

# Junos® OS for EX Series Ethernet Switches

---

## Virtual Chassis User Guide for EX8200 Switches

Published  
2022-06-29

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

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

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

*Junos® OS for EX Series Ethernet Switches Virtual Chassis User Guide for EX8200 Switches*  
Copyright © 2022 Juniper Networks, Inc. All rights reserved.

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

## YEAR 2000 NOTICE

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

## END USER LICENSE AGREEMENT

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

# Table of Contents

**About This Guide | vii**

1

## **Overview**

**Virtual Chassis Overview | 2**

EX8200 Virtual Chassis Overview | 2

Understanding EX8200 Virtual Chassis Components | 3

Understanding EX8200 Virtual Chassis Topologies | 6

Understanding Virtual Chassis Member ID Numbering in an EX8200 Virtual Chassis | 13

Understanding Virtual Chassis Roles in an EX8200 Virtual Chassis | 13

Understanding Virtual Chassis Ports in an EX8200 Virtual Chassis | 14

Understanding Software Upgrades in an EX8200 Virtual Chassis | 16

Understanding Nonstop Software Upgrade on EX Series Switches | 16

Understanding Virtual Chassis Port Link Aggregation in an EX8200 Virtual Chassis | 26

Understanding Link Aggregation in an EX8200 Virtual Chassis | 26

Understanding File Storage in an EX8200 Virtual Chassis | 27

Understanding EX8200 Virtual Chassis Compatibility Requirements | 28

Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis | 29

Understanding Global Management of an EX8200 Virtual Chassis | 29

Understanding Long-Distance Virtual Chassis Port Configuration for an EX8200 Virtual Chassis | 30

Understanding Configuration File Management in an EX8200 Virtual Chassis | 31

Network Port Interface Names on an EX8200 Virtual Chassis | 31

Virtual Chassis Port (VCP) Interface Names in an EX8200 Virtual Chassis | 32

2

## **Configuration**

**Configuration Examples | 36**

Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines | **36**

Requirements | **36**

Overview and Topology | **37**

Setting the EX8200 Switches as Virtual Chassis Members | **40**

Preprovisioning the Virtual Chassis | **40**

Configuring a VCP Link from Member Switch 0 to Member Switch 1 | **42**

Configuring a VCP Link from Member Switch 1 to Member Switch 0 | **43**

Verification | **44**

## Configuration Tasks and Examples | **46**

Configuring an EX8200 Virtual Chassis (CLI Procedure) | **46**

Adding or Replacing a Member Switch or an External Routing Engine in an EX8200 Virtual Chassis (CLI Procedure) | **49**

Setting a 10-Gigabit Ethernet Port as a Virtual Chassis Port in an EX8200 Virtual Chassis (CLI Procedure) | **51**

Configuring a Long-Distance Virtual Chassis Port Connection for an EX8200 Virtual Chassis | **52**

Installing Software for a Single Device in an EX8200 Virtual Chassis | **56**

Installing Software for All Devices in an EX8200 Virtual Chassis | **58**

Performing a Software Recovery Installation for an XRE200 External Routing Engine | **59**

Configuring Line-Card Upgrade Groups for Nonstop Software Upgrade | **61**

How Line-card Upgrade Groups Work with Nonstop Software Upgrade | **62**

Line-card Upgrade Groups Support | **62**

Configure Line-Card Upgrade Groups on an EX4650 Virtual Chassis, a QFX Series Virtual Chassis or a QFX5100 VCF | **63**

Configure Line-Card Upgrade Groups on Standalone EX6200 or EX8200 Switches | **63**

Configure Line-Card Upgrade Groups on an EX8200 Virtual Chassis | **64**

Example: Configuring Line-Card Upgrade Groups for Nonstop Software Upgrade on EX Series Switches | **66**

Requirements | **66**

Overview and Topology | **66**

Configuration | **68**

Upgrading Software on an EX8200 Virtual Chassis Using Nonstop Software Upgrade (CLI Procedure) | **70**

Preparing the Switch for Software Installation | 70

Upgrading the Software Using NSSU | 71

Using Groups to Configure Power Supply Management, Power Budgets, and LCD Menus on EX8200  
Member Switches in an EX8200 Virtual Chassis (CLI Procedure) | 75

## **Configuration Statements | 78**

fpcs (NSSU Upgrade Groups) | 78

graceful-restart (Enabling Globally) | 80

id | 83

location (Virtual Chassis) | 84

mac-persistence-timer | 86

member | 88

member (NSSU Upgrade Groups) | 90

no-split-detection | 92

nssu | 93

preprovisioned | 95

role | 97

serial-number | 100

traceoptions (Virtual Chassis) | 101

upgrade-group | 105

virtual-chassis | 106

## 3

## **Administration**

### **Routine Monitoring | 111**

Verifying the Member ID, Role, and Neighbor Member Connections of an EX8200 Virtual Chassis  
Member | 111

Verifying Virtual Chassis Ports in an EX8200 Virtual Chassis | 112

Verifying Hard Disk Memory Status on an XRE200 External Routing Engine | 114

### **Operational Commands | 117**

clear virtual-chassis protocol statistics | 118

clear virtual-chassis vc-port statistics | 120

request chassis routing-engine hard-disk-test | 122

request session member | 127

request system software nonstop-upgrade | 129

request system software recovery-package | 142

request virtual-chassis device-reachability | 145

request virtual-chassis reactivate | 149

request virtual-chassis recycle | 150

request virtual-chassis renumber | 152

request virtual-chassis vc-port | 153

set chassis virtual-chassis | 157

show chassis nonstop-upgrade | 158

show system software recovery-package | 161

show virtual-chassis active-topology | 164

show virtual-chassis protocol adjacency | 170

show virtual-chassis protocol database | 175

show virtual-chassis protocol interface | 181

show virtual-chassis protocol route | 184

show virtual-chassis protocol statistics | 188

show virtual-chassis | 193

show virtual-chassis vc-port | 202

show virtual-chassis vc-port statistics | 209

## 4

## Troubleshooting

**Troubleshooting Procedures | 220**

Recovering a Routing Engine in a Member Switch in an EX8200 Virtual Chassis | 220

# About This Guide

Use this guide to set up and configure an EX8200 Virtual Chassis composed of EX8200 switches interconnected together with an XRE200 Routing Engine that operate and are managed as a single network entity. Refer also to the EX8200 switch hardware documentation for more details on how to physically connect EX8200 Virtual Chassis components.

# 1

PART

## Overview

---

[Virtual Chassis Overview](#) | 2

---



# Virtual Chassis Overview

## IN THIS CHAPTER

- EX8200 Virtual Chassis Overview | 2
- Understanding EX8200 Virtual Chassis Components | 3
- Understanding EX8200 Virtual Chassis Topologies | 6
- Understanding Virtual Chassis Member ID Numbering in an EX8200 Virtual Chassis | 13
- Understanding Virtual Chassis Roles in an EX8200 Virtual Chassis | 13
- Understanding Virtual Chassis Ports in an EX8200 Virtual Chassis | 14
- Understanding Software Upgrades in an EX8200 Virtual Chassis | 16
- Understanding Nonstop Software Upgrade on EX Series Switches | 16
- Understanding Virtual Chassis Port Link Aggregation in an EX8200 Virtual Chassis | 26
- Understanding Link Aggregation in an EX8200 Virtual Chassis | 26
- Understanding File Storage in an EX8200 Virtual Chassis | 27
- Understanding EX8200 Virtual Chassis Compatibility Requirements | 28
- Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis | 29
- Understanding Global Management of an EX8200 Virtual Chassis | 29
- Understanding Long-Distance Virtual Chassis Port Configuration for an EX8200 Virtual Chassis | 30
- Understanding Configuration File Management in an EX8200 Virtual Chassis | 31
- Network Port Interface Names on an EX8200 Virtual Chassis | 31
- Virtual Chassis Port (VCP) Interface Names in an EX8200 Virtual Chassis | 32

## 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 6.

## RELATED DOCUMENTATION

[Understanding EX8200 Virtual Chassis Components](#) | 3

[Understanding Virtual Chassis Member ID Numbering in an EX8200 Virtual Chassis](#) | 13

[Understanding Virtual Chassis Ports in an EX8200 Virtual Chassis](#) | 14

[Understanding EX8200 Virtual Chassis Compatibility Requirements](#) | 28

[Understanding Virtual Chassis Roles in an EX8200 Virtual Chassis](#) | 13

[Configuring an EX8200 Virtual Chassis \(CLI Procedure\)](#) | 46

## Understanding EX8200 Virtual Chassis Components

### IN THIS SECTION

- [Virtual Chassis Ports \(VCPs\)](#) | 4
- [Primary External Routing Engine Role](#) | 4
- [Backup External Routing Engine Role](#) | 5
- [Linecard Role](#) | 5
- [Member Switch and Member ID](#) | 5
- [Virtual Chassis Identifier \(VCID\)](#) | 6

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.

## 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 30.](#)

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.

## Primary External Routing Engine Role

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

The primary role in an EX8200 Virtual Chassis is assigned to an XRE200 External Routing Engine only. The XRE200 External Routing Engine in the primary 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 primary role. When a backup external Routing Engine is part of the Virtual Chassis, the external Routing Engine with the most uptime functions as the primary.

## 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 primary role if the primary fails.
- Runs Junos OS.
- Synchronizes with the primary, so that it is prepared to preserve routing information and maintain network connectivity without disruption if the primary 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 primary.

## 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 1 on page 6 summarizes EX8200 Virtual Chassis roles and member IDs.

**Table 1: EX8200 Virtual Chassis Roles and Member IDs**

Hardware	Role	Member IDs
EX8208 or EX8216 switch	Linecard	0-7
XRE200 External Routing Engine	Primary 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 | 2](#)

[Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines | 36](#)

[Configuring an EX8200 Virtual Chassis \(CLI Procedure\) | 46](#)

## Understanding EX8200 Virtual Chassis Topologies

### IN THIS SECTION

- [Understanding the Full Mesh EX8200 Virtual Chassis Topology | 7](#)
- [Understanding the Ring EX8200 Virtual Chassis Topology | 10](#)
- [The Role of XRE200 External Routing Engine Redundancy in a Virtual Chassis Topology | 11](#)
- [EX8200 Virtual Chassis Basic Topology | 12](#)

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.

## **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 9](#) 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

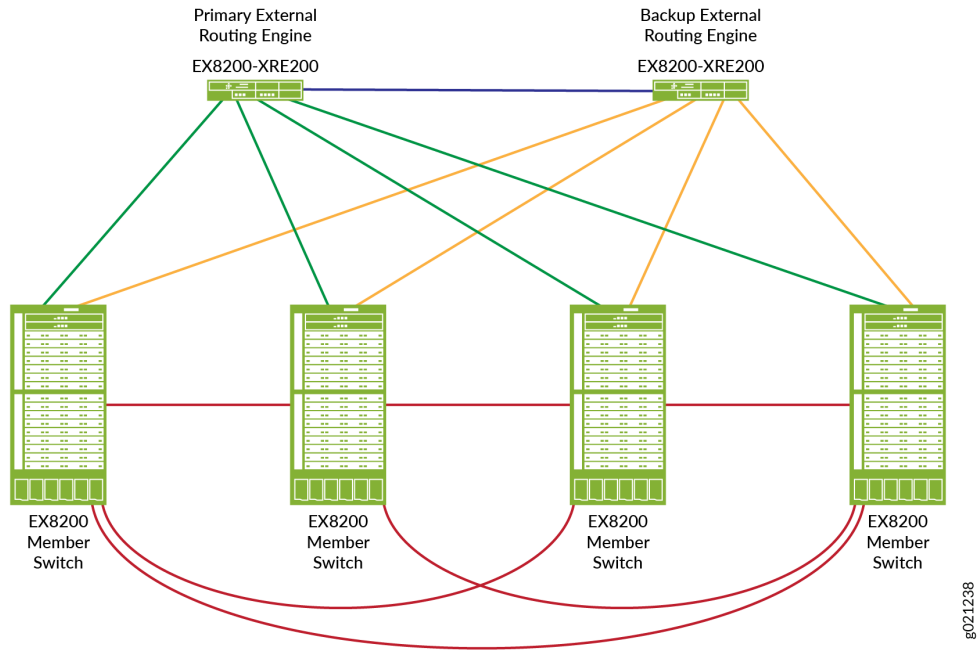
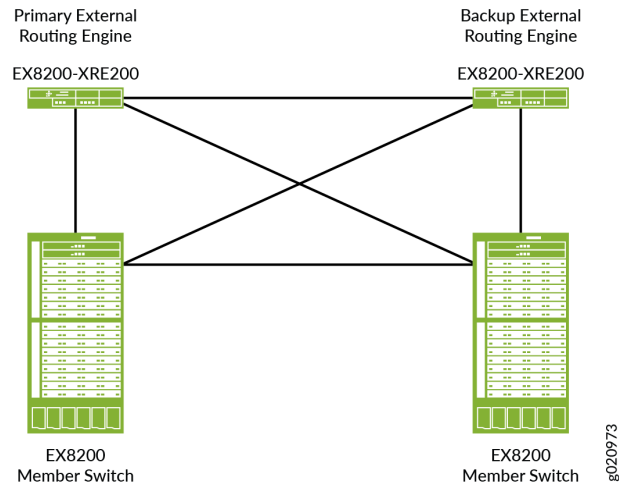




Figure 2 on page 10 shows a full mesh EX8200 Virtual Chassis topology with two member switches.

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



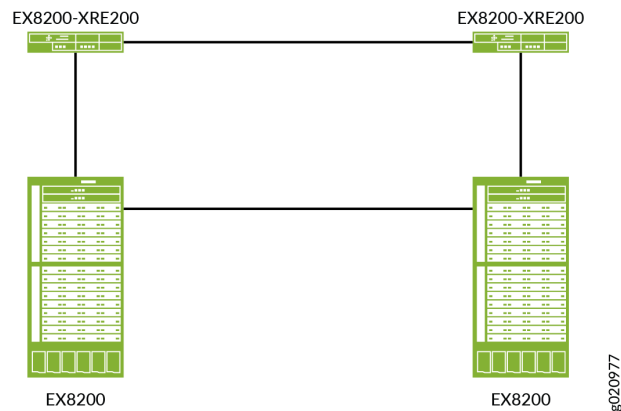
A full mesh Virtual Chassis topology has several advantages over other topologies. Because there are multiple paths to all devices in the Virtual Chassis, a single link failure does not disable any path from one device to another. Because the multiple VCP links connect all devices together, the VCCP 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 11 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 primary XRE200 External Routing Engine. If a primary 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.

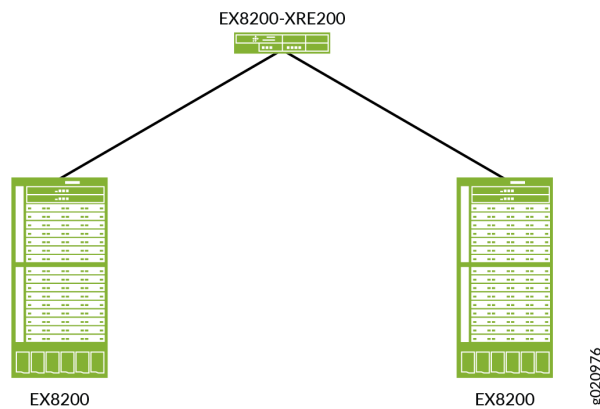
When a backup external Routing Engine is included in the EX8200 Virtual Chassis, one link must be used to connect the primary external Routing Engine to the backup external Routing Engine. Both EX8200 switches must also have connections to the backup external Routing Engine.

## 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 12](#) 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 | 2](#)

[Understanding EX8200 Virtual Chassis Components | 3](#)

[Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines | 36](#)

## 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 primary 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 primary 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 primary role. The role has no correlation to member ID; a primary or backup XRE200 External Routing Engine can be member ID 8 or 9.

[Table 2 on page 13](#) summarizes EX8200 Virtual Chassis member IDs and roles.

**Table 2: EX8200 Virtual Chassis Roles and Member IDs**

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

### RELATED DOCUMENTATION

[EX8200 Virtual Chassis Overview | 2](#)

[Understanding EX8200 Virtual Chassis Components | 3](#)

[Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines | 36](#)

## 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 primary role; the external

Routing Engine with less uptime assumes the backup role. EX8200 member switches are always in the linecard role.

[Table 3 on page 14](#) provides a summary of the EX8200 Virtual Chassis roles.

**Table 3: EX8200 Virtual Chassis Roles**

Role	Description
Primary	<p>The primary XRE200 External Routing Engine.</p> <p>The primary 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 primary 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 | 3](#)

[EX8200 Virtual Chassis Overview | 2](#)

## 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 Routine 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.

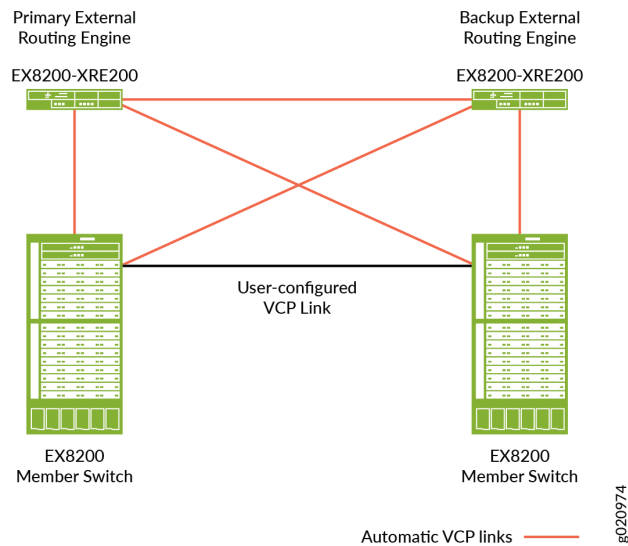
**NOTE:** To connect a 1000BASE-X SFP interface on the XRE200 External Routing Engine to an EX8200 switch, you must use an intermediary switch. The intermediary switch is required because the management interface (labeled **MGMT**) on the Routing Engine (RE) module or the Switch Fabric and Routing Engine (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 30.

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 26.

[Figure 5 on page 15](#) 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 | 2](#)

[Setting a 10-Gigabit Ethernet Port as a Virtual Chassis Port in an EX8200 Virtual Chassis \(CLI Procedure\) | 51](#)

## Understanding Software Upgrades in an EX8200 Virtual Chassis

In an EX8200 *Virtual Chassis*, both XRE200 External Routing Engines and the primary 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 | 58](#)

[Installing Software for a Single Device in an EX8200 Virtual Chassis | 56](#)

[EX8200 Virtual Chassis Overview | 2](#)

## Understanding Nonstop Software Upgrade on EX Series Switches

### SUMMARY

### IN THIS SECTION

[Requirements for Performing an NSSU | 19](#)

Nonstop software upgrade (NSSU) is a feature that enables the upgrade of all supported EX Series switches in a network with a single command.

