



Junos[®] OS for EX Series Ethernet Switches

Virtual Chassis Feature Guide for EX8200 Switches

Release

15.1



Modified: 2016-07-19

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

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

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

Junos[®] OS for EX Series Ethernet Switches Virtual Chassis Feature Guide for EX8200 Switches
Release 15.1
Copyright © 2016, Juniper Networks, Inc.
All rights reserved.

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

Table of Contents

	About the Documentation	xi
	Documentation and Release Notes	xi
	Supported Platforms	xi
	Using the Examples in This Manual	xi
	Merging a Full Example	xii
	Merging a Snippet	xii
	Documentation Conventions	xiii
	Documentation Feedback	xv
	Requesting Technical Support	xv
	Self-Help Online Tools and Resources	xv
	Opening a Case with JTAC	xvi
Part 1	Overview	
Chapter 1	Virtual Chassis Overview	3
	EX8200 Virtual Chassis Overview	4
	Understanding EX8200 Virtual Chassis Components	4
	Virtual Chassis Ports (VCPs)	5
	Master External Routing Engine Role	5
	Backup External Routing Engine Role	6
	Linecard Role	6
	Member Switch and Member ID	6
	Virtual Chassis Identifier (VCID)	7
	Understanding EX8200 Virtual Chassis Topologies	7
	Understanding the Full Mesh EX8200 Virtual Chassis Topology	7
	Understanding the Ring EX8200 Virtual Chassis Topology	9
	The Role of XRE200 External Routing Engine Redundancy in a Virtual Chassis Topology	10
	EX8200 Virtual Chassis Basic Topology	10
	Understanding Virtual Chassis Member ID Numbering in an EX8200 Virtual Chassis	11
	Understanding Virtual Chassis Roles in an EX8200 Virtual Chassis	12
	Understanding Virtual Chassis Ports in an EX8200 Virtual Chassis	13
	Understanding Software Upgrades in an EX8200 Virtual Chassis	14
	Understanding Virtual Chassis Port Link Aggregation in an EX8200 Virtual Chassis	15
	Understanding Link Aggregation in an EX8200 Virtual Chassis	15
	Understanding File Storage in an EX8200 Virtual Chassis	16
	Understanding EX8200 Virtual Chassis Compatibility Requirements	17
	Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis	17

	Understanding Global Management of an EX8200 Virtual Chassis	17
	Understanding Long-Distance Virtual Chassis Port Configuration for an EX8200 Virtual Chassis	18
	Understanding Configuration File Management in an EX8200 Virtual Chassis . . .	19
	Network Port Interface Names on an EX8200 Virtual Chassis	20
	Virtual Chassis Port (VCP) Interface Names in an EX8200 Virtual Chassis	20
	Virtual Chassis Control Interface (VCCI) Module VCP Name on the XRE200 External Routing Engine	21
	Native VCP Name on the XRE200 External Routing Engine	21
	EX8200 Switch to XRE200 External Routing Engine VCP Name	21
	EX8200 Switch to EX8200 Switch VCP Name	21
Part 2	Configuration	
Chapter 2	Configuration Examples	25
	Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines	25
Chapter 3	Configuration Tasks	31
	Configuring an EX8200 Virtual Chassis (CLI Procedure)	32
	Adding or Replacing a Member Switch or an External Routing Engine in an EX8200 Virtual Chassis (CLI Procedure)	34
	Setting a 10-Gigabit Ethernet Port as a Virtual Chassis Port in an EX8200 Virtual Chassis (CLI Procedure)	36
	Configuring a Long-Distance Virtual Chassis Port Connection for an EX8200 Virtual Chassis	37
	Installing Software for All Devices in an EX8200 Virtual Chassis	40
	Installing Software for a Single Device in an EX8200 Virtual Chassis	42
	Performing a Software Recovery Installation for an XRE200 External Routing Engine	43
	Using Groups to Configure Power Supply Management, Power Budgets, and LCD Menus on EX8200 Member Switches in an EX8200 Virtual Chassis (CLI Procedure)	45
Chapter 4	Configuration Statements	47
	[edit virtual-chassis] Configuration Statement Hierarchy	47
	Supported Statements in the [edit virtual-chassis] Hierarchy Level	48
	Unsupported Statements in the [edit virtual-chassis] Hierarchy Level	48
	graceful-restart (Enabling Globally)	50
	id	51
	location (Virtual Chassis)	52
	mac-persistence-timer	53
	member	54
	no-split-detection	55
	preprovisioned	56
	role	57
	serial-number	60
	traceoptions (Virtual Chassis)	61
	virtual-chassis	64

Part 3	Administration	
Chapter 5	Routine Monitoring	69
	Verifying the Member ID, Role, and Neighbor Member Connections of an EX8200 Virtual Chassis Member	69
	Verifying Virtual Chassis Ports in an EX8200 Virtual Chassis	70
	Verifying Hard Disk Memory Status on an XRE200 External Routing Engine	71
Chapter 6	Operational Commands	73
	clear virtual-chassis protocol statistics	74
	clear virtual-chassis vc-port statistics	76
	request chassis routing-engine hard-disk-test	78
	request session member	81
	request system software recovery-package	82
	request virtual-chassis device-reachability	85
	request virtual-chassis reactivate	87
	request virtual-chassis recycle	88
	request virtual-chassis renumber	89
	request virtual-chassis vc-port	90
	set chassis virtual-chassis	92
	show system software recovery-package	93
	show virtual-chassis active-topology	95
	show virtual-chassis protocol adjacency	100
	show virtual-chassis protocol database	104
	show virtual-chassis protocol interface	108
	show virtual-chassis protocol route	111
	show virtual-chassis protocol statistics	114
	show virtual-chassis	117
	show virtual-chassis vc-port	122
	show virtual-chassis vc-port statistics	126
Part 4	Troubleshooting	
Chapter 7	Troubleshooting Procedures	135
	Recovering a Routing Engine in a Member Switch in an EX8200 Virtual Chassis	135

List of Figures

Part 1	Overview	
Chapter 1	Virtual Chassis Overview	3
	Figure 1: EX8200 Virtual Chassis Full Mesh Topology with Four EX8200 Member Switches	8
	Figure 2: EX8200 Virtual Chassis Full Mesh Topology with Two EX8200 Member Switches	9
	Figure 3: EX8200 Virtual Chassis Ring Topology	10
	Figure 4: EX8200 Virtual Chassis Basic Topology	11
	Figure 5: EX8200 Virtual Chassis Ports	14
Part 2	Configuration	
Chapter 3	Configuration Tasks	31
	Figure 6: Long Distance EX8200 Virtual Chassis	38

List of Tables

	About the Documentation	xi
	Table 1: Notice Icons	xiii
	Table 2: Text and Syntax Conventions	xiii
Part 1	Overview	
Chapter 1	Virtual Chassis Overview	3
	Table 3: EX8200 Virtual Chassis Roles and Member IDs	6
	Table 4: EX8200 Virtual Chassis Roles and Member IDs	12
	Table 5: EX8200 Virtual Chassis Roles	12
Part 2	Configuration	
Chapter 2	Configuration Examples	25
	Table 6: Components of the EX8200 Virtual Chassis Topology	26
Part 3	Administration	
Chapter 6	Operational Commands	73
	Table 7: request chassis routing-engine hard-disk-test Output Fields	78
	Table 8: request virtual chassis device reachability Output Fields	86
	Table 9: show virtual-chassis active-topology Output Fields	95
	Table 10: show virtual-chassis protocol adjacency Output Fields	101
	Table 11: show virtual-chassis protocol database Output Fields	105
	Table 12: show virtual-chassis protocol interface Output Fields	109
	Table 13: show virtual-chassis protocol route Output Fields	111
	Table 14: show virtual-chassis protocol statistics Output Fields	114
	Table 15: show virtual-chassis Output Fields	117
	Table 16: show virtual-chassis vc-port Output Fields	122
	Table 17: show virtual-chassis vc-port statistics Output Fields	127

About the Documentation

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

Documentation and Release Notes

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

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

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

Supported Platforms

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

- EX Series

Using the Examples in This Manual

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

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

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

Merging a Full Example

To merge a full example, follow these steps:

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

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

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

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

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

Merging a Snippet

To merge a snippet, follow these steps:

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

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

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

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

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

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

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

For more information about the **load** command, see the *CLI User Guide*.

Documentation Conventions

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

Table 1: Notice Icons

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

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

Table 2: Text and Syntax Conventions

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

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

Convention	Description	Examples
Fixed-width text like this	Represents output that appears on the terminal screen.	<code>user@host> show chassis alarms</code> <code>No alarms currently active</code>
<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 [<i>community-ids</i>]
Indentation and braces ({ })	Identifies a level in the configuration hierarchy.	[edit] routing-options { static { route default { nexthop <i>address</i> ; retain; } } }
;(semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
GUI Conventions		
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.

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

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

Documentation Feedback

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

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

Requesting Technical Support

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

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

Self-Help Online Tools and Resources

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

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

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

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

Opening a Case with JTAC

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

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

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

PART 1

Overview

- [Virtual Chassis Overview on page 3](#)

CHAPTER 1

Virtual Chassis Overview

- [EX8200 Virtual Chassis Overview on page 4](#)
- [Understanding EX8200 Virtual Chassis Components on page 4](#)
- [Understanding EX8200 Virtual Chassis Topologies on page 7](#)
- [Understanding Virtual Chassis Member ID Numbering in an EX8200 Virtual Chassis on page 11](#)
- [Understanding Virtual Chassis Roles in an EX8200 Virtual Chassis on page 12](#)
- [Understanding Virtual Chassis Ports in an EX8200 Virtual Chassis on page 13](#)
- [Understanding Software Upgrades in an EX8200 Virtual Chassis on page 14](#)
- [Understanding Virtual Chassis Port Link Aggregation in an EX8200 Virtual Chassis on page 15](#)
- [Understanding Link Aggregation in an EX8200 Virtual Chassis on page 15](#)
- [Understanding File Storage in an EX8200 Virtual Chassis on page 16](#)
- [Understanding EX8200 Virtual Chassis Compatibility Requirements on page 17](#)
- [Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis on page 17](#)
- [Understanding Global Management of an EX8200 Virtual Chassis on page 17](#)
- [Understanding Long-Distance Virtual Chassis Port Configuration for an EX8200 Virtual Chassis on page 18](#)
- [Understanding Configuration File Management in an EX8200 Virtual Chassis on page 19](#)
- [Network Port Interface Names on an EX8200 Virtual Chassis on page 20](#)
- [Virtual Chassis Port \(VCP\) Interface Names in an EX8200 Virtual Chassis on page 20](#)

EX8200 Virtual Chassis Overview

An EX8200 Virtual Chassis is multiple Juniper Networks EX8200 Ethernet Switches connected together that operate as a single network entity. The advantages of connecting multiple EX8200 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, and a simplified Layer 2 network topology that minimizes or eliminates the need for loop prevention protocols such as Spanning Tree Protocol (STP).

The XRE200 Routing Engine is the device that connects all member switches into a single Virtual Chassis. An EX8200 Virtual Chassis is formed by connecting a Virtual Chassis port (VCP) on an XRE200 External Routing Engine to the management ports (labeled **MGMT**) on the internal Routing Engines of the EX8200 member switches. A member switch is any switch that is part of a Virtual Chassis.

You can add a second XRE200 External Routing Engine to an EX8200 Virtual Chassis topology to provide external Routing Engine redundancy. See [“Understanding EX8200 Virtual Chassis Topologies” on page 7](#).

Related Documentation

- [Understanding EX8200 Virtual Chassis Components on page 4](#)
- [Understanding Virtual Chassis Member ID Numbering in an EX8200 Virtual Chassis on page 11](#)
- [Understanding Virtual Chassis Ports in an EX8200 Virtual Chassis on page 13](#)
- [Understanding EX8200 Virtual Chassis Compatibility Requirements on page 17](#)
- [Understanding Virtual Chassis Roles in an EX8200 Virtual Chassis on page 12](#)
- [Configuring an EX8200 Virtual Chassis \(CLI Procedure\) on page 32](#)

Understanding EX8200 Virtual Chassis Components

An EX8200 Virtual Chassis allows you to interconnect up to four Juniper Networks EX8200 Ethernet Switches and run them as a single network entity. You interconnect the EX8200 switches by connecting them to an XRE200 External Routing Engine.

This topic covers:

- [Virtual Chassis Ports \(VCPs\) on page 5](#)
- [Master External Routing Engine Role on page 5](#)
- [Backup External Routing Engine Role on page 6](#)
- [Linecard Role on page 6](#)
- [Member Switch and Member ID on page 6](#)
- [Virtual Chassis Identifier \(VCID\) on page 7](#)

Virtual Chassis Ports (VCPs)

A Virtual Chassis port (VCP) in an EX8200 Virtual Chassis is any interface whose function is to send and receive Virtual Chassis Control Protocol (VCCP) traffic to create, monitor, and maintain the Virtual Chassis. VCPs are used for creating, monitoring, and maintaining the Virtual Chassis as well as carrying data traffic through the Virtual Chassis.

All Gigabit Ethernet ports on the Virtual Chassis Control Interface (VCCI) modules on the XRE200 External Routine Engine are VCPs. Port 0 on the XRE200 External Routing Engine is also a VCP. You use VCPs to connect EX8200 switches to the external Routing Engine to form a Virtual Chassis and also to connect XRE200 External Routing Engines together to provide external Routing Engine redundancy within the Virtual Chassis. Any link connecting an XRE200 External Routing Engine to an EX8200 switch or to another XRE200 External Routing Engine, therefore, is automatically a VCP link. No user configuration is required to configure these VCP links.

The small form-factor pluggable (SFP) ports on the 4-port 1000BASE-X SFP VCCI module are also VCPs. You can use these ports to directly connect XRE200 External Routing Engines together within an EX8200 Virtual Chassis with no user configuration. You can also use these SFP ports to connect an XRE200 External Routing Engine to an EX8200 switch within a Virtual Chassis when an intermediary switch is used to convert the SFP fiber-optic connection to a copper connection. This conversion is necessary because the management ports (labeled **MGMT**) on EX8200 switches do not support fiber-optic connections. See [“Configuring a Long-Distance Virtual Chassis Port Connection for an EX8200 Virtual Chassis” on page 37](#).

VCP links are also needed in an EX8200 Virtual Chassis to send and receive VCCP traffic between two EX8200 switches. You must explicitly configure VCP links between two EX8200 switches; otherwise, the switches detect such links as network links.

Master External Routing Engine Role

The function of each hardware device in a Virtual Chassis is determined by that device's role.

The master role in an EX8200 Virtual Chassis is assigned to an XRE200 External Routing Engine only. The XRE200 External Routing Engine in the master role:

- Controls most Routing Engine functions for all switches in the Virtual Chassis.
- Manages the Virtual Chassis configuration.
- Provides a single point where you can view all functionality for all devices in the Virtual Chassis.
- Runs the Juniper Networks Junos operating system (Junos OS).

When a single XRE200 External Routing Engine is connecting an EX8200 Virtual Chassis and no backup external Routing Engine is configured, the single external Routing Engine is assigned the master role. When a backup external Routing Engine is part of the Virtual Chassis, the external Routing Engine with the most uptime functions as the master.

Backup External Routing Engine Role

The function of each hardware device in a Virtual Chassis is determined by that device's role.

The external Routing Engine that functions in the backup role:

- Maintains a state of readiness to take over the master role if the master fails.
- Runs Junos OS.
- Synchronizes with the master, so that it is prepared to preserve routing information and maintain network connectivity without disruption if the master external Routing Engine becomes unavailable.

You must have two external Routing Engines in an EX8200 Virtual Chassis configuration to have an external Routing Engine in the backup role. When a backup external Routing Engine is part of the Virtual Chassis, the external Routing Engine with the lesser amount of uptime functions as the backup external Routing Engine.

Linecard Role

All EX8200 switches in an EX8200 Virtual Chassis are assigned the linecard role. Switches in the linecard role:

- Forward network traffic.
- Do not run the chassis control protocols.
- Can detect certain error conditions (such as an unplugged cable) on any interfaces that have been configured on them through the master.

Member Switch and Member ID

An EX8200 Virtual Chassis contains member IDs 0 through 9. Member IDs 0 through 7 must be configured as EX8200 switches.

Member IDs 8 and 9 must be configured as XRE200 External Routing Engines. An XRE200 External Routing Engine automatically assumes the next available member ID if a Virtual Chassis configuration has not been configured and committed. The external Routing Engine does not function properly until it is configured as member 8 or 9, however.

[Table 3 on page 6](#) summarizes EX8200 Virtual Chassis roles and member IDs.

Table 3: EX8200 Virtual Chassis Roles and Member IDs

Hardware	Role	Member IDs
EX8208 or EX8216 switch	Linecard	0–7
XRE200 External Routing Engine	Master or backup	8 or 9

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 the user interface.

Related Documentation

- [EX8200 Virtual Chassis Overview on page 4](#)
- [Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines on page 25](#)
- [Configuring an EX8200 Virtual Chassis \(CLI Procedure\) on page 32](#)

Understanding EX8200 Virtual Chassis Topologies

An EX8200 Virtual Chassis is composed of one or two XRE200 External Routing Engines interconnected with up to four Juniper Networks EX8200 Ethernet Switches.

This topic explains the topologies that you can use to connect these devices.

The topic covers:

- [Understanding the Full Mesh EX8200 Virtual Chassis Topology on page 7](#)
- [Understanding the Ring EX8200 Virtual Chassis Topology on page 9](#)
- [The Role of XRE200 External Routing Engine Redundancy in a Virtual Chassis Topology on page 10](#)
- [EX8200 Virtual Chassis Basic Topology on page 10](#)

Understanding the Full Mesh EX8200 Virtual Chassis Topology

We recommend that you use a full mesh topology for all EX8200 Virtual Chassis whenever possible.

In a full mesh topology, every possible Virtual Chassis port (VCP) connection within the Virtual Chassis is established. Both external Routing Engines have connections to all switches in addition to having connections to each other. The switches are connected together by a user-configured VCP link—preferably multiple VCP links, which will automatically form a VCP link aggregation group (LAG)—to provide another link for Virtual Chassis Control Protocol (VCCP) traffic within the Virtual Chassis. Each member switch has a VCP link to every other member switch.

[Figure 1 on page 8](#) shows a full mesh EX8200 Virtual Chassis topology with four member switches.

Figure 1: EX8200 Virtual Chassis Full Mesh Topology with Four EX8200 Member Switches

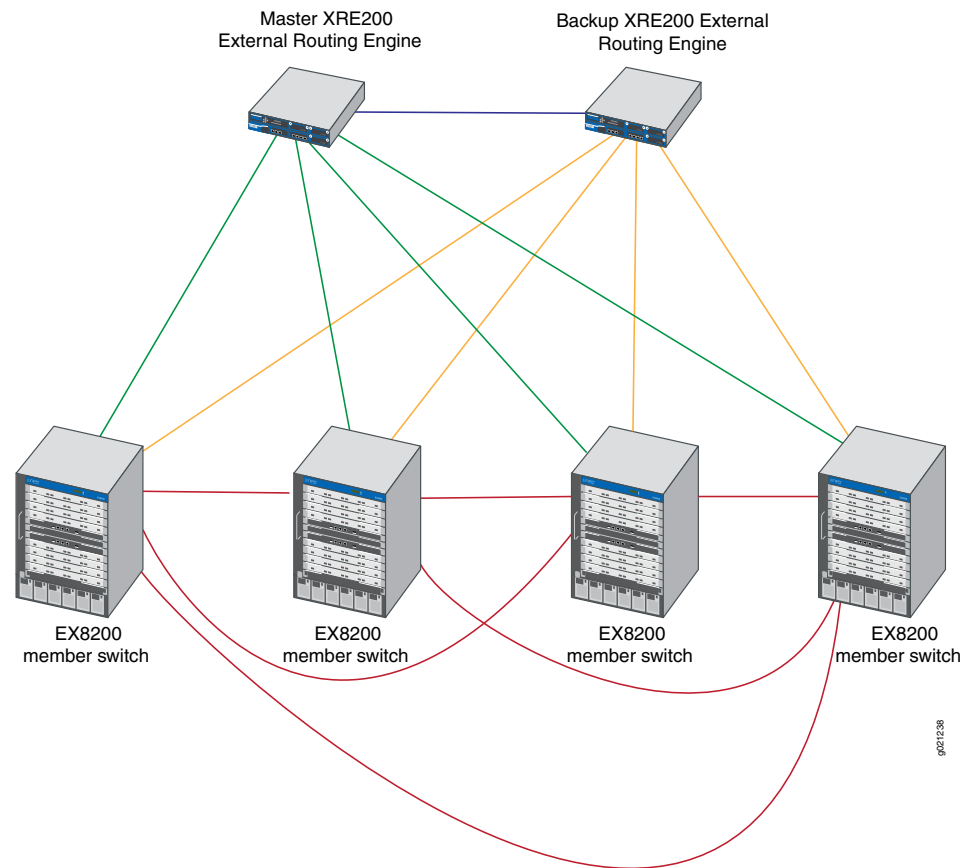
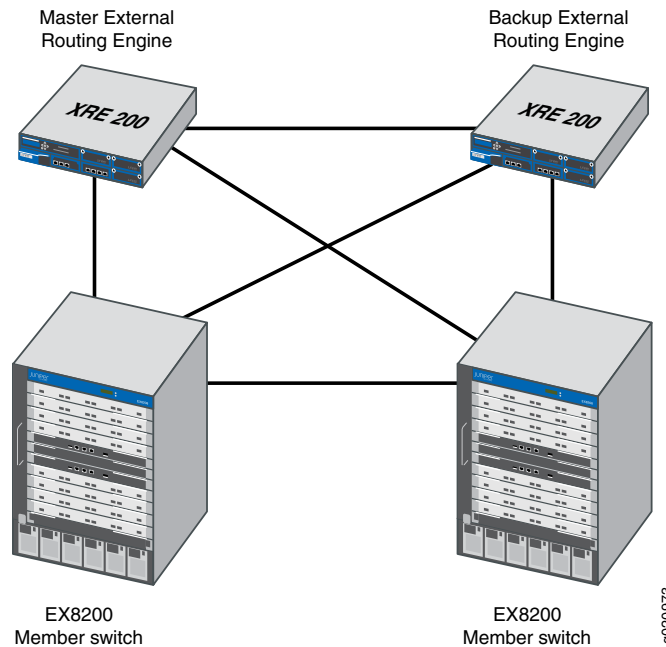


Figure 2 on page 9 shows a full mesh EX8200 Virtual Chassis topology with two member switches.

