedicated	-1	Up	1000	8	vcp-1/2
3/0/0	Configured	-1	Down	10000		
3/0/1	Configured	-1	Down	10000		
3/0/4	Configured	-1	Up	10000	0	vcp-4/0/4
3/0/5	Configured		Absent			
4/0/5	Configured		Absent			
4/0/4	Configured		Absent			

```
member8:
```

Interface or Slot/PIC/Port	Type	Trunk ID	Status	Speed (mbps)	Neighbor ID	Interface
vcp-0/0	Dedicated	-1	Down	1000		
vcp-1/0	Dedicated	-1	Up	1000	9	vcp-1/0
vcp-1/1	Dedicated	-1	Up	1000	0	vcp-0/0
vcp-1/2	Dedicated	-1	Up	1000	1	vcp-0/0
vcp-1/3	Dedicated	-1	Up	1000	9	vcp-1/3

vcp-2/0	Dedicated	-1	Up	1000	0	vcp-0/1
vcp-2/1	Dedicated	-1	Up	1000	9	vcp-1/2
vcp-2/2	Dedicated	-1	Down	1000		
vcp-2/3	Dedicated	-1	Down	1000		

member9:

Interface or Slot/PIC/Port	Type	Trunk ID	Status	Speed (mbps)	Neighbor ID	Interface
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 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
1/1	Auto-Configured	3	Up	1000	2	vcp-255/1/1

fpc1:

Interface or PIC / Port	Type	Trunk ID	Status	Speed (mbps)	Neighbor ID	Interface
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 or PIC / Port	Type	Trunk ID	Status	Speed (mbps)	Neighbor ID	Interface
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 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	Auto-Configured	-1	Up	1000	1	vcp-255/1/0