- [How an NSSU Works | 20](#)
- [NSSU Limitations | 23](#)
- [NSSU and Junos OS Release Support | 23](#)
- [Overview of NSSU Configuration and Operation | 25](#)

Nonstop software upgrade (NSSU) enables you to upgrade the software running on Juniper Networks EX Series Ethernet Switches with redundant Routing Engines and all member switches in EX Series Virtual Chassis using a single command. During the upgrade there might be minimal network traffic disruption during primary-role switchover, and the extent of disruption could be dependent on the network topology, configuration, network traffic, and other environment factors .

**NOTE:** When an EX Series switch in a mixed Virtual Chassis is upgraded to Junos OS Release 15.1 or later from a release earlier than Release 15.1, there might be a drop in traffic for up to 60 seconds.

The following EX Series Virtual Chassis support NSSU:

- EX3300 Virtual Chassis
- EX3400 Virtual Chassis
- EX4200 Virtual Chassis
- EX4300 Virtual Chassis
- EX4400 Virtual Chassis
- EX4500 Virtual Chassis
- EX4550 Virtual Chassis
- All mixed Virtual Chassis composed of EX4200, EX4500, and EX4550 switches
- EX4600 Virtual Chassis
- EX4650 Virtual Chassis



**NOTE:** An EX4650 Virtual Chassis operates the same as a QFX5120 Virtual Chassis, so for details on upgrading an EX4650 Virtual Chassis using NSSU, see *Understanding Nonstop Software Upgrade on a Virtual Chassis and Mixed Virtual Chassis* and *Upgrading Software on a Virtual Chassis and Mixed Virtual Chassis Using Nonstop Software Upgrade* instead of this topic.

- EX6200 switches
- EX8200 switches
- EX8200 Virtual Chassis

Performing an NSSU provides these benefits:

- No disruption to the control plane—An NSSU takes advantage of *graceful Routing Engine switchover* (GRES) and *nonstop active routing* (NSR) to ensure no disruption to the control plane. During the upgrade process, interface, kernel, and routing protocol information is preserved.
- Minimal disruption to network traffic—An NSSU minimizes network traffic disruption by:
  - Upgrading line cards one at a time in an EX6200 switch, EX8200 switch, or EX8200 Virtual Chassis while permitting traffic to continue to flow through the line cards that are not being upgraded.
  - Upgrading member switches one at a time in other EX Series Virtual Chassis while permitting traffic to continue to flow through the members that are not being upgraded.

To achieve minimal disruption to traffic, you must configure link aggregation groups (LAGs) such that the member links of each LAG reside on different line cards or Virtual Chassis members. When one member link of a LAG is down, the remaining links are up, and traffic continues to flow through the LAG.

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

In releases prior to Junos OS Release 16.1, for EX6200 switches, EX8200 switches, and EX8200 Virtual Chassis, you can reduce the amount of time an upgrade takes by configuring line-card upgrade groups. The line cards in an upgrade group are upgraded simultaneously, reducing the amount of time it takes to complete an upgrade. See *Configuring Line-Card Upgrade Groups for Nonstop Software Upgrade*.

## Requirements for Performing an NSSU

The following requirements apply to all switches and Virtual Chassis:

**NOTE:** NSSU can only upgrade up to three major releases ahead of the current release on a device. To upgrade to a release more than three releases ahead of the current release on a device, use the NSSU process to upgrade the switch to one or more intermediate releases until the switch is within three major releases of the target release.

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

**NOTE:** Although nonstop bridging (NSB) does not have to be enabled to perform an NSSU, we recommend enabling NSB before performing an NSSU. Enabling NSB ensures that all NSB-supported Layer 2 protocols operate seamlessly during the Routing Engine switchover that is part of the NSSU. In releases prior to Junos OS Release 16.1, see [Configuring Nonstop Bridging on Switches \(CLI Procedure\)](#).

- For minimal traffic disruption, you must define link aggregation groups (LAGs) such that the member links reside on different Virtual Chassis members or on different line cards.

**NOTE:** During an NSSU operation, if you try to view LAG interface status on the primary Routing Engine member using the `show interfaces ae-ae-interface-number` CLI command, you might see incorrect or zero traffic counts. To work around this problem, run the command on the backup Routing Engine member instead if that member is already loaded and running.

The following are requirements for performing NSSU on an EX Series Virtual Chassis (excluding EX6200 or EX8200 Virtual Chassis):

- The Virtual Chassis members must be connected in a ring topology so that no member is isolated as a result of another member being rebooted. This topology prevents the Virtual Chassis from splitting during an NSSU.
- The Virtual Chassis primary and backup must be adjacent to each other in the ring topology. Adjacency permits the primary and backup to always be in sync, even when the switches in linecard roles are rebooting.

- The Virtual Chassis must be preprovisioned so that the linecard role has been explicitly assigned to member switches acting in a linecard role. During an NSSU, the Virtual Chassis members must maintain their roles—the primary and backup must maintain their primary and backup roles (although primary role will change), and the remaining switches must maintain their linecard roles.
- A two-member Virtual Chassis must have `no-split-detection` configured so that the Virtual Chassis does not split when an NSSU upgrades a member.

**NOTE:** For the EX4300 Virtual Chassis, you should enable the `vcp-no-hold-time` statement at the `[edit virtual-chassis]` hierarchy level before performing a software upgrade using NSSU. If you do not enable the `vcp-no-hold-time` statement, the Virtual Chassis might split during the upgrade. A split Virtual Chassis can cause disruptions to your network, and you might have to manually reconfigure your Virtual Chassis after the NSSU if the split and merge feature was disabled. For more information about a split Virtual Chassis, see *Understanding Split and Merge in a Virtual Chassis*

## How an NSSU Works

This section describes what happens when you request an NSSU on EX Series switches and Virtual Chassis.

**NOTE:** An EX4650 Virtual Chassis operates the same as a QFX5120 Virtual Chassis, so for details on upgrading an EX4650 Virtual Chassis using NSSU, see *Understanding Nonstop Software Upgrade on a Virtual Chassis and Mixed Virtual Chassis* and *Upgrading Software on a Virtual Chassis and Mixed Virtual Chassis Using Nonstop Software Upgrade* instead of this topic.

### EX3300, EX3400, EX4200, EX4300, EX4400, EX4500, EX4600, and Mixed Virtual Chassis

When you request an NSSU on an EX3300, EX3400, EX4200, EX4300, EX4400, EX4500, or mixed Virtual Chassis:

1. The Virtual Chassis primary verifies that:
  - The backup is online and running the same software version.
  - Graceful Routing Engine switchover (GRES) and nonstop active routing (NSR) are enabled.
  - The Virtual Chassis has a preprovisioned configuration.
2. The primary installs the new software image on the backup and reboots it.

3. The primary resynchronizes the backup.
4. The primary installs the new software image on member switches that are in the linecard role and reboots them, one at a time. The primary waits for each member to become online and active before starting the software upgrade on the next member.
5. When all members that are in the linecard role have been upgraded, the primary performs a graceful Routing Engine switchover, and the upgraded backup becomes the primary.
6. The software on the original primary is upgraded and the original primary is automatically rebooted. After the original primary has rejoined the Virtual Chassis, you can optionally return control to it by requesting a graceful Routing Engine switchover.

### EX6200 and EX8200 Switches

When you request an NSSU on a standalone switch with redundant Routing Engines:

1. The switch verifies that:
  - Both Routing Engines are online and running the same software version.
  - Both Routing Engines have sufficient storage space for the new software image.
  - Graceful Routing Engine switchover and nonstop active routing are enabled.
2. The switch installs the new software image on the backup Routing Engine and reboots it.
3. The switch resynchronizes the backup Routing Engine to the primary Routing Engine.
4. The line cards in the first upgrade group (or the line card in slot 0, if no upgrade groups are defined) download the new image and then restart. Traffic continues to flow through the line cards in the other upgrade groups during this process.
5. When line cards restarted in Step 4 are online again, the line cards in the next upgrade group download the new image and restart. This process continues until all online line cards have restarted with the new software.

**NOTE:** If you have taken a line card offline with the CLI before you start the NSSU, the line card is not restarted and remains offline.

6. The switch performs a graceful Routing Engine switchover, so that the upgraded backup Routing Engine becomes the primary.
7. The switch installs the new software on the original primary Routing Engine.

To complete the upgrade process, the original primary Routing Engine must be rebooted. You can do so manually or have the switch perform an automatic reboot by including the `reboot` option when you request the NSSU. After the original primary has been rebooted, you can optionally return control to it by requesting a graceful Routing Engine switchover.

8. (EX6200 switch only) The original primary Routing Engine reboots to complete the software upgrade.

**NOTE:** To complete the upgrade process on an EX8200 switch, you must intervene to reboot the original primary Routing Engine. You can reboot the original primary Routing Engine manually or have the switch perform an automatic reboot by including the `reboot` option when you request the NSSU.

9. (Optional) After the original primary has been rebooted, you can return control to it by requesting a graceful Routing Engine switchover.

The switch can maintain normal operations with either Routing Engine acting as the primary Routing Engine after the software upgrade, so you only have to perform this switchover if you want to return Routing Engine control to the original primary Routing Engine.

## EX8200 Virtual Chassis

When you request an NSSU on an EX8200 Virtual Chassis:

1. The primary external Routing Engine verifies that:
  - It has a backup external Routing Engine that is online.
  - All Virtual Chassis members have redundant Routing Engines and the Routing Engines are online.
  - All Routing Engines are running the same software version.
  - All Routing Engines have sufficient storage space for the new software image.
  - Graceful Routing Engine switchover and nonstop active routing (NSR) are enabled.
2. The primary external Routing Engine installs the new software image on the backup external Routing Engine and reboots it.
3. The backup external Routing Engine resynchronizes with the primary external Routing Engine.
4. The primary external Routing Engine installs the new software on the backup Routing Engines in the member switches and reboots the backup Routing Engines.
5. When the reboot of the backup Routing Engines complete, the line cards in the first upgrade group download the new image and then restart. (If no upgrade groups are defined, the line card in slot 0 of

member 0 downloads the new image and restarts.) Traffic continues to flow through the line cards in the other upgrade groups during this process.

6. When line cards restarted in Step 5 are online again, the line cards in the next upgrade group (or the next sequential line card) download the new image and restart. This process continues until all online line cards have restarted with the new software.

**NOTE:** If you have taken a line card offline with the CLI before you start the NSSU, the line card is not restarted and remains offline.

7. The new software image is installed on the primary Routing Engines, both external and internal.
8. The member switches perform a graceful Routing Engine switchover, so that the upgraded backup Routing Engines become primaries.
9. The primary external Routing Engine performs a graceful Routing Engine switchover so that the backup external Routing Engine is now the primary.

To complete the upgrade process, the original primary Routing Engines, both external and internal, must be rebooted. You can do so manually by establishing a console connection to each Routing Engine or have the reboot performed automatically by including the `reboot` option when you request the NSSU. After the original primary external Routing Engine has been rebooted, you can optionally return control to it by requesting a graceful Routing Engine switchover.

## NSSU Limitations

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

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

## NSSU and Junos OS Release Support

A Virtual Chassis must be running a Junos OS release that supports NSSU before you can perform an NSSU. If a Virtual Chassis is running a software version that does not support NSSU, use the `request system software add` command.

[Table 4 on page 24](#) lists the EX Series switches and Virtual Chassis that support NSSU and the Junos OS release at which they began supporting it.

**Table 4: Platform and Release Support for NSSU**

Platform	Junos OS Release
EX3300 Virtual Chassis	12.2 or later
EX3400 Virtual Chassis	15.1X53-D55
EX4200 Virtual Chassis	12.1 or later
EX4300 Virtual Chassis	13.2X51-D20 or later
EX4300 Multigigabit Virtual Chassis	18.2R1 or later
EX4400 Virtual Chassis	21.1 or later
EX4400 Multigigabit Virtual Chassis	21.2 or later
EX4500 Virtual Chassis	12.1 or later
EX4550 Virtual Chassis	12.2 or later
Mixed EX4200 and EX4500 Virtual Chassis	12.1 or later
Mixed EX4200 and EX4550 Virtual Chassis	12.2 or later
Mixed EX4200, EX4500, and EX4550 Virtual Chassis	12.2 or later
Mixed EX4500 and EX4550 Virtual Chassis	12.2 or later
Mixed EX4300 and EX4600 Virtual Chassis	13.2X51-D25 or later
EX6200 switch	12.2 or later
EX8200 switch	10.4 or later

**Table 4: Platform and Release Support for NSSU (Continued)**

Platform	Junos OS Release
EX8200 Virtual Chassis	11.1 or later

## Overview of NSSU Configuration and Operation

You must ensure that the configuration of the switch or Virtual Chassis meets the requirements described in ["Requirements for Performing an NSSU" on page 19](#). NSSU requires no additional configuration.

In releases prior to Junos OS Release 16.1, for EX6200 switches, EX8200 switches, and EX8200 Virtual Chassis, you can optionally configure line-card upgrade groups using the CLI. See *Example: Configuring Line-Card Upgrade Groups for Nonstop Software Upgrade on EX Series Switches*.

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

### Release History Table

Release	Description
16.1	In releases prior to Junos OS Release 16.1, for EX6200 switches, EX8200 switches, and EX8200 Virtual Chassis, you can reduce the amount of time an upgrade takes by configuring line-card upgrade groups.

## RELATED DOCUMENTATION

*Upgrading Software Using Nonstop Software Upgrade on EX Series Virtual Chassis and Mixed Virtual Chassis (CLI Procedure)*

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

*Upgrading Software on an EX8200 Virtual Chassis Using Nonstop Software Upgrade (CLI Procedure)*

[Configuring Nonstop Active Routing on Switches](#)

*Configuring Graceful Routing Engine Switchover in a Virtual Chassis*

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



## 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 26.

### RELATED DOCUMENTATION

[EX8200 Virtual Chassis Overview | 2](#)

[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 26](#).

## RELATED DOCUMENTATION

| [EX8200 Virtual Chassis Overview](#) | 2

## 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 primary 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 114.

## RELATED DOCUMENTATION

[EX8200 Virtual Chassis Overview | 2](#)

[Understanding Configuration File Management in an EX8200 Virtual Chassis | 31](#)

[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 primary 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 16.

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 | 2](#)

[Understanding EX8200 Virtual Chassis Components | 3](#)

## 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 | 2](#)

## 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 primary 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 primary external Routing Engine or issue the `request session member` command to redirect your session to the primary external Routing Engine.

## RELATED DOCUMENTATION

[EX8200 Virtual Chassis Overview | 2](#)

*Connect a Device 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 30](#).

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

[Understanding Long-Distance Virtual Chassis Port Configuration for an EX8200 Virtual Chassis | 30](#)

## Understanding Configuration File Management in an EX8200 Virtual Chassis

You configure all devices in an EX8200 *Virtual Chassis* by logging in to the primary external Routing Engine. The configuration file on the primary 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 primary 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 primary 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 75.

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 | 2](#)

[Understanding File Storage in an EX8200 Virtual Chassis | 27](#)

## 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 32.

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](#) | 2

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

### IN THIS SECTION

- [Virtual Chassis Control Interface \(VCCI\) Module VCP Name on the XRE200 External Routing Engine](#) | 33
- [Native VCP Name on the XRE200 External Routing Engine](#) | 33
- [EX8200 Switch to XRE200 External Routing Engine VCP Name](#) | 33
- [EX8200 Switch to EX8200 Switch VCP Name](#) | 34

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, which the following sections describe.

### 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 | 2](#)

[Setting a 10-Gigabit Ethernet Port as a Virtual Chassis Port in an EX8200 Virtual Chassis \(CLI Procedure\) | 51](#)

# 2

PART

## Configuration

---

[Configuration Examples](#) | 36

[Configuration Tasks and Examples](#) | 46

[Configuration Statements](#) | 78

---

# Configuration Examples

## IN THIS CHAPTER

- [Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines | 36](#)

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

### IN THIS SECTION

- [Requirements | 36](#)
- [Overview and Topology | 37](#)
- [Setting the EX8200 Switches as Virtual Chassis Members | 40](#)
- [Preprovisioning the Virtual Chassis | 40](#)
- [Configuring a VCP Link from Member Switch 0 to Member Switch 1 | 42](#)
- [Configuring a VCP Link from Member Switch 1 to Member Switch 0 | 43](#)
- [Verification | 44](#)

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

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 Virtual Chassis or 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 primary 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 primary XRE200 External Routing Engine, a backup XRE200 External Routing engine, an EX8208 member switch, and an EX8216 member switch.

[Table 5 on page 37](#) explains the components of the example EX8200 Virtual Chassis topology.