Figure 2: EX8200 Virtual Chassis Full Mesh Topology with Two EX8200 Member Switches



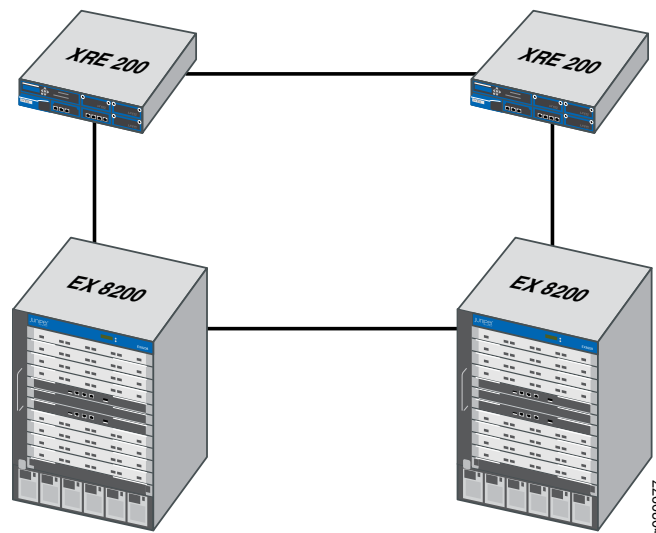
A full mesh Virtual Chassis topology has several advantages over other topologies. Because there are multiple paths to all devices in the Virtual Chassis, a single link failure does not disable any path from one device to another. Because the multiple VCP links connect all devices together, the VCCP has more flexibility when sending traffic through the Virtual Chassis than it does in other topologies.

Understanding the Ring EX8200 Virtual Chassis Topology

In a ring topology, each device in the EX8200 Virtual Chassis has at least one path to every other device in the Virtual Chassis. The paths are not necessarily direct and traffic might be required to travel across multiple hops to reach another device within the same Virtual Chassis.

Figure 3 on page 10 shows a ring EX8200 Virtual Chassis topology.

Figure 3: EX8200 Virtual Chassis Ring Topology



VCCP traffic often has to travel multiple hops to communicate with other devices in the Virtual Chassis. When a VCP link fails in a ring topology, additional hops are added for VCCP traffic in the Virtual Chassis ring topology. VCP link failures are problematic in a ring topology, as the shortage of VCP links in the topology increases the potential for a single link failure to cause the EX8200 member switches to split from the Virtual Chassis.

An EX8200 Virtual Chassis full mesh topology is always recommended. Use the ring topology in cases where some of the VCP links in a full mesh topology have failed or when the equipment to configure a full mesh topology is unavailable.

The Role of XRE200 External Routing Engine Redundancy in a Virtual Chassis Topology

A backup XRE200 External Routing Engine provides redundancy to a master XRE200 External Routing Engine. If a master external Routing Engine fails, the backup external Routing Engine takes control of the Virtual Chassis, allowing network traffic to continue to pass through the Virtual Chassis with minimal network disruption.



NOTE: A backup XRE200 External Routing Engine is recommended in all EX8200 Virtual Chassis configurations.

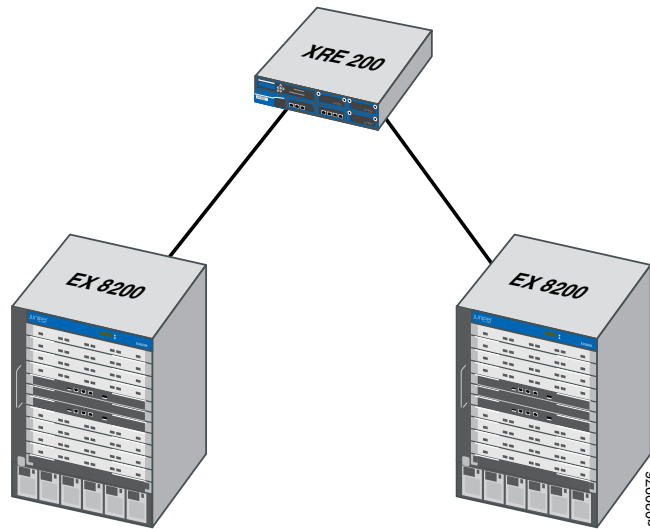
When a backup external Routing Engine is included in the EX8200 Virtual Chassis, one link must be used to connect the master external Routing Engine to the backup external Routing Engine. Both EX8200 switches must also have connections to the backup external Routing Engine.

EX8200 Virtual Chassis Basic Topology

The most basic EX8200 Virtual Chassis topology has one XRE200 External Routing Engine connecting two EX8200 switches into a Virtual Chassis.

Figure 4 on page 11 shows a basic EX8200 Virtual Chassis topology.

Figure 4: EX8200 Virtual Chassis Basic Topology



In this topology, all VCCP traffic has to travel on one of the two links, which is problematic because the switches have to send traffic through the external Routing Engine to communicate with each other.

There is no link redundancy in this topology. If either link fails, one of the switches is no longer participating in the Virtual Chassis.

There is no redundant external Routing Engine in this topology. If the external Routing Engine fails, the Virtual Chassis is no longer active.

An EX8200 Virtual Chassis full mesh topology is always recommended. Use the basic topology only when VCP links have failed or when the equipment to configure a full mesh topology is unavailable.

Related Documentation

- [EX8200 Virtual Chassis Overview on page 4](#)
- [Understanding EX8200 Virtual Chassis Components on page 4](#)
- [Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines on page 25](#)

Understanding Virtual Chassis Member ID Numbering in an EX8200 Virtual Chassis

An EX8200 Virtual Chassis contains member IDs 0 through 9. Member IDs 0 through 7 are reserved for Juniper Networks EX8200 Ethernet Switches, which always assume the linecard role in the EX8200 Virtual Chassis.

Member IDs 8 and 9 are reserved for the XRE200 External Routing Engines, which assume either the master or backup role in the EX8200 Virtual Chassis. These member IDs must be configured before configuring an EX8200 Virtual Chassis. If an XRE200 is not configured

as member ID 8 or 9 before connecting to a Virtual Chassis, it will not work until one of these member IDs is configured.

The master and backup roles for XRE200 External Routing Engines are determined by external Routing Engine uptime; the external Routing Engine that has been up the longest is selected for the master role. The role has no correlation to member ID; a master or backup XRE200 External Routing Engine can be member ID 8 or 9.

[Table 4 on page 12](#) summarizes EX8200 Virtual Chassis member IDs and roles.

Table 4: EX8200 Virtual Chassis Roles and Member IDs

Hardware	Role	Member IDs
EX8208 or EX8216 switch	Linecard	0–7
XRE200 External Routing Engine	Master, Backup	8–9

Related Documentation

- [EX8200 Virtual Chassis Overview on page 4](#)
- [Understanding EX8200 Virtual Chassis Components on page 4](#)
- [Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines on page 25](#)

Understanding Virtual Chassis Roles in an EX8200 Virtual Chassis

The roles of Virtual Chassis members in an EX8200 Virtual Chassis are defined by uptime and hardware type. The XRE200 External Routing Engine with the most uptime assumes the master role; the external Routing Engine with less uptime assumes the backup role. EX8200 member switches are always in the linecard role.

[Table 5 on page 12](#) provides a summary of the EX8200 Virtual Chassis roles.

Table 5: EX8200 Virtual Chassis Roles

Role	Description
Master	<p>The master XRE200 External Routing Engine.</p> <p>The master XRE200 External Routing Engine is responsible for maintaining an active EX8200 Virtual Chassis.</p>
Backup	<p>The backup XRE200 External Routing Engine.</p> <p>The backup external Routing Engine takes control of the Virtual Chassis when the master external Routing Engine fails.</p>
Linecard	<p>EX8208 or EX8216 switch.</p> <p>All EX8200 member switches in the Virtual Chassis are in the linecard role.</p>

Related Documentation

- [Understanding EX8200 Virtual Chassis Components on page 4](#)

- [EX8200 Virtual Chassis Overview on page 4](#)

Understanding Virtual Chassis Ports in an EX8200 Virtual Chassis

A Virtual Chassis port (VCP) in an EX8200 Virtual Chassis is any port whose function is to send and receive Virtual Chassis Control Protocol (VCCP) traffic to create, monitor, and maintain the Virtual Chassis. VCPs are responsible for creating, monitoring, and maintaining the Virtual Chassis as well as carrying data traffic through the Virtual Chassis.

All Gigabit Ethernet ports on the 10/100/100BASE-T RJ-45 Virtual Chassis Control Interface (VCCI) modules and all small form-factor pluggable (SFP) ports on the 1000BASE-X SFP VCCI modules on the XRE200 External Routing Engine are VCPs. Port 0 on the XRE200 External Routing Engine is also a VCP. You can use VCPs to connect Juniper Networks EX8200 Ethernet Switches to the external Routing Engine to form a Virtual Chassis and also to connect XRE200 External Routing Engines together to provide external Routing Engine redundancy within the Virtual Chassis. Any link connecting an XRE200 External Routing Engine to an EX8200 switch or to another XRE200 External Routing Engine, therefore, is automatically a VCP link. No user configuration is required to configure these VCP links.



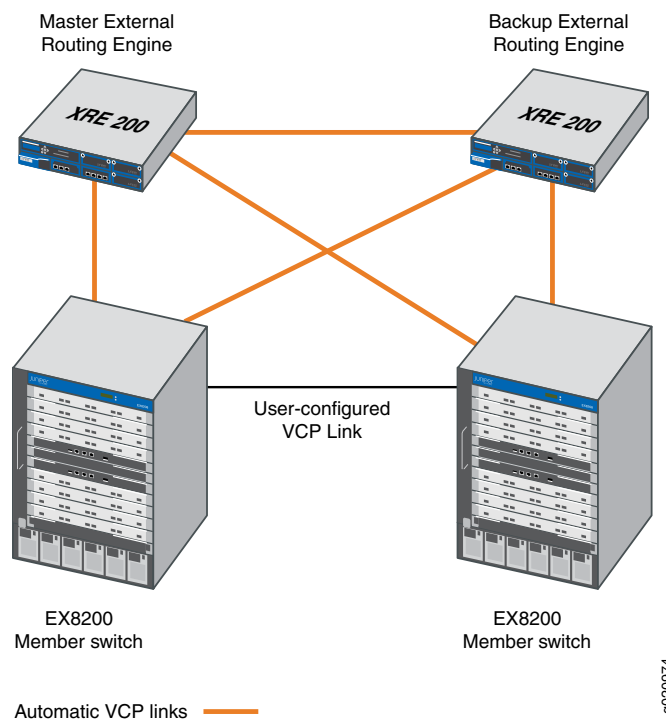
NOTE: To connect a 1000BASE-X SFP interface on the XRE200 External Routing Engine to an EX8200 switch, you must use an intermediary switch. The intermediary switch is required because the management interface (labeled **MGMT**) on the Routing Engine (RE) module or the Switch Fabric and Routing Engine (SRE) module in the EX8200 switch does not support SFP connections. See [“Understanding Long-Distance Virtual Chassis Port Configuration for an EX8200 Virtual Chassis” on page 18](#).

VCP links are also needed in an EX8200 Virtual Chassis to send and receive VCCP traffic between two EX8200 member switches. You must explicitly configure VCP links between two EX8200 switches; otherwise, the switches detect the link as a network link. For information regarding which ports on an EX8200 switch can be configured as VCPs, see *Line Card Model and Version Compatibility in an EX8200 Switch*.

Multiple VCP links can be configured between two EX8200 member switches; these links will automatically form a link aggregation group (LAG). See [“Understanding Virtual Chassis Port Link Aggregation in an EX8200 Virtual Chassis” on page 15](#).

[Figure 5 on page 14](#) shows a two member Virtual Chassis configured in a full mesh topology and highlights which VCP links must be user-configured.

Figure 5: EX8200 Virtual Chassis Ports



Related Documentation

- [EX8200 Virtual Chassis Overview on page 4](#)
- [Setting a 10-Gigabit Ethernet Port as a Virtual Chassis Port in an EX8200 Virtual Chassis \(CLI Procedure\) on page 36](#)

Understanding Software Upgrades in an EX8200 Virtual Chassis

In an EX8200 Virtual Chassis, both XRE200 External Routing Engines and the master Routing Engines of the member switches must be running the same version of Juniper Networks Junos OS Software. You must upgrade all switches and external Routing Engine to the same version of Junos OS before connecting the Virtual Chassis for the first time.

After the Virtual Chassis is connected, you can upgrade the software for all switches and external Routing Engines in the EX8200 Virtual Chassis through the use of the XRE200 External Routing Engine software package. You identify this software package on the software download site by confirming that “xre200” is listed in the filename as the *package-name*, for instance, `jinstall-ex-xre200-10.4R1.6-domestic-signed.tgz`. This software package contains the Junos OS image for the external Routing Engines as well as the software packages for the EX8200 switches. The software packages in the EX8200 Virtual Chassis software package are identical to the software packages for the stand-alone Juniper Networks EX8200 Ethernet Switches; the switches are running the same software regardless of whether they were upgraded with or without the XRE200 External Routing Engine software package.

- Related Documentation**
- [Installing Software for All Devices in an EX8200 Virtual Chassis on page 40](#)
 - [Installing Software for a Single Device in an EX8200 Virtual Chassis on page 42](#)
 - [EX8200 Virtual Chassis Overview on page 4](#)

Understanding Virtual Chassis Port Link Aggregation in an EX8200 Virtual Chassis

In an EX8200 Virtual Chassis, physical 10–GigabitEthernet ports that are configured as Virtual Chassis port (VCP) links that connect member switches together automatically form a logical point-to-point link, known as a link aggregation group (LAG) or bundle. A LAG provides more bandwidth than a single Ethernet link can provide. Additionally, link aggregation provides network redundancy by load-balancing traffic across all available links. If one link fails, the system automatically load-balances traffic across all remaining links.

VCP links between member switches in the Virtual Chassis are automatically connected into a single LAG; no user configuration is required to configure the LAG. You can configure up to 12 Ethernet ports as VCPs to form a LAG between two member switches in a Virtual Chassis.

The 12 Ethernet links as VCP LAG restriction is between member switches. Each member switch in an EX8200 Virtual Chassis can have a VCP LAG to every other member switch in the Virtual Chassis. For instance, in a Virtual Chassis with four member switches, the switch configured as member Id 0 can have one VCP LAG to the switch configured as member id 1, another VCP LAG to the switch configured as member id 2, and another VCP LAG to the switch configured as member id 3. Each VCP LAG must have no more than 12 links. Only one VCP LAG is permitted between the same member switches in the Virtual Chassis.

LAGs are also useful for connecting the network ports of directly-connected devices into an EX8200 Virtual Chassis. See [“Understanding Link Aggregation in an EX8200 Virtual Chassis” on page 15](#).

- Related Documentation**
- [EX8200 Virtual Chassis Overview on page 4](#)
 - [Configuring Aggregated Ethernet Links \(CLI Procedure\)](#)

Understanding Link Aggregation in an EX8200 Virtual Chassis

An EX8200 Virtual Chassis is typically used to aggregate standalone EX8200 switches at the distribution layer into a Virtual Chassis. The advantages of connecting multiple EX8200 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, and a simplified Layer 2 network topology that minimizes or eliminates the need for loop prevention protocols such as Spanning Tree Protocol (STP).

You can extend the benefits of a network using an EX8200 Virtual Chassis at the distribution layer by configuring the network connections of the devices connecting to the EX8200 Virtual Chassis as link aggregation group (LAG) bundles. A LAG balances

traffic across the member links within an aggregated Ethernet bundle and effectively increases the uplink bandwidth of network traffic into the Virtual Chassis. An EX8200 Virtual Chassis can support up to 239 LAGs and up to 12 interfaces per LAG. Member links of a single LAG can span between multiple switches that are part of the same EX8200 Virtual Chassis.

Configuring LAGs into the Virtual Chassis minimizes the need for spanning-tree protocols by bundling multiple interfaces into a single interface. Another advantage of link aggregation is increased link availability compared to that of links not configured in LAGs. Because the LAG is composed of multiple member links, the LAG continues to carry traffic over the remaining links when a link fails.

Link aggregation is also useful for combining Virtual Chassis Ports (VCPs) within an EX8200 Virtual Chassis. See [“Understanding Virtual Chassis Port Link Aggregation in an EX8200 Virtual Chassis” on page 15](#).

Related Documentation

- [EX8200 Virtual Chassis Overview on page 4](#)

Understanding File Storage in an EX8200 Virtual Chassis

An XRE200 External Routing Engine is shipped with two 160 GB Hard Disk Drive (HDD) modules. The HDD modules are fully redundant file storage units; all content that is stored on one HDD module is stored on the other HDD module. This redundancy helps ensure that no files are lost if one HDD module is unusable for any reason.

In an EX8200 Virtual Chassis, the master external Routing Engine is responsible for most Routing Engine functions for all switches in the Virtual Chassis. This responsibility extends to file storage; the external Routing Engine stores many of the files related to the Virtual Chassis, including:

- Software packages—You can use the XRE200 External Routing Engine software package to upgrade all devices in the Virtual Chassis simultaneously. You can store these packages on the external Routing Engine for the entire Virtual Chassis.
- Log files—All log files related to all devices in the EX8200 Virtual Chassis are stored on the XRE200 External Routing Engines. The EX8200 member switches continue to store logs related to their local functions.
- Configuration files—The XRE200 External Routing Engine has its own configuration file. This configuration file configures the Virtual Chassis itself and configures the network parameters for all network traffic entering and leaving the Virtual Chassis.

Starting in Junos OS Release 12.3R1, configuration files are saved to both XRE200 External Routing Engines and to both Routing Engines or Switch Fabric and Routing Engines on all EX8200 member switches each time a configuration is committed.

The XRE200 External Routing Engine uses RAID storage technology to manage file storage between the HDD modules. The RAID file storage management is handled automatically and is not user-configurable. If you suspect a memory problem with your external Routing Engine, you can run an HDD memory and diagnostics monitoring test.

The test results provide information on the health of RAID and the HDD modules. See [“Verifying Hard Disk Memory Status on an XRE200 External Routing Engine” on page 71](#).

- Related Documentation**
- [EX8200 Virtual Chassis Overview on page 4](#)
 - [Understanding Configuration File Management in an EX8200 Virtual Chassis on page 19](#)
 - *Hard Disk Drive (HDD) Module in an XRE200 External Routing Engine*

Understanding EX8200 Virtual Chassis Compatibility Requirements

For Juniper Networks EX8200 Ethernet Switches to be interconnected into a Virtual Chassis, the switches and the XRE200 External Routing Engines must be running the same software versions. The master external Routing Engine checks the hardware version, the Juniper Networks Junos operating system (Junos OS) version, and other component versions running in a switch that is physically interconnected to its Virtual Chassis port (VCP). A switch that is running a different version of software will not be allowed to join the Virtual Chassis configuration.