**Table 5: Components of the EX8200 Virtual Chassis Topology**

Property	Settings
Hardware	<ul style="list-style-type: none"> <li>• Two XRE200 External Routing Engines</li> <li>• Member EX8208 switch</li> <li>• Member EX8216 switch</li> </ul>

Table 5: Components of the EX8200 Virtual Chassis Topology (Continued)

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

Table 5: Components of the EX8200 Virtual Chassis Topology (Continued)

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

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

### IN THIS SECTION

- [Procedure | 40](#)

### Procedure

#### Step-by-Step Procedure

To configure the EX8200 switches as Virtual Chassis members:

On the EX8208 switch:

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

On the EX8216 switch:

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

## Preprovisioning the Virtual Chassis

### IN THIS SECTION

- [CLI Quick Configuration | 41](#)
- [Procedure | 41](#)
- [Results | 42](#)

## CLI Quick Configuration

To quickly preprovision the Virtual Chassis, log in to the primary 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
```

## Procedure

### Step-by-Step Procedure

To preprovision the EX8200 Virtual Chassis:

1. Log in to the primary 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

### IN THIS SECTION

- [Procedure | 43](#)

## Procedure

### 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

#### IN THIS SECTION

- [Procedure | 43](#)

## Procedure

### 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

### IN THIS SECTION

- [Verifying That the Virtual Chassis Is Operational | 44](#)

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

### 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
```

Member ID	Status	Serial No	Model	Mastership priority	Role	Neighbor List ID Interface
0 (FPC 0-15)	Prsnt	ghi789	ex8208	0	Linecard	8 vcp-0/0
						9 vcp-0/1
						1 vcp-3/0/2
1 (FPC 16-31)	Prsnt	jkl112	ex8216	0	Linecard	8 vcp-0/0
						9 vcp-0/1
						0 vcp-4/0/2
8 (FPC 128-143)	Prsnt	abc123	ex-xre	129	Master*	9 vcp-1/0
						0 vcp-1/1
						1 vcp-1/2
9 (FPC 144-159)	Prsnt	def456	ex-xre	129	Backup	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 primary 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 \(CLI Procedure\) | 46](#)

[Setting a 10-Gigabit Ethernet Port as a Virtual Chassis Port in an EX8200 Virtual Chassis \(CLI Procedure\) | 51](#)

# Configuration Tasks and Examples

## IN THIS CHAPTER

- [Configuring an EX8200 Virtual Chassis \(CLI Procedure\) | 46](#)
- [Adding or Replacing a Member Switch or an External Routing Engine in an EX8200 Virtual Chassis \(CLI Procedure\) | 49](#)
- [Setting a 10-Gigabit Ethernet Port as a Virtual Chassis Port in an EX8200 Virtual Chassis \(CLI Procedure\) | 51](#)
- [Configuring a Long-Distance Virtual Chassis Port Connection for an EX8200 Virtual Chassis | 52](#)
- [Installing Software for a Single Device in an EX8200 Virtual Chassis | 56](#)
- [Installing Software for All Devices in an EX8200 Virtual Chassis | 58](#)
- [Performing a Software Recovery Installation for an XRE200 External Routing Engine | 59](#)
- [Configuring Line-Card Upgrade Groups for Nonstop Software Upgrade | 61](#)
- [Example: Configuring Line-Card Upgrade Groups for Nonstop Software Upgrade on EX Series Switches | 66](#)
- [Upgrading Software on an EX8200 Virtual Chassis Using Nonstop Software Upgrade \(CLI Procedure\) | 70](#)
- [Using Groups to Configure Power Supply Management, Power Budgets, and LCD Menus on EX8200 Member Switches in an EX8200 Virtual Chassis \(CLI Procedure\) | 75](#)

## 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 51.](#)

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 primary 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 Virtual Chassis or 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 primary role, so the external Routing Engine that is powered on in this step is the primary external Routing Engine.
7. Log in to the primary external Routing Engine and specify the preprovisioned configuration mode. If the external Routing Engine primary role 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
```

- 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.

- 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 51](#).
- 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](#) | 36

[Adding or Replacing a Member Switch or an External Routing Engine in an EX8200 Virtual Chassis \(CLI Procedure\)](#) | 49

[Verifying the Member ID, Role, and Neighbor Member Connections of an EX8200 Virtual Chassis Member](#) | 111

## 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. Unconfigure 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 Virtual Chassis or 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 primary 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 \(CLI Procedure\) | 46](#)

[Verifying the Member ID, Role, and Neighbor Member Connections of an EX8200 Virtual Chassis Member | 111](#)

## 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 14

**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 14.

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 26.

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 | 36](#)

[Configuring an EX8200 Virtual Chassis \(CLI Procedure\) | 46](#)

[Verifying the Member ID, Role, and Neighbor Member Connections of an EX8200 Virtual Chassis Member | 111](#)

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

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.

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 100BASE-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 51.

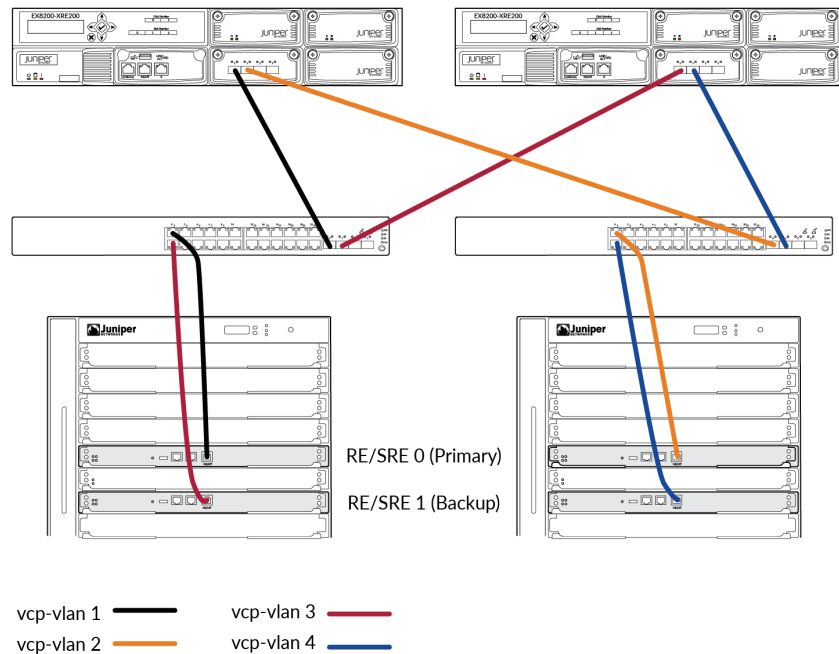
You can connect an XRE200 External Routing Engine to an EX8200 switch to form a Virtual Chassis over a longer distance by using an intermediary switch. The intermediary switch should be located within 328 feet (100 meters) of the EX8200 member switch to ensure a Gigabit Ethernet connection to the management 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.

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 54 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



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 primary XRE200 External Routing Engine connected to intermediary switch 0.
- Gigabit Ethernet port on intermediary switch 0 to the management port (labeled **MGMT**) on the primary 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 primary XRE200 External Routing Engine connected to intermediary switch 1.
- Gigabit Ethernet port on intermediary switch 1 to the management port (labeled **MGMT**) on the primary 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 | 30](#)  
[Virtual Chassis Control Interface \(VCCI\) Modules in an XRE200 External Routing Engine](#)

## 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 58](#)

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 49](#).

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

1. Download the software package. 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 primary 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](#) | 16



## 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 primary 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**).
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 primary 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 | 56](#)

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

[Understanding Software Upgrades in an EX8200 Virtual Chassis | 16](#)

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

## 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.

**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](#)" on [page 142](#) 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](#) | 16

## Configuring Line-Card Upgrade Groups for Nonstop Software Upgrade

### IN THIS SECTION

- [How Line-card Upgrade Groups Work with Nonstop Software Upgrade](#) | 62
- [Line-card Upgrade Groups Support](#) | 62
- [Configure Line-Card Upgrade Groups on an EX4650 Virtual Chassis, a QFX Series Virtual Chassis or a QFX5100 VCF](#) | 63
- [Configure Line-Card Upgrade Groups on Standalone EX6200 or EX8200 Switches](#) | 63
- [Configure Line-Card Upgrade Groups on an EX8200 Virtual Chassis](#) | 64

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

## How Line-card Upgrade Groups Work with Nonstop Software Upgrade

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

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

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

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

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

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

## Line-card Upgrade Groups Support

The following platforms support NSSU line-card upgrade groups:

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

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

When you configure line-card upgrade groups on an EX4650 Virtual Chassis, a QFX Series Virtual Chassis, or a QFX5100 VCF, whose switches do not have separate line cards, you use only the `fpcs` option to specify the Virtual Chassis or VCF member IDs that you want to include in an upgrade group. You don't need to use the `member` option.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

## SEE ALSO

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

## Configure Line-Card Upgrade Groups on an EX8200 Virtual Chassis

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

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

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

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

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

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

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

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

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

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

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

## SEE ALSO

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

## RELATED DOCUMENTATION

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

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

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

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

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

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

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



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

### IN THIS SECTION

- Requirements | 66
- Overview and Topology | 66
- Configuration | 68

Nonstop software upgrade (NSSU) enables you to upgrade the software running on an EX Series switch with redundant Routing Engines or on most EX Series Virtual Chassis by using a single command and with minimal disruption to network traffic. By default, NSSU upgrades the software running on line cards one line card at a time.

To reduce the time an NSSU takes, you can configure line-card upgrade groups on an EX6200 or EX8200 switch with redundant Routing Engines or on an EX8200 Virtual Chassis.

This example shows how to configure NSSU to use line-card upgrade groups:

### Requirements

This example uses the following hardware and software components:

- An EX8200 switch with redundant Routing Engines
- Junos OS Release 10.4 or later for EX Series switches

Before you begin to configure line-card upgrade groups, ensure that you have configured the link aggregation groups (LAGs) as described in [Configuring Aggregated Ethernet Links \(CLI Procedure\)](#). See "[Overview and Topology](#)" on [page 66](#) for details about the LAG configurations for this example.

### Overview and Topology

#### IN THIS SECTION

- Topology | 67

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

Because the default configuration upgrades each line card one at a time, the upgrade can take some time to complete. You can reduce the time it takes to perform an NSSU by configuring line-card upgrade groups. Instead of being upgraded sequentially, the line cards in an upgrade group are upgraded simultaneously. To achieve minimal traffic disruption, you must define the line-card upgrade groups such that the member links of the LAGs reside on line cards that are in different upgrade groups.

**NOTE:** NSSU upgrades the groups in the order that they appear in the configuration (in other words, in the order you configure them).

## Topology

This example uses an EX8200 switch that has five line cards installed in slots 0 through 4. Two LAGs have been configured:

- ae0—Has two member links, one on the line card in slot 0 and one on the line card in slot 1.
- ae1—Has two member links, one on the line card in slot 2 and one on the line card in slot 3.

The interfaces on the line card in slot 4 are not part of either LAG.

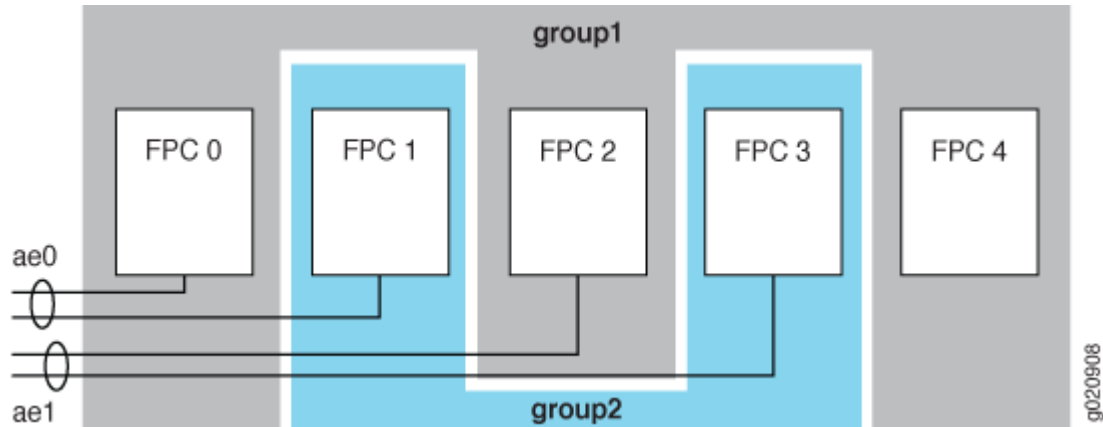
To minimize the time an upgrade takes and to ensure that the member links of each LAG are in different upgrade groups, this example configures the following two line-card upgrade groups:

- group1—Contains the line cards in slots 0, 2, and 4.
- group2—Contains the line cards in slots 1 and 3.

The line card in slot 4 could be put in either group. It could also be left out of an upgrade group entirely, and it would be upgraded separately after the line cards in the upgrade groups have been upgraded. However, it is more efficient to include it in an upgrade group.

Figure 7 on page 68 illustrates the topology.

Figure 7: Example Line-Card Upgrade Group Topology



## Configuration

### IN THIS SECTION

- [CLI Quick Configuration | 68](#)
- [Procedure | 69](#)

To create line-card upgrade groups, perform these tasks:

### CLI Quick Configuration

To quickly create the line-card upgrade groups, copy the following commands and paste them into the switch terminal window:

```
[edit]
set chassis nssu upgrade-group group1 fpcs [0 2 4]
set chassis nssu upgrade-group group2 fpcs [1 3]
```

## Procedure

### Step-by-Step Procedure

To create the line-card upgrade groups for an NSSU:

1. Create the first line-card upgrade group:

```
[edit chassis]
user@switch# set nssu upgrade-group group1 fpcs [0 2 4]
```

2. Create the second line-card upgrade group:

```
[edit chassis]
user@switch# set nssu upgrade-group group2 fpcs (NSSU Upgrade Groups) [1
3]
```

## Results

Display the results of the configuration:

```
[edit chassis]
user@switch# show
nssu {
  upgrade-group group1 {
    fpcs [ 0 2 4 ];
  }
  upgrade-group group2 {
    fpcs [ 1 3 ];
  }
}
```

## RELATED DOCUMENTATION

*Configuring Line-Card Upgrade Groups for Nonstop Software Upgrade*

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

*Upgrading Software on an EX8200 Virtual Chassis Using Nonstop Software Upgrade (CLI Procedure)*

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

### IN THIS SECTION

- [Preparing the Switch for Software Installation | 70](#)
- [Upgrading the Software Using NSSU | 71](#)

You can use nonstop software upgrade (NSSU) to upgrade the software on an EX8200 Virtual Chassis. NSSU upgrades the software running on all Routing Engines with minimal traffic disruption during the upgrade. NSSU is supported on EX8200 Virtual Chassis with redundant XRE200 External Routing Engines running Junos OS Release 11.1 or later.

**NOTE:** NSSU upgrades all Routing Engines on all members of the Virtual Chassis and on the XRE200 External Routing Engines. Using NSSU, you cannot choose to upgrade the backup Routing Engines only, nor can you choose to upgrade a specific member of the Virtual Chassis. If you need to upgrade a specific member of the Virtual Chassis, see ["Installing Software for a Single Device in an EX8200 Virtual Chassis"](#) on page 56.

This topic covers:

### Preparing the Switch for Software Installation

Before you begin software installation using NSSU:

- (Optional) Configure line-card upgrade groups as described in *Configuring Line-Card Upgrade Groups for Nonstop Software Upgrade*. By default, NSSU upgrades line cards one at a time, starting with the line card in slot 0 of member 0. This permits aggregated Ethernet links that have members on different line cards remain up through the upgrade process. Configuring line-card upgrade groups reduces the time an upgrade takes because the line cards in each upgrade group are upgraded at the same time rather than sequentially.
- Verify that the members are running the same version of the software:

```
{master:8}
user@external-routing-engine> show version all-members
```

If the Virtual Chassis members are not running the same version of the software, use the `request system software add` command to upgrade the software on the inconsistent members. For instructions, see ["Installing Software for a Single Device in an EX8200 Virtual Chassis" on page 56](#).

- Ensure that nonstop active routing (NSR) and graceful Routing Engine switchover (GRES) are enabled. To verify that they are enabled, you need to check only the state of nonstop active routing— if nonstop active routing is enabled, then graceful Routing Engine switchover is enabled.

To verify that nonstop active routing is enabled:

```
{master:8}
user@switch> show task replication
    Stateful Replication: Enabled
    RE mode: Master

Protocol                Synchronization Status
-----                -
PIM                      Complete
```

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

## Upgrading the Software Using NSSU

This procedure describes how to upgrade the software running on all Routing Engines using NSSU. When the upgrade completes, all Routing Engines are running the new version of the software. The backup external Routing Engine is now the primary external Routing Engine, and the internal backup Routing Engines in the member switches are now the internal primary Routing Engines in those member switches.

To upgrade all Routing Engines using NSSU:

1. Download the software package for the XRE200 External Routing Engine by following one of the procedures in [Downloading Software](#). The name of the software package for the XRE200 External Routing Engine contains the term **xre200**.
2. Copy the software package to the switch. We recommend that you use FTP to copy the file to the `/var/tmp` directory.
3. Log in to the primary external Routing Engine using the console connection. You can perform an NSSU from the management interface, but a console connection allows you to monitor the progress of the primary Routing Engine reboot.

#### 4. Install the new software package:

```
{master:8}
user@external-routing-engine> request system software nonstop-upgrade reboot
/var/tmp/package-name-m.nZx-distribution.tgz
```

where *package-name-m.nZx-distribution.tgz* is, for example, *jinstall-ex-xre200-11.1R2.5-domestic-signed.tgz*.

**NOTE:** You can omit **reboot** option. When you include the **reboot** option, NSSU automatically reboots the original primary Routing Engines after the new image has been installed on them. If you omit the **reboot** option, you must manually reboot the original primary Routing Engines (now the backup Routing Engines) to complete the upgrade. To perform the reboot, you must establish a connection to the console port on the Switch Fabric and Routing Engine (SRE) module or Routing Engine (RE) module.

The switch displays status messages similar to the following messages as the upgrade executes:

```
Chassis ISSU Check Done
ISSU: Validating Image
ISSU: Preparing LCC Backup REs
ISSU: Preparing Backup RE
Pushing bundle /var/tmp/jinstall-ex-xre200-11.1-20110208.0-domestic-signed.tgz to member9
member9:
-----
WARNING: A reboot is required to install the software
WARNING: Use the 'request system reboot' command immediately
VC Backup upgrade done
Rebooting VC Backup RE

Rebooting member9
ISSU: Backup RE Prepare Done
Waiting for VC Backup RE reboot
Pushing bundle to member0-backup
Pushing bundle to member1-backup
WARNING: A reboot is required to install the software
WARNING: Use the 'request system reboot' command immediately
WARNING: A reboot is required to install the software
WARNING: Use the 'request system reboot' command immediately
```

Rebooting member0-backup

Rebooting LCC [member0-backup]

Rebooting member1-backup

Rebooting LCC [member1-backup]

ISSU: LCC Backup REs Prepare Done

GRES operational

Initiating Chassis Nonstop-Software-Upgrade

Chassis ISSU Started

ISSU: Preparing Daemons

ISSU: Daemons Ready for ISSU

ISSU: Starting Upgrade for FRUs

ISSU: Preparing for Switchover

ISSU: Ready for Switchover

Checking Nonstop-Upgrade status

member0:

```
-----
Item          Status          Reason
FPC 0         Online (ISSU)
FPC 1         Online (ISSU)
FPC 2         Online (ISSU)
FPC 5         Online (ISSU)
```

member1:

```
-----
Item          Status          Reason
FPC 0         Online (ISSU)
FPC 1         Online (ISSU)
FPC 2         Online (ISSU)
FPC 5         Online (ISSU)
```

member0:

```
-----
Item          Status          Reason
FPC 0         Online (ISSU)
FPC 1         Online (ISSU)
FPC 2         Online (ISSU)
FPC 5         Online (ISSU)
```

member1:

```
-----
Item          Status          Reason
FPC 0         Online (ISSU)
```



```

FPC 1      Online (ISSU)
FPC 2      Online (ISSU)
FPC 5      Online (ISSU)
ISSU: Upgrading Old Master RE
Pushing bundle /var/tmp/incoming-package-8200.tgz to member0-master
Pushing bundle /var/tmp/incoming-package-8200.tgz to member1-master
ISSU: RE switchover Done
WARNING: A reboot is required to install the software
WARNING: Use the 'request system reboot' command immediately
ISSU: Old Master Upgrade Done
ISSU: IDLE

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

Shutdown NOW!

```

**NOTE:** If you omit the **reboot** option in this step, you must complete the upgrade by separately rebooting the original primary Routing Engine on each Virtual Chassis member and the original primary external Routing Engine. To reboot the original primary Routing Engine on a Virtual Chassis member, you must establish a connection to the console port on the Switch Fabric and Routing Engine (SRE) module or Routing Engine (RE) module.

5. Log in after the reboot completes. To verify that the software on all Routing Engines in the Virtual Chassis members has been upgraded, enter the following command:

```

{backup:8}
user@external-routing-engine> show version all-members

```

6. Verify that the line cards that were online before the upgrade are online after the upgrade by entering the `show chassis nonstop-upgrade` command:

```

{backup:8}
user@external-routing-engine> show chassis nonstop-upgrade
member0:
-----

```

Item	Status	Reason
FPC 0	Online	
FPC 1	Online	
FPC 2	Online	
FPC 5	Online	

member1:

---

Item	Status	Reason
FPC 0	Online	
FPC 1	Online	
FPC 2	Online	
FPC 5	Online	

## RELATED DOCUMENTATION

*Upgrading Software Using Nonstop Software Upgrade on EX Series Virtual Chassis and Mixed Virtual Chassis (CLI Procedure)*

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

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

[Configuring Dual-Root Partitions](#)

[Troubleshooting Software Installation on EX Series Switches](#)

*Understanding Nonstop Software Upgrade on EX Series Switches*

[Understanding Software Installation on EX Series Switches](#)

## 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 primary external Routing Engine. The configuration file that is saved on the primary 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 `member`*member-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 primary external Routing Engine. You also have to apply the configured groups from primary external Routing Engine CLI on the primary 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 primary 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 membermember-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 membermember-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 membermember-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](#) | 31

# Configuration Statements

## IN THIS CHAPTER

- `fpcs` (NSSU Upgrade Groups) | 78
- `graceful-restart` (Enabling Globally) | 80
- `id` | 83
- `location` (Virtual Chassis) | 84
- `mac-persistence-timer` | 86
- `member` | 88
- `member` (NSSU Upgrade Groups) | 90
- `no-split-detection` | 92
- `nssu` | 93
- `preprovisioned` | 95
- `role` | 97
- `serial-number` | 100
- `traceoptions` (Virtual Chassis) | 101
- `upgrade-group` | 105
- `virtual-chassis` | 106

## fpcs (NSSU Upgrade Groups)

### IN THIS SECTION

- Syntax | 79
- Hierarchy Level | 79
- Description | 79
- Options | 80

- Required Privilege Level | 80
- Release Information | 80

## Syntax

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

## Hierarchy Level

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

## Description

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

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

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

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

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

To configure an upgrade group that includes line cards on switches that support multiple line cards and comprise a Virtual Chassis, use this statement with the `member` option to specify the Virtual Chassis

member ID and the desired line card slot number or numbers on that member switch to include in the upgrade group. Use multiple statements to add line cards from different Virtual Chassis members to the upgrade group.

## Options

### *list-of-slot-numbers*

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

### *slot-number*

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

## Required Privilege Level

interface—To view this statement in the configuration.

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

## Release Information

Statement introduced in Junos OS Release 10.4.

## RELATED DOCUMENTATION

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

*Configuring Line-Card Upgrade Groups for Nonstop Software Upgrade*

## graceful-restart (Enabling Globally)

### IN THIS SECTION

- [Syntax | 81](#)
- [Hierarchy Level | 81](#)
- [Description | 81](#)

- Default | 82
- Options | 82
- Required Privilege Level | 82
- Release Information | 82

## Syntax

```
graceful-restart {  
    disable;  
    helper-disable;  
    maximum-helper-recovery-time seconds;  
    maximum-helper-restart-time seconds;  
    notify-duration seconds;  
    recovery-time seconds;  
    restart-duration seconds;  
    stale-routes-time seconds;  
}
```

## Hierarchy Level

```
[edit logical-systems logical-system-name routing-options],  
[edit logical-systems logical-system-name routing-instances routing-instance-name routing-  
options],  
[edit routing-options],  
[edit routing-instances routing-instance-name routing-options]
```

## Description

You configure the graceful restart routing option globally to enable the feature, but not to enable graceful restart for all routing protocols in a routing instance. To enable graceful restart globally, include the `graceful-restart` statement under the `[edit routing options]` hierarchy level. This enables graceful restart globally for all routing protocols. You can, optionally, modify the global settings at the individual protocol level.



**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. See [CLI Explorer](#).

**Required Privilege Level**

routing—To view this statement in the configuration.

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

**Release Information**

Statement introduced before Junos OS Release 7.4.

**RELATED DOCUMENTATION**

---

[Enabling Graceful Restart](#)

---

[Configuring Routing Protocols Graceful Restart](#)

---

[Configuring Graceful Restart for MPLS-Related Protocols](#)

---

[Configuring VPN Graceful Restart](#)

---

[Configuring Logical System Graceful Restart](#)

---

[Configuring Graceful Restart for QFabric Systems](#)

# id

## IN THIS SECTION

- Syntax | 83
- Hierarchy Level | 83
- Description | 83
- Options | 83
- Required Privilege Level | 83
- Release Information | 84

## Syntax

```
id id;
```

## Hierarchy Level

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

## Description

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

## Options

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

## Required Privilege Level

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

## Release Information

Statement introduced in Junos OS Release 9.3.

### RELATED DOCUMENTATION

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

[Assigning the Virtual Chassis ID to Determine Precedence During a Virtual Chassis Merge](#)

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

[Autoprovisioning a Virtual Chassis Fabric](#)

[Preprovisioning a Virtual Chassis Fabric](#)

[Configuring an EX8200 Virtual Chassis \(CLI Procedure\)](#)

[Understanding Virtual Chassis Member ID Numbering in an EX8200 Virtual Chassis](#)

## location (Virtual Chassis)

### IN THIS SECTION

- [Syntax | 84](#)
- [Hierarchy Level | 85](#)
- [Description | 85](#)
- [Options | 85](#)
- [Required Privilege Level | 85](#)
- [Release Information | 85](#)

### Syntax

```
location location;
```

## Hierarchy Level

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

## Description

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

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

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

## Options

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

## Required Privilege Level

system—To view this statement in the configuration.

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

## Release Information

Statement introduced in Junos OS Release 11.1.

## RELATED DOCUMENTATION

---

*Autoprovisioning a Virtual Chassis Fabric*

---

*Preprovisioning a Virtual Chassis Fabric*

---

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

---

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

---

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

## mac-persistence-timer

### IN THIS SECTION

- [Syntax | 86](#)
- [Hierarchy Level | 86](#)
- [Description | 86](#)
- [Default | 86](#)
- [Options | 87](#)
- [Required Privilege Level | 87](#)
- [Release Information | 87](#)

### Syntax

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

### Hierarchy Level

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

### Description

Specify how long the Virtual Chassis or VCF continues to use the MAC address of the switch that was originally configured in the primary role as the system MAC base address after the original primary 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 primary role remains a member of the Virtual Chassis or VCF.

The maximum timer limit is 60 minutes starting in Junos OS Release 12.2. There are no minimum or maximum timer limits in prior Junos OS releases.

### Default

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

## Options

**minutes** Time in minutes that the member switch in the backup role continues to use the system MAC base address of the old primary before using its own system MAC base address after the switch in the primary role is physically disconnected or removed from the Virtual Chassis or VCF.

**disable** Disable the MAC persistence timer. The system MAC base address never changes when the MAC persistence timer is disabled, even when the switch in the primary role is physically disconnected or removed from the Virtual Chassis or VCF.

## Required Privilege Level

system—To view this statement in the configuration.

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

## Release Information

Statement introduced in Junos OS Release 9.0.

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

The maximum timer limit changed from no maximum timer limit to 60 minutes in Junos OS Release 12.2 for EX Series switches.

## RELATED DOCUMENTATION

[Configuring the Timer for the Backup Member to Start Using Its Own MAC Address as Primary of a Virtual Chassis](#)

---

*Autoprovisioning a Virtual Chassis Fabric*

---

*Preprovisioning a Virtual Chassis Fabric*

## member

### IN THIS SECTION

- [Syntax | 88](#)
- [Hierarchy Level | 88](#)
- [Description | 88](#)
- [Default | 89](#)
- [Options | 89](#)
- [Required Privilege Level | 89](#)
- [Release Information | 89](#)

### Syntax

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

### Hierarchy Level

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

### Description

Configure a switch or an XRE200 External Routing Engine as a member of a Virtual Chassis or a Virtual Chassis Fabric (VCF) with characteristics specified by the available options.

## Default

When an EX Series switch or a QFX Series devices configured in standalone mode is powered on but not interconnected through its Virtual Chassis ports (VCPs) with other member switches, its default member ID is 0.

There is no default member ID in an EX8200 or EX9200 Virtual Chassis. An EX8200 or EX9200 Virtual Chassis must be preprovisioned, and that process configures the member IDs.

## Options

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

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

In an EX8200 Virtual Chassis, member IDs 0 through 7 are reserved for EX8200 member switches and member IDs 8 and 9 are reserved for the primary and backup external Routing Engines.

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

## Required Privilege Level

system—To view this statement in the configuration.

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

## Release Information

Statement introduced in Junos OS Release 9.0.

## RELATED DOCUMENTATION

---

*Autoprovisioning a Virtual Chassis Fabric*

---

*Preprovisioning a Virtual Chassis Fabric*

---

*Adding a Device to a Virtual Chassis Fabric*

---

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

---

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

---

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

---

[Configuring an EX9200 Virtual Chassis](#)

---



## member (NSSU Upgrade Groups)

### IN THIS SECTION

- Syntax | 90
- Hierarchy Level | 90
- Description | 90
- Options | 91
- Required Privilege Level | 91
- Release Information | 91

### Syntax

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

### Hierarchy Level

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

### Description

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

**NOTE:** This statement is not applicable to Virtual Chassis or VCF member switches that do not support separate line cards. To configure Virtual Chassis or VCF member switches that do not

have separate line cards into an NSSU upgrade group, use the `fpcs` statement alone, and specify the Virtual Chassis or VCF member IDs to include in the upgrade group in place of line card slot numbers.

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

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

## Options

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

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

## Required Privilege Level

interface—To view this statement in the configuration.

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

## Release Information

Statement introduced in Junos OS Release 11.1.

## RELATED DOCUMENTATION

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

*Configuring Line-Card Upgrade Groups for Nonstop Software Upgrade*

## no-split-detection

### IN THIS SECTION

- [Syntax | 92](#)
- [Hierarchy Level | 92](#)
- [Description | 92](#)
- [Default | 92](#)
- [Required Privilege Level | 93](#)
- [Release Information | 93](#)

### Syntax

```
no-split-detection;
```

### Hierarchy Level

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

### Description

Disable the split and merge feature in a Virtual Chassis or VCF configuration.

The split and merge feature is enabled by default when you initially set up a Virtual Chassis. If you have only two members in your Virtual Chassis, we strongly recommend that you configure `no-split-detection` to disable the split and merge feature. This feature makes sure both switches remain in the correct Virtual Chassis roles in the event of a Virtual Chassis split. If you expand the two-member Virtual Chassis later to add more members, delete the `no-split-detection` configuration item to re-enable the split and merge feature again.

### 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.

## Release Information

Statement introduced in Junos OS Release 9.3.

## RELATED DOCUMENTATION

*Understanding Split and Merge in a Virtual Chassis*

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

*Disabling Split and Merge in a Virtual Chassis*

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

## nssu

### IN THIS SECTION

- [Syntax | 93](#)
- [Hierarchy Level | 94](#)
- [Description | 94](#)
- [Default | 94](#)
- [Required Privilege Level | 95](#)
- [Release Information | 95](#)

## Syntax

```
nssu {
  rcp-count number;
  upgrade-group group-name {
```

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

## Hierarchy Level

[edit chassis]

## Description

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

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

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

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

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

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

## Default

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

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

## Required Privilege Level

interface—To view this statement in the configuration.

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

## Release Information

Statement introduced in Junos OS Release 10.4.

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

## RELATED DOCUMENTATION

*Configuring Line-Card Upgrade Groups for Nonstop Software Upgrade*

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

## preprovisioned

### IN THIS SECTION

- [Syntax | 95](#)
- [Hierarchy Level | 96](#)
- [Description | 96](#)
- [Required Privilege Level | 96](#)
- [Release Information | 96](#)

## Syntax

```
preprovisioned;
```

## Hierarchy Level

[edit virtual-chassis]

## Description

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

When the preprovisioned configuration mode is enabled, you cannot use the CLI or the J-Web interface to change the primary-role priority or member ID of member switches.

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

## Required Privilege Level

system—To view this statement in the configuration.

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

## Release Information

Statement introduced in Junos OS Release 9.0.

## RELATED DOCUMENTATION

*Preprovisioning a Virtual Chassis Fabric*

[Example: Configuring an EX4200 Virtual Chassis Using a Preprovisioned Configuration File](#)

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

[Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\)](#)

[Configuring an EX8200 Virtual Chassis \(CLI Procedure\)](#)

[Configuring an EX9200 Virtual Chassis](#)

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

*Removing or Replacing a Member Switch of a Virtual Chassis Configuration*

## role

### IN THIS SECTION

- [Syntax | 97](#)
- [Hierarchy Level | 97](#)
- [Description | 97](#)
- [Options | 98](#)
- [Required Privilege Level | 99](#)
- [Release Information | 99](#)

### Syntax

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

### Hierarchy Level

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

### Description

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

#### Virtual Chassis Fabric

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

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



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

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

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

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

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

### EX8200 Virtual Chassis

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

### Options

- `line-card`—Enables the member to be eligible to function only in the linecard role. Any member of the Virtual Chassis or VCF configuration other than the primary 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 primary or backup Routing Engine of the Virtual Chassis or VCF configuration. The primary manages all members and runs the chassis management processes and control protocols. The backup synchronizes with the primary 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 primary is unavailable.

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

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

## Required Privilege Level

system—To view this statement in the configuration.

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

## Release Information

Statement introduced in Junos OS Release 9.0.

## RELATED DOCUMENTATION

---

*Autoprovisioning a Virtual Chassis Fabric*

---

*Preprovisioning a Virtual Chassis Fabric*

---

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

---

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

---

[Configuring an EX9200 Virtual Chassis](#)

---

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

---

*Removing or Replacing a Member Switch of a Virtual Chassis Configuration*

## serial-number

### IN THIS SECTION

- Syntax | 100
- Hierarchy Level | 100
- Description | 100
- Options | 100
- Required Privilege Level | 101
- Release Information | 101

### Syntax

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

### Hierarchy Level

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

### Description

In a preprovisioned Virtual Chassis or Virtual Chassis Fabric (VCF), specify the serial number of each member switch to be included in the configuration. If you do not include the serial number within the configuration, the switch cannot be recognized as a member of a preprovisioned configuration. Serial number values are case-sensitive.

In an EX8200 Virtual Chassis configuration, specify the serial number of each XRE200 External Routing Engine and each EX8200 member switch to be included in the Virtual Chassis configuration. If you do not include the serial number within the Virtual Chassis configuration, the external Routing Engine or switch cannot be recognized as a member of the configuration.

### Options

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

## Required Privilege Level

system—To view this statement in the configuration.

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

## Release Information

Statement introduced in Junos OS Release 9.0.

## RELATED DOCUMENTATION

*[Autoprovisioning a Virtual Chassis Fabric](#)*

*[Preprovisioning a Virtual Chassis Fabric](#)*

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

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

[Configuring an EX9200 Virtual Chassis](#)

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

## traceoptions (Virtual Chassis)

### IN THIS SECTION

- [Syntax | 102](#)
- [Hierarchy Level | 102](#)
- [Description | 102](#)
- [Default | 102](#)
- [Options | 102](#)
- [Required Privilege Level | 104](#)
- [Release Information | 104](#)

## Syntax

```

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

```

## Hierarchy Level

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

## 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 primary 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:** `yk` to specify KB, `ym` to specify MB, or `yg` 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.

## Release Information

Statement introduced in Junos OS Release 9.0.

Option detail added in Junos OS Release 9.2 for EX Series switches.

## RELATED DOCUMENTATION

[Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis](#)

[Verifying the Member ID, Role, and Neighbor Member Connections of a Virtual Chassis Member](#)

[Verifying That Virtual Chassis Ports Are Operational](#)

[Troubleshooting an EX Series Virtual Chassis](#)

*Troubleshooting Virtual Chassis Fabric*

## upgrade-group

### IN THIS SECTION

- [Syntax | 105](#)
- [Hierarchy Level | 105](#)
- [Description | 105](#)
- [Options | 106](#)
- [Required Privilege Level | 106](#)
- [Release Information | 106](#)

### Syntax

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

### Hierarchy Level

```
[edit chassis nssu]
```

### Description

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



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

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

## Options

*group-name*                                  Name of the upgrade group.

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

## Required Privilege Level

interface—To view this statement in the configuration.

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

## Release Information

Statement introduced in Junos OS Release 10.4.

## RELATED DOCUMENTATION

*Configuring Line-Card Upgrade Groups for Nonstop Software Upgrade*

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

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

## virtual-chassis

### IN THIS SECTION

● [Syntax](#) | 107

● [Hierarchy Level](#) | 108

- Description | 108
- Default | 108
- Required Privilege Level | 108
- Release Information | 108

## Syntax

```

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

```

```

regex>;
    flag flag ;
}
vc-port {
    lag-hash (packet-based | source-port-based);
}
vcp-no-hold-time;
vcp-snmp-statistics;
}

```

## Hierarchy Level

[edit]

## Description

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

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

## Default

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

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

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.

## Release Information

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

Statement introduced in Junos OS Release 13.2X50-D15.

## RELATED DOCUMENTATION

*Autoprovisioning a Virtual Chassis Fabric*

---

*Preprovisioning a Virtual Chassis Fabric*

---

*Adding a Device to a Virtual Chassis Fabric*

---

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

---

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

---

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

---

[Configuring an EX9200 Virtual Chassis](#)

# 3

PART

## Administration

---

[Routine Monitoring | 111](#)

[Operational Commands | 117](#)

---

# Routine Monitoring

## IN THIS CHAPTER

- Verifying the Member ID, Role, and Neighbor Member Connections of an EX8200 Virtual Chassis Member | 111
- Verifying Virtual Chassis Ports in an EX8200 Virtual Chassis | 112
- Verifying Hard Disk Memory Status on an XRE200 External Routing Engine | 114

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

### IN THIS SECTION

- Purpose | 111
- Action | 111
- Meaning | 112

### 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

Member ID	Status	Serial No	Model	Mastership		Neighbor List	
				priority	Role	ID	Interface
0 (FPC 0-15)	Prsnt	ghi789	ex8208	0	Linecard	8	vcp-0/0
						9	vcp-0/1
						1	vcp-3/0/2
1 (FPC 16-31)	Prsnt	jk1112	ex8216	0	Linecard	8	vcp-0/0
						9	vcp-0/1
						0	vcp-4/0/2
8 (FPC 128-143)	Prsnt	abc123	ex-xre	129	Master*	9	vcp-1/0
						0	vcp-1/1
						1	vcp-1/2
9 (FPC 144-159)	Prsnt	def456	ex-xre	129	Backup	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 primary 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 \(CLI Procedure\) | 46](#)

[Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines | 36](#)

## Verifying Virtual Chassis Ports in an EX8200 Virtual Chassis

### IN THIS SECTION

- [Purpose | 113](#)
- [Action | 113](#)
- [Meaning | 113](#)

## 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      Type           Trunk  Status   Speed   Neighbor
or             or             ID     or      (mbps)  ID  Interface
Slot/PIC/Port
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 \(CLI Procedure\) | 51](#)

# Verifying Hard Disk Memory Status on an XRE200 External Routing Engine

## IN THIS SECTION

- [Purpose | 114](#)
- [Action | 114](#)
- [Meaning | 115](#)

## 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 *Contact Customer Support to Obtain Return Material Authorization*.

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](#) | 27

Hard Disk Drive (HDD) Module in an XRE200 External Routing Engine

# Operational Commands

## IN THIS CHAPTER

- clear virtual-chassis protocol statistics | 118
- clear virtual-chassis vc-port statistics | 120
- request chassis routing-engine hard-disk-test | 122
- request session member | 127
- request system software nonstop-upgrade | 129
- request system software recovery-package | 142
- request virtual-chassis device-reachability | 145
- request virtual-chassis reactivate | 149
- request virtual-chassis recycle | 150
- request virtual-chassis renumber | 152
- request virtual-chassis vc-port | 153
- set chassis virtual-chassis | 157
- show chassis nonstop-upgrade | 158
- show system software recovery-package | 161
- show virtual-chassis active-topology | 164
- show virtual-chassis protocol adjacency | 170
- show virtual-chassis protocol database | 175
- show virtual-chassis protocol interface | 181
- show virtual-chassis protocol route | 184
- show virtual-chassis protocol statistics | 188
- show virtual-chassis | 193
- show virtual-chassis vc-port | 202
- show virtual-chassis vc-port statistics | 209

## clear virtual-chassis protocol statistics

### IN THIS SECTION

- [Syntax | 118](#)
- [Description | 118](#)
- [Options | 118](#)
- [Required Privilege Level | 119](#)
- [Sample Output | 119](#)
- [Release Information | 119](#)

### Syntax

```
clear virtual-chassis protocol statistics
<all-members>
<local>
<member member-id>
```

### Description

Clear—reset to zero (0)—all Virtual Chassis Control Protocol (VCCP) traffic counters for the Virtual Chassis.

### Options

- |                                |   |
|--------------------------------|---|
| <b>none</b>                    | Clear VCCP counters for all members of the Virtual Chassis.   |
| <b>all-members</b>             | (Optional) Clear VCCP traffic counters for all members of the Virtual Chassis.                                    |
| <b>local</b>                   | (Optional) Clear VCCP traffic counters on the switch or external Routing Engine on which this command is entered. |
| <b>member <i>member-id</i></b> | (Optional) Clear VCCP traffic counters for the specified member of the Virtual Chassis.                           |

## Required Privilege Level

clear

## 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.

## Release Information

Command introduced in Junos OS Release 10.4.

## RELATED DOCUMENTATION

*show virtual-chassis protocol statistics*

[Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis](#) | 29

## clear virtual-chassis vc-port statistics

### IN THIS SECTION

- [Syntax | 120](#)
- [Description | 120](#)
- [Options | 120](#)
- [Required Privilege Level | 121](#)
- [Sample Output | 121](#)
- [Release Information | 122](#)

### Syntax

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

### Description

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

### Options

- |                              |   |
|------------------------------|---|
| <b>none</b>                  | Clear traffic statistics for VCPs of all members of a Virtual Chassis or VCF.   |
| <b>all-members</b>           | (Optional) Clear traffic statistics for VCPs of all members of a Virtual Chassis or VCF.                                  |
| <b><i>interface-name</i></b> | (Optional) Clear traffic statistics for the specified VCP.  |
| <b>local</b>                 | (Optional) Clear traffic statistics for VCPs from the switch or external Routing Engine on which this command is entered. |

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

## Required Privilege Level

clear

## 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
```

### Release Information

Command introduced in Junos OS Release 9.0.