After an EX8200 Virtual Chassis is connected, software upgrades for all switches and external Routing Engines in the Virtual Chassis can be performed simultaneously. This process ensures all devices are running the same version of Junos OS. See [“Understanding Software Upgrades in an EX8200 Virtual Chassis” on page 14](#).

Different models in the EX8200 line of switches can be members of the same Virtual Chassis; for instance, an XRE200 External Routing Engine can be used to connect a Virtual Chassis composed of one EX8208 switch and one EX8216 switch.

- Related Documentation**
- [EX8200 Virtual Chassis Overview on page 4](#)
 - [Understanding EX8200 Virtual Chassis Components on page 4](#)

Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis

An EX8200 Virtual Chassis uses the Virtual Chassis Control Protocol (VCCP) to create and maintain a Virtual Chassis.

VCCP is enabled automatically when an EX8200 Virtual Chassis is formed. No user configuration is needed or available.

VCCP is based on the IS-IS protocol. You can monitor VCCP packet activity by using the **show virtual-chassis protocol** commands.

- Related Documentation**
- [EX8200 Virtual Chassis Overview on page 4](#)

Understanding Global Management of an EX8200 Virtual Chassis

An EX8200 Virtual Chassis is composed of up to four EX8200 switches and up to two XRE200 External Routing Engines, so it has up to 10 console (**CON**) ports and accessible out-of-band management (**MGMT**) ports. The XRE200 External Routing Engines **MGMT**

ports are accessible through the network; the EX8200 switch management ports are connected to the external Routing Engines.

You can configure an EX8200 Virtual Chassis by logging in to the out-of-band management (**MGMT**) port or to the console (**CON**) port of the master external Routing Engine.

If you log in to the backup external Routing Engine or in to a member switch and then want to configure the EX8200 Virtual Chassis, either cancel your session and log in to the master external Routing Engine or issue the **request session member** command to redirect your session to the master external Routing Engine.

**Related
Documentation**

- [EX8200 Virtual Chassis Overview on page 4](#)
- [Connecting a Switch to a Network for Out-of-Band Management](#)

Understanding Long-Distance Virtual Chassis Port Configuration for an EX8200 Virtual Chassis

An EX8200 Virtual Chassis is formed when XRE200 External Routing Engines and EX8200 switches are interconnected to form a Virtual Chassis.

The maximum length supported to connect an XRE200 External Routing Engine to another XRE200 External Routing Engine is 43.5 miles (70 kilometers). You must use the 4-port 1000BASE-X small form-factor pluggable (SFP) module on each XRE200 External Routing Engine to establish this VCP connection over any distance greater than 328 feet (100 meters). See *Virtual Chassis Control Interface (VCCI) Modules in an XRE200 External Routing Engine*.

An XRE200 External Routing Engine can connect to an EX8200 switch over a distance up to 43.5 miles using an SFP connection only if an intermediary switch that supports SFPs is used to convert the SFP optical connection into a fiber connection. This conversion is necessary because the management interface (labeled **MGMT**) on the Routing Engine (RE) or Switch Fabric and Routing Engine (SRE) in the EX8200 switch does not support SFP connections. See [“Configuring a Long-Distance Virtual Chassis Port Connection for an EX8200 Virtual Chassis” on page 37](#).

An EX Series switch used as an intermediary switch for this purpose must be dedicated to this task. VLANs must be created on the intermediary switch to properly direct VCCP traffic between the XRE200 External Routing Engines and the EX8200 switches. This switch cannot receive or transmit any other traffic. Any EX Series switch can be used for this purpose.



NOTE: The most affordable EX Series switch is an EX2200 switch with 24 total ports and no Power over Ethernet (PoE) ports (model number EX2200-24T-4G). If you need to purchase a new switch to extend a VCP for an EX8200 Virtual Chassis, we recommend purchasing this switch with an SFP uplink module and an SFP cable whose length is appropriate for your environment. The EX2200 switch supports uplink modules with SFP connections that span up to 43.5 miles (70 km).

Related Documentation

- [Configuring a Long-Distance Virtual Chassis Port Connection for an EX8200 Virtual Chassis on page 37](#)

Understanding Configuration File Management in an EX8200 Virtual Chassis

You configure all devices in an EX8200 Virtual Chassis by logging in to the master external Routing Engine. The configuration file on the master XRE200 External Routing Engine controls the configuration for all member switches and external Routing Engines in the Virtual Chassis.

In Juniper Networks Junos operating system (Junos OS) Release 12.2 or earlier releases, the configuration file used to configure the EX8200 Virtual Chassis was saved on the master and backup external Routing Engines only when configuration changes were committed.

Beginning with Junos OS Release 12.3, the configuration file used to configure the EX8200 Virtual Chassis is saved on the master and the backup external Routing Engines, on all Switch Fabric and Routing Engine (SRE) modules of all EX8208 member switches, and on all Routing Engine (RE) modules of all member EX8216 switches whenever it is committed. This configuration file saving behavior happens by default and is not user configurable.

The configuration changes allow you to configure parameters specific for an individual member switch in an EX8200 Virtual Chassis. You can now use groups to set and apply configuration parameters to a member switch only. For information about configuring parameters specific to a member switch in the configuration file for the EX8200 Virtual Chassis, see [“Using Groups to Configure Power Supply Management, Power Budgets, and LCD Menus on EX8200 Member Switches in an EX8200 Virtual Chassis \(CLI Procedure\)” on page 45](#).

The configuration file from the EX8200 Virtual Chassis remains on both SRE modules of all EX8208 member switches and on both RE modules of all EX8216 member switches even after the switch is removed from the Virtual Chassis.

Related Documentation

- [EX8200 Virtual Chassis Overview on page 4](#)
- [Understanding File Storage in an EX8200 Virtual Chassis on page 16](#)

Network Port Interface Names on an EX8200 Virtual Chassis

Network interfaces in Junos OS are specified as follows:

type-fpc / pic / port

This topic discusses network-port naming conventions for the network-facing interfaces on EX8200 member switches in an EX8200 Virtual Chassis. For information on interface naming conventions for Virtual Chassis Ports (VCPs) in an EX8200 Virtual Chassis, see [“Virtual Chassis Port \(VCP\) Interface Names in an EX8200 Virtual Chassis” on page 20](#).

An EX8200 Virtual Chassis network interface name uses these conventions:

- *type*—EX8200 member switch interfaces use the following media types:
 - **ge**—Gigabit Ethernet interface
 - **xe**—10-Gigabit Ethernet interface
- *fpc*—Flexible PIC Concentrator. In an EX8200 Virtual Chassis, 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 EX8200 Virtual Chassis reserves 16 FPC slot numbers for each member switch, regardless of whether the member switch is an EX8208 or an EX8216 switch. For EX8208 switches, therefore, FPC numbers 8 through 15 and 24 through 31 are not used.

- *pic*—PIC (Physical Interface Card) number in interface names. On EX8200 member switches in an EX8200 Virtual Chassis, the PIC number is always 0.
- *port*—Port number. On EX8200 switches, the network ports are numbered from left to right on each line card. On line cards that have two rows of ports, the ports on the top row start with 0 followed by the remaining even-numbered ports, and the ports on the bottom row start with 1 followed by the remaining odd-numbered ports.

Related Documentation • [EX8200 Virtual Chassis Overview on page 4](#)

Virtual Chassis Port (VCP) Interface Names in an EX8200 Virtual Chassis

A Virtual Chassis port (VCP) in an EX8200 Virtual Chassis is any port whose function is to send and receive Virtual Chassis Control Protocol (VCCP) traffic to create, monitor, and maintain the Virtual Chassis.

An EX8200 Virtual Chassis has four types of interfaces that can form a VCP link.

This topic describes the naming conventions for these interface types:

- [Virtual Chassis Control Interface \(VCCI\) Module VCP Name on the XRE200 External Routing Engine on page 21](#)
- [Native VCP Name on the XRE200 External Routing Engine on page 21](#)

- [EX8200 Switch to XRE200 External Routing Engine VCP Name on page 21](#)
- [EX8200 Switch to EX8200 Switch VCP Name on page 21](#)

Virtual Chassis Control Interface (VCCI) Module VCP Name on the XRE200 External Routing Engine

The interface names for the Gigabit Ethernet or small form-factor pluggable (SFP) VCPs on the VCCI modules in the XRE200 External Routing Engine follow this format:

vcp-slot-number/port-number

where:

- *slot-number*—Slot on the XRE200 External Routing Engine where the VCCI module is installed. VCCI modules can be installed in slot 1 and slot 2 on the external Routing Engine.
- *port-number*—Port number on the VCCI module. Each VCCI module has four ports numbered 0 through 3.

Native VCP Name on the XRE200 External Routing Engine

The native VCP on the XRE200 External Routing Engine—the port located next to the management (**MGMT**) port labeled **0**—is always named **vcp-0/0**.

EX8200 Switch to XRE200 External Routing Engine VCP Name

An XRE200 External Routing Engine forms a VCP when it connects to the management (**MGMT**) port on the Routing Engine (RE) module or the Switch Fabric and Routing Engine (SRE) module in an EX8200 switch.

The naming for the management port on the EX8200 switch when used as a VCP follows this format:

vcp-pic/slot

where:

- *pic* is the PIC number, which is always 0 for an EX8200 switch.
- *slot* is the number of the slot containing the RE module or the SRE module.

An EX8200 switch VCP name is always **vcp-0/0** or **vcp-0/1**.

EX8200 Switch to EX8200 Switch VCP Name

The names for the Gigabit Ethernet ports configured as VCP links on the EX8200 member switches follow this format:

vcp-fpc/pic/port

where:

- *fpc* is the slot number of the line card that contains the physical interface.

- *pic* is the PIC number, which is always 0 for an EX8200 switch.
- *port* is the network port number.

vcp-fpc/pic/port names apply to the device that the switch is connected to, not to the switch itself. Therefore, a VCP link is named according to the connected interface. For instance, if an external Routing Engine connects to interface **4/0/3** on an EX8200 switch, that VCP link is detected as **vcp-4/0/3** by the external Routing Engine.

**Related
Documentation**

- [EX8200 Virtual Chassis Overview on page 4](#)
- [Setting a 10-Gigabit Ethernet Port as a Virtual Chassis Port in an EX8200 Virtual Chassis \(CLI Procedure\) on page 36](#)

PART 2

Configuration

- [Configuration Examples on page 25](#)
- [Configuration Tasks on page 31](#)
- [Configuration Statements on page 47](#)

CHAPTER 2

Configuration Examples

- [Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines on page 25](#)

Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines

An EX8200 Virtual Chassis connects up to four EX8200 switches into a single network entity. The Virtual Chassis is formed by connecting the switches to an XRE200 External Routing Engine.

This example describes how to connect two EX8200 switches and two XRE200 External Routing Engines into a Virtual Chassis:

- [Requirements on page 25](#)
- [Overview and Topology on page 26](#)
- [Setting the EX8200 Switches as Virtual Chassis Members on page 27](#)
- [Preprovisioning the Virtual Chassis on page 28](#)
- [Configuring a VCP Link from Member Switch 0 to Member Switch 1 on page 29](#)
- [Configuring a VCP Link from Member Switch 1 to Member Switch 0 on page 29](#)
- [Verification on page 29](#)

Requirements

This example uses the following hardware and software components:

- Two XRE200 External Routing Engines
- One EX8208 switch
- One EX8216 switch

Before you begin:

Ensure that the external Routing Engines and the member switches are running the same version of Junos OS Release 10.4 or later. See *Installing Software on an EX Series Switch with a Single Routing Engine (CLI Procedure)*, *Installing Software on an EX Series Switch with Redundant Routing Engines (CLI Procedure)*, or *Upgrading Software on an EX6200*

or *EX8200 Standalone Switch Using Nonstop Software Upgrade (CLI Procedure)* to upgrade software on EX8200 switches.

Overview and Topology

An EX8200 Virtual Chassis connects up to four EX8200 switches into a single virtual device, thereby simplifying network management, configuration, and troubleshooting.

In this example, two EX8200 switches are connected to an XRE200 External Routing Engine to form an EX8200 Virtual Chassis. A backup XRE200 External Routing Engine is also part of the topology and has connections to the master external Routing Engine and both member switches. A Virtual Chassis port (VCP) link is also configured to connect the switches, thereby forming a full mesh topology that utilizes all possible VCP links within the Virtual Chassis topology.

This example shows an EX8200 Virtual Chassis composed of four devices: a master XRE200 External Routing Engine, a backup XRE200 External Routing engine, an EX8208 member switch, and an EX8216 member switch.

[Table 6 on page 26](#) explains the components of the example EX8200 Virtual Chassis topology.

Table 6: Components of the EX8200 Virtual Chassis Topology

Property	Settings
Hardware	<ul style="list-style-type: none"> Two XRE200 External Routing Engines Member EX8208 switch Member EX8216 switch
Member IDs and Roles	<ul style="list-style-type: none"> Master external Routing Engine <ul style="list-style-type: none"> Member ID: 8 Role: routing-engine <p>NOTE: The master external Routing Engine earned the mastership role by having the longest uptime. This example assumes this external Routing Engine was up before the backup external Routing Engine.</p> Backup external Routing Engine <ul style="list-style-type: none"> Member ID: 9 Role: routing-engine Member EX8208 switch <ul style="list-style-type: none"> Member ID: 0 Role: line-card Member EX8216 switch <ul style="list-style-type: none"> Member ID: 1 Role: line-card

Table 6: Components of the EX8200 Virtual Chassis Topology (*continued*)

Property	Settings
Interfaces	<p>The master XRE200 External Routing Engine interfaces:</p> <ul style="list-style-type: none"> • vcp-1/0—Connects to vcp-1/0 on the backup external Routing Engine (member 9). • vcp-1/1—Connects to vcp-0/0 (the MGMT port on the module in slot RE0) on the EX8208 member switch (member 0). • vcp-1/2—Connects to vcp-0/0 (the MGMT port on the module in slot RE0) on the EX8216 member switch (member 1). <p>The backup XRE200 External Routing Engine interfaces:</p> <ul style="list-style-type: none"> • vcp-1/0—Connects to vcp-1/0 on the master external Routing Engine (member 8). • vcp-1/1—Connects to vcp-0/1 (the MGMT port on the module in slot RE1) on the EX8208 member switch (member 0). • vcp-1/2—Connects to vcp-0/1 (the MGMT port on the module in slot RE1) on the EX8216 member switch (member 1). <p>The EX8208 member switch interfaces:</p> <ul style="list-style-type: none"> • vcp-0/0 (the MGMT port on the module in slot RE0)—Connects to vcp-1/1 on the master external Routing Engine (member 8). • vcp-0/1 (the MGMT port on the module in slot RE1)—Connects to vcp-1/1 on the backup external Routing Engine (member 9). • xe-3/0/2—Connects to xe-4/0/2 on the EX8216 member switch (member 1). <p>The EX8216 member switch interfaces:</p> <ul style="list-style-type: none"> • vcp-0/0 (the MGMT port on the module in slot RE0)—Connects to vcp-1/2 on the master external Routing Engine (member 8). • vcp-0/1 (the MGMT port on the module in slot RE1)—Connects to vcp-1/2 on the backup external Routing Engine (member 9). • xe-4/0/2—Connects to xe-3/0/2 on the EX8208 member switch (member 0).

This example assumes that all devices in the Virtual Chassis are running the same version of Junos OS Release 10.4 or later. If you need to upgrade a device in the Virtual Chassis configuration to meet this requirement, perform that upgrade before performing this procedure.

Setting the EX8200 Switches as Virtual Chassis Members

Step-by-Step Procedure	<p>To configure the EX8200 switches as Virtual Chassis members:</p> <p>On the EX8208 switch:</p> <pre>user@switch> set chassis virtual-chassis</pre> <p>On the EX8216 switch:</p> <pre>user@switch> set chassis virtual-chassis</pre>
-------------------------------	---

Preprovisioning the Virtual Chassis

CLI Quick Configuration To quickly preprovision the Virtual Chassis, log in to the master external Routing Engine and enter the following commands:

```
[edit]
set virtual-chassis preprovisioned
set virtual-chassis member 0 serial-number ghi789 role line-card
set virtual-chassis member 1 serial-number jkl112 role line-card
set virtual-chassis member 8 serial-number abc123 role routing-engine
set virtual-chassis member 9 serial-number def456 role routing-engine
```

Step-by-Step Procedure To preprovision the EX8200 Virtual Chassis:

1. Log in to the master external Routing Engine and specify the use of a preprovisioned configuration:



NOTE: You must use a preprovisioned configuration to configure an EX8200 Virtual Chassis. Nonprovisioned configuration is not supported.

```
[edit virtual-chassis]
user@external-routing-engine# set preprovisioned
```

2. Specify all members that you want to include in the Virtual Chassis configuration, listing each switch's or external Routing Engine's serial number with its desired member ID and role:



NOTE: You must use the backplane serial number for EX8208 switches, and the midplane serial number for EX8216 switches.

```
[edit virtual-chassis]
user@external-routing-engine# set member 0 serial-number ghi789 role line-card
user@external-routing-engine# set member 1 serial-number jkl112 role line-card
user@external-routing-engine# set member 8 serial-number abc123 role
routing-engine
user@external-routing-engine# set member 9 serial-number def456 role
routing-engine
```

Results

Check the results of the configuration:

```
[edit virtual-chassis]
user@external-routing-engine# show
preprovisioned;
member 0 {
  role line-card;
  serial-number ghi789;
}
member 1 {
```

```

    role line-card;
    serial-number jkl112;
  }
  member 8 {
    role routing-engine;
    serial-number abc123;
  }
  member 9 {
    role routing-engine;
    serial-number def456;
  }
}

```

Configuring a VCP Link from Member Switch 0 to Member Switch 1

Step-by-Step Procedure To configure the VCP link between the member switches, log in to the EX8208 member switch (member ID 0) and enter



NOTE: For information on which ports on EX8200 line cards can be configured as VCPs, see *Line Card Model and Version Compatibility in an EX8200 Switch*.

```

user@external-routing-engine> request virtual-chassis vc-port set fpc-slot 4 pic-slot 0 port
2 member 1

```

Configuring a VCP Link from Member Switch 1 to Member Switch 0

Step-by-Step Procedure To configure the VCP link between the member switches, log in to the EX8216 member switch (member ID 1) and enter:



NOTE: For information on which ports on EX8200 line cards can be configured as VCPs, see *Line Card Model and Version Compatibility in an EX8200 Switch*.

```

user@switch> request virtual-chassis vc-port set fpc-slot 3 pic-slot 0 port 2 member 0

```

Verification

To verify that the Virtual Chassis is operational, perform this task:

- [Verifying That the Virtual Chassis Is Operational on page 29](#)

Verifying That the Virtual Chassis Is Operational

Purpose Verify that the Virtual Chassis is operational and that all VCP links in the Virtual Chassis are functioning properly.

Action Enter the **show virtual-chassis** command:

```

user@external-routing-engine> show virtual-chassis

```

```

Preprovisioned Virtual Chassis
Virtual Chassis ID: c806.0842.de51

```

List Member ID	Status	Serial No	Model	Mastership		Neighbor	
				priority	Role	ID	Interface
0 (FPC 0-15)	Prsnt	ghi789	ex8208		0 Linecard	8	vcp-0/0
						9	vcp-0/1
1 (FPC 16-31)	Prsnt	jkl112	ex8216		0 Linecard	1	vcp-3/0/2
						8	vcp-0/0
						9	vcp-0/1
8 (FPC 128-143)	Prsnt	abc123	ex-xre	129	Master*	0	vcp-4/0/2
						9	vcp-1/0
						0	vcp-1/1
9 (FPC 144-159)	Prsnt	def456	ex-xre	129	Backup	1	vcp-1/2
						8	vcp-1/0
						0	vcp-1/1
						1	vcp-1/2

Meaning This output verifies that a Virtual Chassis composed of one EX8208 switch (member 0), one EX8216 switch (member 1), and master and backup XRE200 External Routing Engines (members 8 and 9) is operational. The display shows the VCP connections and confirms that each member has a connection to all three other members to form a full-mesh EX8200 Virtual Chassis.

Related Documentation

- [Configuring an EX8200 Virtual Chassis \(CLI Procedure\) on page 32](#)
- [Setting a 10-Gigabit Ethernet Port as a Virtual Chassis Port in an EX8200 Virtual Chassis \(CLI Procedure\) on page 36](#)

CHAPTER 3

Configuration Tasks

- [Configuring an EX8200 Virtual Chassis \(CLI Procedure\) on page 32](#)
- [Adding or Replacing a Member Switch or an External Routing Engine in an EX8200 Virtual Chassis \(CLI Procedure\) on page 34](#)
- [Setting a 10-Gigabit Ethernet Port as a Virtual Chassis Port in an EX8200 Virtual Chassis \(CLI Procedure\) on page 36](#)
- [Configuring a Long-Distance Virtual Chassis Port Connection for an EX8200 Virtual Chassis on page 37](#)
- [Installing Software for All Devices in an EX8200 Virtual Chassis on page 40](#)
- [Installing Software for a Single Device in an EX8200 Virtual Chassis on page 42](#)
- [Performing a Software Recovery Installation for an XRE200 External Routing Engine on page 43](#)
- [Using Groups to Configure Power Supply Management, Power Budgets, and LCD Menus on EX8200 Member Switches in an EX8200 Virtual Chassis \(CLI Procedure\) on page 45](#)

Configuring an EX8200 Virtual Chassis (CLI Procedure)

You can configure an EX8200 Virtual Chassis to include up to four EX8200 switches and one or two XRE200 External Routing Engines. You interconnect the member switches by connecting them to the external Routing Engines, whose ports automatically function as Virtual Chassis ports (VCPs). A VCP is any port whose function is to send and receive Virtual Chassis Control Protocol (VCCP) traffic to create, monitor, and maintain the Virtual Chassis. VCPs also carry data traffic through the Virtual Chassis.

If you want to add paths for VCCP traffic within an EX8200 Virtual Chassis, you can also configure VCP links between member switches. See [“Setting a 10-Gigabit Ethernet Port as a Virtual Chassis Port in an EX8200 Virtual Chassis \(CLI Procedure\)”](#) on page 36.

To configure an EX8200 Virtual Chassis:

1. Log into each EX8200 member switch and revert the switch to the factory default configuration:

```
user@switch# load factory-default
```

See [Reverting to the Default Factory Configuration for the EX Series Switch](#) for additional information on this process.
2. Configure a password for the EX8200 switch. See [Connecting and Configuring an EX Series Switch \(CLI Procedure\)](#).



BEST PRACTICE: You should always access the EX8200 Virtual Chassis through the master XRE200 External Routing Engine. However, configuring the password ensures that you can access the EX8200 member switch for purposes such as local monitoring or debugging.

3. Set each EX8200 switch as a Virtual Chassis member:

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

Follow the CLI prompts. You must reboot the switch to complete this procedure.
4. Upgrade the software on all switches and external Routing Engines in the Virtual Chassis to the same Junos OS release. All switches and external Routing Engines must be running the same version of Junos OS before being connected into a Virtual Chassis. See [Installing Software on an EX Series Switch with a Single Routing Engine \(CLI Procedure\)](#), [Installing Software on an EX Series Switch with Redundant Routing Engines \(CLI Procedure\)](#), or [Upgrading Software on an EX6200 or EX8200 Standalone Switch Using Nonstop Software Upgrade \(CLI Procedure\)](#).
5. Make a list of the serial numbers of all switches and external Routing Engines to be connected in the Virtual Chassis configuration. For EX8208 switches, use the backplane serial number. For EX8216 switches, use the midplane serial number. See [Locating the Serial Number on an EX8200 Switch or Component](#) and [Locating the Serial Number on an XRE200 External Routing Engine or Component](#).
6. If both external Routing Engines are already powered on, skip this step.

Power on one of the external Routing Engines. The external Routing Engine with the most uptime is assigned the master role, so the external Routing Engine that is powered on in this step is the master external Routing Engine.

7. Log in to the master external Routing Engine and specify the preprovisioned configuration mode. If the external Routing Engine mastership has not been determined, log in to the external Routing Engine that has been powered on the longest.



NOTE: You must use a preprovisioned configuration to configure an EX8200 Virtual Chassis. Nonprovisioned configuration is not supported.

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

```
user@external-routing-engine# set preprovisioned
```

8. Specify all members for the Virtual Chassis, listing each switch's or external Routing Engine's serial number, member ID, and role:

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

```
user@external-routing-engine# set member member-id serial-number serial-number role
role
```

For example:

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

```
user@external-routing-engine# set member 0 serial-number abc123 role line-card
```

```
user@external-routing-engine# set member 1 serial-number def456 role line-card
```

```
user@external-routing-engine# set member 2 serial-number ghi789 role line-card
```

```
user@external-routing-engine# set member 3 serial-number jkl101 role line-card
```

```
user@external-routing-engine# set member 8 serial-number yza567 role routing-engine
```

```
user@external-routing-engine# set member 9 serial-number bcd890 role routing-engine
```



NOTE: An EX8200 Virtual Chassis does not function properly when an external Routing Engine is configured to use a member ID other than 8 or 9. When the EX8200 Virtual Chassis configuration is not preprovisioned, the external Routing Engine assumes a member ID that is not 8 or 9. The EX8200 Virtual Chassis cannot be formed until this member ID configuration is changed using this procedure.

9. Interconnect the member switches and external Routing Engines. See *Connecting an EX8200 Switch to an XRE200 External Routing Engine*, *Interconnecting XRE200 External Routing Engines*, and ["Setting a 10-Gigabit Ethernet Port as a Virtual Chassis Port in an EX8200 Virtual Chassis \(CLI Procedure\)"](#) on page 36.

10. Commit the configuration:

```
[edit]
```

```
user@external-routing-engine# commit synchronize
```

11. Power on the backup external Routing Engine if it is not already powered on.

Related Documentation

- [Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines](#) on page 25
- [Adding or Replacing a Member Switch or an External Routing Engine in an EX8200 Virtual Chassis \(CLI Procedure\)](#) on page 34

- [Verifying the Member ID, Role, and Neighbor Member Connections of an EX8200 Virtual Chassis Member on page 69](#)

Adding or Replacing a Member Switch or an External Routing Engine in an EX8200 Virtual Chassis (CLI Procedure)

You can configure an EX8200 Virtual Chassis to include up to four EX8200 switches and one or two XRE200 External Routing Engines. You can add or replace a member switch or an external Routing Engine if one needs to be replaced for any reason.

To add or replace a member switch or external Routing Engine in an EX8200 Virtual Chassis:

1. If you are replacing an existing switch or external Routing Engine, power off the device. See *Powering Off an EX8200 Switch* or *Powering Off an XRE200 External Routing Engine*.
2. Uncable the device that you will remove from the Virtual Chassis.
3. Power on the new switch or external Routing Engine. See *Powering On an EX8200 Switch* or *Powering On an XRE200 External Routing Engine*.
4. Note and record the serial number of the new switch or external Routing Engine. See *Locating the Serial Number on an EX8200 Switch or Component* and *Locating the Serial Number on an XRE200 External Routing Engine or Component*.
5. Upgrade the software on the new switch or external Routing Engine to the Junos OS release currently running in the Virtual Chassis, unless the switch or external Routing Engine is already running that release. See *Installing Software on an EX Series Switch with a Single Routing Engine (CLI Procedure)*, *Installing Software on an EX Series Switch with Redundant Routing Engines (CLI Procedure)*, or *Upgrading Software on an EX6200 or EX8200 Standalone Switch Using Nonstop Software Upgrade (CLI Procedure)*.



NOTE: If you are adding an EX8200 switch to the Virtual Chassis, reboot the switch to complete the software upgrade after completing step 6.

If you are adding an XRE200 External Routing Engine to the Virtual Chassis, reboot the device to complete the software upgrade.

6. (EX8200 switch only) Set the EX8200 switch as a Virtual Chassis member:

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

Follow the CLI prompts. You must reboot the switch to complete this step of the procedure.

7. Log in to the master external Routing Engine on the Virtual Chassis.

8. Delete the device that you uncabled from the Virtual Chassis configuration:

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

```
user@external-routing-engine# delete member member-id serial-number serial-number  
role role
```

For example:

- To delete a switch from the Virtual Chassis configuration:

```
[edit virtual-chassis]
user@external-routing-engine# delete member 1 serial-number abc123 role line-card
```

- To delete an external Routing Engine from the Virtual Chassis configuration:

```
[edit virtual-chassis]
user@external-routing-engine# delete member 8 serial-number yza567 role
routing-engine
```

9. Add the new device to the Virtual Chassis configuration:

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

For example:

- To add a switch to the Virtual Chassis configuration:

```
[edit virtual-chassis]
user@external-routing-engine# set member 1 serial-number lmn910 role line-card
```

- To add an external Routing Engine to the Virtual Chassis configuration:

```
[edit virtual-chassis]
user@external-routing-engine# set member 8 serial-number opq112 role routing-engine
```

10. Commit the configuration:

```
[edit]
user@external-routing-engine# commit synchronize
```

11. Connect the new device to the EX8200 Virtual Chassis. See *Connecting an EX8200 Switch to an XRE200 External Routing Engine*.

Related Documentation

- [Configuring an EX8200 Virtual Chassis \(CLI Procedure\) on page 32](#)
- [Verifying the Member ID, Role, and Neighbor Member Connections of an EX8200 Virtual Chassis Member on page 69](#)

Setting a 10-Gigabit Ethernet Port as a Virtual Chassis Port in an EX8200 Virtual Chassis (CLI Procedure)

You can connect member switches together in an EX8200 Virtual Chassis by configuring 10-Gigabit Ethernet ports as Virtual Chassis ports (VCPs). Member switch to member switch VCPs provide additional paths within the Virtual Chassis for Virtual Chassis Control Protocol (VCCP) traffic to monitor, maintain, or carry data traffic through the Virtual Chassis. See [“Understanding Virtual Chassis Ports in an EX8200 Virtual Chassis” on page 13](#)



NOTE: For information on which ports on EX8200 line cards can be configured as VCPs, see [Line Card Model and Version Compatibility in an EX8200 Switch](#).

All connections between EX8200 member switches and XRE200 External Routing Engines are automatically configured as dedicated VCP links, so you do not need to perform this procedure on those connections. See [“Understanding Virtual Chassis Ports in an EX8200 Virtual Chassis” on page 13](#).

VCP links between EX8200 member switches automatically form a link aggregation group (LAG). Up to 12 VCP links can be part of a LAG of VCP links between EX8200 member switches. No user configuration is required to create this LAG. See [“Understanding Virtual Chassis Port Link Aggregation in an EX8200 Virtual Chassis” on page 15](#).

You must use this procedure to configure VCPs between member switches in an EX8200 Virtual Chassis.

To configure a VCP link between two EX8200 switches:

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

When you use this command to create the VCP link, a paired 10-Gigabit Ethernet interface is also automatically configured as a VCP. For instance, when you enter **request virtual-chassis vc-port set fpc-slot 5 pic-slot 0 port 0** to configure interface **xe-5/0/0** as a VCP, interface **xe-5/0/1** is also configured as a VCP and automatically becomes part of the LAG.

To unconfigure a VCP link between two EX8200 switches and make it a Gigabit Ethernet network link:

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

Related Documentation

- [Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines on page 25](#)
- [Configuring an EX8200 Virtual Chassis \(CLI Procedure\) on page 32](#)
- [Verifying the Member ID, Role, and Neighbor Member Connections of an EX8200 Virtual Chassis Member on page 69](#)

Configuring a Long-Distance Virtual Chassis Port Connection for an EX8200 Virtual Chassis

This procedure describes how to connect XRE200 External Routing Engine to EX8200 switch connections over long distances in an EX8200 Virtual Chassis only.

You can connect XRE200 External Routing Engines together over distances up to 43.5 miles (70 km) within an EX8200 Virtual Chassis using the 1000BASE-X SFP Virtual Chassis Control Interface (VCCI) module ports. No software configuration is needed to make these connections.

You can connect EX8200 switches together over longer distances within an EX8200 Virtual Chassis by configuring a 10-Gigabit Ethernet port as a Virtual Chassis port (VCP). See [“Setting a 10-Gigabit Ethernet Port as a Virtual Chassis Port in an EX8200 Virtual Chassis \(CLI Procedure\)”](#) on page 36.

You can connect an XRE200 External Routing Engine to an EX8200 switch to form a Virtual Chassis over a longer distance by using an intermediary switch. The intermediary switch should be located within 328 feet (100 meters) of the EX8200 member switch to ensure a Gigabit Ethernet connection to the management port (labeled **MGMT**) can be made. The intermediary switch is required because an SFP connection from an external Routing Engine cannot directly connect to the management port of an EX8200 switch. Virtual Chassis Control Protocol (VCCP) traffic traversing the intermediary switches must be placed into VLANs to ensure it is directed to the correct destination.

This procedure shows how to connect two XRE200 External Routing Engines to two EX8200 member switches together over a long-distance connection.

Ensure that you have the following parts and tools available:

- Two XRE200 External Routing Engines, each with at least one installed 1000BASE-X SFP VCCI module
- Two EX8200 switches
- Two EX Series switches



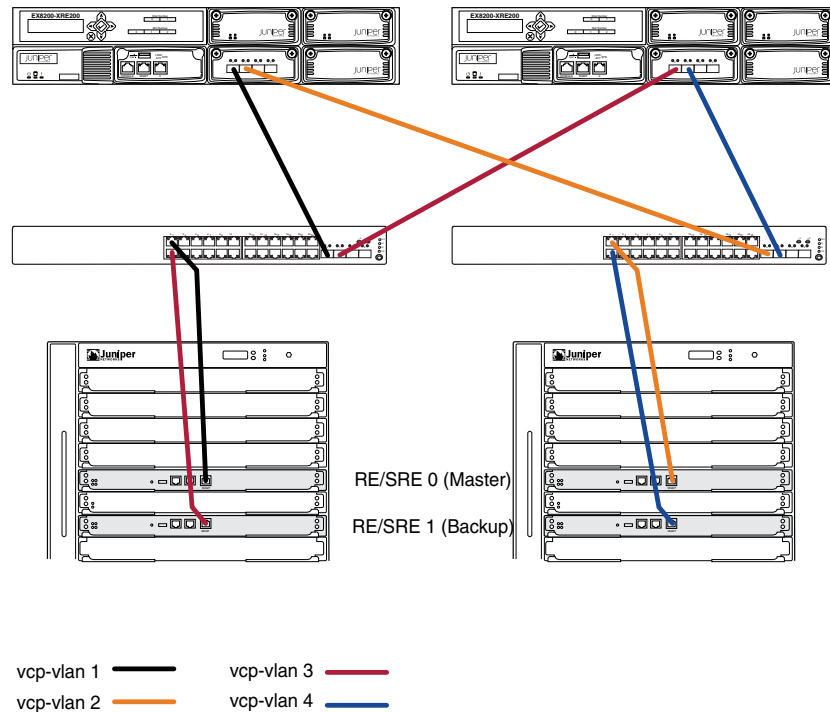
NOTE: Any EX Series switch can be used as an intermediary switch in an EX8200 Virtual Chassis. The most affordable EX Series switch is an EX2200 switch with 24 total ports and no Power over Ethernet (PoE) ports (model number EX2200-24T-4G). If you need to purchase a new switch to extend a VCP for an EX8200 Virtual Chassis, we recommend purchasing this switch with an SFP uplink module that supports the same SFPs as an XRE200 External Routing Engine and an SFP cable whose length is appropriate for your environment. The EX2200 switch supports uplink modules with SFP connections that span up to 43.5 miles (70 km).

See *Pluggable Transceivers Supported on EX2200 Switches* and *Pluggable Transceivers Supported on XRE200 External Routing Engines* for information on supported optical uplinks.

The EX Series switches used as intermediary switches in this procedure must be dedicated to this task. These switches cannot receive or forward any other network traffic.

Figure 6 on page 38 provides a diagram of the XRE200 External Routing Engine-to-EX8200 switch long-distance connections in the EX8200 Virtual Chassis. The diagram shows the physical and VLAN setup of the Virtual Chassis.

Figure 6: Long Distance EX8200 Virtual Chassis



g021230

To configure a long-distance VCP using two intermediary EX Series switches in an EX8200 Virtual Chassis:

1. Cable the devices. The following connections must be cabled:

VLAN 1:

- SFP VCP on the master XRE200 External Routing Engine connected to intermediary switch 0.
- Gigabit Ethernet port on intermediary switch 0 to the management port (labeled **MGMT**) on the master Switch Fabric and Routing Engine (SRE) module or Routing Engine (RE) module on the EX8200 member switch ID 0.

VLAN 2:

- SFP VCP on the master XRE200 External Routing Engine connected to intermediary switch 1.
- Gigabit Ethernet port on intermediary switch 1 to the management port (labeled **MGMT**) on the master SRE module or RE module on the EX8200 member switch ID 1.

VLAN 3:

- SFP VCP on the backup XRE200 External Routing Engine connected to intermediary switch 0.
- Gigabit Ethernet port on intermediary switch 0 to the management port (labeled **MGMT**) on the backup SRE module or RE module on the EX8200 member switch ID 0.

VLAN 4:

- SFP VCP on the backup XRE200 External Routing Engine connected to intermediary switch 1.
 - Gigabit Ethernet port on intermediary switch 1 to the management port (labeled **MGMT**) on the backup SRE module or RE module on the EX8200 member switch ID 1.
2. Configure the VLANs on the intermediary switches. Each VLAN must contain the interface on the intermediary switch connecting to the XRE200 External Routing Engine and the interface connecting to the EX8200 member switch.



NOTE: You only need to configure the VLANs on the intermediary switches. You do not need to place any interfaces on the XRE200 External Routing Engine or on the EX8200 member switches into the VLANs.

VLAN 1:

```
user@switch-0# set vlans vcp-vlan-1 interface ge-0/1/0
user@switch-0# set vlans vcp-vlan-1 interface ge-0/0/0
```

VLAN 2:

```
user@switch-1# set vlans vcp-vlan-2 interface ge-0/1/0
user@switch-1# set vlans vcp-vlan-2 interface ge-0/0/0
```

VLAN 3:

```
user@switch-0# set vlans vcp-vlan-3 interface ge-0/1/1
user@switch-0# set vlans vcp-vlan-3 interface ge-0/0/1
```

VLAN 4:

```
user@switch-1# set vlans vcp-vlan-4 interface ge-0/1/1
user@switch-1# set vlans vcp-vlan-4 interface ge-0/0/1
```

3. Set the maximum transmission units (MTU) value on all interfaces on both intermediary switches transmitting VCCP traffic to 3000 bytes or more.

On switch 0:

```
user@switch-0# set interfaces ge-0/0/0 mtu 3000
user@switch-0# set interfaces ge-0/0/1 mtu 3000
user@switch-0# set interfaces ge-0/1/0 mtu 3000
user@switch-0# set interfaces ge-0/1/1 mtu 3000
```

On switch 1:

```
user@switch-1# set interfaces ge-0/0/0 mtu 3000
user@switch-1# set interfaces ge-0/0/1 mtu 3000
user@switch-1# set interfaces ge-0/1/0 mtu 3000
```

```
user@switch-1# set interfaces ge-0/1/1 mtu 3000
```

Related Documentation

- [Understanding Long-Distance Virtual Chassis Port Configuration for an EX8200 Virtual Chassis on page 18](#)
- *Virtual Chassis Control Interface (VCCI) Modules in an XRE200 External Routing Engine*

Installing Software for All Devices in an EX8200 Virtual Chassis

We recommend that you use nonstop software upgrade (NSSU) to upgrade the software for an EX8200 Virtual Chassis. NSSU upgrades all Routing Engines in the Virtual Chassis with minimal disruption of traffic. For more details, see *Upgrading Software on an EX8200 Virtual Chassis Using Nonstop Software Upgrade (CLI Procedure)*.

Use the procedure described in this topic to upgrade the software for the EX8200 Virtual Chassis when you cannot use NSSU—for example, when the software version you are running does not support NSSU.

You can use this procedure to upgrade the software on all devices in the EX8200 Virtual Chassis, including all EX8200 member switches and both the master and backup external Routing Engines. This procedure requires a reboot of the Virtual Chassis.

To upgrade software for all devices in the EX8200 Virtual Chassis:

1. Download the software package for the XRE200 External Routing Engine (the software package whose package name contains “xre200”—for instance, **jinstall-ex-xre200-10.4R1.8-domestic-signed.tgz**—as described in *Downloading Software Packages from Juniper Networks*.
2. (Optional) Back up the current software configuration to a second storage option. See the *Junos OS Installation and Upgrade Guide*.
3. (Optional) Copy the software package to the master external Routing Engine. We recommend that you use FTP to copy the file to the **/var/tmp** directory.

This step is optional because you can upgrade Junos OS when the software image is stored at a remote location. This procedure describes the software upgrade process for both scenarios.

4. Install the new package:



NOTE: A reboot, which occurs as part of the execution of the following command, is required on each external Routing Engine or switch to complete the software upgrade. If you use this procedure as outlined, all Switch Fabric and Routing Engine (SRE) modules or Routing Engine (RE) modules on the EX8200 switches that are connected into the EX8200 Virtual Chassis will reboot simultaneously when the command is entered. Network traffic cannot be forwarded during the reboot.