The options **all-members** and **local** were added in Junos OS Release 9.3 for EX Series switches.

### RELATED DOCUMENTATION

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

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

[Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis](#)

## request chassis routing-engine hard-disk-test

### IN THIS SECTION

- [Syntax | 122](#)
- [Description | 123](#)
- [Required Privilege Level | 123](#)
- [Output Fields | 123](#)
- [Sample Output | 126](#)
- [Release Information | 127](#)

### Syntax

```
request chassis routing-engine hard-disk-test
```

## 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

## Output Fields

[Table 6 on page 123](#) 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 6: request chassis routing-engine hard-disk-test Output Fields**

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

Table 6: request chassis routing-engine hard-disk-test Output Fields (Continued)

Field Name	Field Description
<b>Drive0 model</b>	The model number of the HDD module in HDD slot 0. This output is generally only useful for customer support personnel.
<b>Drive1 model</b>	The model number of the HDD module in HDD slot 1. This output is generally only useful for customer support personnel.
<b>Drive0 s/n</b>	The serial number of the HDD module in HDD slot 0. This output is generally only useful for customer support personnel.
<b>Drive1 s/n</b>	The serial number of the HDD module in HDD slot 1. This output is generally only useful for customer support personnel.
<b>Drive0 capacity</b>	The memory capacity of the HDD module in HDD slot 0, in gigabytes.
<b>Drive1 capacity</b>	The memory capacity of the HDD module in HDD slot 1, in gigabytes.
<b>Drive0 (RAID STATUS)</b>	<p>The RAID status of the HDD module in HDD slot 0.</p> <p>Outputs include:</p> <ul style="list-style-type: none"> <li>• <b>On-line</b>—The HDD module is online.</li> <li>• <b>Off-line</b>—The HDD module is offline.</li> </ul> <p>If the status is <b>Off-line</b>, 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).</p> <ul style="list-style-type: none"> <li>• <b>Unknown</b>—HDD module RAID status is unknown.</li> </ul> <p>If the status is <b>Unknown</b>, wait a few minutes then retry the test. If the status remains <b>Unknown</b> after the second test, contact Juniper Networks customer support (JTAC).</p>

Table 6: request chassis routing-engine hard-disk-test Output Fields (Continued)

Field Name	Field Description
<b>Drive1 (RAID STATUS)</b>	<p>The RAID status of the HDD module in HDD slot 1.</p> <p>Outputs include:</p> <ul style="list-style-type: none"> <li>• <b>On-line</b>—HDD module is online.</li> <li>• <b>Off-line</b>—HDD module is offline.</li> </ul> <p>If the status is <b>Off-line</b>, 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).</p> <ul style="list-style-type: none"> <li>• <b>Unknown</b>—HDD module RAID status is unknown.</li> </ul> <p>If the status is <b>Unknown</b>, wait a few minutes then retry the test. If the status remains <b>Unknown</b> after the second test, contact Juniper Networks customer support (JTAC).</p>
<b>Number of partitions</b>	The number of RAID memory partitions.
<b>Size of Partitions</b>	The size of each partition, in gigabytes.

Table 6: request chassis routing-engine hard-disk-test Output Fields (Continued)

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

## 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
```

## Release Information

Command introduced in Junos OS Release 12.1.

## RELATED DOCUMENTATION

[Hard Disk Drive \(HDD\) Module in an XRE200 External Routing Engine](#)

[XRE200 External Routing Engine Hardware Overview](#)

## request session member

### IN THIS SECTION

- [Syntax | 128](#)
- [Description | 128](#)
- [Options | 128](#)

- Required Privilege Level | 128
- Release Information | 128

## Syntax

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

## Description

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

## Options

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

## Required Privilege Level

maintenance

## Release Information

Command introduced in Junos OS Release 9.0.

## RELATED DOCUMENTATION

*member*

[Understanding Virtual Chassis Components](#)

## request system software nonstop-upgrade

### IN THIS SECTION

- [Syntax | 129](#)
- [Description | 129](#)
- [Options | 131](#)
- [Required Privilege Level | 133](#)
- [Output Fields | 133](#)
- [Sample Output | 133](#)
- [Release Information | 141](#)

### Syntax

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

### Description

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

- When you run this command on any of the following Virtual Chassis or VCF configurations, NSSU upgrades all members of the Virtual Chassis:
  - EX3300, EX3400, EX4200, EX4300, EX4400, EX4500, EX4550, EX4600, or EX4650-48Y Virtual Chassis
  - Mixed Virtual Chassis composed of any combination of EX4200, EX4500, and EX4550 switches, or EX4300 and EX4600 switches
  - QFX3500 and QFX3600 Virtual Chassis



- QFX5100 Virtual Chassis
- QFX5120-48Y, QFX5120-48T or QFX5120-32C Virtual Chassis
- Fixed configuration of switches in a VCF (QFX3500/QFX3600 and QFX5100 switches)
- Mixed VCF composed of any combination of QFX3500/QFX3600, QFX5100, and EX4300 switches

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

- When you run this command on an EX6200 or EX8200 switch, NSSU upgrades both the backup and primary Routing Engines. The original backup Routing Engine becomes the new primary at the end of the upgrade.
  - On an EX6200 switch, NSSU automatically reboots the original primary Routing Engine.
  - On an EX8200 switch, NSSU does not automatically reboot the original primary Routing Engine unless you specify the reboot option.
- When you run this command on an EX8200 Virtual Chassis, NSSU upgrades all primary and backup Routing Engines in the Virtual Chassis, including the external Routing Engines. The original backup Routing Engines become the new primary Routing Engines. NSSU does not automatically reboot the original primary Routing Engines unless you specify the reboot option.

This command has the following requirements:

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

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

- You must run the command from the primary Routing Engine on a standalone switch or from the primary on a Virtual Chassis.

- For minimal traffic disruption, you must define link aggregation groups (LAGs) such that the member links reside on different Virtual Chassis or VCF member switches (or on different line cards for EX6200 and EX8200 switches and EX8200 Virtual Chassis).
- For all Virtual Chassis (except EX8200 Virtual Chassis):
  - The Virtual Chassis members must be connected in a ring topology. A ring topology prevents the Virtual Chassis from splitting during an NSSU.
  - The Virtual Chassis primary and backup must be adjacent to each other in the ring topology. With adjacent placement, the primary and backup are always in sync while the switches in line-card roles are rebooting.
  - The Virtual Chassis must be preprovisioned so the line-card role is explicitly assigned to member switches acting in a line-card role. During an NSSU, the primary and backup member switches must maintain their Routing Engine roles (although the primary role switches to the backup), and the remaining switches must maintain their line-card roles.
  - In a two-member Virtual Chassis, you must configure `no-split-detection` so the Virtual Chassis doesn't split during NSSU.
- For Virtual Chassis Fabric:
  - You can only have two members preprovisioned in the Routing Engine role. If more than two Routing Engines are configured, NSSU issues a warning message and the NSSU process stops.
  - The VCF members should be connected in a spine and leaf topology. A spine and leaf topology prevents the VCF from splitting during NSSU. Each leaf device must be connected to both spine devices.
  - The VCF must be preprovisioned so that the line-card role has been explicitly assigned to member switches acting in a line-card role, and likewise the Routing Engine role has been explicitly assigned to the member switches acting in a Routing Engine role. During an NSSU, the primary and backup member switches must maintain their Routing Engine roles (although the primary role switches to the backup), and the remaining switches must maintain their line-card roles.
  - You must configure `no-split-detection` in a two-member VCF so the VCF does not split during NSSU.

## Options

*package-name*

Location of the software package or bundle to be installed. For example:

- `/var/tmp/package-name`—For a software package or bundle installed from a local directory on the switch.

- *protocol://hostname/pathname/package-name*—For a software package or bundle downloaded and installed from a remote location. Replace *protocol* with one of the following:
  - **ftp**—File Transfer Protocol.  
Use `ftp://hostname/pathname/package-name`.  
To specify authentication credentials, use `ftp://<username>:<password>@hostname/pathname/package-name`.  
To have the system prompt you for the password, specify `prompt` in place of the password.  
The command displays an error message if a password is required and you do not specify the password or prompt.
  - **http**—Hypertext Transfer Protocol.  
Use `http://hostname/pathname/package-name`.  
To specify authentication credentials, use `http://<username>:<password>@hostname/pathname/package-name`.  
The command prompts you for a password if one is required and you didn't include it.
  - **scp**—Secure copy (available only for Canada and U.S. version).  
Use `scp://hostname/pathname/package-name`.  
To specify authentication credentials, use `scp://<username>:<password>@hostname/pathname/package-name`.

**NOTE:** The *pathname* in the protocol is the relative path to the user home directory on the remote system and not the root directory.

<b>set</b> <b>[package-name package-name]</b>	(Mixed Virtual Chassis only) Locations of the different installation packages required by the different types of member switches. These packages must be for the same Junos OS release. See this command's <i>package-name</i> option for information about how to specify the installation packages.
<b>force-host</b>	(Optional) Force adding the host software package or bundle (and ignore warnings) on EX4650, QFX5100, or QFX5120 devices.
<b>no-copy</b>	(Optional) Install a software package or bundle, but do not save copies of the package or bundle files.
<b>no-old-master-upgrade</b>	(Optional) (EX8200 switches only) Upgrade the backup Routing Engine only. After the upgrade completes, the original primary Routing Engine becomes the backup Routing Engine and continues running the previous software version.

**reboot** (Optional) (EX8200 switches and EX8200 Virtual Chassis only) When you include the reboot option, NSSU automatically reboots the original primary (new backup) Routing Engine after being upgraded to the new software. When you omit the reboot option, you must manually reboot the original primary (new backup) Routing Engine using the `request system reboot` command.

**NOTE:** If you do not use the reboot option on an EX8200 Virtual Chassis, you must establish a connection to the console port on the Switch Fabric and Routing Engine (SRE) module or Routing Engine (RE) module to manually reboot the backup Routing Engines.

**unlink** (Optional) Remove the software package after a successful upgrade.

## Required Privilege Level

maintenance

## Output Fields

This command reports feedback on the status of the request. Some functions are shared between NSSU and the in-service software upgrade (ISSU) feature, so you might see what appear to be ISSU messages as well as NSSU messages in the output from this command.

## Sample Output

### **request system software nonstop-upgrade (EX4200 Virtual Chassis)**

```
user@switch> request system software nonstop-upgrade
/var/tmp/jinstall-ex-4200-12.1R5.5-domestic-signed.tgz
Chassis ISSU Check Done
ISSU: Validating Image
ISSU: Preparing Backup RE
Installing image on other FPC's along with the backup

Checking pending install on fpc1
Pushing bundle to fpc1
WARNING: A reboot is required to install the software
WARNING: Use the 'request system reboot' command immediately
```

Completed install on fpc1

Checking pending install on fpc2

Pushing bundle to fpc2

WARNING: A reboot is required to install the software

WARNING: Use the 'request system reboot' command immediately

Completed install on fpc2

Checking pending install on fpc3

Pushing bundle to fpc3

WARNING: A reboot is required to install the software

WARNING: Use the 'request system reboot' command immediately

Completed install on fpc3

Checking pending install on fpc4

Pushing bundle to fpc4

WARNING: A reboot is required to install the software

WARNING: Use the 'request system reboot' command immediately

Completed install on fpc4

Checking pending install on fpc5

Pushing bundle to fpc5

WARNING: A reboot is required to install the software

WARNING: Use the 'request system reboot' command immediately

Completed install on fpc5

Checking pending install on fpc6

Pushing bundle to fpc6

WARNING: A reboot is required to install the software

WARNING: Use the 'request system reboot' command immediately

Completed install on fpc6

Checking pending install on fpc7

Pushing bundle to fpc7

WARNING: A reboot is required to install the software

WARNING: Use the 'request system reboot' command immediately

Completed install on fpc7

Backup upgrade done

Rebooting Backup RE

Rebooting fpc1

ISSU: Backup RE Prepare Done

Waiting for Backup RE reboot

```

GRES operational
Initiating Chassis In-Service-Upgrade
Chassis ISSU Started
ISSU: Preparing Daemons
ISSU: Daemons Ready for ISSU
ISSU: Starting Upgrade for FRUs
ISSU: Preparing for Switchover
ISSU: Ready for Switchover
Checking In-Service-Upgrade status
  Item          Status          Reason
  FPC 0         Online
  FPC 1         Online
  FPC 2         Online (ISSU)
  FPC 3         Online (ISSU)
  FPC 4         Online (ISSU)
  FPC 5         Online (ISSU)
  FPC 6         Online (ISSU)
  FPC 7         Online (ISSU)
Going to install image on master
WARNING: A reboot is required to install the software
WARNING: Use the 'request system reboot' command immediately
relinquish mastership
ISSU: IDLE

*** FINAL System shutdown message from root@switch ***

System going down IMMEDIATELY

Shutdown NOW!
[pid 9336]

```

### request system software nonstop-upgrade (EX6200 Switch)

```

{master}
user@switch> request system software nonstop-upgrade
/var/tmp/jinstall-ex-6200-12.2R5.5-domestic-signed.tgz
Chassis ISSU Check Done
ISSU: Validating Image
ISSU: Preparing Backup RE

```

Pushing bundle to re0

NOTICE: Validating configuration against jinstall-ex-6200-12.2R5.5-domestic-signed.tgz.

NOTICE: Use the 'no-validate' option to skip this if desired.

WARNING: A reboot is required to install the software

WARNING: Use the 'request system reboot' command immediately

Backup upgrade done

Rebooting Backup RE

Rebooting re0

ISSU: Backup RE Prepare Done

Waiting for Backup RE reboot

GRES operational

Initiating Chassis In-Service-Upgrade

Chassis ISSU Started

ISSU: Preparing Daemons

ISSU: Daemons Ready for ISSU

ISSU: Starting Upgrade for FRUs

ISSU: Preparing for Switchover

ISSU: Ready for Switchover

Checking In-Service-Upgrade status

Item	Status	Reason
FPC 0	Online (ISSU)	
FPC 1	Online (ISSU)	
FPC 2	Online (ISSU)	
FPC 3	Online (ISSU)	
FPC 4	Online	
FPC 5	Online	
FPC 6	Online (ISSU)	
FPC 7	Online (ISSU)	
FPC 8	Online (ISSU)	
FPC 9	Online (ISSU)	

Going to install image on master

NOTICE: Validating configuration against jinstall-ex-6200-12.2R5.5-domestic-signed.tgz.

NOTICE: Use the 'no-validate' option to skip this if desired.

WARNING: A reboot is required to install the software

WARNING: Use the 'request system reboot' command immediately

relinquish mastership

ISSU: IDLE

Trying to relinquish mastership before rebooting...

Resolving mastership...

Complete. The other routing engine becomes the master.

\*\*\* FINAL System shutdown message from user@switch \*\*\*

System going down IMMEDIATELY

### request system software nonstop-upgrade reboot (EX8200 Switch)

```
{master}
user@switch> request system software nonstop-upgrade reboot
/var/tmp/jinstall-ex-8200-10.4R1.5-domestic-signed.tgz
Chassis ISSU Check Done
ISSU: Validating Image
ISSU: Preparing Backup RE
Pushing bundle to re1
WARNING: A reboot is required to install the software
WARNING: Use the 'request system reboot' command immediately
Backup upgrade done
Rebooting Backup RE

Rebooting re1
ISSU: Backup RE Prepare Done
Waiting for Backup RE reboot
GRES operational
Initiating Chassis In-Service-Upgrade
Chassis ISSU Started
ISSU: Preparing Daemons
ISSU: Daemons Ready for ISSU
ISSU: Starting Upgrade for FRUs
ISSU: Preparing for Switchover
ISSU: Ready for Switchover
Checking In-Service-Upgrade status
  Item          Status          Reason
  FPC 0         Online (ISSU)
  FPC 2         Offline          Offlined by CLI command
  FPC 3         Online (ISSU)
Resolving mastership...
Complete. The other routing engine becomes the master.
ISSU: RE switchover Done
ISSU: Upgrading Old Master RE
WARNING: A reboot is required to install the software
WARNING: Use the 'request system reboot' command immediately
ISSU: Old Master Upgrade Done
ISSU: IDLE
```



Shutdown NOW!

[pid 2635]

\*\*\* FINAL System shutdown message from user@switch \*\*\*

System going down IMMEDIATELY

## request system software nonstop-upgrade no-old-master-upgrade (EX8200 Switch)