If you want to reboot a device at a later time or reboot the SRE modules or RE modules individually, do not use the `reboot` option at this point of the procedure. Enter the `request system reboot` command at a later time to reboot the external Routing Engine or SRE module or RE module to complete the procedure.

```
user@switch> request system software add source reboot
```

Replace *source* with one of the following paths:

- For a software package that is installed from a local directory on the external Routing Engine—*/pathname/package-name-m.nZx-distribution.tgz*.
- For a software package that is downloaded and installed from a remote location:
 - *ftp://hostname/pathname/package-name-m.nZx-distribution.tgz*
 - *http://hostname/pathname/package-name-m.nZx-distribution.tgz*

where *package-name-m.nZx-distribution.tgz* is, for example, *jinstall-ex-xre200-10.4R1.8-domestic-signed.tgz*.

Related Documentation

- [Installing Software for a Single Device in an EX8200 Virtual Chassis on page 42](#)
- [Upgrading Software on an EX8200 Virtual Chassis Using Nonstop Software Upgrade \(CLI Procedure\)](#)
- [Understanding Software Upgrades in an EX8200 Virtual Chassis on page 14](#)
- [Understanding Nonstop Software Upgrade on EX Series Switches](#)

Installing Software for a Single Device in an EX8200 Virtual Chassis

You can use this procedure to upgrade the software on a single device in an EX8200 Virtual Chassis. This procedure is most useful when one device that was previously recognized as active in the Virtual Chassis is no longer recognized as active and the reason for that lack of recognition is likely related to a software version mismatch between this device and the other Virtual Chassis members.

Generally, the preferred method of upgrading a Virtual Chassis is to upgrade all devices in the Virtual Chassis at the same time to ensure all members are running the same version of Junos OS. See [“Installing Software for All Devices in an EX8200 Virtual Chassis” on page 40](#)

If you are adding a single device to an existing Virtual Chassis, upgrade that device to the version of Junos OS currently running in the Virtual Chassis before you connect the device. The following procedure cannot be used in that scenario, as the Virtual Chassis will not recognize the new device until the software mismatch is corrected. See [“Adding or Replacing a Member Switch or an External Routing Engine in an EX8200 Virtual Chassis \(CLI Procedure\)” on page 34](#).

To upgrade software for a single device in an EX8200 Virtual Chassis:

1. Download the software package as described in *Downloading Software Packages from Juniper Networks*. The software package in this procedure is always the XRE200 software package, even if you are upgrading the software on a member switch.
2. (Optional) Back up the current software configuration to a second storage option. See the *Junos OS Installation and Upgrade Guide*.
3. (Optional) Copy the software package to the master external Routing Engine. We recommend that you use FTP to copy the file to the `/var/tmp` directory.

This step is optional because you can also upgrade Junos OS when the software image is stored at a remote location. This procedure describes the software upgrade process for both scenarios.

4. Install the new package:



NOTE: A reboot, which occurs as part of the execution of the following command, is required to complete the software upgrade. If you want to reboot the external Routing Engine or switch at a later time, do not use the `reboot` option at this point of the procedure and enter the `request system reboot` command at a later time to reboot the device.

```
user@switch> request system software add source member member-id reboot
```

Replace **source** with one of the following paths:

- For a software package that is installed from a local directory on the external Routing Engine—`/pathname/package-name-m.nZx-distribution.tgz`.

- For a software package that is downloaded and installed from a remote location:
 - `ftp://hostname/pathname/package-name-m.nZx-distribution.tgz`
 - `http://hostname/pathname/package-name-m.nZx-distribution.tgz`

where `package-name-m.nZx-distribution.tgz` is, for example, `jinstall-ex-xre200-10.4R1.8-domestic-signed.tgz`.

Replace *member-id* with the member ID of the device receiving the upgrade.

Related Documentation • [Understanding Software Upgrades in an EX8200 Virtual Chassis on page 14](#)

Performing a Software Recovery Installation for an XRE200 External Routing Engine

This process can be used in any scenario when a software recovery upgrade of an XRE200 External Routing Engine is needed. A software recovery upgrade might be needed in several scenarios, including cases when the CLI prompt is unreachable or the external Routing Engine is continually rebooting and other troubleshooting methods have failed to address the problem.

This procedure copies the Junos OS image on USB flash memory onto an XRE200 External Routing Engine. The external Routing Engine then boots using that Junos OS image.

Ensure that you have the following tools and parts available:

- A USB flash drive that meets the EX Series switch USB port specifications. See *USB Port Specifications for an XRE200 External Routing Engine*.
- A computer or other device that you can use to download the software package from the Internet and copy it to the USB flash drive.

To perform a software recovery installation:

1. Download the Junos OS package that you would like to place onto the external Routing Engine from the Internet onto the USB flash drive by using your computer or other device. See *Downloading Software Packages from Juniper Networks*.



NOTE: In most cases, you should download the Junos OS image running on the other devices in your EX8200 Virtual Chassis onto USB flash memory. Booting the external Routing Engine with the same software currently running within the EX8200 Virtual Chassis ensures that the external Routing Engine will be recognized by the other Virtual Chassis member devices after the recovery upgrade.

2. Remove the USB flash drive from the computer or other device.
3. Insert the USB flash drive into the USB port on the external Routing Engine.
4. Power off the external Routing Engine. See *Powering Off an XRE200 External Routing Engine*.

5. Reconnect power to the external Routing Engine: When you see the following prompt:

```
Hit [Enter] to boot Immediately or [R] key to start recovery ...
press the R key to select the recovery software upgrade procedure.
```

6. Start the recovery: When the following prompt appears on your external Routing Engine:

```
Select one of the following options ...
1. Reboot the system with out making any changes.
2. Start Recovery.
Choice :
```

press the the 2 key.



NOTE: If you have loaded multiple software recovery packages onto the USB flash drive, you are prompted to select which package to use for the recovery. See [request system software recovery-package](#) for additional information.

Press the 2 key.

7. Monitor the output:

```
Starting Recovery Process ...
(user@switch, Wed Aug 18 16:30:02 IST 2010)
Loading /boot/defaults/loader.conf
/kernel text=0x4a6878 data=0x354e0+0x5edf4 syms=[0x4+0x601a0 +0x4+0x8a749
/
Hit [Enter] to boot immediately, or space bar for command prompt.
OK boot
GDB: debug ports: uart
GDB: current port: uart
KDB: debugger backends: ddb gdb
KDB: current backend: ddb
262144K of memory above 4GB ignored
Copyright (c) 1996-2010, Juniper Networks, Inc.
All rights reserved.
Copyright (c) 1992-2006 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994
The Regents of the University of California. All rights reserved.
JUNOS 10.4I #0: 2010-07-20 04:46:37 UTC
user@switch:/b/user/current/XRE-200/obj-i386/bsd/sys/compile/INSTALL-EXInstall
EXInstall File: /tmp/cf/packages/jinstall-ex-xre200-10.4R1.5-domestic-signed.tgz
Rebooting to complete the installation. Please wait...
```

The external Routing Engine reboots to complete the procedure.

8. When you see the CLI prompt, remove the USB flash memory card from the USB port on the external Routing Engine.

Related Documentation

- [Understanding Software Upgrades in an EX8200 Virtual Chassis on page 14](#)

Using Groups to Configure Power Supply Management, Power Budgets, and LCD Menus on EX8200 Member Switches in an EX8200 Virtual Chassis (CLI Procedure)

You configure an EX8200 Virtual Chassis from the master external Routing Engine. The configuration file that is saved on the master external Routing Engine is also saved to the backup external Routing Engine as well as to each Routing Engine or each Switch Fabric and Routing Engine on all EX8200 member switches in the Virtual Chassis.

For most features in an EX8200 Virtual Chassis, this file saving architecture allows you to configure features for the entire Virtual Chassis from a single interface.

For managing the power supplies or the LCD panel interfaces for specific member switches in the Virtual Chassis, however, you cannot use this file saving architecture to precisely set the configuration parameters. You must instead use groups to configure power supply management or the LCD interface for individual member switches in an EX8200 Virtual Chassis.

An EX8200 Virtual Chassis has a default group for each member switch that allows you to configure LCD interfaces or power supply management. The group is named **membermember-number**, where **member-number** is the member ID number of the switch. For instance, you configure features in the group named **member1** to configure power supply management or the LCD interface for the switch assigned member ID 1 in the Virtual Chassis.

You configure groups using the CLI interface on the master external Routing Engine. You also have to apply the configured groups from master external Routing Engine CLI on the master external Routing Engines for the group configuration to be applied to the member switch.



NOTE: You cannot use groups to configure member switch interfaces in an EX8200 Virtual Chassis.

To use groups to manage the power supply configuration or to disable a menu on the LCD interface for a specific member switch within an EX8200 Virtual Chassis:

1. Log into the master external Routing Engine.
2. Use the group configuration to disable a menu on the LCD interface:

```
[edit]
user@external-routing-engine# set groups membermember-number chassis lcd-menu
menu-item maintenance-menu disable
```

For example, to disable the maintenance menu on the LCD interface on member switch 1:

```
[edit]
user@external-routing-engine# set groups member1 chassis lcd-menu menu-item
maintenance-menu disable
```

3. Use the group configuration to enable power supply redundancy:

```
[edit]
```

```
user@external-routing-engine# set groups member member-number chassis psu redundancy  
n-plus-n
```

For example, to enable power supply redundancy on member switch 1:

```
[edit]  
user@external-routing-engine# set groups member1 chassis psu redundancy n-plus-n
```

4. Use the group configuration to assign a power budget priority for a line card on a member switch:

```
[edit]  
user@external-routing-engine# set groups member member-number chassis fpc fpc-number  
power-budget-priority power-budget-priority
```

For example, to assign a power budget priority of 7 to the line card in slot 6 of the EX8200 member switch assigned as member ID 1:

```
[edit]  
user@external-routing-engine# set groups member1 chassis fpc 6 power-budget-priority  
7
```

5. Apply the group to the EX8200 Virtual Chassis configuration:

```
[edit]  
user@external-routing-engine# set apply-groups member member-number
```

For instance, to apply the group assigned to member 1 to the configuration:

```
[edit]  
user@external-routing-engine# set apply-groups member1
```

Related Documentation

- [Understanding Configuration File Management in an EX8200 Virtual Chassis on page 19](#)

CHAPTER 4

Configuration Statements

- [\[edit virtual-chassis\] Configuration Statement Hierarchy on page 47](#)
- [graceful-restart \(Enabling Globally\) on page 50](#)
- [id on page 51](#)
- [location \(Virtual Chassis\) on page 52](#)
- [mac-persistence-timer on page 53](#)
- [member on page 54](#)
- [no-split-detection on page 55](#)
- [preprovisioned on page 56](#)
- [role on page 57](#)
- [serial-number on page 60](#)
- [traceoptions \(Virtual Chassis\) on page 61](#)
- [virtual-chassis on page 64](#)

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

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

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

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

This topic lists:

- [Supported Statements in the \[edit virtual-chassis\] Hierarchy Level on page 48](#)
- [Unsupported Statements in the \[edit virtual-chassis\] Hierarchy Level on page 48](#)

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

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

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

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

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

- Related Documentation**
- *Preprovisioning a Virtual Chassis Fabric*
 - *Autoprovisioning a Virtual Chassis Fabric*
 - *Adding a Device to a Virtual Chassis Fabric*

- *Configuring a QFX Series Virtual Chassis (CLI Procedure)*
- *Configuring an EX4300 Virtual Chassis (CLI Procedure)*
- *Configuring an EX2200 Virtual Chassis (CLI Procedure)*
- *Configuring an EX3300 Virtual Chassis (CLI Procedure)*
- *Configuring an EX4200, EX4500, or EX4550 Virtual Chassis (CLI Procedure)*
- *Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches (CLI Procedure)*
- [Configuring an EX8200 Virtual Chassis \(CLI Procedure\) on page 32](#)

graceful-restart (Enabling Globally)

Syntax	<pre> graceful-restart { disable; helper-disable; maximum-helper-recovery-time <i>seconds</i>; maximum-helper-restart-time <i>seconds</i>; notify-duration <i>seconds</i>; recovery-time <i>seconds</i>; restart-duration <i>seconds</i>; stale-routes-time <i>seconds</i>; }</pre>
Hierarchy Level	<pre> [edit logical-systems <i>logical-system-name</i> routing-options], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> routing-options], [edit routing-options], [edit routing-instances <i>routing-instance-name</i> routing-options]</pre>
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.</p>
Description	<p>Configure graceful restart globally to enable the feature. You cannot enable graceful restart for specific protocols unless graceful restart is also enabled globally. You can, optionally, modify the global settings at the individual protocol level.</p>



NOTE:

- For VPNs, the **graceful-restart** statement allows a router whose VPN control plane is undergoing a restart to continue to forward traffic while recovering its state from neighboring routers.
- For BGP, if you configure graceful restart after a BGP session has been established, the BGP session restarts and the peers negotiate graceful restart capabilities.
- LDP sessions flap when **graceful-restart** configurations change.

Default	Graceful restart is disabled by default.
Options	The remaining statements are explained separately.
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • <i>Enabling Graceful Restart</i> • <i>Configuring Routing Protocols Graceful Restart</i>

- *Configuring Graceful Restart for MPLS-Related Protocols*
- *Configuring VPN Graceful Restart*
- *Configuring Logical System Graceful Restart*
- *Graceful Restart Configuration Statements*
- *Configuring Graceful Restart for QFabric Systems*

id

Syntax	<code>id id;</code>
Hierarchy Level	[edit virtual-chassis]
Release Information	Statement introduced in Junos OS Release 9.3 for EX Series switches. Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
Description	Configure the alphanumeric string that identifies a Virtual Chassis or Virtual Chassis Fabric (VCF) configuration.
Options	<i>id</i> —Virtual Chassis ID (VCID), which uses the ISO family address format—for example, 9622.6ac8.5345 .
Required Privilege Level	system—To view this statement in the configuration. system-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> • <i>Example: Assigning the Virtual Chassis ID to Determine Precedence During an EX4200 Virtual Chassis Merge</i> • <i>Assigning the Virtual Chassis ID to Determine Precedence During a Virtual Chassis Merge (CLI Procedure)</i> • <i>Configuring a QFX Series Virtual Chassis (CLI Procedure)</i> • <i>Autoprovisioning a Virtual Chassis Fabric</i> • <i>Preprovisioning a Virtual Chassis Fabric</i> • Configuring an EX8200 Virtual Chassis (CLI Procedure) on page 32 • Understanding Virtual Chassis Member ID Numbering in an EX8200 Virtual Chassis on page 11

location (Virtual Chassis)

Syntax	<code>location location;</code>
Hierarchy Level	[edit virtual-chassis member member-id]
Release Information	<p>Statement introduced in Junos OS Release 11.1 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).</p>
Description	<p>Set a description of the location of the Virtual Chassis or VCF member switch or external Routing Engine.</p> <p>The Location field is visible to users who enter the show virtual-chassis status detail command.</p> <p>Setting this description has no effect on the operation of the member device.</p>
Options	location —Location of the current member switch or external Routing Engine. The location can be any single word.
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 a QFX Series Virtual Chassis (CLI Procedure)</i>• <i>Example: Configuring an EX4200 Virtual Chassis Using a Preprovisioned Configuration File</i>• <i>Example: Configuring a Preprovisioned Mixed EX4200 and EX4500 Virtual Chassis</i>• Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines on page 25• <i>Configuring an EX4200, EX4500, or EX4550 Virtual Chassis (CLI Procedure)</i>• <i>Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches (CLI Procedure)</i>• Configuring an EX8200 Virtual Chassis (CLI Procedure) on page 32

mac-persistence-timer

Syntax	<code>mac-persistence-timer [<i>minutes</i> disable];</code>
Hierarchy Level	[edit virtual-chassis]
Release Information	<p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Option disable introduced in Junos OS Release 12.2 for EX Series switches.</p> <p>The maximum timer limit changed from no maximum timer limit to 60 minutes in Junos OS Release 12.2 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).</p>
Description	<p>Specify how long the Virtual Chassis or VCF continues to use the MAC address of the switch that was originally configured in the master role as the system MAC base address after the original master switch is removed from the Virtual Chassis or VCF. The system MAC base address does not change in the event of a switchover provided the switch originally configured in the master role remains a member of the Virtual Chassis or VCF.</p> <p>The maximum timer limit is 60 minutes starting in Junos OS Release 12.2. There are no minimum or maximum timer limits in prior Junos OS releases.</p>
Default	The MAC persistence timer is set to 10 minutes by default.
Options	<p>minutes—Time in minutes that the member switch in the backup role continues to use the system MAC base address of the old master before using its own system MAC base address after the switch in the master role is physically disconnected or removed from the Virtual Chassis or VCF.</p> <p>disable—Disable the MAC persistence timer. The system MAC base address never changes when the MAC persistence timer is disabled, even when the switch in the master role is physically disconnected or removed from the Virtual Chassis or VCF.</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>Configuring the Timer for the Backup Member to Start Using Its Own MAC Address, as Master of a Virtual Chassis (CLI Procedure)</i> • <i>Autoprovisioning a Virtual Chassis Fabric</i> • <i>Preprovisioning a Virtual Chassis Fabric</i>

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	Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
Description	Configure a switch or an XRE200 External Routing Engine as a member of a Virtual Chassis or a Virtual Chassis Fabric (VCF) with characteristics specified by the available options.
Default	<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	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>Autoprovisioning a Virtual Chassis Fabric</i>• <i>Preprovisioning a Virtual Chassis Fabric</i>• <i>Adding a Device to a Virtual Chassis Fabric</i>• <i>Configuring a QFX Series Virtual Chassis (CLI Procedure)</i>

- *Example: Configuring an EX4200 Virtual Chassis Using a Preprovisioned Configuration File*
- [Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines on page 25](#)
- *Configuring an EX3300 Virtual Chassis (CLI Procedure)*
- *Configuring an EX4200, EX4500, or EX4550 Virtual Chassis (CLI Procedure)*
- [Configuring an EX8200 Virtual Chassis \(CLI Procedure\) on page 32](#)
- *Configuring an EX9200 Virtual Chassis*
- *Configuring a QFX Series Virtual Chassis (CLI Procedure)*

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>
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 (CLI Procedure)</i> • <i>Assigning the Virtual Chassis ID to Determine Precedence During a Virtual Chassis Merge (CLI Procedure)</i> • <i>Disabling Split Detection in a Virtual Chassis Configuration</i> • <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>• Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines on page 25• <i>Configuring an EX4200, EX4500, or EX4550 Virtual Chassis (CLI Procedure)</i>• Configuring an EX8200 Virtual Chassis (CLI Procedure) on page 32• <i>Configuring an EX9200 Virtual Chassis</i>• <i>Configuring a QFX Series Virtual Chassis (CLI Procedure)</i>• <i>Replacing a Member Switch of a Virtual Chassis Configuration (CLI Procedure)</i>

role

Syntax	<code>role (line-card routing-engine);</code>
Hierarchy Level	[edit virtual-chassis preprovisioned member <i>member-id</i>]
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.

Virtual Chassis Fabric

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

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

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

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

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

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

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

EX8200 Virtual Chassis

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

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

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

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

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

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

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

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

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*
- *Example: Configuring an EX4200 Virtual Chassis Using a Preprovisioned Configuration File*
- [Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines on page 25](#)
- *Configuring an EX3300 Virtual Chassis (CLI Procedure)*
- *Configuring an EX4200, EX4500, or EX4550 Virtual Chassis (CLI Procedure)*
- [Configuring an EX8200 Virtual Chassis \(CLI Procedure\) on page 32](#)
- *Configuring an EX9200 Virtual Chassis*
- *Configuring a QFX Series Virtual Chassis (CLI Procedure)*
- *Configuring a Virtual Chassis on an EX Series Switch (J-Web Procedure)*
- *Adding a New EX4200 Switch to an Existing EX4200 Virtual Chassis (CLI Procedure)*
- *Replacing a Member Switch of a Virtual Chassis Configuration (CLI Procedure)*

serial-number

Syntax	<code>serial-number serial-number;</code>
Hierarchy Level	[edit virtual-chassis preprovisioned 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.</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 EX2200 Virtual Chassis (CLI Procedure)</i>• <i>Configuring an EX3300 Virtual Chassis (CLI Procedure)</i>• <i>Configuring an EX4300 Virtual Chassis (CLI Procedure)</i>• <i>Configuring an EX4200, EX4500, or EX4550 Virtual Chassis (CLI Procedure)</i>• Configuring an EX8200 Virtual Chassis (CLI Procedure) on page 32• <i>Configuring an EX9200 Virtual Chassis</i>• <i>Configuring a QFX Series Virtual Chassis (CLI Procedure)</i>• <i>Configuring a Virtual Chassis on an EX Series Switch (J-Web Procedure)</i>

tracoptions (Virtual Chassis)

Syntax tracoptions {
 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> • Verifying Virtual Chassis Ports in an EX8200 Virtual Chassis on page 70 • <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-provisioned;
        auto-sw-update {
            (ex-4200 | ex-4300 | ex-4500 | ex-4600 | qfx-3 | qfx-5)
            package-name package-name;
        }
        fast-failover (ge | vcp disable | xe);
        graceful-restart {
            disable;
        }
        id id;
        mac-persistence-timer seconds;
        member member-id {
            fabric-tree-root;
            location location;
            mastership-priority number;
            no-management-vlan;
            serial-number;
            role;
        }
        no-split-detection;
        preprovisioned;
        traceoptions (Virtual Chassis) {
            file filename <files number> <size size> <world-readable | no-world-readable> <match
                regex>;
            flag flag ;
        }
        vc-port {
            lag-hash (packet-based | source-port-based);
        }
        vcp-no-hold-time;
        vcp-snmp-statistics;
    }
```

Hierarchy Level [edit]

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

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

The remaining statements are explained separately.

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

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

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

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>Autoprovisioning a Virtual Chassis Fabric</i>• <i>Preprovisioning a Virtual Chassis Fabric</i>• <i>Adding a Device to a Virtual Chassis Fabric</i>• <i>Configuring a QFX Series Virtual Chassis (CLI Procedure)</i>• <i>Example: Configuring an EX3300 Virtual Chassis with a Master and Backup</i>• <i>Example: Configuring an EX4200 Virtual Chassis with a Master and Backup in a Single Wiring Closet</i>• Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines on page 25• <i>Configuring an EX3300 Virtual Chassis (CLI Procedure)</i>• <i>Configuring an EX4200, EX4500, or EX4550 Virtual Chassis (CLI Procedure)</i>• Configuring an EX8200 Virtual Chassis (CLI Procedure) on page 32• <i>Configuring an EX9200 Virtual Chassis</i>

PART 3

Administration

- [Routine Monitoring on page 69](#)
- [Operational Commands on page 73](#)

CHAPTER 5

Routine Monitoring

- [Verifying the Member ID, Role, and Neighbor Member Connections of an EX8200 Virtual Chassis Member on page 69](#)
- [Verifying Virtual Chassis Ports in an EX8200 Virtual Chassis on page 70](#)
- [Verifying Hard Disk Memory Status on an XRE200 External Routing Engine on page 71](#)

Verifying the Member ID, Role, and Neighbor Member Connections of an EX8200 Virtual Chassis Member

Purpose View the member ID, role, and VCP links in an EX8200 Virtual Chassis.

Action To display the role and member ID assignments, use the **show virtual-chassis** command:

```
user@external-routing-engine> show virtual-chassis
```

```
Preprovisioned Virtual Chassis
Virtual Chassis ID: c806.0842.de51
```

List				Mastership		Neighbor	
Member ID	Status	Serial No	Model	priority	Role	ID	Interface
0 (FPC 0-15)	Prsnt	ghi789	ex8208		0 Linecard	8	vcp-0/0
						9	vcp-0/1
1 (FPC 16-31)	Prsnt	jkl112	ex8216		0 Linecard	1	vcp-3/0/2
						8	vcp-0/0
						9	vcp-0/1
8 (FPC 128-143)	Prsnt	abc123	ex-xre	129	Master*	0	vcp-4/0/2
						9	vcp-1/0
						0	vcp-1/1
9 (FPC 144-159)	Prsnt	def456	ex-xre	129	Backup	1	vcp-1/2
						8	vcp-1/0
						0	vcp-1/1
						1	vcp-1/2

Meaning This output verifies that a Virtual Chassis composed of one EX8208 switch (member 0), one EX8216 switch (member 1), and master and backup XRE200 External Routing Engines

(members 8 and 9) is operational by showing that all devices are in the **Prsnt** state. The display shows the VCP connections and confirms that each member has an active connection to all three other members.

- Related Documentation**
- [Configuring an EX8200 Virtual Chassis \(CLI Procedure\) on page 32](#)
 - [Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines on page 25](#)

Verifying Virtual Chassis Ports in an EX8200 Virtual Chassis

Purpose Verify that the Virtual Chassis ports (VCPs) in the EX8200 Virtual Chassis are properly configured. This task is particularly important as a method of confirming the user-configured VCP connection between two EX8200 switches.

Action Verify the type, status, speed, and neighbor ID and interface of each VCP on an EX8200 switch or an XRE200 External Routing Engine:

```
user@external-routing-engine> show virtual-chassis vc-port member 0
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	9	vcp-1/1
3/0/2	Configured	-1	Up	10000	1	vcp-4/0/2
3/0/3	Configured		Absent			
3/0/4	Configured		Absent			
3/0/5	Configured		Absent			
3/0/6	Configured		Absent			
3/0/7	Configured		Absent			

Meaning Members 0 through 7 in an EX8200 Virtual Chassis are reserved for EX8200 switches, so we know this output from member 0 shows the VCP connections from an EX8200 member switch.

The output shows that the switch has a dedicated VCP connection to each external Routing Engine—member IDs 8 and 9, which are always external Routing Engines. These links are dedicated because they are connections to an external Routing Engine.

The output also shows a VCP link between two EX8200 switches (member 0 and member 1). VCP links between EX8200 member switches must be user-configured, and the **Configured** Type output is a result of this user configuration. This VCP link is from interface **vcp-3/0/2** on the switch to interface **vcp-4/0/2** of the switch acting as member 1.

Trunk ID -1 is used to indicate a link is not part of a link aggregation group (LAG). This output shows that no links in this configuration are part of a LAG.

- Related Documentation**
- [Setting a 10-Gigabit Ethernet Port as a Virtual Chassis Port in an EX8200 Virtual Chassis \(CLI Procedure\) on page 36](#)

Verifying Hard Disk Memory Status on an XRE200 External Routing Engine

Purpose Verify that the hard disk memory on your XRE200 External Routing Engine is functioning properly.

- Action**
1. Run a Hard Disk Drive (HDD) memory and diagnostics monitoring test:

```
user@external-routing-engine> request chassis routing-engine hard-disk-test
```
 2. View the test results that appear on the screen:

```
RAID INFORMATION
RAID device path: /dev/ad4
Firmware Version: 11594
RAID controller s/n: 12345678
RAID Chip ID: 123
RAID policy: SAFE

Drive0 model: WDC WD123AAJS-4567A0
Drive1 model: WDC WD345JD-18MSA1
Drive0 s/n: WD-WCAT30214999
Drive1 s/n: WD-WMAM9DTK4111
Drive0 capacity: 74(GB)
Drive1 capacity: 74(GB)

RAID STATUS
Drive0: On-line
Drive1: On-line
Number of partitions: 1
Size of Partitions:
  Partition 0: 74(GB)
RAID Status: Healthy
```

Meaning These results confirm that both HDD modules are functioning properly, as shown by the status of **On-line** for both **Drive0** and **Drive1**. If your HDD module is not functioning properly, the status is **Off-line** or **Unknown**.

If the status is **Off-line** for **Drive0** or **Drive1**, verify that the HDD module is securely seated in its slot. If the status is **Unknown** for either drive, wait a few minutes then retry the test. If you are unable to return the status to **On-line**, contact Juniper Networks customer support (JTAC). See *Contacting Customer Support to Obtain Return Material Authorization for Switches*.

The results also show that RAID storage, as indicated in the **RAID Status** output, is **Healthy**. RAID storage technology manages storage between the redundant HDD modules in the XRE200 External Routing Engine. If RAID storage is not functioning properly, the status is **Degraded** or **Unknown**.

If **Degraded** appears in the RAID Status field, contact customer support.

If the **RAID Status** is **Unknown**, wait a few minutes, then retry the test. If the status remains **Unknown** after the second test, contact JTAC.



NOTE: The remainder of the test results output is either informational or intended for customer support purposes only.

**Related
Documentation**

- [Understanding File Storage in an EX8200 Virtual Chassis on page 16](#)
- *Hard Disk Drive (HDD) Module in an XRE200 External Routing Engine*

CHAPTER 6

Operational Commands

- clear virtual-chassis protocol statistics
- clear virtual-chassis vc-port statistics
- request chassis routing-engine hard-disk-test
- request session member
- request system software recovery-package
- request virtual-chassis device-reachability
- request virtual-chassis reactivate
- request virtual-chassis recycle
- request virtual-chassis renumber
- request virtual-chassis vc-port
- set chassis virtual-chassis
- show system software recovery-package
- show virtual-chassis active-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 vc-port statistics

clear virtual-chassis protocol statistics

Syntax	clear virtual-chassis protocol statistics <all-members> <local> <member <i>member-id</i> >
Release Information	Command introduced in Junos OS Release 10.4 for EX Series switches.
Description	Clear—reset to zero (0)—all Virtual Chassis Control Protocol (VCCP) traffic counters for the Virtual Chassis.
Options	<p>none—Clear VCCP counters for all members of the Virtual Chassis.</p> <p>all-members—(Optional) Clear VCCP traffic counters for all members of the Virtual Chassis.</p> <p>local—(Optional) Clear VCCP traffic counters on the switch or external Routing Engine on which this command is entered.</p> <p>member <i>member-id</i>—(Optional) Clear VCCP traffic counters for the specified member of the Virtual Chassis.</p>
Required Privilege Level	clear
Related Documentation	<ul style="list-style-type: none">• show virtual-chassis protocol statistics on page 114• Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis on page 17

Sample Output

clear virtual-chassis protocol statistics

```
user@switch> clear virtual-chassis protocol statistics
member0:
-----
Statistics cleared

member1:
-----
Statistics cleared

member8:
-----
Statistics cleared

member9:
-----
Statistics cleared
```

This output shows that the VCCP counters have been cleared for both member switches (member 0 and member 1) and both XRE200 External Routing Engines (member 8 and member 9) in the EX8200 Virtual Chassis.

clear virtual-chassis vc-port statistics

Syntax	<code>clear virtual-chassis vc-port statistics</code> <code><all-members></code> <code><interface-name></code> <code><local></code> <code><member member-id></code>
Release Information	Command introduced in Junos OS Release 9.0 for EX Series switches. The options all-members and local were added in Junos OS Release 9.3 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric.
Description	Clear—reset to zero (0)—the traffic statistics counters on Virtual Chassis ports (VCPs).
Options	none —Clear traffic statistics for VCPs of all members of a Virtual Chassis or VCF. all-members —(Optional) Clear traffic statistics for VCPs of all members of a Virtual Chassis or VCF. interface-name —(Optional) Clear traffic statistics for the specified VCP. local —(Optional) Clear traffic statistics for VCPs from the switch or external Routing Engine on which this command is entered. member member-id —(Optional) Clear traffic statistics for VCPs from the specified member of a Virtual Chassis or VCF.
Required Privilege Level	clear
Related Documentation	<ul style="list-style-type: none">• show virtual-chassis vc-port statistics on page 126• show virtual-chassis vc-port on page 122• <i>Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis</i>
List of Sample Output	clear virtual-chassis vc-port statistics (EX4200 Virtual Chassis) on page 76 clear virtual-chassis vc-port statistics (EX8200 Virtual Chassis) on page 77 clear virtual-chassis vc-port statistics member 3 on page 77

Sample Output

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

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

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

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

```
member0:
```

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

```
member1:
```

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

```
member8:
```

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

```
member9:
```

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

clear virtual-chassis vc-port statistics member 3

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

```
Cleared statistics on member 3
```

request chassis routing-engine hard-disk-test

Syntax	request chassis routing-engine hard-disk-test
Release Information	Command introduced in Junos OS Release 12.1 for XRE200 External Routing Engines.
Description	Run a Hard Disk Drive (HDD) memory and diagnostics monitoring test for an XRE200 External Routing Engine. The test provides output that is helpful in monitoring the status of the hard disk memory on the XRE200 External Routing Engine.
Required Privilege Level	system-control
Related Documentation	<ul style="list-style-type: none"> <i>Hard Disk Drive (HDD) Module in an XRE200 External Routing Engine</i> <i>XRE200 External Routing Engine Hardware Overview</i>
List of Sample Output	request chassis routing-engine hard-disk-test on page 80
Output Fields	Table 7 on page 78 lists the output fields for the request chassis routing-engine hard-disk-test command. Output fields are listed in the approximate order in which they appear.

Table 7: request chassis routing-engine hard-disk-test Output Fields

Field Name	Field Description
RAID device path	<p>The device path to the RAID.</p> <p>This output is generally only useful for customer support personnel.</p>
Firmware Version	<p>The firmware version of the RAID.</p> <p>This output is generally only useful for customer support personnel.</p>
RAID controller s/n	<p>The serial number of the RAID controller.</p> <p>This output is generally only useful for customer support personnel.</p>
RAID Chip ID	<p>The RAID chip identification number.</p> <p>This output is generally only useful for customer support personnel.</p>
RAID policy	<p>The RAID policy status.</p> <p>SAFE is the only RAID policy used by an XRE200 External Routing Engine.</p> <p>This output is generally only useful for customer support personnel.</p>
Drive0 model	<p>The model number of the HDD module in HDD slot 0.</p> <p>This output is generally only useful for customer support personnel.</p>

Table 7: request chassis routing-engine hard-disk-test Output Fields (*continued*)

Field Name	Field Description
Drive1 model	<p>The model number of the HDD module in HDD slot 1.</p> <p>This output is generally only useful for customer support personnel.</p>
Drive0 s/n	<p>The serial number of the HDD module in HDD slot 0.</p> <p>This output is generally only useful for customer support personnel.</p>
Drive1 s/n	<p>The serial number of the HDD module in HDD slot 1.</p> <p>This output is generally only useful for customer support personnel.</p>
Drive0 capacity	The memory capacity of the HDD module in HDD slot 0, in gigabytes.
Drive1 capacity	The memory capacity of the HDD module in HDD slot 1, in gigabytes.
Drive0 (RAID STATUS)	<p>The RAID status of the HDD module in HDD slot 0.</p> <p>Outputs include:</p> <ul style="list-style-type: none"> • On-line—The HDD module is online. • Off-line—The HDD module is offline. If the status is Off-line, verify that the HDD module is securely seated in its slot. If the HDD module is securely seated and you cannot identify the reason for this status, contact Juniper Networks customer support (JTAC). • Unknown—HDD module RAID status is unknown. If the status is Unknown, wait a few minutes then retry the test. If the status remains Unknown after the second test, contact Juniper Networks customer support (JTAC).
Drive1 (RAID STATUS)	<p>The RAID status of the HDD module in HDD slot 1.</p> <p>Outputs include:</p> <ul style="list-style-type: none"> • On-line—HDD module is online. • Off-line—HDD module is offline. If the status is Off-line, verify that the HDD module is securely seated in its slot. If the HDD module is securely seated and you cannot identify the reason for this status, contact Juniper Networks customer support (JTAC). • Unknown—HDD module RAID status is unknown. If the status is Unknown, wait a few minutes then retry the test. If the status remains Unknown after the second test, contact Juniper Networks customer support (JTAC).
Number of partitions	The number of RAID memory partitions.
Size of Partitions	The size of each partition, in gigabytes.

Table 7: request chassis routing-engine hard-disk-test Output Fields (*continued*)

Field Name	Field Description
RAID Status	<p>The RAID status.</p> <p>Outputs include:</p> <ul style="list-style-type: none"> • Healthy—RAID is healthy. • Degraded—RAID has detected a problem with at least one HDD module. If Degraded appears in the RAID Status field, contact customer support. • Unknown—RAID status is unknown. If the status is Unknown, wait a few minutes then retry the test. If the status remains Unknown after the second test, contact Juniper Networks customer support (JTAC). • Rebuilding—The RAID is rebuilding. The output includes a drive number and a percentage complete indication. The RAID rebuilding process can take up to 40 minutes. Wait several minutes after the process has finished and then retry the test. • Verifying—The RAID is rebuilt and being verified by the system. The output includes a drive number and a percentage complete indication. The RAID verification process can take up to 40 minutes. Wait several minutes after the process has finished and then retry the test.

Sample Output

request chassis routing-engine hard-disk-test

```

user@switch> request chassis routing-engine hard-disk-test
RAID INFORMATION
RAID device path: /dev/ad4
Firmware Version: 11594
RAID controller s/n: 12345678
RAID Chip ID: 123
RAID policy: SAFE

Drive0 model: WDC WD123AAJS-4567A0
Drive1 model: WDC WD345JD-18MSA1
Drive0 s/n: WD-WCAT30214999
Drive1 s/n: WD-WMAM9DTK4111
Drive0 capacity: 74(GB)
Drive1 capacity: 74(GB)

RAID STATUS
Drive0: On-line
Drive1: On-line
Number of partitions: 1
Size of Partitions:
  Partition 0: 74(GB)
RAID Status: Healthy

```


request session member

Syntax	<code>request session member <i>member-id</i></code>
Release Information	Command introduced in Junos OS Release 9.0 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
Description	Start a session with the specified member of a Virtual Chassis or a VCF.
Options	<i>member-id</i> —Member ID for the specific member of the Virtual Chassis or VCF.
Required Privilege Level	maintenance
Related Documentation	<ul style="list-style-type: none">• member on page 54• <i>Understanding EX Series Virtual Chassis Components</i>• <i>Understanding QFX Series Virtual Chassis Components</i>

request system software recovery-package

Syntax request system software recovery-package (add *package-name* | delete *package-name* |
extract (ex-xre200-package | ex-8200-package) *package-name*)
<backup>
<master>

Release Information Command introduced in Junos OS Release 12.2 for EX Series switches.

Description Add, remove, or extract a software recovery package for the EX8200 switch or the XRE200 External Routing Engine in an EX8200 Virtual Chassis.

Use this command to load a software recovery package onto the USB flash drive of the external Routing Engine and then extract it to the recovery partition of the external Routing Engine—so that it is available for use during a software recovery installation. During a software recovery installation procedure, you are prompted to select which software recovery package to use for the recovery.



NOTE: This command is available only on the external Routing Engine.

If you do not have the desired software recovery package already installed in the recovery partition of the external Routing Engine, download the software package from the Internet. See *Downloading Software Packages from Juniper Networks*.

Options An EX8200 Virtual Chassis supports up to two external Routing Engines for the purpose of external Routing Engine redundancy. If you have configured two external Routing Engines, one functions as the master external Routing Engine and the other functions as the backup. If you have not configured two external Routing Engines, the **backup** and **master** options are not relevant.

add *package-name* backup | master—Load the EX8200 software recovery package or the XRE200 software recovery package onto the USB flash drive of the external Routing Engine.



NOTE: You can have up to five software recovery packages loaded in the USB flash drive.

- ***package-name***—path name of the software recovery package.
- **backup**—(Optional) Load the specified package onto the USB flash drive of the backup external Routing Engine.
- **master**—(Optional) Load the specified package onto the USB flash drive of the master external Routing Engine.

delete *package-name* backup | master—Delete the EX8200 software recovery package or the XRE200 software recovery package from the USB drive of the external Routing Engine.

- ***package-name***—Path name of the software package.
- **backup**—(Optional) Delete the specified package from the USB flash drive of the backup external Routing Engine.
- **master**—(Optional) Delete the specified package from the USB flash drive of the master external Routing Engine.

extract (ex-xre200-package | ex-8200-package) *package-name* backup | master—Extract either the XRE200 software recovery package or the EX8200 software recovery package from the USB flash drive of the external Routing Engine and put it into the recovery partition of the external Routing Engine.

- ***package-name***—Path name of the software recovery package.
- **backup**—(Optional) Extract the specified package from the USB flash drive to the recovery partition of the backup external Routing Engine.
- **master**—(Optional) Extract the specified package from the USB flash drive to the recovery partition of the master external Routing Engine.