```
{master}
user@switch> request system software nonstop-upgrade no-old-master-upgrade
/var/tmp/jinstall-ex-8200-10.4R1.5-domestic-signed.tgz
```

Chassis ISSU Check Done

ISSU: Validating Image

ISSU: Preparing Backup RE

Pushing bundle to re1

WARNING: A reboot is required to install the software

WARNING: Use the 'request system reboot' command immediately

Backup upgrade done

Rebooting Backup RE

Rebooting re1

ISSU: Backup RE Prepare Done

Waiting for Backup RE reboot

GRES operational

Initiating Chassis In-Service-Upgrade

Chassis ISSU Started

ISSU: Preparing Daemons

ISSU: Daemons Ready for ISSU

ISSU: Starting Upgrade for FRUs

ISSU: Preparing for Switchover

ISSU: Ready for Switchover

Checking In-Service-Upgrade status

Item	Status	Reason
FPC 0	Online (ISSU)	
FPC 1	Online (ISSU)	
FPC 2	Online (ISSU)	
FPC 3	Offline	Offlined by CLI command
FPC 4	Online (ISSU)	
FPC 5	Online (ISSU)	
FPC 6	Online (ISSU)	

```

FPC 7      Online (ISSU)
Resolving mastership...
Complete. The other routing engine becomes the master.
ISSU: RE switchover Done
Skipping Old Master Upgrade
ISSU: IDLE

```

### request system software nonstop-upgrade reboot (EX8200 Virtual Chassis)

```

{master:9}
user@external-routing-engine> request system software nonstop-upgrade reboot
/var/tmp/jinstall-ex-xre200-11.1-20101130.0-domestic-signed.tgz
Chassis ISSU Check Done
ISSU: Validating Image
ISSU: Preparing LCC Backup REs
ISSU: Preparing Backup RE
Pushing bundle /var/tmp/jinstall-ex-xre200-11.1-20101130.0-domestic-signed.tgz to member8
-----
WARNING: A reboot is required to install the software
WARNING:   Use the 'request system reboot' command immediately
VC Backup upgrade done
Rebooting VC Backup RE

Rebooting member8
ISSU: Backup RE Prepare Done
Waiting for VC Backup RE reboot
Pushing bundle to member0-backup
Pushing bundle to member1-backup
WARNING: A reboot is required to install the software
WARNING:   Use the 'request system reboot' command immediately
WARNING: A reboot is required to install the software
WARNING:   Use the 'request system reboot' command immediately

Rebooting member0-backup
Rebooting LCC [member0-backup]

Rebooting member1-backup
Rebooting LCC [member1-backup]
ISSU: LCC Backup REs Prepare Done
GRES operational
Initiating Chassis Nonstop-Software-Upgrade

```

Chassis ISSU Started

ISSU: Preparing Daemons

ISSU: Daemons Ready for ISSU

ISSU: Starting Upgrade for FRUs

ISSU: Preparing for Switchover

ISSU: Ready for Switchover

Checking Nonstop-Upgrade status

member0:

---

Item	Status	Reason
FPC 0	Online (ISSU)	
FPC 1	Online (ISSU)	
FPC 2	Online (ISSU)	
FPC 5	Online (ISSU)	

member1:

---

Item	Status	Reason
FPC 0	Online (ISSU)	
FPC 1	Offline	Offlined due to config
FPC 2	Online (ISSU)	
FPC 3	Online (ISSU)	
FPC 4	Online (ISSU)	
FPC 5	Online (ISSU)	
FPC 7	Online (ISSU)	

member0:

---

Item	Status	Reason
FPC 0	Online (ISSU)	
FPC 1	Online (ISSU)	
FPC 2	Online (ISSU)	
FPC 5	Online (ISSU)	

member1:

---

Item	Status	Reason
FPC 0	Online (ISSU)	
FPC 1	Offline	Offlined due to config
FPC 2	Online (ISSU)	
FPC 3	Online (ISSU)	
FPC 4	Online (ISSU)	
FPC 5	Online (ISSU)	

```
FPC 7          Online (ISSU)
ISSU: Upgrading Old Master RE
Pushing bundle /var/tmp/incoming-package-8200.tgz to member0-master
Pushing bundle /var/tmp/incoming-package-8200.tgz to member1-master

ISSU: RE switchover Done
WARNING: A reboot is required to install the software
WARNING:   Use the 'request system reboot' command immediately
Rebooting ...
shutdown: [pid 2188]
Shutdown NOW!
ISSU: Old Master Upgrade Done
ISSU: IDLE
Shutdown NOW!

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

## Release Information

Command introduced in Junos OS Release 10.4.

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

## RELATED DOCUMENTATION

*show chassis nonstop-upgrade*

---

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

---

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

---

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

---

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

## request system software recovery-package

### IN THIS SECTION

- [Syntax | 142](#)
- [Description | 142](#)
- [Options | 143](#)
- [Required Privilege Level | 144](#)
- [Output Fields | 144](#)
- [Sample Output | 144](#)
- [Release Information | 145](#)

### Syntax

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

### 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.

## 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 primary 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 primary 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 primary 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 primary external Routing Engine.

## Required Privilege Level

maintenance

## 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
```

## Release Information

Command introduced in Junos OS Release 12.2.

### RELATED DOCUMENTATION

| [show system software recovery-package](#) | 161

## request virtual-chassis device-reachability

### IN THIS SECTION

- [Syntax](#) | 145
- [Description](#) | 146
- [Options](#) | 146
- [Required Privilege Level](#) | 147
- [Output Fields](#) | 147
- [Sample Output](#) | 148
- [Release Information](#) | 148

## 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>
```

## 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

<b>test-name</b> <i>name</i>	Specify the name of the test. The name can be any single-word character string. Spaces are not allowed.
<b>source-fpc</b> <i>source-fpc-id</i>	Specify the FPC that sends the first ping message.
<b>source-ip-address</b> <i>source-ip-address</i>	Specify the source IP address that receives the ping message.
<b>destination-device</b> <i>device-id</i>	Specify the device in the Virtual Chassis that receives the ping message.
<b>destination-fpc</b> <i>destination-fpc-id</i>	Specify the FPC that receives the first ping message.
<b>destination-ip-address</b> <i>destination-ip-address</i>	Specify the destination IP address that receives the ping message.
<b>probe-count</b> <i>count</i>	(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.
<b>probe-interval</b> <i>interval</i>	(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.
<b>probe-pattern</b> <i>pattern</i>	(Optional) Specify a payload pattern in the ping message. Enter the pattern in hexadecimal format.
<b>test-count</b> <i>count</i>	(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.
<b>test-interval</b> <i>interval</i>	(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

## Output Fields

Table 7 on page 147 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 7: request virtual chassis device reachability Output Fields**

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

## 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
--- sping statistics ---
Session ID                : 0
Packets Count/Sent/Received/Sendfail : 5/5/5/0
Unknown/Timeout/Duplicate packets received : 0/0/0
Round-trip Min/Avg/Max    : 181/807/3010 usec
```

## Release Information

Command introduced in Junos OS Release 11.2.

## RELATED DOCUMENTATION

| *Verifying Connectivity Between Virtual Chassis Member Devices*

## request virtual-chassis reactivate

### IN THIS SECTION

- [Syntax | 149](#)
- [Description | 149](#)
- [Required Privilege Level | 149](#)
- [Sample Output | 149](#)
- [Release Information | 150](#)

### Syntax

```
request virtual-chassis reactivate
```

### Description

Reactivate a device that has been assigned a member ID but is not currently connected to the Virtual Chassis or VCF.

You can use this command to reactivate a device that was previously part of the Virtual Chassis or VCF but whose status is no longer **Prsnt**.

### Required Privilege Level

system-control

### Sample Output

```
request virtual-chassis reactivate
```

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

## Release Information

Command introduced in Junos OS Release 9.3.

### RELATED DOCUMENTATION

[Verifying the Member ID, Role, and Neighbor Member Connections of a Virtual Chassis Member](#)

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

## request virtual-chassis recycle

### IN THIS SECTION

- [Syntax | 150](#)
- [Description | 150](#)
- [Options | 151](#)
- [Required Privilege Level | 151](#)
- [Sample Output | 151](#)
- [Sample Output | 151](#)
- [Release Information | 151](#)

## Syntax

```
request virtual-chassis recycle member-id member-id
```

## Description

Make a previously used member ID available for reassignment.

When you remove a member switch from the Virtual Chassis configuration, the primary reserves that member ID. To make the member ID available for reassignment, you must use this command.

**NOTE:** You must run this command from the Virtual Chassis member in the primary role.

## Options

**member-id** *member-id* Specify the member ID that you want to make available for reassignment to a different member.

## Required Privilege Level

system-control

## 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
```

## Release Information

Command introduced in Junos OS Release 9.0.

## RELATED DOCUMENTATION

[request virtual-chassis renumber](#) | [152](#)

*Removing or Replacing a Member Switch of a Virtual Chassis Configuration*

## request virtual-chassis renumber

### IN THIS SECTION

- [Syntax | 152](#)
- [Description | 152](#)
- [Options | 152](#)
- [Required Privilege Level | 152](#)
- [Sample Output | 153](#)
- [Release Information | 153](#)

### Syntax

```
request virtual-chassis renumber member-id old-member-id new-member-id new-member-id
```

### Description

Renumber a member of a Virtual Chassis configuration.

**NOTE:** You must run this command from the Virtual Chassis member in the primary role.

### Options

- |   |   |
|---|---|
| <b>member-id</b> <i>old-member-id</i>     | Specify the ID of the member that you wish to renumber. |
| <b>new-member-id</b> <i>new-member-id</i> | Specify an unassigned member ID.                        |

### Required Privilege Level

system-control

## 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
```

## Release Information

Command introduced in Junos OS Release 9.0.

## RELATED DOCUMENTATION

[request virtual-chassis recycle](#) | 150

*Removing or Replacing a Member Switch of a Virtual Chassis Configuration*

## request virtual-chassis vc-port

### IN THIS SECTION

- [Syntax](#) | 154
- [Description](#) | 154
- [Options](#) | 155
- [Required Privilege Level](#) | 155
- [Sample Output](#) | 156
- [Release Information](#) | 156



## Syntax

```
request virtual-chassis vc-port (set | delete)
<fpc-slot fpc-slot> pic-slot pic-slot port port-number
interface interface-name
<member | local>
```

*member-id*

## Description

Set a port to operate as a Virtual Chassis port (VCP), or delete the VCP setting on a port. See [Virtual Chassis Port Options](#) for details on which ports you can set as VCPs on different switches. After setting a port as a VCP, you can't use the port for any other purpose unless you remove the VCP setting.

You can specify the port using the `interface` option or the `pic-slot` and `port` options.

If you don't include the `member member-id` option, this command defaults to setting the port as a VCP or deleting the VCP setting on the switch where you run the command. You can alternatively set the `local` option to ensure the command applies to the specified port on the local switch where you run it.

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

Some switches have ports that are dedicated VCPs (you can only use them as VCPs and for no other purpose) or that are configured as VCPs by default (set as VCPs in the default factory configuration). You do not need to explicitly set those ports as VCPs to use them to interconnect the switch into a Virtual Chassis. See [Virtual Chassis Port Options](#) for details.

If you don't need a default-configured VCP to interconnect Virtual Chassis member switches, you can run this command with the `delete` option to remove the VCP setting and use the port as a regular uplink or network port. If you want to use the port as a VCP again later, then you need to run this command with the `set` option to reapply the VCP setting.

## Special Behavior for VCPs on EX4400 Switches

On EX4400 switches, you don't use this command to convert ports on the switch into VCPs. EX4400 switches have default VCPs which are the only ports you can use as VCPs on the switch. These are the two 100-Gbps QSFP28 ports on the rear panel in PIC slot 1, which are set by default as four logical 50-Gbps VCP interfaces (`vcp-255/1/0` through `vcp-255/1/3`). You can convert them into network ports by enabling network port mode on the switch (`request virtual-chassis mode network-port <reboot>`). In that case, to subsequently use the switch in a Virtual Chassis, you must disable network port mode (`request virtual-`

`chassis mode network-port disable <reboot>`), which converts those ports back into VCPs. (Port mode changes apply to all VCPs on the switch,)

You can enter the `request virtual-chassis vc-port delete` command to disable these ports as VCPs, but, as mentioned above, that action doesn't convert them into network ports.

**NOTE:** On EX4400 switches, if you use the `request virtual-chassis vc-port delete interface interface-name` command to disable logical VCP interface `vcp-255/1/0`, the switch disables both logical ports 0 and 1 (`vcp-255/1/0` and `vcp-255/1/1`). Similarly, if you disable logical VCP interface `vcp-255/1/2`, that action disables both logical ports 2 and 3 (`vcp-255/1/2` and `vcp-255/1/3`).

## Options

<b>set</b>	Set a port as a VCP to convert an uplink or network port into a VCP.
<b>delete</b>	Delete the VCP setting on a port to convert a VCP back into an uplink or network port.
<b>pic-slot <i>pic-slot</i></b>	Number of the PIC slot for the port on the switch.
<b>port <i>port-number</i></b>	Number of the port that you want to enable or disable as a VCP.
<b>interface <i>interface-name</i></b>	Interface name of the port that you want to enable or disable as a VCP. You can use this option to specify the port instead of using the <code>pic-slot</code> and <code>port</code> options.
<b>member <i>member-id</i></b>	(Optional) Enable or disable the specified VCP on the specified member of the Virtual Chassis or VCF.
<b>local</b>	(Optional) Enable or disable the specified VCP on the local switch where you run the command.

## Required Privilege Level

system-control

## 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.

## Release Information

Command introduced in Junos OS Release 9.0.

Option `fpc-slot` introduced in Junos OS Release 10.4 for EX Series switches.

## RELATED DOCUMENTATION

[request virtual-chassis vc-port \(Dedicated VCP\)](#)

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

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

[clear virtual-chassis vc-port statistics](#)

[Virtual Chassis Port \(VCP\) Interface Names in an EX8200 Virtual Chassis](#)

[Understanding Virtual Chassis Components](#)

## set chassis virtual-chassis

### IN THIS SECTION

- [Syntax | 157](#)
- [Description | 157](#)
- [Options | 157](#)
- [Required Privilege Level | 158](#)
- [Sample Output | 158](#)
- [Release Information | 158](#)

### Syntax

```
set chassis virtual-chassis  
<disable>  
<local>
```

### Description

Set the EX8200 switch into Virtual Chassis mode.

An EX8200 switch cannot be a member switch in an EX8200 Virtual Chassis until the switch is in Virtual Chassis mode.

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.

### Options

- none**     Set the internal Routing Engines into Virtual Chassis mode.
- disable**     Disable Virtual Chassis mode for all internal REs on the switch.
- local**     Set the Routing Engine that you are currently logged in to into Virtual Chassis mode.

## Required Privilege Level

view

## 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
```

## Release Information

Command introduced in Junos OS Release 10.4.

## RELATED DOCUMENTATION

[\*show virtual-chassis\*](#)

[Configuring an EX8200 Virtual Chassis \(CLI Procedure\) | 46](#)

## show chassis nonstop-upgrade

### IN THIS SECTION

- [Syntax | 159](#)
- [Description | 159](#)
- [Required Privilege Level | 159](#)
- [Output Fields | 159](#)
- [Sample Output | 160](#)
- [Release Information | 161](#)

## Syntax

```
show chassis nonstop-upgrade
```

## Description

(EX6200 switches, EX8200 switches, EX8200 Virtual Chassis, QFX3500 and QFX3600 Virtual Chassis, and Virtual Chassis Fabric only) Display the status of the line cards or Virtual Chassis members in the linecard role after the most recent nonstop software upgrade (NSSU). You must issue this command on the primary Routing Engine.

## Required Privilege Level

view

## Output Fields

[Table 8 on page 159](#) lists the output fields for the `show chassis nonstop-upgrade` command. Output fields are listed in the approximate order in which they appear.

**Table 8: show chassis nonstop-upgrade Output Fields**

Field Name	Field Description
<b>Item</b>	Line card slot number.
<b>Status</b>	State of line card: <ul style="list-style-type: none"> <li>• <b>Error</b>—Line card is in an error state.</li> <li>• <b>Offline</b>—Line card is powered down.</li> <li>• <b>Online</b>—Line card is online and running.</li> </ul>
<b>Reason</b>	Reason for the state (if the line card is offline).

## Sample Output

### show chassis nonstop-upgrade (EX8200 Switch)

```
user@switch> show chassis nonstop-upgrade
  Item          Status          Reason
  FPC 0         Online
  FPC 1         Online
  FPC 2         Online
  FPC 3         Offline         Offlined by CLI command
  FPC 4         Online
  FPC 5         Online
  FPC 6         Online
  FPC 7         Online
```

### show chassis nonstop-upgrade (EX8200 Virtual Chassis)

```
user@external-routing-engine> show chassis nonstop-upgrade
member0:
-----
  Item          Status          Reason
  FPC 0         Online
  FPC 1         Online
  FPC 2         Online
  FPC 5         Online

member1:
-----
  Item          Status          Reason
  FPC 0         Online
  FPC 1         Offline         Offlined due to config
  FPC 2         Online
  FPC 3         Online
  FPC 4         Online
  FPC 5         Online
  FPC 7         Online
```

## show chassis nonstop-upgrade (Virtual Chassis Fabric)

Item	Status	Reason
FPC 0	Online	
FPC 1	Online	
FPC 2	Online	
FPC 3	Online	
FPC 4	Online	
FPC 5	Online	

## Release Information

Command introduced in Junos OS Release 10.4.

## RELATED DOCUMENTATION

*request system software nonstop-upgrade*

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

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

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

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

## show system software recovery-package

### IN THIS SECTION

- [Syntax | 162](#)
- [Description | 162](#)
- [Options | 162](#)
- [Required Privilege Level | 162](#)
- [Output Fields | 162](#)
- [Sample Output | 163](#)



## Syntax

```
show system software recovery-package  
<backup>  
<master>
```

## Description

Display the software recovery packages loaded on an XRE200 External Routing Engine in an EX8200 Virtual Chassis.