Required Privilege Level maintenance

Related Documentation

- [show system software recovery-package on page 93](#)
- *XRE200 External Routing Engine Hardware Overview*

List of Sample Output

- [request system software recovery-package add on page 83](#)
- [request system software recovery-package delete on page 84](#)
- [request system software recovery-package extract on page 84](#)

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

[request system software recovery-package add](#)

```
user@external-routing-engine> request system software recovery-package add
/tftpboot/jinstall-ex-xre200-11.1I20110608_1356_pravasp-domestic.tgz
...-ex-xre200-11.1I20110608_1356_pravasp-domestic.tgz
Member 1 :
This feature is not supported on this device
Inorder to activate this feature, please reinstall the device from USB with a
package not lesser than 11.4
Member 0 :
Validating package... Done.
Please wait while the package is being uploaded.
jinstall-ex-xre200-11.1I20110608_1356_pravasp-domestic.tgz was added successfully.
```

request system software recovery-package delete

```
user@external-routing-engine> request system software recovery-package delete  
jinstall-ex-xre200-11.1I20110608_1356_pravasp-domestic.tgz
```

```
Member 1 : jinstall-ex-xre200-11.1I20110608_1356_pravasp-domestic.tgz was removed  
successfully.
```

```
Member 0 : Requested package doesn't exist in the recovery media.
```

request system software recovery-package extract

```
user@external-routing-engine> request system software recovery-package extract  
ex-xre200-package jinstall-ex-xre200-11.1I20110608_1356_pravasp-domestic.tgz  
...kage jinstall-ex-xre200-11.1I20110608_1356_pravasp-domestic.tgz  
Package extracted to  
member1:/tftpboot/jinstall-ex-xre200-11.1I20110608_1356_pravasp-domestic.tgz
```

request virtual-chassis device-reachability

Syntax `request virtual-chassis device-reachability test-name name (source-fpc source-fpc-id | source-ip-address source-ip-address) (destination-device device-id | destination-fpc destination-fpc-id | destination-ip-address destination-ip-address)`
`<probe-count count>`
`<probe-interval interval>`
`<probe-pattern pattern>`
`<test-count count>`
`<test-interval interval>`

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

Description Run a Virtual Chassis device reachability test. A Virtual Chassis device reachability test sends ping packets from one member of a Virtual Chassis to another member of a Virtual Chassis directly through the Virtual Chassis connections. The receiving Virtual Chassis member sends reply packets to confirm receipt of the ping packets from the sending device. The results of the test immediately provide information that is helpful in verifying connectivity between Virtual Chassis member devices.

Options **test-name *name***—Specify the name of the test. The name can be any single-word character string. Spaces are not allowed.

source-fpc *source-fpc-id*—Specify the FPC that sends the first ping message.

source-ip-address *source-ip-address*—Specify the source IP address that receives the ping message.

destination-device *device-id*—Specify the device in the Virtual Chassis that receives the ping message.

destination-fpc *destination-fpc-id*—Specify the FPC that receives the first ping message.

destination-ip-address *destination-ip-address*—Specify the destination IP address that receives the ping message.

probe-count *count*—(Optional) Specify the number of ping messages to send for the test. The count can be any number from 1 through 10. The default count is 5.

probe-interval *interval*—(Optional) Specify the time in seconds between the ping messages that are sent during the test. The interval can be any number from 1 through 3. The default interval is 1 second.

probe-pattern *pattern*—(Optional) Specify a payload pattern in the ping message. Enter the pattern in hexadecimal format.

test-count *count*—(Optional) Specify the number of times to run the test. The count can be any number from 1 through 3. The default count is 1.

test-interval *interval*—(Optional) Specify the time in seconds between tests when multiple tests are specified. The interval can be any number from 1 through 3. The default interval is 1.

Required Privilege Level system-control

Related Documentation • *Verifying Connectivity Between Virtual Chassis Member Devices*

Output Fields [Table 8 on page 86](#) lists the output fields for the **request virtual chassis device reachability** command. Output fields are listed in the approximate order in which they appear.

Table 8: request virtual chassis device reachability Output Fields

Field Name	Field Description
Test Name	The name of the test.
Performing Test	The test number.
Session ID	The session ID. This output is always 0 and not useful for Virtual Chassis device reachability tests.
Packets Count/Sent/Received/Sendfail	The number of total ping packets sent, including the total number of ping packets counted (Count), sent (Sent), received (Received), and packets that could not be sent (Sendfail).
Unknown/Timedout/Duplicate packets received	The failed pings, including the number of ping packets that failed for an unknown reason (Unknown), timed out (Timed Out), and the number of duplicate received ping packets (Duplicate packets received).
Round-trip Min/Avg/Max	The average round-trip ping time, including the shortest ping time (Min), the average ping time (Avg), and the maximum ping time (Max).

Sample Output

request virtual-chassis device-reachability test-name member0-to-member2 source-fpc 0 destination-fpc 2

```

user@switch> request virtual-chassis device-reachability test-name member0-to-member2
source-fpc 0 destination-fpc 2
Device Reachability Statistics:
Test Name                : member0-to-member2

Performing Test: 1

56 bytes from 0: session-id 0 seq-id 0
56 bytes from 0: session-id 0 seq-id 1
56 bytes from 0: session-id 0 seq-id 2
56 bytes from 0: session-id 0 seq-id 3
56 bytes from 0: session-id 0 seq-id 4
--- ping statistics ---
Session ID                : 0
Packets Count/Sent/Received/Sendfail : 5/5/5/0
Unknown/Timedout/Duplicate packets received : 0/0/0
Round-trip Min/Avg/Max    : 181/807/3010 usec

```

request virtual-chassis reactivate


Syntax	<code>request virtual-chassis reactivate</code>
Release Information	<p>Command introduced in Junos OS Release 9.3 for EX Series switches.</p> <p>Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).</p>
Description	<p>Reactivate a device that has been assigned a member ID but is not currently connected to the Virtual Chassis or VCF.</p> <p>You can use this command to reactivate a device that was previously part of the Virtual Chassis or VCF but whose status is no longer Prsnt.</p>
Required Privilege Level	system-control
Related Documentation	<ul style="list-style-type: none"> • <i>Verifying the Member ID, Role, and Neighbor Member Connections of a Virtual Chassis Member</i> • Verifying the Member ID, Role, and Neighbor Member Connections of an EX8200 Virtual Chassis Member on page 69
List of Sample Output	request virtual-chassis reactivate on page 87

Sample Output

request virtual-chassis reactivate

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

request virtual-chassis recycle

Syntax	request virtual-chassis recycle member-id <i>member-id</i>
Release Information	Command introduced in Junos OS Release 9.0 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series.
Description	<p>Make a previously used member ID available for reassignment.</p> <p>When you remove a member switch from the Virtual Chassis configuration, the master reserves that member ID. To make the member ID available for reassignment, you must use this command.</p>
	<div> NOTE: You must run this command from the Virtual Chassis member in the master role.</div>
Options	member-id <i>member-id</i> —Specify the member ID that you want to make available for reassignment to a different member.
Required Privilege Level	system-control
Related Documentation	<ul style="list-style-type: none">• request virtual-chassis renumber on page 89• <i>Replacing a Member Switch of a Virtual Chassis Configuration (CLI Procedure)</i>• <i>Adding or Replacing a Member Switch or an External Routing Engine in an EX8200 Virtual Chassis (CLI Procedure) on page 34</i>
List of Sample Output	request virtual-chassis recycle member-id 3 on page 88 request virtual-chassis recycle member-id 1 on page 88

Sample Output

request virtual-chassis recycle member-id 3


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

Sample Output

request virtual-chassis recycle member-id 1

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


request virtual-chassis renumber

Syntax	<code>request virtual-chassis renumber member-id <i>old-member-id</i> new-member-id <i>new-member-id</i></code>
Release Information	Command introduced in Junos OS Release 9.0 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series.
Description	Renumber a member of a Virtual Chassis configuration.
<div>  NOTE: You must run this command from the Virtual Chassis member in the master role. </div>	
Options	<p><code>member-id <i>old-member-id</i></code>—Specify the ID of the member that you wish to renumber.</p> <p><code>new-member-id <i>new-member-id</i></code>—Specify an unassigned member ID.</p>
Required Privilege Level	system-control
Related Documentation	<ul style="list-style-type: none"> • request virtual-chassis recycle on page 88 • <i>Replacing a Member Switch of a Virtual Chassis Configuration (CLI Procedure)</i> • <i>Adding or Replacing a Member Switch or an External Routing Engine in an EX8200 Virtual Chassis (CLI Procedure) on page 34</i>
List of Sample Output	<p>request virtual-chassis renumber member-id 5 new-member-id 4 on page 89</p> <p>request virtual-chassis renumber member-id 1 new-member-id 0 on page 89</p>

Sample Output

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

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

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

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

request virtual-chassis vc-port

Syntax	<code>request virtual-chassis vc-port [set delete] <fpc-slot <i>fpc-slot</i>> pic-slot <i>pic-slot</i> port <i>port-number</i> <member <i>member-id</i>></code>
Release Information	<p>Command introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Option fpc-slot 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	<p>Enable or disable a port as a Virtual Chassis port (VCP).</p> <p>If you omit member <i>member-id</i>, this command defaults to enabling or disabling the uplink VCP or SFP network port configured as a VCP on the switch where the command is issued.</p> <p>On an EX3300 switch, uplink ports 2 and 3 are configured as VCPs by default. No other uplink ports on any other EX Series switches are configured as VCPs by default.</p> <p>You might experience a temporary traffic disruption immediately after creating or deleting a user-configured VCP in an EX8200 Virtual Chassis.</p>
Options	<p>set—Set a network port into a VCP to convert a network port into a VCP.</p> <p>delete—Delete the VCP setting on a port to convert a VCP into a network port.</p> <p>pic-slot <i>pic-slot</i>—Number of the PIC slot for the port on the switch.</p> <p>port <i>port-number</i>—Number of the port that is to be enabled or disabled as a VCP.</p> <p>member <i>member-id</i>—(Optional) Enable or disable the specified VCP on the specified member of the Virtual Chassis or VCF.</p>
Required Privilege Level	system-control
Related Documentation	<ul style="list-style-type: none"> • request virtual-chassis vc-port (dedicated port) • show virtual-chassis vc-port on page 122 • show virtual-chassis vc-port statistics on page 126 • clear virtual-chassis vc-port statistics on page 76 • Virtual Chassis Port (VCP) Interface Names in an EX8200 Virtual Chassis on page 20 • Understanding EX Series Virtual Chassis Components • Understanding QFX Series Virtual Chassis Components
List of Sample Output	<p>request virtual-chassis vc-port set pic-slot 1 port 0 on page 91</p> <p>request virtual-chassis vc-port set pic-slot 1 port 1 member 3 on page 91</p> <p>request virtual-chassis vc-port delete pic-slot 1 port 1 member 3 on page 91</p>

Sample Output

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

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

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

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

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

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

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

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

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

set chassis virtual-chassis

Syntax	set chassis virtual-chassis <disable> <local>
Release Information	Command introduced in Junos OS Release 10.4 for EX Series switches.
Description	<p>Set the EX8200 switch into Virtual Chassis mode.</p> <p>An EX8200 switch cannot be a member switch in an EX8200 Virtual Chassis until the switch is in Virtual Chassis mode.</p> <p>This command is not needed when configuring an EX3300 Virtual Chassis, EX4200 Virtual Chassis, EX4500 Virtual Chassis, or a mixed EX4200 and EX4500 Virtual Chassis.</p>
Options	<p>none—Set the internal Routing Engines into Virtual Chassis mode.</p> <p>disable—Disable Virtual Chassis mode for all internal REs on the switch.</p> <p>local—Set the Routing Engine that you are currently logged in to into Virtual Chassis mode.</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none">• show virtual-chassis on page 117• Configuring an EX8200 Virtual Chassis (CLI Procedure) on page 32
List of Sample Output	set chassis virtual-chassis on page 92

Sample Output

set chassis virtual-chassis

```
user@switch> set chassis virtual-chassis
This will reboot the RE(s)
Do you want to continue ? [yes,no] (no) yes
```

show system software recovery-package

Syntax	show system software recovery-package <backup> <master>
Release Information	Command introduced in Junos OS Release 12.2 for EX Series switches.
Description	Display the software recovery packages loaded on an XRE200 External Routing Engine in an EX8200 Virtual Chassis.
Options	<p>An EX8200 Virtual Chassis supports up to two external Routing Engines for the purpose of external Routing Engine redundancy. If you have configured two external Routing Engines, one functions as the master external Routing Engine and the other functions as the backup. If you have not configured two external Routing Engines, the backup and master options are not relevant.</p> <p>none—Display the software recovery packages available on the external Routing Engine.</p> <p>backup—(Optional) Display the software recovery packages available on the backup external Routing Engine.</p> <p>master—(Optional) Display the software recovery packages available on the master external Routing Engine.</p>
Required Privilege Level	maintenance
Related Documentation	<ul style="list-style-type: none"> • <i>XRE200 External Routing Engine Hardware Overview</i> • request system software recovery-package on page 82
List of Sample Output	show system software recovery-package on page 93 show system software recovery-package backup on page 94 show system software recovery-package master on page 94
Output Fields	When you enter this command, you are provided a list of software recovery packages, including the software release number, installed on the external Routing Engine.

Sample Output

show system software recovery-package

```

user@external-routing-engine> show system software recovery-package
Member 1 :
-----
1. jinstall-ex-xre200-11.1I20110608_1356_pravasp-domestic.tgz
Member 0 :
-----
1. jinstall-ex-xre200-11.4I20110608_1356_pravasp-domestic.tgz

```

show system software recovery-package backup

```
user@external-routing-engine> show system software recovery-package backup
Member 0 :
```

```
-----
1. jinstall-ex-xre200-11.4I20110608_1356_pravasp-domestic.tgz
```

show system software recovery-package master

```
user@external-routing-engine> show system software recovery-package master
Member 1 :
```

```
-----
1. jinstall-ex-xre200-11.1I20110608_1356_pravasp-domestic.tgz
```

show virtual-chassis active-topology

Syntax	show virtual-chassis active-topology <all-members> <local> <member <i>member-id</i> >
Release Information	Command introduced in Junos OS Release 9.0 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
Description	Display the active topology of the Virtual Chassis or VCF with next-hop reachability information.
Options	<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> • <i>Understanding EX Series Virtual Chassis Configuration</i>
List of Sample Output	show virtual-chassis active-topology (EX4200 Virtual Chassis) on page 96 show virtual-chassis active-topology (EX8200 Virtual Chassis) on page 96 show virtual-chassis active-topology (Virtual Chassis Fabric) on page 97
Output Fields	Table 9 on page 95 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 9: 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 6(vcp-255/0/1.32768)	4(vcp-255/0/2.32768) 5(vcp-255/0/3.32768)
1 6(vcp-255/0/1.32768)	4(vcp-255/0/2.32768) 5(vcp-255/0/3.32768)
3 6(vcp-255/0/1.32768)	4(vcp-255/0/2.32768) 5(vcp-255/0/3.32768)
4	4(vcp-255/0/2.32768)
5	5(vcp-255/0/3.32768)
6	6(vcp-255/0/1.32768)

fpc3:

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

fpc4:

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

fpc5:

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

```

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

```

fpc6:

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

show virtual-chassis protocol adjacency

Syntax	<code>show virtual-chassis protocol adjacency</code> <code><brief detail extensive></code> <code><all-members></code> <code><local></code> <code><member <i>member-id</i>></code> <code><system-id></code>
Release Information	Command introduced in Junos OS Release 10.4 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
Description	Display the Virtual Chassis Control Protocol (VCCP) adjacency statistics in the Virtual Chassis or VCF for all hardware devices.
Options	<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">• Understanding EX Series Virtual Chassis Port Link Aggregation• Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis on page 17
List of Sample Output	show virtual-chassis protocol adjacency on page 101 show virtual-chassis protocol adjacency detail on page 102
Output Fields	Table 10 on page 101 lists the output fields for the show virtual-chassis protocol adjacency command. Output fields are listed in the approximate order in which they appear.

Table 10: show virtual-chassis protocol adjacency Output Fields

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

Sample Output

show virtual-chassis protocol adjacency

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

```
member0:
```

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

```
member1:
```

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

```
member8:
```

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

```
member9:
```

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

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

show virtual-chassis protocol adjacency detail

```
user@switch> show virtual-chassis protocol adjacency detail
member0:
```

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

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

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

```
member1:
```

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

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

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

```
member8:
```

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

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

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

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

```
member9:
```

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

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

show virtual-chassis protocol database

Syntax	<code>show virtual-chassis protocol database</code> <code><brief detail extensive></code> <code><all-members></code> <code><local></code> <code><member <i>member-id</i>></code>
Release Information	Command introduced in Junos OS Release 10.4 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
Description	Display the Virtual Chassis Control Protocol (VCCP) database statistics for all hardware devices within the Virtual Chassis or VCF.
Options	<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 on page 17• <i>Understanding EX Series Virtual Chassis Components</i>• <i>Understanding QFX Series Virtual Chassis Components</i>
List of Sample Output	show virtual-chassis protocol database on page 105 show virtual-chassis protocol database detail on page 106
Output Fields	Table 11 on page 105 lists the output fields for the show virtual-chassis protocol database command. Output fields are listed in the approximate order in which they appear.

Table 11: show virtual-chassis protocol database Output Fields

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

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

Sample Output

show virtual-chassis protocol database

```
user@switch> show virtual-chassis protocol database
member0:
```

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

```
member1:
```

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

```
member8:
```

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

```
member9:
```

```
-----
LSP ID          Sequence Checksum Lifetime
```

0000.4a75.9b7c.00-00	0x1dd80	0xc2e3	116
0000.73e9.9a57.00-00	0xf361	0x27e8	116
0021.59f7.d000.00-00	0x16883	0x289	114
0026.888d.6800.00-00	0x16920	0xa335	116

4 LSPs

show virtual-chassis protocol database detail

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

```
member0:
```

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

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

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

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

```
member1:
```

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

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

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

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

```
member8:
```

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

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

```

Neighbor: 0026.888d.6800.00 Interface: vcp-1/1.32768 Metric: 150

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

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

```

member9:

```

-----

0000.4a75.9b7c.00-00 Sequence: 0x1ddbd, Checksum: 0xfd83, Lifetime: 116 secs
Neighbor: 0000.73e9.9a57.00 Interface: vcp-1/0.32768 Metric: 150
Neighbor: 0021.59f7.d000.00 Interface: vcp-1/1.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-1/2.32768 Metric: 150

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

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

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

```

show virtual-chassis protocol interface

Syntax	<code>show virtual-chassis protocol interface</code> <code><brief detail></code> <code><all-members></code> <code><interface-name></code> <code><local></code> <code><member member-id></code>
Release Information	Command introduced in Junos OS Release 10.4 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
Description	Display information about Virtual Chassis Control Protocol (VCCP) statistics for VCCP-enabled interfaces within the Virtual Chassis or VCF.
Options	<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>EX Series Virtual Chassis Overview</i>• <i>Understanding QFX Series Virtual Chassis</i>• Understanding Virtual Chassis Ports in an EX8200 Virtual Chassis on page 13• Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis on page 17
List of Sample Output	show virtual-chassis protocol interface on page 109
Output Fields	Table 12 on page 109 lists the output fields for the show virtual-chassis protocol interface command. Output fields are listed in the approximate order in which they appear.

Table 12: show virtual-chassis protocol interface Output Fields

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

Sample Output

show virtual-chassis protocol interface

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

```
member0:
```

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

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

```
member1:
```

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

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

```
member8:
```

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

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

```
member9:
```

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

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

show virtual-chassis protocol route

Syntax	show virtual-chassis protocol route <all-members> <destination-id> <local> <member <i>member-id</i> >
Release Information	Command introduced in Junos OS Release 10.4 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
Description	Display the unicast and multicast Virtual Chassis Control Protocol (VCCP) routing tables within the Virtual Chassis or VCF.
Options	<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 <i>member-id</i>—(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>EX Series Virtual Chassis Overview</i> • <i>Understanding QFX Series Virtual Chassis</i> • Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis on page 17
List of Sample Output	show virtual-chassis protocol route on page 112
Output Fields	Table 13 on page 111 lists the output fields for the show virtual-chassis protocol route command. Output fields are listed in the approximate order in which they appear.

Table 13: show virtual-chassis protocol route Output Fields

Field Name	Field Description
Dev	MAC address of the member storing the VCCP routing table.
Version	Version of the shortest-path-first algorithm that generated the routing table.

Table 13: show virtual-chassis protocol route Output Fields (*continued*)

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

Sample Output

show virtual-chassis protocol route

```

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

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

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

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

```


vcp-0/1.32768

member8:

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

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

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

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

member9:

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

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

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

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

show virtual-chassis protocol statistics

Syntax	show virtual-chassis protocol statistics <all-members> <interface-name> <local> <member member-id>
Release Information	Command introduced in Junos OS Release 10.4 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
Description	Display the Virtual Chassis Control Protocol (VCCP) 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>EX Series Virtual Chassis Overview</i> • <i>Understanding QFX Series Virtual Chassis</i> • Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis on page 17
List of Sample Output	show virtual-chassis protocol statistics on page 115
Output Fields	Table 14 on page 114 lists the output fields for the show virtual-chassis protocol interface command. Output fields are listed in the approximate order in which they appear.

Table 14: show virtual-chassis protocol statistics Output Fields

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

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

Field Name	Field Description
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).</p> <p>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.</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • show virtual-chassis active-topology on page 95 • show virtual-chassis protocol adjacency on page 100 • <i>show virtual-chassis vc-path</i> • <i>Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis</i>
List of Sample Output	<p>show virtual-chassis (EX4200 Virtual Chassis) on page 119</p> <p>show virtual-chassis (EX8200 Virtual Chassis) on page 120</p> <p>show virtual-chassis (Virtual Chassis Fabric) on page 120</p>
Output Fields	Table 15 on page 117 lists the output fields for the show virtual-chassis command. Output fields are listed in the approximate order in which they appear.