## 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 primary 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.

**none** Display the software recovery packages available on the external Routing Engine.

**backup** (Optional) Display the software recovery packages available on the backup external Routing Engine.

**master** (Optional) Display the software recovery packages available on the primary external Routing Engine.

## Required Privilege Level

maintenance

## 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
```

## Release Information

Command introduced in Junos OS Release 12.2.

## RELATED DOCUMENTATION

[request system software recovery-package](#) | 142

# show virtual-chassis active-topology

## IN THIS SECTION

- [Syntax](#) | 164
- [Description](#) | 164
- [Options](#) | 164
- [Required Privilege Level](#) | 165
- [Output Fields](#) | 165
- [Sample Output](#) | 165
- [Release Information](#) | 170

## Syntax

```
show virtual-chassis active-topology  
<all-members | local | member member-id>
```

## Description

Display the active topology of the Virtual Chassis or VCF with next-hop reachability information.

## Options

- |                    |   |
|--------------------|---|
| <b>none</b>        | Display the active topology of the member switch where you enter this command.                                |
| <b>all-members</b> | (Optional) Display the active topology of all members of the Virtual Chassis or VCF.                          |
| <b>local</b>       | (Optional) Display the active topology of the switch or external Routing Engine where you enter this command. |

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

## Required Privilege Level

view

## Output Fields

[Table 9 on page 165](#) 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
<b>Destination ID</b>	Specifies the member ID of the destination.
<b>Next-hop</b>	Specifies the member ID and Virtual Chassis port (VCP) of the next hop to which packets for the destination ID are forwarded.  The next hop can be more than one device in a VCF.

## Sample Output

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

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

```
1          1(vcp-1)
```

```
2          1(vcp-1)
```

```
3          1(vcp-1)
```

```

4          1(vcp-1)

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

6          8(vcp-0)

7          8(vcp-0)

8          8(vcp-0)

```

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

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

```
member0:
```

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

```
member1:
```

```
-----
Destination ID      Next-hop
0                   0(vcp-3/0/4.32768)
```

```
8          8(vcp-0/0.32768)
```

```
9          8(vcp-0/0.32768)
```

```
member8:
```

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

```
member9:
```

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

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

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

```
fpc0:
```

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

6 6(vcp-255/0/1.32768)

fpc1:

---

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

fpc2:

---

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

fpc3:

---

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

2 4(vcp-255/0/2.32768) 5(vcp-255/0/3.32768) 6(vcp-255/0/1.32768)

4 4(vcp-255/0/2.32768)

5 5(vcp-255/0/3.32768)

6 6(vcp-255/0/1.32768)

fpc4:

---

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

fpc5:

---

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



```
fpc6:
```

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

## Release Information

Command introduced in Junos OS Release 9.0.

## RELATED DOCUMENTATION

| [Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis](#)

## show virtual-chassis protocol adjacency

### IN THIS SECTION

- [Syntax | 171](#)
- [Description | 171](#)
- [Options | 171](#)
- [Required Privilege Level | 172](#)

- [Output Fields | 172](#)
- [Sample Output | 173](#)
- [Release Information | 175](#)

## Syntax

```
show virtual-chassis protocol adjacency
<brief | detail | extensive>
<all-members | local | member member-id>
<system-id>
```

## Description

Display the Virtual Chassis Control Protocol (VCCP) adjacency statistics in the Virtual Chassis or VCF for all hardware devices.

## Options

<b>none</b>	Display VCCP adjacency statistics in brief form for all members of the Virtual Chassis or VCF.
<b>brief   detail   extensive</b>	(Optional) Display the specified level of output. Using the <code>brief</code> option is equivalent to entering the command with no options (the default). The <code>detail</code> and <code>extensive</code> options provide identical displays.
<b>all-members</b>	(Optional) Display VCCP adjacency statistics in brief form for all members of the Virtual Chassis or VCF.
<b>local</b>	(Optional) Display VCCP adjacency statistics for the switch or external Routing Engine on which this command is entered.
<b>member <i>member-id</i></b>	(Optional) Display VCCP adjacency statistics for the specified member of the Virtual Chassis or VCF.
<b><i>system-id</i></b>	(Optional) Display VCCP adjacency statistics for the specified member of the Virtual Chassis or VCF.

## Required Privilege Level

clear

## Output Fields

Table 10 on page 172 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"> <li>• Up—The link is up.</li> <li>• Down—The link is down.</li> <li>• New—The link is new.</li> <li>• One-way—The link is transmitting traffic in one direction.</li> <li>• Initializing—The link is initializing.</li> <li>• Rejected—The link is rejected.</li> </ul>	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
```

## Release Information

Command introduced in Junos OS Release 10.4.

## RELATED DOCUMENTATION

[Understanding Virtual Chassis Port Link Aggregation](#)

[Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis](#)

## show virtual-chassis protocol database

### IN THIS SECTION

- [Syntax | 176](#)
- [Description | 176](#)
- [Options | 176](#)
- [Required Privilege Level | 176](#)
- [Output Fields | 176](#)
- [Sample Output | 177](#)
- [Release Information | 180](#)

## Syntax

```
show virtual-chassis protocol database
<brief | detail | extensive>
<all-members | local | member member-id>
```

## Description

Display the Virtual Chassis Control Protocol (VCCP) database statistics for all hardware devices within the Virtual Chassis or VCF.

## Options

<b>none</b>	Display VCCP database statistics in brief form for all members of the Virtual Chassis or VCF.
<b>brief   detail   extensive</b>	(Optional) Display the specified level of output. Using the <b>brief</b> option is equivalent to entering the command with no options (the default). The <b>detail</b> option provides more output than the <b>brief</b> option. The <b>extensive</b> option provides all output and is most useful for customer support personnel.
<b>all-members</b>	(Optional) Display VCCP database statistics in brief form for all members of the Virtual Chassis or VCF.
<b>local</b>	(Optional) Display VCCP database statistics for the switch or external Routing Engine where you enter this command.
<b>member <i>member-id</i></b>	(Optional) Display VCCP database statistics for the specified member of the Virtual Chassis or VCF.

## Required Privilege Level

clear

## Output Fields

[Table 11 on page 177](#) 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: 0x1ddb, Checksum: 0xfd83, Lifetime: 118 secs

Neighbor: 0000.73e9.9a57.00 Interface: vcp-1/0.32768 Metric: 150  
 Neighbor: 0021.59f7.d000.00 Interface: vcp-1/1.32768 Metric: 150  
 Neighbor: 0026.888d.6800.00 Interface: vcp-1/2.32768 Metric: 150

0000.73e9.9a57.00-00 Sequence: 0xf381, Checksum: 0xe065, Lifetime: 115 secs

Neighbor: 0000.4a75.9b7c.00 Interface: vcp-1/0.32768 Metric: 150  
 Neighbor: 0026.888d.6800.00 Interface: vcp-1/1.32768 Metric: 150

0021.59f7.d000.00-00 Sequence: 0x168af, Checksum: 0x8b0b, Lifetime: 116 secs

Neighbor: 0000.4a75.9b7c.00 Interface: vcp-0/0.32768 Metric: 150

```

Neighbor: 0026.888d.6800.00 Interface: vcp-4/0/1.32768 Metric:      15

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

member9:
-----

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

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

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

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

```

## Release Information

Command introduced in Junos OS Release 10.4.

## RELATED DOCUMENTATION

[Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis](#)

[Understanding Virtual Chassis Components](#)

## show virtual-chassis protocol interface

### IN THIS SECTION

- [Syntax | 181](#)
- [Description | 181](#)
- [Options | 181](#)
- [Required Privilege Level | 182](#)
- [Output Fields | 182](#)
- [Sample Output | 182](#)
- [Release Information | 183](#)

### Syntax

```
show virtual-chassis protocol interface  
<brief | detail>  
<all-members | local | member member-id>  
<interface-name>
```

### Description

Display information about Virtual Chassis Control Protocol (VCCP) statistics for VCCP-enabled interfaces within the Virtual Chassis or VCF.

### Options

- |                       |  |
|-----------------------|--|
| <b>none</b>           | Display the VCCP interface statistics in brief form for all members of the Virtual Chassis or VCF.   |
| <b>brief   detail</b> | (Optional) Display the specified level of output. Using the <code>brief</code> option is equivalent to entering the command with no options (the default). The <code>detail</code> option provides more output than the <code>brief</code> option. |
| <b>all-members</b>    | (Optional) Display VCCP interface statistics for all members of the Virtual Chassis or VCF.  |

- interface-name*** (Optional) Display VCCP interface statistics for the specified interface.
- local** (Optional) Display VCCP interface statistics for the switch or external Routing Engine where you enter this command.
- member** (Optional) Display VCCP interface statistics for the specified member of the Virtual  
***member-id*** Chassis or VCF.

## Required Privilege Level

clear

## Output Fields

[Table 12 on page 182](#) 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"> <li>• Up—The link is up.</li> <li>• Down—The link is down.</li> </ul>	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

## Release Information

Command introduced in Junos OS Release 10.4.

## RELATED DOCUMENTATION

[Understanding EX Series Virtual Chassis](#)

[Understanding QFX Series Virtual Chassis](#)

[Understanding Virtual Chassis Ports in an EX8200 Virtual Chassis](#)

[Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis](#)

## show virtual-chassis protocol route

### IN THIS SECTION

- [Syntax | 184](#)
- [Description | 184](#)
- [Options | 185](#)
- [Required Privilege Level | 185](#)
- [Output Fields | 185](#)
- [Sample Output | 186](#)
- [Release Information | 188](#)

### Syntax

```
show virtual-chassis protocol route  
<all-members | local | member member-id>  
<destination-id>
```

### Description

Display the unicast and multicast Virtual Chassis Control Protocol (VCCP) routing tables within the Virtual Chassis or VCF.

## Options

<b>none</b>	Display the unicast and multicast routing tables for all members of the Virtual Chassis.
<b>all-members</b>	(Optional) Display the unicast and multicast routing tables for all members of the Virtual Chassis or VCF.
<b><i>destination-id</i></b>	(Optional) Display the unicast and multicast routing tables to the specified destination member ID for each member of the Virtual Chassis or VCF.
<b>local</b>	(Optional) Display the unicast and multicast routing tables on the device where you enter this command.
<b>member <i>member-id</i></b>	(Optional) Display the unicast and multicast routing tables for the specified member of the Virtual Chassis or VCF.

## Required Privilege Level

clear

## Output Fields

[Table 13 on page 185](#) lists the output fields for the `show virtual-chassis protocol route` command. Output fields are listed in the approximate order in which they appear.

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

Field Name	Field Description
Dev	MAC address of the member storing the VCCP routing table.
Version	Version of the shortest-path-first algorithm that generated the routing table.
System ID	MAC address of the device.
Version	Version of the shortest-path-first (SPF) algorithm that generated the route.



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

Field Name	Field Description
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

```

## Release Information

Command introduced in Junos OS Release 10.4.

## RELATED DOCUMENTATION

[Understanding EX Series Virtual Chassis](#)

[Understanding QFX Series Virtual Chassis](#)

[Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis](#)

## show virtual-chassis protocol statistics

### IN THIS SECTION

- [Syntax | 189](#)
- [Description | 189](#)
- [Options | 189](#)
- [Required Privilege Level | 189](#)
- [Output Fields | 189](#)
- [Sample Output | 191](#)

## Syntax

```
show virtual-chassis protocol statistics
<all-members | local | member member-id>
<interface-name>
```

## Description

Display the Virtual Chassis Control Protocol (VCCP) statistics for all hardware devices within the Virtual Chassis or VCF.

## Options

<b>none</b>	Display VCCP statistics for all members of the Virtual Chassis or VCF.
<b>all-members</b>	(Optional) Display VCCP statistics for all members of the Virtual Chassis or VCF.
<b><i>interface-name</i></b>	(Optional) Display VCCP statistics for the specified interface.
<b>local</b>	(Optional) Display VCCP statistics for the switch or external Routing Engine where you run this command.
<b>member <i>member-id</i></b>	(Optional) Display VCCP statistics for the specified member of the Virtual Chassis or VCF.

## Required Privilege Level

clear

## Output Fields

[Table 14 on page 190](#) lists the output fields for the `show virtual-chassis protocol interface` command. Output fields are listed in the approximate order in which they appear.

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

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

## Sample Output

### show virtual-chassis protocol statistics

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

```
member0:
```

```
-----
```

```
IS-IS statistics for 0021.59f7.d000:
```

PDU type	Received	Processed	Drops	Sent	Rexmit
LSP	8166	8166	0	4551	0
HELLO	1659	1659	0	1693	0
CSNP	2	2	0	3	0
PSNP	1909	1909	0	2293	0
Unknown	0	0	0	0	0
Totals	11736	11736	0	8540	0

```
Total packets received: 11736 Sent: 8540
```

```
LSP queue length: 0 Drops: 0
```

```
SPF runs: 9
```

```
Fragments rebuilt: 1640
```

```
LSP regenerations: 1
```

```
Purges initiated: 0
```

```
member1:
```

```
-----
```

```
IS-IS statistics for 0026.888d.6800:
```

PDU type	Received	Processed	Drops	Sent	Rexmit
LSP	10909	10909	0	12088	0
HELLO	1877	1877	0	2251	0
CSNP	3	3	0	3	0
PSNP	3846	3846	0	3732	0
Unknown	0	0	0	0	0
Totals	16635	16635	0	18074	0

```
Total packets received: 16635 Sent: 18074
```

```
LSP queue length: 0 Drops: 0
```

```
SPF runs: 13
```

```
Fragments rebuilt: 1871
```

LSP regenerations: 2

Purges initiated: 0

member8:

-----

IS-IS statistics for 0000.4a75.9b7c:

PDU type	Received	Processed	Drops	Sent	Rexmit
LSP	7935	7935	0	14865	0
HELLO	2695	2695	0	7124	0
CSNP	4	4	0	4	0
PSNP	4398	4398	0	3666	0
Unknown	0	0	0	0	0
Totals	15032	15032	0	25659	0

Total packets received: 15032 Sent: 25659

LSP queue length: 0 Drops: 0

SPF runs: 26

Fragments rebuilt: 2666

LSP regenerations: 4

Purges initiated: 0

member9:

-----

IS-IS statistics for 0000.73e9.9a57:

PDU type	Received	Processed	Drops	Sent	Rexmit
LSP	10800	10800	0	6327	0
HELLO	1492	1492	0	2356	0
CSNP	2	2	0	2	0
PSNP	2683	2683	0	3149	0
Unknown	0	0	0	0	0
Totals	14977	14977	0	11834	0

Total packets received: 14977 Sent: 11834

LSP queue length: 0 Drops: 0

SPF runs: 19

Fragments rebuilt: 1510

LSP regenerations: 6

Purges initiated: 0

## Release Information

Command introduced in Junos OS Release 10.4.

### RELATED DOCUMENTATION

[Understanding EX Series Virtual Chassis](#)

[Understanding QFX Series Virtual Chassis](#)

[Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis](#)

## show virtual-chassis

### IN THIS SECTION

- [Syntax | 193](#)
- [Description | 193](#)
- [Options | 194](#)
- [Required Privilege Level | 194](#)
- [Output Fields | 194](#)
- [Sample Output | 197](#)
- [Release Information | 201](#)

### Syntax

```
show virtual-chassis <status>
```

### Description

Display information about all members of the Virtual Chassis or VCF.



## Options

- none**      Display information about all Virtual Chassis or VCF member devices.
- status**     Same output as for `show virtual-chassis` without any options.

## Required Privilege Level

view

## Output Fields

[Table 15 on page 194](#) 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
<b>Fabric ID</b>	(VCF only) Assigned ID used to identify the VCF.
<b>Fabric Mode</b>	(VCF only) Mode of the VCF: Enabled, Disabled, or Mixed.
<b>Preprovisioned Virtual Chassis or Preprovisioned Virtual Chassis Fabric</b>	Virtual Chassis or VCF is configured using preprovisioning.
<b>Virtual Chassis ID</b>	Assigned ID that applies to the entire Virtual Chassis or VCF.

Table 15: show virtual-chassis Output Fields (Continued)

Field Name	Field Description
<b>Virtual Chassis Mode</b>	<p>Mode of the Virtual Chassis or VCF. This field indicates support for the Virtual Chassis feature and, if a Virtual Chassis is configured, if it is a mixed or homogenous Virtual Chassis. Values can be:</p> <ul style="list-style-type: none"> <li>• 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).</li> <li>• Disabled—The switch does not support the Virtual Chassis feature.</li> </ul> <p><b>NOTE:</b> Switches that support the Virtual Chassis feature do not display this value. Even if a Virtual Chassis is not currently configured, those switches display Enabled in this field.</p> <ul style="list-style-type: none"> <li>• 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).</li> </ul>
<b>Member ID</b>	<p>Assigned member ID and FPC:</p> <ul style="list-style-type: none"> <li>• On all EX Series Virtual Chassis except EX8200 Virtual Chassis, and on a VCF, the FPC number refers to the member ID assigned to the switch.</li> <li>• On EX8200 Virtual Chassis, member IDs are numbered 0 through 9. The FPC number indicates the slot number of the line card within the Virtual Chassis. The FPC number on member 0 is always 0 through 15. The FPC number on member 1 is always 16 through 31. The FPC number on member 2 is always 32 through 47; and so on for the members.</li> </ul>
<b>Status</b>	<p>For a nonprovisioned configuration:</p> <ul style="list-style-type: none"> <li>• Prsnt for a member that is currently connected to the Virtual Chassis or VCF configuration.</li> <li>• NotPrsnt for a member ID that has been assigned but is not currently connected.</li> </ul> <p>For a preprovisioned configuration:</p> <ul style="list-style-type: none"> <li>• Prsnt for a member that is specified in the preprovisioned configuration file and is currently connected to the Virtual Chassis or VCF.</li> <li>• Unprvsnd for a member that is interconnected with the Virtual Chassis or VCF configuration but is not specified in the preprovisioned configuration file.</li> </ul>

Table 15: show virtual-chassis Output Fields (Continued)

Field Name	Field Description
<b>Serial No</b>	Serial number of the member device.
<b>Alias-Name</b>	<p>The user-configured alias of the member device.</p> <p>The <b>Alias-Name</b> field appears only if an alias has been configured for at least one device in the Virtual Chassis or VCF. Aliases are configured using the alias-name statement in the [edit virtual-chassis aliases serial-number <i>serial-number</i>] hierarchy.</p>
<b>Model</b>	Model number of the member device.
<b>Mastership Priority or Mstr prio</b>	Primary-role priority value of the member device.
<b>Role</b>	<p>Role of the member device: primary, backup, or linecard.</p> <p>An asterisk ( * ) following the Role denotes the member device on which the show virtual-chassis &lt;status&gt; command was issued.</p>
<b>Mixed Mode</b>	<p>Mixed mode configuration status:</p> <ul style="list-style-type: none"> <li>• Y for a member device configured in mixed mode.</li> <li>• N for a member device not configured in mixed mode.</li> <li>• NA for a member device that cannot be configured in mixed mode.</li> </ul>
<b>Route Mode</b>	<p>The route mode of the member device:</p> <ul style="list-style-type: none"> <li>• F for fabric (VCF) mode</li> <li>• VC for Virtual Chassis mode</li> </ul>
<b>Location</b>	<p>Location of the member device.</p> <p>If this field is empty, the location field was not set for the device.</p>

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

Field Name	Field Description
<b>Neighbor List ID and Interface</b>	Member ID of the neighbor member to which this member's Virtual Chassis port (VCP) is connected, and the VCP interface name.

## Sample Output

### show virtual-chassis (EX2300 multigigabit model—EX2300-24MP and EX2300-48MP—Virtual Chassis)

```

user@switch> show virtual-chassis
Virtual Chassis ID: cdef.789a.bc12
Virtual Chassis Mode: Enabled

Member ID  Status  Serial No  Model          Mstr    Mixed Route  Neighbor List
prio Role   Mode  Mode  ID  Interface
0 (FPC 0)  Prsnt   ...       ex2300-24mp   128    Backup      N    VC    3    vcp-255/1/0
                                     1    vcp-255/1/2
1 (FPC 1)  Prsnt   ...       ex2300-48mp   128    Master*     N    VC    2    vcp-255/1/4
                                     0    vcp-255/1/1
2 (FPC 2)  Prsnt   ...       ex2300-24mp   128    Linecard   N    VC    3    vcp-255/1/0
                                     1    vcp-255/1/1
3 (FPC 3)  Prsnt   ...       ex2300-48mp   128    Linecard   N    VC    0    vcp-255/1/4
                                     2    vcp-255/1/5

```

### show virtual-chassis (EX2300 and EX2300 multigigabit model (EX2300-24MP, EX2300-48MP) Virtual Chassis)