Table 15: show virtual-chassis Output Fields

Field Name	Field Description
Fabric ID	Assigned ID used to identify the VCF.
Fabric Mode	Mode of the VCF: Enabled, Disabled, or Mixed.
Virtual Chassis ID	Assigned ID that applies to the entire Virtual Chassis or VCF.

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

Field Name	Field Description
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.
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	Mastership priority value of the member device.
Role	Role of the member device: master, backup, or linecard.

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

Field Name	Field Description
Mixed Mode	Mixed mode configuration status: <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	Location of the member device. If this field is empty, the location field was not set for the device.
Neighbor List	Member ID of the neighbor member to which this member's Virtual Chassis port (VCP) is connected.

Sample Output

show virtual-chassis (EX4200 Virtual Chassis)

```

user@switch> show virtual-chassis
Virtual Chassis ID: 0019.e250.47a0
Virtual Chassis Mode: Enabled

```

Member ID	Status	Serial No	Model	Mastership priority	Role	Mixed Mode	Neighbor List ID	Interface
0 (FPC 0)	Prsnt	AK0207360276	ex4200-24t	249	Master*	N	8	vcp-0
							1	vcp-1
1 (FPC 1)	Prsnt	AK0207360281	ex4200-24t	248	Backup	N	0	vcp-0
							2	vcp-1
2 (FPC 2)	Prsnt	AJ0207391130	ex4200-48p	247	Linecard	N	1	vcp-0
							3	vcp-1
3 (FPC 3)	Prsnt	AK0207360280	ex4200-24t	246	Linecard	N	2	vcp-0
							4	vcp-1
4 (FPC 4)	Prsnt	AJ0207391113	ex4200-48p	245	Linecard	N	3	vcp-0
							5	vcp-1
5 (FPC 5)	Prsnt	BP0207452204	ex4200-48t	244	Linecard	N	4	vcp-0
							6	vcp-1
6 (FPC 6)	Prsnt	BP0207452222	ex4200-48t	243	Linecard	N	5	vcp-0
							7	vcp-1
7 (FPC 7)	Prsnt	BR0207432028	ex4200-24f	242	Linecard	N	6	vcp-0

```

8 vcp-1
8 (FPC 8) Prsnt BR0207431996 ex4200-24f 241 Linecard N 7 vcp-0
0 vcp-1

Member ID for next new member: 9 (FPC 9)

```

show virtual-chassis (EX8200 Virtual Chassis)

```

user@external-routing-engine> show virtual-chassis
Virtual Chassis ID: c806.0842.de51
Virtual Chassis Mode: Enabled

```

Member ID	Status	Serial No	Model	Mastership priority	Role	Neighbor List ID Interface
0 (FPC 0-15)	Prsnt	BA0908380001	ex8216	0	Linecard	8 vcp-0/0 8 vcp-0/1 1 vcp-4/0/4
1 (FPC 16-31)	Prsnt	BT0909411634	ex8208	0	Linecard	8 vcp-0/0 0 vcp-3/0/4
8 (FPC 128-143)	Prsnt	062009000021	ex-xre	128	Master	9 vcp-1/0 1 vcp-1/2 9 vcp-1/3 0 vcp-2/0 9 vcp-2/1 0 vcp-1/1
9 (FPC 144-159)	Prsnt	062009000022	ex-xre	128	Backup*	8 vcp-1/0 8 vcp-1/2 8 vcp-1/3 8 vcp-1/3

show virtual-chassis (Virtual Chassis Fabric)

```

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

```

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

3 (FPC 3) Prsnt	AB3112460011 qfx5100-48s	0	Linecard	N	F	1
vcp-255/1/0						0
vcp-255/1/1						
4 (FPC 4) Prsnt	AB3112430011 qfx5100-48s	0	Linecard	N	F	1
vcp-255/1/0						0
vcp-255/1/1						

show virtual-chassis vc-port

Syntax	show virtual-chassis vc-port <all-members> <local> <member <i>member-id</i> >
Release Information	Command introduced in Junos OS Release 9.0 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
Description	Display the status of the Virtual Chassis ports (VCPs), including both the dedicated VCPs and the uplink ports configured as VCPs.
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 on page 126 • <i>Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis</i> • Verifying Virtual Chassis Ports in an EX8200 Virtual Chassis on page 70
List of Sample Output	show virtual-chassis vc-port (EX4200 Virtual Chassis) on page 124 show virtual-chassis vc-port (EX8200 Virtual Chassis) on page 124 show virtual-chassis vc-port all-members on page 125
Output Fields	Table 16 on page 122 lists the output fields for the show virtual-chassis vc-port command. Output fields are listed in the approximate order in which they appear.

Table 16: show virtual-chassis vc-port Output Fields

Field Name	Field Description
<i>fpcnumber</i>	The FPC number is the same as the member ID.

Table 16: 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 Switch as a Virtual Chassis Port (CLI Procedure)</i> or <i>“Setting a 10-Gigabit Ethernet Port as a Virtual Chassis Port in an EX8200 Virtual Chassis (CLI Procedure)” on page 36</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    Type           Trunk   Status    Speed    Neighbor
or           or              ID      (mbps)    ID       Interface
Slot/PIC/Port
vcp-0/0      Dedicated      -1      Disabled  1000
vcp-1/0      Dedicated      -1      Up        1000      8   vcp-1/0
vcp-1/1      Dedicated      -1      Down      1000
vcp-1/2      Dedicated      -1      Up        1000      8   vcp-2/1
vcp-1/3      Dedicated      -1      Up        1000      8   vcp-1/3
```

show virtual-chassis vc-port all-members

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

```
fpc0:
```

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

```
fpc1:
```

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

```
fpc2:
```

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

```
fpc3:
```

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

show virtual-chassis vc-port statistics

Syntax	<pre>show virtual-chassis vc-port statistics <all-members> <brief detail extensive > <interface-name> <local> <member member-id></pre>
Release Information	<p>Command introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>The options all-members, brief, detail, extensive, and local were added in Junos OS Release 9.3 for EX Series switches.</p> <p>Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series.</p> <p>Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).</p>
Description	Display the traffic statistics collected on Virtual Chassis ports (VCPs).
Options	<p>none—Display traffic statistics for VCPs of all members of a 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 traffic statistics for VCPs of all members of a Virtual Chassis or VCF.</p> <p>interface-name—(Optional) Display traffic statistics for the specified VCP.</p> <p>local—(Optional) Display traffic statistics for VCPs on the switch or external Routing Engine on which this command is entered.</p> <p>member member-id—(Optional) Display traffic statistics for VCPs on the specified member of a Virtual Chassis or VCF.</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none">• clear virtual-chassis vc-port statistics on page 76• show virtual-chassis vc-port on page 122• <i>Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis</i>• Verifying Virtual Chassis Ports in an EX8200 Virtual Chassis on page 70
List of Sample Output	<p>show virtual-chassis vc-port statistics on page 129</p> <p>show virtual-chassis vc-port statistics (EX8200 Virtual Chassis) on page 130</p> <p>show virtual-chassis vc-port statistics brief on page 130</p> <p>show virtual-chassis vc-port statistics extensive on page 130</p> <p>show virtual-chassis vc-port statistics member 0 on page 132</p>

Output Fields Table 17 on page 127 lists the output fields for the **show virtual-chassis vc-port statistics** command. Output fields are listed in the approximate order in which they appear.

Table 17: show virtual-chassis vc-port statistics Output Fields

Field Name	Field Description	Level of Output
fpcnumber	(All Virtual Chassis except EX8200 Virtual Chassis. VCF) ID of the Virtual Chassis member. The FPC number is the same as the member ID.	All levels
member number	(EX8200 Virtual Chassis only) Member ID of the Virtual Chassis member.	All levels
Interface	VCP name.	brief
Input Octets/Packets	Number of octets and packets received on the VCP.	brief, member, none
Output Octets/Packets	Number of octets and packets transmitted on the VCP.	brief, member, none
master: number	Member ID of the master Routing Engine.	All levels
Port	VCP for which RX (Receive) statistics, TX (Transmit) statistics, or both are reported by the VCP subsystem during a sampling interval—since the statistics counter was last cleared.	detail, extensive
Total octets	Total number of octets received and transmitted on the VCP.	detail, extensive
Total packets	Total number of packets received and transmitted on the VCP.	detail, extensive
Unicast packets	Number of unicast packets received and transmitted on the VCP.	detail, extensive
Broadcast packets	Number of broadcast packets received and transmitted on the VCP.	detail, extensive
Multicast packets	Number of multicast packets received and transmitted on the VCP.	detail, extensive
MAC control frames	Number of media access control (MAC) control frames received and transmitted on the VCP.	detail, extensive

Table 17: show virtual-chassis vc-port statistics Output Fields (*continued*)

Field Name	Field Description	Level of Output
CRC alignment errors	<p>Number of packets received on the VCP that had a length—excluding framing bits, but including frame check sequence (FCS) octets—of between 64 and 1518 octets, inclusive, and had one of the following errors:</p> <ul style="list-style-type: none"> Invalid FCS with an integral number of octets (FCS error) Invalid FCS with a nonintegral number of octets (alignment error) 	detail, extensive
Oversize packets	Number of packets received on the VCP that were longer than 1518 octets (excluding framing bits, but including FCS octets) but were otherwise well formed.	detail, extensive
Undersize packets	Number of packets received on the VCP that were shorter than 64 octets (excluding framing bits but including FCS octets) and were otherwise well formed..	detail, extensive
Jabber packets	<p>Number of packets received on the VCP that were longer than 1518 octets—excluding framing bits, but including FCS octets—and that had either an FCS error or an alignment error.</p> <p>NOTE: This definition of <i>jabber</i> is different from the definition in IEEE-802.3 section 8.2.1.5 (10Base5) and section 10.3.1.4 (10Base2). These documents define <i>jabber</i> as the condition in which any packet exceeds 20 ms. The allowed range to detect jabber is between 20 ms and 150 ms.</p>	detail, extensive
Fragments received	<p>Number of packets received on the VCP that were shorter than 64 octets (excluding framing bits, but including FCS octets), and had either an FCS error or an alignment error.</p> <p>Fragment frames normally increment because both runs (which are normal occurrences caused by collisions) and noise hits are counted.</p>	detail, extensive
Ifout errors	Number of outbound packets received on the VCP that could not be transmitted because of errors.	detail, extensive
Packet drop events	Number of outbound packets received on the VCP that were dropped, rather than being encapsulated and sent out of the switch as fragments. The packet drop counter is incremented if a temporary shortage of packet memory causes packet fragmentation to fail.	detail, extensive
64 octets frames	Number of packets received on the VCP (including invalid packets) that were 64 octets in length (excluding framing bits, but including FCS octets).	detail, extensive

Table 17: show virtual-chassis vc-port statistics Output Fields (*continued*)

Field Name	Field Description	Level of Output
65–127 octets frames	Number of packets received on the VCP (including invalid packets) that were between 65 and 127 octets in length, inclusive (excluding framing bits, but including FCS octets).	detail, extensive
128–255 octets frames	Number of packets received on the VCP (including invalid packets) that were between 128 and 255 octets in length, inclusive (excluding framing bits, but including FCS octets).	detail, extensive
256–511 octets frames	Number of packets received on the VCP (including invalid packets) that were between 256 and 511 octets in length, inclusive (excluding framing bits, but including FCS octets).	detail, extensive
512–1023 octets frames	Number of packets received on the VCP (including invalid packets) that were between 512 and 1023 octets in length, inclusive (excluding framing bits, but including FCS octets).	detail, extensive
1024–1518 octets frames	Number of packets received on the VCP (including invalid packets) that were between 1024 and 1518 octets in length, inclusive (excluding framing bits, but including FCS octets).	detail, extensive
Rate packets per second	Number of packets per second received and transmitted on the VCP.	detail, extensive
Rate bytes per second	Number of bytes per second received and transmitted on the VCP.	detail, extensive

Sample Output

show virtual-chassis vc-port statistics

```
user@switch> show virtual-chassis vc-port statistics
fpc0:
```

```
-----
Interface          Input  Octets/Packets      Output  Octets/Packets
internal-0/24       0      / 0                0      / 0
internal-0/25       0      / 0                0      / 0
internal-1/26       0      / 0                0      / 0
internal-1/27       0      / 0                0      / 0
vcp-0               0      / 0                0      / 0
vcp-1               0      / 0                0      / 0
internal-0/26       0      / 0                0      / 0
internal-0/27       0      / 0                0      / 0
internal-1/24       0      / 0                0      / 0
internal-1/25       0      / 0                0      / 0
```

```
{master:0}
```

show virtual-chassis vc-port statistics (EX8200 Virtual Chassis)

```

user@external-routing-engine> show virtual-chassis vc-port statistics
member0:
-----
Interface          Input Octets/Packets      Output Octets/Packets
vcp-4/0/4           43171238 / 48152          47687133 / 51891
vcp-4/0/7           0 / 0                     0 / 0

member1:
-----
Interface          Input Octets/Packets      Output Octets/Packets
vcp-3/0/0           0 / 0                     0 / 0
vcp-3/0/1           0 / 0                     0 / 0
vcp-3/0/4           47695376 / 51899          43180556 / 48160

member8:
-----

member9:
-----

```

show virtual-chassis vc-port statistics brief

```

user@switch> show virtual-chassis vc-port statistics brief
fpc0:
-----
Interface          Input Octets/Packets      Output Octets/Packets
internal-0/24       0 / 0                     0 / 0
internal-0/25       0 / 0                     0 / 0
internal-1/26       0 / 0                     0 / 0
internal-1/27       0 / 0                     0 / 0
vcp-0               0 / 0                     0 / 0
vcp-1               0 / 0                     0 / 0
internal-0/26       0 / 0                     0 / 0
internal-0/27       0 / 0                     0 / 0
internal-1/24       0 / 0                     0 / 0
internal-1/25       0 / 0                     0 / 0

{master:0}

```

show virtual-chassis vc-port statistics extensive

```

user@switch> show virtual-chassis vc-port statistics extensive
fpc0:
-----

```

	RX	TX
Port: internal-0/24		
Total octets:	0	0
Total packets:	0	0
Unicast packets:	0	0
Broadcast packets:	0	0
Multicast packets:	0	0
MAC control frames:	0	0
CRC alignment errors:	0	
Oversize packets:	0	
Undersize packets:	0	
Jabber packets:	0	
Fragments received:	0	

```

Ifout errors:          0
Packet drop events:    0
64      octets frames: 0
65-127   octets frames: 0
128-255  octets frames: 0
256-511  octets frames: 0
512-1023 octets frames: 0
1024-1518 octets frames: 0
Rate packets per second: 0          0
Rate bytes per second:   0          0

...

Port: vcp-0
Total octets:          0          0
Total packets:         0          0
Unicast packets:       0          0
Broadcast packets:     0          0
Multicast packets:     0          0
MAC control frames:    0          0
CRC alignment errors:  0
Oversize packets:     0
Undersize packets:     0
Jabber packets:        0
Fragments received:    0
Ifout errors:          0
Packet drop events:    0
64      octets frames: 0
65-127   octets frames: 0
128-255  octets frames: 0
256-511  octets frames: 0
512-1023 octets frames: 0
1024-1518 octets frames: 0
Rate packets per second: 0          0
Rate bytes per second:   0          0

Port: vcp-1
Total octets:          0          0
Total packets:         0          0
Unicast packets:       0          0
Broadcast packets:     0          0
Multicast packets:     0          0
MAC control frames:    0          0
CRC alignment errors:  0
Oversize packets:     0
Undersize packets:     0
Jabber packets:        0
Fragments received:    0
Ifout errors:          0
Packet drop events:    0
64      octets frames: 0
65-127   octets frames: 0
128-255  octets frames: 0
256-511  octets frames: 0
512-1023 octets frames: 0
1024-1518 octets frames: 0
Rate packets per second: 0          0
Rate bytes per second:   0          0

...

```

```
{master:0}
```

show virtual-chassis vc-port statistics member 0

```
user@switch>show virtual-chassis vc-port statistics member 0  
fpc0:
```

```
-----  
Interface           Input  Octets/Packets      Output  Octets/Packets  
internal-0/24        0           / 0                0           / 0  
internal-0/25        0           / 0                0           / 0  
internal-1/26        0           / 0                0           / 0  
internal-1/27        0           / 0                0           / 0  
vcp-0                0           / 0                0           / 0  
vcp-1                0           / 0                0           / 0  
internal-0/26        0           / 0                0           / 0  
internal-0/27        0           / 0                0           / 0  
internal-1/24        0           / 0                0           / 0  
internal-1/25        0           / 0                0           / 0
```

```
{master:0}
```

PART 4

Troubleshooting

- [Troubleshooting Procedures on page 135](#)

CHAPTER 7

Troubleshooting Procedures

- Recovering a Routing Engine in a Member Switch in an EX8200 Virtual Chassis on page 135

Recovering a Routing Engine in a Member Switch in an EX8200 Virtual Chassis

Problem **Description:** A single Routing Engine in a member switch within an EX8200 Virtual Chassis might not boot. Some typical reasons for Routing Engine boot failure include uninstalled or corrupted software.

When the boot failure occurs, the member switch remains in the u-boot prompt (= >) or in the loader prompt (loader>).

This procedure can be used to update software on a single Routing Engine of a member switch in an EX8200 Virtual Chassis whenever that Routing Engine is in the loader or u-boot prompt. It is especially useful after a Routing Engine in a member switch within an EX8200 Virtual Chassis is replaced and the software on the new Routing Engine has not been installed.

You can use this recovery installation procedure to install Junos OS to the Routing Engine of the member switch only without disrupting all traffic to the EX8200 Virtual Chassis by updating the software for the other EX8200 Virtual Chassis devices.

Environment:

Symptoms:

Solution You can resolve a boot failure on the Routing Engine by downloading a Junos OS image for an EX8200 switch into the /tftpboot directory on a directly-connected XRE200 External Routing Engine. You can then boot the switch using the downloaded Junos OS image.

To upgrade the software on the Routing Engine:

1. Log in to the master XRE200 External Routing Engine.
2. Copy a Junos OS image for an EX8200 switch into the /tftpboot directory on the external Routing Engine. The Junos OS image for the EX8200 switch should have the same release as the software currently running on the Virtual Chassis.



NOTE: You can download a Junos OS image for a standalone EX8200 switch that is independent of the Junos OS image for the XRE200 External Routing Engine as long as the image that you download has the same release as the software currently running on the Virtual Chassis.

```
user@external-routing-engine> file copy source /tftpboot
```

where ***source*** represents the name and location of the Junos OS image; for example, **tftp://192.0.2.28/junos/jinstall-ex-8200-11.4R1.5-domestic-signed.tgz**

3. Log in to the Routing Engine of the member switch in the loader or u-boot prompt.
4. From the loader or u-boot prompt, set the IP address to any address in the 128.0.0.0/24 network, set the gateway IP address to 128.0.0.1 (the external Routing Engine), and set the network mask to 255.255.254.0. Save the configuration.

From the loader prompt:

```
loader> set ipaddr=128.0.0.10
loader> set gatewayip=128.0.0.1
loader> set netmask=255.255.254.0
loader> save
```



NOTE: You can enter the **show** command in the loader> prompt to confirm these changed settings.

From the u-boot prompt:

```
=> setenv ipaddr 128.0.0.10
=> setenv gatewayip 128.0.0.1
=> setenv netmask 255.255.254.0
=> save
```

5. (u-boot prompt only) Access the loader prompt by booting the switch and pressing the Spacebar during the boot:

```
=> boot
[Space Bar]
loader>
```

6. Install the Junos OS image:

```
loader> install tftp://128.0.0.1/image-name
```

where ***image-name*** represents the name of the Junos OS image copied to the /tftpboot directory in step 2; for example, **jinstall-ex-8200-11.4R1.5-domestic-signed.tgz**.

7. Configure the switch into Virtual Chassis mode, if necessary:



NOTE: This step is only necessary if the switch fails to join the Virtual Chassis. An EX8200 member switch joins the Virtual Chassis if the Virtual Chassis mode has been enabled.

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

A reboot of the member switch is required to change the Virtual Chassis mode. You can reboot the switch by answering the prompts after entering the **set chassis virtual-chassis** command or by entering the **request system reboot** command.

**Related
Documentation**

- [Installing Software for All Devices in an EX8200 Virtual Chassis on page 40](#)
- [Installing Software for a Single Device in an EX8200 Virtual Chassis on page 42](#)
- [Understanding Software Upgrades in an EX8200 Virtual Chassis on page 14](#)