```

user@switch> show virtual-chassis status
Preprovisioned Virtual Chassis
Virtual Chassis ID: 9876.5432.abcd
Virtual Chassis Mode: Enabled

Member ID  Status  Serial No  Model          Mstr    Mixed Route  Neighbor List
prio Role   Mode  Mode  ID  Interface
0 (FPC 0)  Prsnt   ...       ex2300-24mp   129    Backup      N    VC    1    vcp-255/1/3
                                     3    vcp-255/1/0
1 (FPC 1)  Prsnt   ...       ex2300-24p    129    Master*     N    VC    2    vcp-255/1/0

```

```

0 vcp-255/1/1
2 (FPC 2) Prsnt ... ex2300-24p 0 Linecard N VC 3 vcp-255/1/2
1 vcp-255/1/3
3 (FPC 3) Prsnt ... ex2300-48mp 0 Linecard N VC 2 vcp-255/1/5
0 vcp-255/1/0

```

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

```

user@switch> show virtual-chassis
Virtual Chassis ID: 00ab.cdef.1234
Virtual Chassis Mode: Enabled

Member ID  Status  Serial No  Model      Mastership  Mixed  Neighbor List
           priority Role Mode ID  Interface
0 (FPC 0)  Prsnt   ...       ex4200-24t  249  Master*   N   8  vcp-0
           1  vcp-1
1 (FPC 1)  Prsnt   ...       ex4200-24t  248  Backup    N   0  vcp-0
           2  vcp-1
2 (FPC 2)  Prsnt   ...       ex4200-48p  247  Linecard  N   1  vcp-0
           3  vcp-1
3 (FPC 3)  Prsnt   ...       ex4200-24t  246  Linecard  N   2  vcp-0
           4  vcp-1
4 (FPC 4)  Prsnt   ...       ex4200-48p  245  Linecard  N   3  vcp-0
           5  vcp-1
5 (FPC 5)  Prsnt   ...       ex4200-48t  244  Linecard  N   4  vcp-0
           6  vcp-1
6 (FPC 6)  Prsnt   ...       ex4200-48t  243  Linecard  N   5  vcp-0
           7  vcp-1
7 (FPC 7)  Prsnt   ...       ex4200-24f  242  Linecard  N   6  vcp-0
           8  vcp-1
8 (FPC 8)  Prsnt   ...       ex4200-24f  241  Linecard  N   7  vcp-0
           0  vcp-1

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

```

### show virtual-chassis (Mixed EX4300 multigigabit model—EX4300-48MP—Virtual Chassis)

```

user@switch> show virtual-chassis
Preprovisioned Virtual Chassis
Virtual Chassis ID: abcd.ef00.1234
Virtual Chassis Mode: Mixed

```

Member ID	Status	Serial No	Model	Mstr prio	Role	Mixed Mode	Route Mode	Neighbor ID	List Interface
0 (FPC 0)	Prsnt	...	ex4300-48mp	129	Master*	Y	VC	1	vcp-255/1/0
								3	vcp-255/1/3
1 (FPC 1)	Prsnt	...	ex4300-48mp	129	Backup	Y	VC	0	vcp-255/1/1
								2	vcp-255/1/3
2 (FPC 2)	Prsnt	...	ex4300-48mp	0	Linecard	Y	VC	3	vcp-255/1/1
								1	vcp-255/1/3
3 (FPC 3)	Prsnt	...	ex4300-48t	0	Linecard	Y	VC	0	vcp-255/1/0
								2	vcp-255/1/2

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

```

user@external-routing-engine> show virtual-chassis
Virtual Chassis ID: cdc.d.1212.efef
Virtual Chassis Mode: Enabled

```

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

### show virtual-chassis (QFX5110 Virtual Chassis)

```

user@switch> show virtual-chassis
Preprovisioned Virtual Chassis
Virtual Chassis ID: abab.1212.cdc.d

```

Virtual Chassis Mode: Enabled

MemberID	Status	Serial No	Model	Mstr prio	Role	Mixed Mode	Route Mode	Neighbor List ID	Interface
0 (FPC 0)	Prsnt	...	qfx5110-32q	129	Backup	Y	VC	9	vcp-255/0/12 1 vcp-255/0/16
1 (FPC 1)	Prsn	...	qfx5110-32q	129	Master*	Y	VC	2	vcp-255/0/16 0 vcp-255/0/17
2 (FPC 2)	Prsnt	...	qfx5110-32q	0	Linecard	Y	VC	3	vcp-255/0/11 1 vcp-255/0/14
3 (FPC 3)	Prsnt	...	qfx5110-32q	0	Linecard	Y	VC	4	vcp-255/0/14 2 vcp-255/0/31
4 (FPC 4)	Prsnt	...	qfx5110-32q	0	Linecard	Y	VC	5	vcp-255/0/10 3 vcp-255/0/5
5 (FPC 5)	Prsnt	...	qfx5100e-24q-2p	0	Linecard	Y	VC	6	vcp-255/0/12 4 vcp-255/0/20
6 (FPC 6)	Prsnt	...	qfx5110-48s-4c	0	Linecard	Y	VC	7	vcp-255/0/10 5 vcp-255/0/49
7 (FPC 7)	Prsnt	...	qfx5100e-48s-6q	0	Linecard	Y	VC	6	vcp-255/0/18 8 vcp-255/0/31
8 (FPC 8)	Prsnt	...	qfx5110-48s-4c	0	Linecard	Y	VC	7	vcp-255/0/21 9 vcp-255/0/49
9 (FPC 9)	Prsnt	...	qfx5110-32q	0	Linecard	Y	VC	8	vcp-255/0/13 0 vcp-255/0/17

### show virtual-chassis (QFX5200 Virtual Chassis)

user@switch> show virtual-chassis

Virtual Chassis ID: abab.1212.cdcd

Virtual Chassis Mode: Enabled

Member ID	Status	Serial No	Model	Mstr prio	Role	Mixed Mode	Route Mode	Neighbor List ID	Interface
0 (FPC 0)	Prsnt	...	qfx5200-32c-r	128	Master*	N	VC	1	vcp-255/0/19 2 vcp-255/0/26
1 (FPC 1)	Prsnt	...	qfx5200-32c-r	128	Linecard	N	VC	0	vcp-255/0/20 2 vcp-255/0/16
2 (FPC 2)	Prsnt	...	qfx5200-32c-32q	128	Backup	N	VC	0	vcp-255/0/15 1 vcp-255/0/18

## show virtual-chassis (QFX5100 Virtual Chassis Fabric)

```

user@switch> show virtual-chassis
Preprovisioned Virtual Chassis Fabric
Fabric ID: 0123.abcd.4567
Fabric Mode: Enabled

```

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

## Release Information

Command introduced in Junos OS Release 9.2.

Fabric ID, Fabric Mode, and Route Mode output fields introduced in Junos OS Release 13.2X51-D20.

Alias-Name output field introduced in Junos OS Release 14.1X53-D10.

## RELATED DOCUMENTATION

[\*show virtual-chassis active-topology\*](#)

[\*show virtual-chassis protocol adjacency\*](#)

[\*show virtual-chassis vc-path\*](#)

[Understanding Mixed EX Series and QFX Series Virtual Chassis](#)

[Understanding Mixed Virtual Chassis Fabric](#)

[Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis](#)



## show virtual-chassis vc-port

### IN THIS SECTION

- [Syntax | 202](#)
- [Description | 202](#)
- [Options | 202](#)
- [Required Privilege Level | 203](#)
- [Output Fields | 203](#)
- [Sample Output | 205](#)
- [Release Information | 208](#)

### Syntax

```
show virtual-chassis vc-port  
<all-members | local | member member-id>
```

### Description

Display the status of the Virtual Chassis ports (VCPs), including dedicated VCPs, default-configured VCPs, and uplink ports configured as VCPs, if present.

### Options

<b>none</b>	Display the operational status of all VCPs of the member switch where you enter the command.
<b>all-members</b>	(Optional) Display the operational status of all VCPs on all members of the Virtual Chassis or VCF.
<b>local</b>	(Optional) Display the operational status of the switch or external Routing Engine where you enter this command.
<b>member <i>member-id</i></b>	(Optional) Display the operational status of all VCPs for the specified member of the Virtual Chassis or VCF.

## Required Privilege Level

view

## Output Fields

Table 16 on page 203 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>member name</i>	Device or member number in the Virtual Chassis, represented as <i>fpc number</i> , <i>member number</i> , or <i>local-re</i> (the primary member).
Interface or PIC/ Port	<p>VCP name.</p> <ul style="list-style-type: none"> <li>The dedicated VCPs in an EX4200 or EX4500 Virtual Chassis are <i>vcp-0</i> and <i>vcp-1</i>. The dedicated VCPs in an EX4550 Virtual Chassis are <i>VCP-1/0</i>, <i>VCP-1/1</i>, <i>VCP-2/0</i>, and <i>VCP-2/1</i>.</li> <li>Optical ports set as VCPs are named <i>1/0</i> and <i>1/1</i>, representing the PIC number and the port number.</li> <li>The native VCP (port 0) on an XRE200 External Routing Engine in an EX8200 Virtual Chassis is named <i>vcp-0</i>.</li> <li>The VCPs on each Virtual Chassis Control Interface (VCCI) module in an XRE200 External Routing Engine are named using the <i>vcp-slot-number/port-number</i> convention; for instance, <i>vcp-1/0</i>.</li> <li>The VCPs on EX8200 member switches are named using the <i>vcp-slot-number/pic-number/interface-number</i> convention; for instance, <i>vcp-3/0/2</i>.</li> <li>A 255 as the first number in your port number indicates that your VCP is part of a Link Aggregation group (LAG) bundle. For example, <i>vcp-255/1/0</i> indicates that the dedicated VCP named <i>vcp-1/0</i> is part of a LAG bundle, and <i>vcp-255/1/0</i> represents an uplink port previously named <i>xe-0/1/0</i> that is now part of a VCP LAG bundle.</li> </ul>

Table 16: show virtual-chassis vc-port Output Fields (Continued)

Field Name	Field Description
Type	<p>Type of VCP:</p> <ul style="list-style-type: none"> <li>• Dedicated—Rear panel dedicated VCP on an EX4200 or EX4300 multigigabit model switch, a Virtual Chassis module port on EX4500 or EX4550 switch, or any VCP link connected to an XRE200 External Routing Engine in an EX8200 Virtual Chassis.</li> <li>• Configured—Optical port configured as a VCP, which includes default-configured VCPs (set in the default factory configuration) or those that are set by a user.</li> <li>• Auto-Configured—Optical port automatically converted into a VCP.</li> </ul> <p>See <a href="#">Setting an Uplink Port on an EX Series or QFX Series Switch as a Virtual Chassis Port</a> or <a href="#">Configuring an EX4650 or a QFX Series Virtual Chassis</a> for information about configuring VCPs, and <a href="#">Automatic Virtual Chassis Port (VCP) Conversion</a> for details on how a port is automatically converted into a VCP.</p>
Trunk ID	<p>A positive-number ID assigned to a link aggregation group (LAG) formed by the Virtual Chassis. The trunk ID value is -1 if no trunk is formed. A LAG between uplink VCPs requires that the link speed be the same on connected interfaces and that at least two VCPs on one member be connected to at least two VCPs on the other member in an EX4200 or EX4500 Virtual Chassis.</p> <p>Dedicated VCP LAGs are assigned trunk IDs 1 and 2. Trunk IDs for LAGs formed with uplink VCPs therefore have values of 3 or greater.</p> <p>The trunk ID value changes if the link-adjacency state between LAG members changes; trunk membership is then allocated or deallocated.</p>
Status	<p>Interface status:</p> <ul style="list-style-type: none"> <li>• absent—Interface is not a VCP link.</li> <li>• down—VCP link is down.</li> <li>• up—VCP link is up.</li> </ul>
Speed (mbps)	Speed of the interface in megabits per second.
Neighbor ID/ Interface	The Virtual Chassis member ID and interface of a VCP on a member that is connected to the interface or PIC/Port field in the same row as this interface.

## Sample Output

### show virtual-chassis vc-port (EX4200 Virtual Chassis)

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

```
fpc0:
```

```
-----
```

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

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

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

```
member0:
```

```
-----
```

Interface or Slot/PIC/Port	Type	Trunk ID	Status	Speed (mbps)	Neighbor ID Interface
vcp-0/0	Dedicated	-1	Up	1000	8 vcp-1/1
vcp-0/1	Dedicated	-1	Up	1000	8 vcp-2/0
4/0/4	Configured	-1	Up	10000	1 vcp-3/0/4
4/0/7	Configured	-1	Down	10000	
4/0/3	Configured		Absent		
4/0/2	Configured		Absent		
4/0/5	Configured		Absent		
4/0/6	Configured		Absent		
4/0/1	Configured		Absent		
4/0/0	Configured		Absent		

```
member1:
```

```
-----
```

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

```

vcp-0/0      Dedicated      -1  Up      1000      8  vcp-1/2
3/0/0       Configured     -1  Down    10000
3/0/1       Configured     -1  Down    10000
3/0/4       Configured     -1  Up      10000      0  vcp-4/0/4
3/0/5       Configured
4/0/5       Configured
4/0/4       Configured

```

member8:

```

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

```

fpc1:

```

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

```

fpc2:

```

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

```

fpc3:

```

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

```

### show virtual-chassis vc-port (QFX5120-32C Virtual Chassis)

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

```
localre:
```

```

-
Interface  Type          Trunk  Status  Speed  Neighbor
or        ID          (mbps) ID  Interface
PIC / Port
0/13      Auto-Configured  6    Up    100000  1  vcp-255/0/13
0/23      Auto-Configured  6    Up    100000  1  vcp-255/0/23
0/26      Auto-Configured  5    Up    40000   1  vcp-255/0/26
0/27      Configured       5    Up    40000   1  vcp-255/0/27
0/6       Configured       5    Up    40000   1  vcp-255/0/6
0/8       Auto-Configured  5    Up    40000   1  vcp-255/0/8
0/9       Auto-Configured  6    Up    100000  1  vcp-255/0/9

fpc1:
-
Interface  Type          Trunk  Status  Speed  Neighbor
or        ID          (mbps) ID  Interface
PIC / Port
0/13      Auto-Configured  6    Up    100000  0  vcp-255/0/13
0/23      Auto-Configured  6    Up    100000  0  vcp-255/0/23
0/26      Auto-Configured  5    Up    40000   0  vcp-255/0/26
0/27      Configured       5    Up    40000   0  vcp-255/0/27
0/6       Configured       5    Up    40000   0  vcp-255/0/6
0/8       Auto-Configured  5    Up    40000   0  vcp-255/0/8
0/9       Auto-Configured  6    Up    100000  0  vcp-255/0/9

```

## Release Information

Command introduced in Junos OS Release 9.0.

## RELATED DOCUMENTATION

*show virtual-chassis vc-port statistics*

[Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis](#)

## show virtual-chassis vc-port statistics

### IN THIS SECTION

- [Syntax | 209](#)
- [Description | 209](#)
- [Options | 209](#)
- [Required Privilege Level | 210](#)
- [Output Fields | 210](#)
- [Sample Output | 214](#)
- [Release Information | 218](#)

### Syntax

```
show virtual-chassis vc-port statistics
<all-members | local | member member-id>
<brief | detail | extensive >
<interface-name>
```

### Description

Display the traffic statistics collected on Virtual Chassis ports (VCPs).

### Options

<b>none</b>	Display traffic statistics for VCPs of all members of a Virtual Chassis or VCF.
<b>brief   detail   extensive</b>	(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.
<b>all-members</b>	(Optional) Display traffic statistics for VCPs of all members of a Virtual Chassis or VCF
<b><i>interface-name</i></b>	(Optional) Display traffic statistics for the specified VCP.



- local** (Optional) Display traffic statistics for VCPs on the switch or external Routing Engine where you enter this command.
- member *member-id*** (Optional) Display traffic statistics for VCPs on the specified member of a Virtual Chassis or VCF.

## Required Privilege Level

view

## Output Fields

[Table 17 on page 210](#) 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
<i>fpcnumber</i>	(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
<i>member number</i>	(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
<i>master: number</i>	Member ID of the primary Routing Engine.	All levels

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

Field Name	Field Description	Level of Output
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
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"> <li>• Invalid FCS with an integral number of octets (FCS error)</li> <li>• Invalid FCS with a nonintegral number of octets (alignment error)</li> </ul>	detail, extensive

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

Field Name	Field Description	Level of Output
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><b>NOTE:</b> This definition of <i>jabber</i> is different from the definition in IEEE-802.3 section 8.2.1.5 (10Base5) and section 10.3.1.4 (10Base2). These documents define <i>jabber</i> as the condition in which any packet exceeds 20 ms. The allowed range to detect jabber is between 20 ms and 150 ms.</p>	detail, extensive
Fragments received	<p>Number of packets received on the VCP that were shorter than 64 octets (excluding framing bits, but including FCS octets), and had either an FCS error or an alignment error.</p> <p>Fragment frames normally increment because both runts (which are normal occurrences caused by collisions) and noise hits are counted.</p>	detail, extensive
Ifout errors	Number of outbound packets received on the VCP that could not be transmitted because of errors.	detail, extensive

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

Field Name	Field Description	Level of Output
Packet drop events	Number of outbound packets received on the VCP that were dropped, rather than being encapsulated and sent out of the switch as fragments. The packet drop counter is incremented if a temporary shortage of packet memory causes packet fragmentation to fail.	detail, extensive
64 octets frames	Number of packets received on the VCP (including invalid packets) that were 64 octets in length (excluding framing bits, but including FCS octets).	detail, extensive
65-127 octets frames	Number of packets received on the VCP (including invalid packets) that were between 65 and 127 octets in length, inclusive (excluding framing bits, but including FCS octets).	detail, extensive
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

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

Field Name	Field Description	Level of Output
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}

```

## Release Information

Command introduced in Junos OS Release 9.0.

The options `all-members`, `brief`, `detail`, `extensive`, and `local` were added in Junos OS Release 9.3 for EX Series switches.

## RELATED DOCUMENTATION

*clear virtual-chassis vc-port statistics*

*show virtual-chassis vc-port*

[Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis](#)

[Verifying Virtual Chassis Ports in an EX8200 Virtual Chassis](#)

# 4

PART

## Troubleshooting

---

[Troubleshooting Procedures | 220](#)

---

# Troubleshooting Procedures

## IN THIS CHAPTER

- [Recovering a Routing Engine in a Member Switch in an EX8200 Virtual Chassis | 220](#)

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

### IN THIS SECTION

- [Problem | 220](#)
- [Solution | 221](#)

### 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.

## 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 primary 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 | 58](#)

---

[Installing Software for a Single Device in an EX8200 Virtual Chassis | 56](#)

---

[Understanding Software Upgrades in an EX8200 Virtual Chassis | 16](#)